Information Systems Education Journal

Volume 24, No. 2
March 2026
ISSN: 1545-679X

In this issue:

iscap
Information Systems & Computing Academic Professionals

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **ISCAP** (Information Systems and Computing Academic Professionals). Publishing frequency is five times per year. The first year of publication was 2003.

ISEDJ is published online (https://isedj.org). Our sister publication, the Proceedings of the ISCAP Conference (https://iscap.us/proceedings) features all papers, abstracts, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the ISCAP conference. All papers, whether award-winners or not, are invited to resubmit for journal consideration after applying feedback from the Conference presentation. Award winning papers are assured of a publication slot; however, all re-submitted papers including award winners are subjected to a second round of three blind peer reviews to improve quality and make final accept/reject decisions. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 35%.

Information Systems Education Journal is pleased to be listed in the Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to volunteer members of ISCAP who perform the editorial and review processes for ISEDJ.

2026 ISCAP Board of Directors

Amy Connolly James Madison University President	Michael Smith Georgia Institute of Technology Vice President	Jeff Cummings Univ of NC Wilmington Past President
David Firth University of Montana Director	Mark Frydenberg Bentley University Director/Secretary	Leigh Mutchler James Madison University Director
RJ Podeschi Millikin University Director/Treasurer	Bryan Reinicke Rochester Institute of Technology / Director	Jeffry Babb West Texas A&M University Director/Curricular Matters
Eric Breimer Siena University Director/2026 Conf Chair	Tom Janicki Univ of NC Wilmington Director/Meeting Planner	Xihui “Paul” Zhang University of North Alabama Director/JISE Editor

Copyright © 2026 by Information Systems and Computing Academic Professionals (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to Paul Witman, Editor, editor@isedj.org.

INFORMATION SYSTEMS EDUCATION JOURNAL

Editors

Kevin Mentzer
Editor
Nichols College

Ira Goldman
Associate Editor
Siena University

David Yates
Associate Editor
Bentley University

Michelle Louch
Teaching Cases & Exercises Editor
University of Pittsburgh - Greensburg

Mark Pisano
Teaching Cases & Exercises Associate Editor
Southern Connecticut State University

Thomas Janicki
Publisher
U of North Carolina Wilmington

David Woods
Assistant Publisher
Miami University Regionals

Paul Witman
Emeritus Editor
(2021-2026)
California Lutheran University

Jeffry Babb
Emeritus Editor
(2016-2021)
West Texas A&M University

Donald Colton
Emeritus Editor
(2003-2010)
Brigham Young University Hawaii

Preparing the XR Workforce: Curricular Patterns of Extended Reality in Higher Education

Tan Gürpinar
tan.gurpinar@qu.edu

Sana Quadri
Sana.Quadri@qu.edu

Shizhen (Jasper) Jia
shizhen.jia@quinnipiac.edu

Luis SaCouto
Luis.SaCouto@qu.edu

Guido Lang
guido.lang@quinnipiac.edu

Quinnipiac University
Hamden, CT 06518

Abstract

As extended reality (XR) technologies, including virtual reality (VR), augmented reality (AR), and mixed reality (MR), become increasingly embedded across industries, higher education institutions are tasked with preparing a workforce capable of designing, deploying, and critically engaging with immersive technologies. In response, universities have introduced a growing number of XR-related courses, programs, and initiatives. However, questions remain regarding how these offerings contribute distinct educational value and align with evolving skill demands. This study examines curricular patterns of XR integration in U.S. higher education, focusing on institutions with AACSB-accredited business schools as an anchor for identifying cross-disciplinary XR adoption. Drawing on a comprehensive dataset of 547 U.S.-based institutions, the analysis assesses XR-related academic offerings across multiple disciplines, including computer science, design, business, health sciences, and engineering. By systematically analyzing curricula, initiatives, and applications, the study identifies areas of overlap, differentiation, and convergence between XR and adjacent domains such as artificial intelligence, robotics, and interactive media. Results indicate that approximately 43% of institutions integrate XR within their academic offerings, combining conceptual instruction with applied, hands-on engagement. At the same time, XR education reveals persistent institutional silos that limit cross-disciplinary integration. These findings provide evidence-based insights for institutions seeking to design or refine XR curricula that support workforce preparation, highlighting prevailing trends, disciplinary contexts, and skill emphases. Overall, this research contributes a structured view of how higher education is positioning XR within broader curricular ecosystems to bridge traditional educational models with the emerging demands of the immersive technology sector.

Keywords: Extended Reality, Virtual Reality, Augmented Reality, Digital Learning, Immersive Technologies, Experiential Learning

Recommended Citation: Gürpinar, T., Quadri, S., Jia, S., SaCouto, L., Lang, G., (2026). Preparing the XR Workforce: Curricular Patterns of Extended Reality in Higher Education. *Information Systems Education Journal* v24(n2), pp 4-15. DOI# https://doi.org/10.62273/NXRK8987

Preparing the XR Workforce: Curricular Patterns of Extended Reality in Higher Education

Tan Gürpinar, Sana Quadri. Shizhen (Jasper) Jia, Luis SaCouto and Guido Lang

1. INTRODUCTION

Extended reality (XR), including virtual reality (VR), augmented reality (AR), and mixed reality (MR), has rapidly evolved from a niche innovation to a set of mature technologies with demonstrable value across different domains, such as healthcare, education, architecture, and business (Dincelli et al. 2026). Rather than supplementing existing workflows, XR is increasingly integrated into core professional practices, facilitating new modes of spatial visualization, embodied interaction, and remote collaboration (Zhou et al., 2025). As XR technologies become more accessible and technically robust, their adoption is prompting substantial shifts in how knowledge is produced, experienced, and transferred within industry and academic contexts (Gürpinar et al. 2025a).

The growing relevance of XR has created a pressing need for educational institutions to prepare students for the emerging demands of the immersive technology sector (Burke et al., 2025; Khlaif et al., 2024; Küpeli & Gürpinar, 2023). While industries increasingly look to XR to improve services and operations, many universities are only beginning to explore how to integrate XR meaningfully into their curricula (El Dandachi et al., 2023). There is, however, an urgent challenge: institutions must not only adopt new technologies but also design programs that cultivate the technical, creative, and interdisciplinary skills required to build, deploy, and manage XR applications effectively (Düdder et al., 2021; Karamitsos et al., 2024; Mentzer et al., 2025).

Despite growing recognition of XR's importance, it remains unclear how it is systematically embedded within higher education. This study addresses the following research questions:

RQ1: What domains of knowledge and skill sets are most frequently emphasized in XR-related curricula across higher education institutions?
RQ2: How do academic programs integrate and balance the technical, creative, and ethical dimensions of XR within course structures?
RQ3: To what extent do current XR course offerings reflect the interdisciplinary nature of the field, how are cross-departmental collaborations facilitated?

This paper seeks to map and analyze the current landscape of XR-related academic offerings in U.S. higher education, with a focus on institutions accredited by the AACSB. By systematically reviewing course offerings, program structures, and curricular emphasis across a representative sample of 547 U.S. universities, we aim to provide ab baseline and insights into how institutions are equipping students with XR-relevant competencies. Our research identifies trends, shared themes, and points of divergence across programs, offering input for universities seeking to expand or refine their XR initiatives. As demand for professionals with XR-related skills accelerates, reflected in industry job postings, research collaborations, and startup activities, academic institutions face the opportunity to position themselves as leaders in immersive technology education. This paper contributes to that effort by offering a structured analysis of current practices and by highlighting areas where academic innovation can help bridge the gap between traditional learning models and the demands of the fast-evolving XR sector.

In the following sections, we present the scientific and industry context, describe our research methodology, and share key findings as well as recommendations to support the development of robust XR education pathways.

2. SCIENTIFIC BACKGROUND

XR technologies offer new modalities for experiential and immersive learning, prompting institutions to rethink traditional models of instruction (Rauschnabel et al., 2022). This section reviews the current landscape of XR integration in higher education, situating it within broader pedagogical and institutional trans-formations. The adoption of XR technologies in higher education is a complex process that requires aligning technological innovations with institutional objectives (El Dandachi et al., 2023). Institutions are increasingly viewing XR not only as a tool for fostering innovation and enhancing student engagement, but also as a strategically aligned initiative that advances research, interdisciplinary collaboration, and workforce development (Mentzer et al., 2025). Strategic alignment refers to the extent to which XR initiatives are integrated into an institution's broader mission, such as enhancing digital infrastructure, promoting cross-disciplinary learning, and addressing evolving industry demands. A study by Huang & Hew (2021) found that higher education institutions with strong leadership and a clear vision for technology integration were more successful in aligning XR projects with their strategic objectives. Successful XR integration, therefore, requires active participation from leadership to create an environment that supports innovation, experimentation, and long-term sustainability.

Industry trends offer important context for understanding which domains of knowledge and skill sets should be emphasized in XR-related curricula. An analysis of XR-related job advertisements by Verma et al. (2021) found that employers consistently seek competencies in UI/UX design, 3D asset creation, real-time graphics rendering, and emerging technologies and systems. These technical and creative skills reflect the interdisciplinary nature of XR development, where design thinking, programming, and visual storytelling converge (King & Gürpinar, 2025). Although not based on educational institutions, this study offers a relevant benchmark for assessing academic alignment with workforce needs.

For XR to be successfully integrated into university curricula, faculty and staff must develop a diverse set of competencies, which are essential to ensuring effective use and application. Following Tusher et al. (2024), these competencies extend beyond technical expertise and should include the following:

- **Pedagogical Competence**: Faculty must be equipped to design and implement XR-based learning modules that meet educational objectives. This includes the ability to integrate immersive experiences with traditional teaching methods, ensuring a seamless learning process.
- **Technological Proficiency**: While XR adoption requires specialized technical knowledge, faculty and staff also need to be familiar with the basic infrastructure, software, and hardware involved in XR applications. Collaboration with IT departments and external partners often plays a key role in addressing these needs.
- **Organizational Strategy**: Business schools need to develop and communicate a strategic vision for XR implementation that aligns with broader academic and institutional goals. This vision should include plans for faculty development, infrastructure investments, and long-term sustainability of XR initiatives.
- **Ethical and Equity Awareness**: As XR technologies have the potential to enhance educational experiences, it is critical to ensure affordability and accessibility to all students, including those with disabilities. Ethical concerns surrounding privacy, data security, and the potential for addiction must also be considered.

Finally, the interdisciplinary nature of XR is increasingly emphasized in both literature and institutional practice. Studies from Radianti et al. (2020) highlight that XR curricula often merge elements from computer science, design, psychology, and subject-specific domains (e.g. medicine, architecture or marketing), underscoring the need for cross-disciplinary fluency. Institutional initiatives further reflect this trend. For example, Yale University's Blended Reality Applied Research Project brings together faculty from engineering, arts, the humanities, and library sciences to co-develop XR applications (CCAM, 2025). Similarly, the University of Michigan's XR Initiative fosters collaboration across departments including education, nursing, design, and computer science, offering centralized support and funding for interdisciplinary XR course development (Georgieva et al., 2024). These models not only facilitate shared resource use and curricular innovation, but also highlight a broader shift in higher education toward breaking down departmental barriers in response to the transdisciplinary demands of emerging tech.

3. RESEARCH METHODOLOGY

This study investigates the extent and nature of XR integration in higher education curricula. To achieve this, a systematic search was conducted in June and July 2025 to identify XR-related course offerings across U.S. universities. The research design builds upon established methodological frameworks used in prior analyses of emerging technologies in higher education, with an emphasis on course identification, classification, and curricular analysis (see Figure 1) (Ceccucci et al., 2020; Gürpinar et al., 2025b). As a starting point, the official AACSB website was queried to identify all U.S.-based institutions with AACSB-accredited business schools offering undergraduate or graduate programs. This yielded a list of 547 universities, representing a diverse and representative sample of higher education

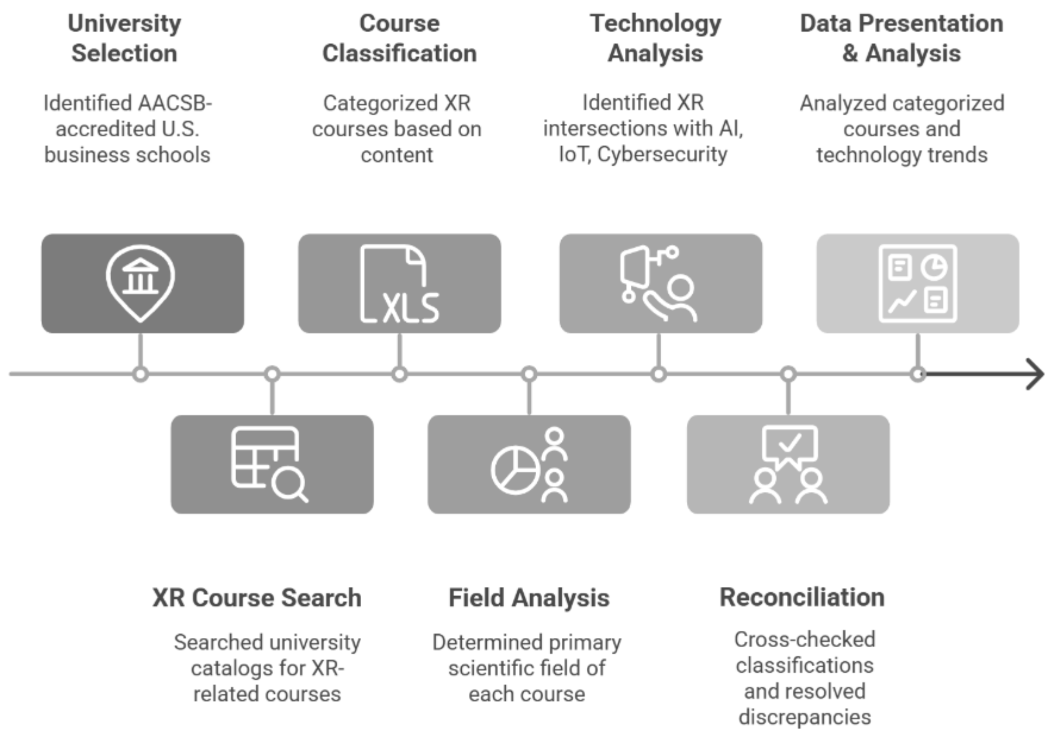


Figure 1: Research Methodology Overview

institutions across the country. The selected sample served as the basis for a comprehensive examination of XR-related course offerings and programmatic integration.

Identification of XR-Related Courses
Following the identification of the institutions, the next step involved searching the websites and academic catalogs of the identified universities for courses related to XR technologies. The search term "extended reality" was used, encompassing virtual reality (VR), augmented reality (AR), and mixed reality (MR). The goal was to identify universities offering at least one course related to XR in their curricula. From this search, 236 universities were identified as offering at least one course on XR technologies. This represents approximately 43% of the total number of universities surveyed. Data such as the course title, department, school, level, and course descriptions were then collected from the universities' catalogs and websites. This data was used for subsequent analysis.

Course Classification
The courses were classified into categories based on their content and subject matter. In the first step, the authors reviewed course descriptions of the identified XR-related courses to determine the primary topics covered. A set of categories was developed from this initial review, ensuring that all relevant XR-related subjects were accounted for. Each course was then mapped to one of these iteratively established categories.

If a course did not fit into any of the existing categories, a new category was created, and all authors were notified of the update. To ensure consistency, multiple authors independently reviewed samples of courses and classified them. This method is similar to the approach employed by Yang & Wen (2017) in their survey of university IS program curricula (Yang & Wen, 2017). Once all courses were independently classified, the authors reconciled their categorization through peer debriefing. If discrepancies in classification arose, the authors discussed and agreed upon the final categorization.

Field Analysis
In addition to classifying the courses, the analysis also examined the associated academic disciplines, intended learning outcomes, and course formats. This was accomplished by systematically reviewing department affiliations and course descriptions to determine the primary focus of each course. Specifically, courses were

categorized based on whether they (1) focused explicitly on XR topics, (2) addressed other subject areas while incorporating XR technologies as tools, or (3) combined both approaches. This framework provided insights into the ways XR is integrated into curricula across disciplines and levels of study.

Technology Analysis
The interplay between XR and other emerging technologies was another criterium and it was analyzed how XR technologies intersect with technologies such as Artificial Intelligence (AI), Internet of Things (IoT), and Cybersecurity. This was accomplished by manually scanning course descriptions for references to these technologies, as well as relevant synonyms and related terms. Understanding the technological intersections allowed the research team to assess how XR is positioned within the broader landscape of digital technologies.

Reconciliation and Data Presentation
To ensure the accuracy and consistency of the data, a final peer review process was implemented similar to one described for the initial course categorization. Samples of the field and technology analysis were also cross-checked by multiple authors to reach consensus on items with unclear allocation.

The data collected from the university catalogs were analyzed and presented using a variety of statistical and qualitative methods. Course classifications were analyzed to determine the prevalence of different XR-related topics across higher education curricula. The field analysis was used to understand how XR is integrated into various academic disciplines. Finally, the technology analysis was conducted to identify trends in how XR is being combined with other emerging technologies, providing insight into the broader direction of innovation in higher education.

4. FINDINGS AND DISCUSSION

XR-related courses are offered across a wide range of university departments and programs, as shown in Table 1. While most are hosted by departments related to Media & Film or Engineering & Computer Science, Engineering, and Media Studies, some also appear in less expected areas like Psychology, Archeology, or Nursing. A significant number of courses are also offered in a multidisciplinary format, cross-listed across several departments. A more detailed distribution of XR courses across scientific disciplines and programs can be obtained from Appendix A.

Department	Count	(%)
Media, Film & Interactive Design	252	27.6
Engineering, Computing & Technology	238	26.1
Fine Arts & Performing Arts	94	10.3
Interdiscipl. / Emerging Fields	86	9.4
Humanities & Languages	61	6.7
Education & Instructional Technology	54	5.9
Business, Management & Policy	26	2.8
Health & Biomedical Sciences	22	2.4
Social Sciences	19	2.1
Science & Math	17	1.9
Architecture & Planning	14	1.5
Multidiscipl. / University Offering	30	3.3

Table 1: Departments Offering XR Courses

The number of XR courses offered by universities varies. Most schools (311) do not offer any XR courses (Figure 2). However, 236 schools offer at least one, 142 at least two, and 16 schools offer a portfolio of more than 10 XR courses (Figure 2).

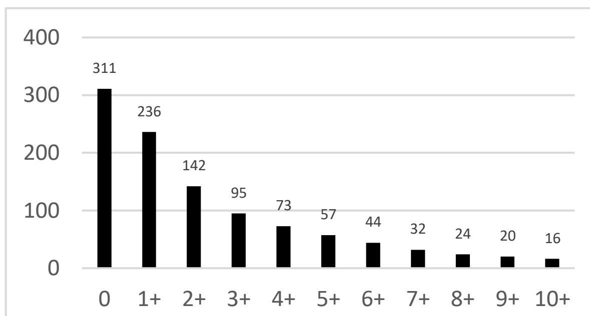


Figure 2: University's Number of XR Courses

The majority (57%) of the XR courses are offered are at the undergraduate level, 26% at the graduate level, and 17% can be taken at both the graduate and undergraduate levels (Figure 3).

©2026 ISCAP (Information Systems and Computing Academic Professionals)

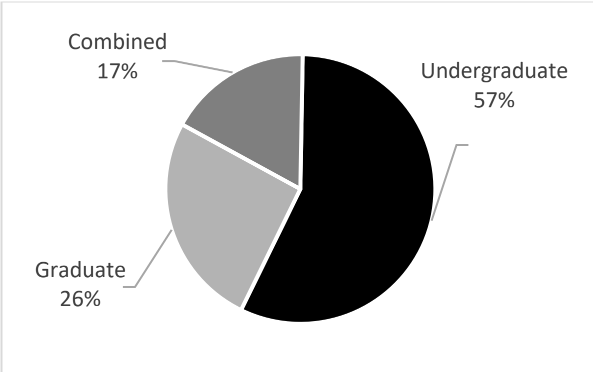


Figure 3: Academic Level of XR Courses

A greater total number of public universities offer XR courses (157 Public vs 79 Private, see Table 2). However, in looking within the university types, a slightly greater percentage of the private universities offer XR courses. The analysis shows that 44.4% of private universities offer at least one XR course, compared to 42.5% of public universities.

	Count	Total	Percentage
Private	79	178	44.4%
Public	157	369	42.5%

Table 2: Universities Offering XR Courses

Universities with larger undergraduate populations are generally more likely to offer XR courses. Among institutions with 25,001 to 30,000 students, 68% provide at least one XR course. This figure increases to 80% for universities enrolling over 45,000 students. In contrast, smaller institutions with fewer than 5,000 undergraduates show a considerably lower participation rate, with 32% offering XR courses (see Table 3).

Appendix A presents a diagram illustrating the distribution of XR course offerings across seven broad scientific fields and the departments within them. The largest share of XR courses falls under Engineering, Technology, and Mathematics, accounting for 26% of all courses. Within this field, Computer Science, Information Technology, and Engineering & Architecture programs are the most common providers.

University Size	Count	Total	Percentage
0 - 5,000	59	186	31.7%
5,001 - 10,000	58	149	38.9%
10,001 - 15,000	28	71	39.4%
15,001 - 20,000	29	45	64.4%
20,001 - 25,000	21	35	60.0%
25,001 - 30,000	17	25	68.0%
30,001 - 35,000	12	19	63.2%
35,001 - 40,000	6	9	66.7%
40,001 - 45,000	2	3	66.7%
>45,000	4	5	80.0%

Table 3: XR Course Offerings by University Size

Arts, Design, and Media make up 18% of XR offerings, with Visual and Performing Arts, Emerging Media Arts, Graphic Design, Game Design, and Interior Design among the primary programs. Humanities and Social Sciences also represent 18%, including departments or programs such as Sociology, Anthropology, Literature and History, and Psychology. Communication and Journalism account for 6%, often through programs focused on teaching and learning, instructional design, and educational psychology. Natural and Life Sciences contribute 3%, with courses linked to Geology, Biology, and Archaeology. Finally, Business and Management, encompassing Innovation and Business Technology, Information Management, and Marketing, make up 3% of the XR courses.

To further understand the focus areas of current XR courses, we categorized them based on the specific thematic content and educational objectives, rather than by scientific field or department as discussed previously. Therefore, Table 4 presents thematic categories along with the number of courses assigned to each.

Course Topic Category	Courses in Category
Arts, Humanities & Culture	177
XR Development & Programming	137
Communication, Journalism	131
XR Design & User Experience	112
Game Design & Interactive Media	99
XR Technological Foundations	80
Specialized & Applied Topics	56
Architecture, Engineering, Built	50
Education, Training, Pedagogy	44
Science & Research Applications	27
Grand Total	**913**

Table 4: XR Course Topic Categories

The largest category is Arts, Humanities & Culture (177 courses), reflecting a strong presence of XR in creative and cultural studies, followed by XR Development & Programming (137) and Communication and Journalism (131), reflecting strong emphasis on creative, technical, and media-oriented applications. XR Design & User Experience comprises 112 courses, Game Design & Interactive Media 99, and XR Technological Foundations 80, covering core principles and highlighting the importance of interactive and user-centered aspects. Other notable areas include Specialized & Applied Topics (56) and Architecture, Engineering, and Built Environment (50).

The category of Specialized and Applied Topics reveals how XR technologies are deeply integrated into broader ecosystems of emerging technologies. In the "Artificial Intelligence for Enterprise Program," students explore concepts like image and video recognition, natural language processing, and robotics process automation, with AR used to simulate AI workflows in business environments. Likewise, "Cyber Science Fundamentals" introduces students to a suite of technologies, including quantum computing, blockchain, and AI, to examine how they collectively shape the future of cybersecurity and data systems. Within this context, XR is explored as a medium for visualizing complex cyber systems, simulating attack scenarios, and fostering experiential understanding of abstract digital infrastructures. The course "Emerging Technologies in Digital Transformation" positions XR alongside cloud computing, IoT, drones, and digital assets, emphasizing its role in transforming business operations and user engagement. In the engineering domain, "Manufacturing Automation" explores XR through virtual environments and simulation in shared production scenarios, industrial design, prototyping, diagnostics and smart maintenance. These courses demonstrate how XR does not stand alone but interacts with a constellation of technologies, enabling immersive, applied learning that prepares students for innovation in the evolving tech landscape. In total, 913 courses were categorized across the introduced thematic areas, demonstrating the breadth and diversity of XR education offerings.

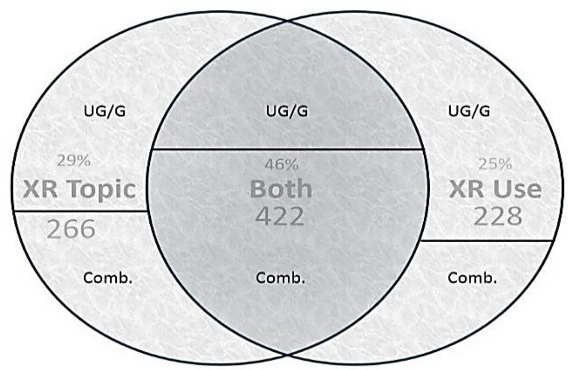


Figure 4: XR Course Distribution by Type

Competency Domain	Key Learning Areas	Representative Learning Objectives
Technical Competency	1. Foundation Knowledge: System architecture, 3D modeling, data formats.	"Describe elements of typical system architecture for virtual environments (VEs)" (Naval Postgraduate School)
	2. Hardware Integration: Tracking technologies, haptic devices, display systems.	"Determine if a task may benefit from a 3DOF or 6DOF haptic device" (Northwestern State University)
	3. Systems Integration: AI integration, data strategies, converged communications.	"Learn best practice data strategies, VR/MR/AR and robotic solutions" (Virginia Tech)
Human-Centered Design and UX	1. Human factors: Perceptual modalities, sensory integration, cognition.	"Describe processes supporting human sensing of visual, auditory, haptic stimuli" (Northwestern State University)
	2. Design Documentation: User tasks, interface techniques, spatial interaction.	"Use design documentation, object-oriented software design, project management, and design iterations through user feedback" (University of Tennessee at Martin)
	3. Application Design: Use case analysis, technology matching, domain specificity.	"Identify applications benefiting from different visual displays" (Northwestern State University)
Ethical and Safety Considerations	1. Health and Safety: Cybersickness, simulator sickness, safety protocols.	"Understand health, social, privacy, and security issues and human factors that influence usability" (Augusta University)
	2. Professional ethics: Research ethics, human subjects, professional conduct	"Describe elements of professional ethics in research domain" (Northwestern State University)
Inter-disciplinary Integration & Research Methods	1. Domain application: Industry contexts, military applications, interdisciplinary use.	"Use diverse interdisciplinary approaches to explore the intersection between society, culture, technology, and digital connectivity" (Norfolk State University)
	2. Research method: User studies, usability evaluation, empirical research	"Design and execute studies evaluating usability of Virtual Environments" (Naval Postgraduate School)

Table 5: XR Learning Areas and Objectives

Regarding the nature of XR courses offered, the analysis identifies three distinct types. A total of 266 courses explicitly focus on XR, providing foundational knowledge and theory, while 228 incorporate XR technologies within broader disciplinary contexts. Notably, 422 courses combine both aspects, covering XR topics in depth while also providing hands-on experience with XR devices and applications, especially in courses designed for both undergraduate and graduate student (indicated as “comb.” for combined courses, see Figure 4). This blend of theoretical and practical engagement highlights the diverse approaches universities take to integrating XR education, balancing conceptual understanding with experiential learning.

Finally, the examination of detailed learning objectives from XR-related courses reveals how institutions are structuring competencies across technical, creative, ethical, and interdisciplinary dimensions. This analysis conducted in Table 5 draws from courses with comprehensive learning outcomes to illustrate the specificities of XR education and provide concrete examples for curriculum designers seeking to develop or refine their programs. The developed competency framework provides curriculum designers with a structured approach to developing comprehensive XR education programs that balance technical depth, creative application, and ethical responsibility while maintaining relevance to evolving industry demands and interdisciplinary collaboration requirements.

5. CONCLUSION

This study examines the current landscape of XR course offerings across AACSB-accredited universities in the US. Among 547 universities reviewed, larger institutions are more likely to offer XR-related courses, reflecting growing institutional capacity and student demand. The courses identified span diverse scientific fields, with a notable concentration in Technology and Engineering programs, alongside significant representation from Arts, Design, Media, Humanities, and Social Sciences. In contrast, only a small share of courses are located within Business Programs, despite XR's growing role in sales, operational workflows, and behavioral data analytics. Our findings highlight how universities are beginning to address the multidimensional nature of XR education by balancing technical, creative, and applied competencies, although such integration through labs and cross-program initiatives remains rare. XR courses frequently emphasize foundational knowledge and development skills, but also incorporate design, user experience, and domain-specific applications such as co-production and maintenance. This distribution reflects efforts to cultivate the varied skill sets required for XR practitioners, from programming and system design to critical thinking and problem-solving.

Contributions

Our study confirms that XR education spans a wide array of disciplines, yet meaningful cross-disciplinary collaboration remains limited. For example, while AR applications in medical training benefit from architectural insights (e.g., spatial design, lighting), and VR simulations of archaeological sites require narrative techniques from journalism, such integration is rarely reflected in curriculum design. By mapping the departmental distribution of XR courses, this study exposes both the promise and the fragmentation of XR education across academic silos.

This fragmentation reflects a broader institutional challenge that our study brings into focus. Higher education institutions tend to be highly siloed, with campus units maintaining their own priorities, cultures, and budgets. Launching collaborations proves difficult because it requires cutting across existing organizational structures and sometimes developing entirely new ones. Nonetheless, leading institutions have begun to overcome these barriers through coordinated initiatives. For instance, the University of Michigan's XR Initiative supports over 40 projects across 17 of the university's 19 schools and colleges, while Yale University's Blended Reality Applied Research Project brings together faculty from engineering, arts, humanities, and library sciences.

Despite these successes, significant barriers persist including the need to align diverse departmental priorities, establish sustainable funding mechanisms, and address the lack of shared technical expertise across departments. Many XR investments remain isolated pilot projects that – while demonstrating potential – are unlikely to scale due to lack of sustainability and collaboration.

The significant number of courses combining theoretical and hands-on learning indicates a trend toward experiential education, essential for mastering immersive technologies. This pedagogical shift reflects recognition that XR enables experiential learning, where learners can practice skills in a safe, controlled environment. Medical students can rehearse clinical procedures through immersive simulations, while simulation-based learning has proven particularly effective in healthcare, engineering, and emergency response fields.

Finally, our analysis shows how XR technologies increasingly support experiential learning across disciplines. Courses in the dataset leverage tools such as 360° video to grant access to environments otherwise unreachable – such as enabling students to explore coral reefs without leaving the classroom. Others incorporate immersive storytelling and gamified elements to deepen engagement, transforming history lessons into interactive encounters with historical figures. These curricular choices reflect a shift from novelty-driven adoption toward pedagogically grounded XR use. As such, they underscore XR's potential not only to enhance learning outcomes, but also to promote inclusion, accessibility, and deeper cognitive engagement in higher education.

Limitations and Future Research

While this study provides a broad overview of XR course offerings and emerging patterns in curricular design, it also carries limitations. The data collection and analysis process were conducted manually, highlighting the need for automated methods to retrieve course information and apply large-scale AI-driven classification and analysis. Such tools would improve scalability and consistency across future studies. Additionally, our scope was limited to AACSB-accredited schools, which future work could extend to a broader range of institutions.

Future research should explore how academic programs translate XR-related competencies into curricula aligned with evolving industry demands, identifying gaps and areas for pedagogical innovation. A key challenge lies in bridging the disconnect between the technological sophistication of XR tools and the pedagogical principles necessary for effective learning – particularly as many XR developers may lack educational design expertise.

Despite promising examples, many XR initiatives remain isolated pilot projects that face significant barriers to scalability. Resource constraints continue to limit adoption, from hardware performance issues and user discomfort to accessibility challenges and integration difficulties within existing educational infrastructures. Developing high-quality XR content also requires specialized technical and instructional expertise that remains scarce in many institutions.

Periodic reassessment will be essential to track how XR education evolves in tandem with advances in technology and labor market expectations. Critical areas for future inquiry include creating more affordable and ergonomic XR devices, establishing effective instructional models tailored to immersive media, and conducting longitudinal research to evaluate XR's long-term impact on learning, employability, and professional development.

Ultimately, universities have a central role in shaping the XR talent pipeline. Doing so requires sustained cross-disciplinary collaboration, long-term investment, and a research agenda that addresses both current limitations and future opportunities in immersive education.

6. REFERENCES

Burke, D., Crompton, H., & Nickel, C. (2025). The Use of Extended Reality (XR) in Higher Education: A Systematic Review. *TechTrends.* Advance online publication. https://doi.org/10.1007/s11528-025-01092-y

CCAM (2025). Center for Collaborative Arts and Media. https://ccam.yale.edu/projects/research-blended-reality

Ceccucci, W., Jones, K., Toskin, K., & Leonard, L. (2020). Undergraduate Business Analytics and the Overlap with Information Systems Programs. *Information Systems Education Journal*, *18*(4), 22–32.

Dincelli, E., Warkentin, M., Lowry, P., Suh, A. (2026). Introduction to the Minitrack on Metaverse for Work and Play. *59th Hawaii International Conference on System Sciences*.

Düdder, B., Fomin, V., Gürpinar, T., Henke, M., Iqbal, M., Janavičienė, V., Matulevičius, R., Straub, N., & Wu, H. (2021). Interdisciplinary Blockchain Education: Utilizing Blockchain Technology From Various Perspectives. *Frontiers in Blockchain*, *3*(1), Article 578022. https://doi.org/10.3389/fbloc.2020.578022

El Dandachi, I., El Nemar, S., & El-Chaarani, H. (2023). XR and the Metaverse: New Opportunities in Education. In R. El Khoury & B. Alareeni (Eds.), *Contributions to Environmental Sciences & Innovative Business Technology. How the Metaverse Will Reshape Business and Sustainability* (pp. 51–61). Springer Nature Singapore. https://doi.org/10.1007/978-981-99-5126-0_6

Georgieva, M., Nelson, J., LaFosse, R., & Contis, D. (2024). XR in Higher Education: Adoption, Considerations, and Recommendations. *Educause Review*.

Gürpinar, T., Quadri, S., Jia, S., Sacouto, L., Lang, G. (2025a). Virtual, Augmented, and Extended Reality in Higher Education: Trends, Applications, Impacts. ISCAP Conference.

Gürpinar, T., Verma, A., Leonard, L., Jones, K., & Ceccucci, W. (2025b). Blockchain Education: Evaluating Programs, Curricula, and Integration with Emerging Technologies in Higher Education. *Information Systems Education Journal*, *23*(5), 20–29. https://doi.org/10.62273/WNRJ3445

Huang, B., & Hew, K. F. (2021). Using Gamification to Design Courses: Lessons Learned in a Three-year Design-based Study.

Educational Technology & Society, *24*(1), 44–63.

Karamitsos, I., User, E., Al-Qadi, S., Attias, S., Belling, A., Grant, J., Gürpinar, T., & et al. (2024). Exploring the Future of the Metaverse Through - A Holistic Introduction of Virtual Worlds for Policy Makers. *INATBA Report*.

Khlaif, Z. N., Mousa, A., & Sanmugam, M. (2024). Immersive Extended Reality (XR) Technology in Engineering Education: Opportunities and Challenges. *Technology, Knowledge and Learning*, *29*(2), 803–826. https://doi.org/10.1007/s10758-023-09719-w

King, A., Gürpinar, T. (2025). Industrial Metaverse und Extended Reality (XR): Anwendungen in Unternehmensnetzwerken und intelligenter Instandhaltung. Instand-haltungsforum 2025.

Küpeli, O., & Gürpinar, T. (2023). Towards a definition of the industrial metaverse applied in context of the blockchain and web3 ecosystem. *Blockchain and Cryptocurrency Conference*.

Mentzer, K., Frydenberg, M., & Li, S. (2025). Exploring VR-Enhanced Learning in Business Education: A Multi-Site Study. *Information Systems Education Journal*, *23*(5), 30–43. https://doi.org/10.62273/WOFR1838

Radianti, J., Majchrzak, T. A., Fromm, J., & Wohlgenannt, I. (2020). A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, *147*, 103778. https://doi.org/10.1016/j.compedu.2019.103778

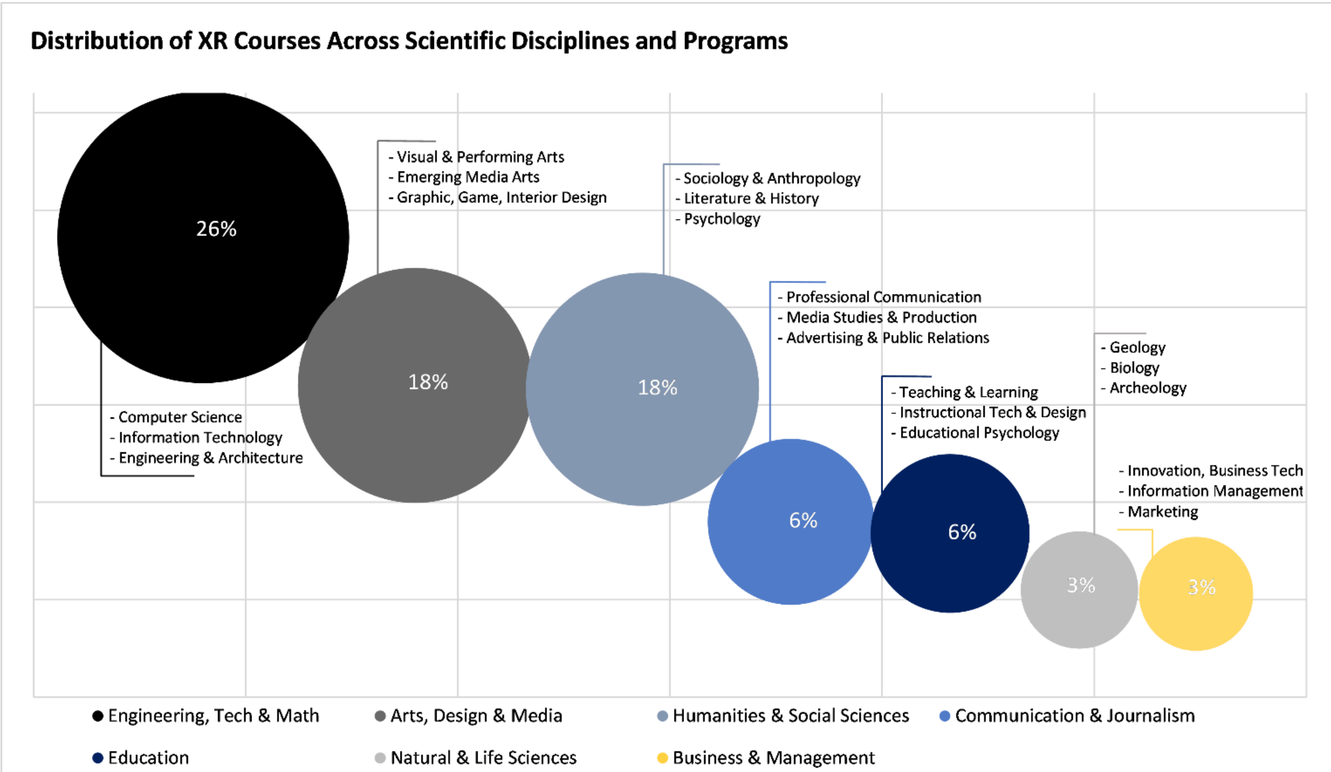
Tusher, H. M., Mallam, S., & Nazir, S. (2024). A Systematic Review of Virtual Reality Features for Skill Training. Technology, Knowledge and Learning, 29(2), 843–878. https://doi.org/10.1007/s10758-023-09713-2

Verma, A., Purohit, P., Thornton, T., & Lamsal, K. (2021). An examination of skill requirements for Augmented Reality and Virtual Reality job advertisements. https://doi.org/10.48550/arXiv.2108.04946

Yang, S. C., & Wen, B. (2017). Toward a cybersecurity curriculum model for undergraduate business schools: A survey of AACSB-accredited institutions in the United States. Journal of Education for Business, 92(1), 1–8. https://doi.org/10.1080/08832323.2016.1261790

Zhou, D., Globa, A., & McLaughlan, R. (2025). A Systematic Review of XR, Spatial, and Physical Interaction in Remote Communication for Emotional Well-Being in Older Adults. In Q. Gao & J. Zhou (Eds.), Lecture Notes in Computer Science. Human Aspects of IT for the Aged Population (Vol. 15809, pp. 427–448). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-92707-2_28

Appendix A. Distribution of XR Courses Across Scientific Disciplines and Programs



Teaching Case

Our Interconnected, Resilient, Modern World ... is Still Painfully and Remarkably Brittle: A Case Study of the Risks of Technology Failures

Paul Witman
witman@ieee.org

Jim Prior
jr2dg.consulting@gmail.com

California Lutheran University
Thousand Oaks, CA USA

Hook

Systems of all kinds are pervasive in everyday life. And when systems fail, often due to a lack of resilience, many aspects of daily life, including life-critical activities, may be impacted or destroyed. As future technologists and business leaders, understanding the importance and mechanisms of resilience is critical to ensuring that your organization's systems are appropriately resilient, and that trade-offs are appropriately considered.

Abstract

8.5 million Windows PCs failed to restart in July of 2024. CrowdStrike, an IT security company, delivered updates with a faulty configuration file to millions of Windows PCs around the world. The failure disrupted or halted the global operations of airlines, banks, medical facilities, and much more. Many PCs were unable to reboot without direct human intervention. This resulted in delays that impacted the recovery time for all types of businesses, as well as their employees, supply chains, and consumers. The paper also provides several other failure scenarios for student analysis, including cyber-attacks on health care and government, and a broader system failure in air traffic control, as well as a detailed background in the underlying principles of technological resilience. This teaching case for undergraduate and graduate information systems courses encourages students to examine key questions for each event: What went wrong, what was the root cause, and why didn't we see this failure coming? What can we learn from each case that might help technology leaders and their business counterparts to further mitigate the risk of any similar events?

Keywords: Teaching Case, System Resilience, Risk Management, Complexity, Testing, Organizational Preparedness, Critical Infrastructure, Business Continuity

Recommended Citation: Witman, P.D., Prior, J., (2026). Our Interconnected, Resilient, Modern World Is Still Painfully and Remarkably Brittle: A Case Study of the Risks of Technology Failures. *Information Systems Education Journal*, v24(n2) pp 16-26. DOI: https://doi.org/10.62273/PSAS2989

Our Interconnected, Resilient, Modern World ... is Still Painfully and Remarkably Brittle

Paul Witman and Jim Prior

1. INTRODUCTION AND OPENING STORY

The trigger for the case
On July 19, 2024, Paul (one of the authors) was traveling by plane to visit family. On arrival at the airport, he went to check the monitor near the entrance to find his flight's gate information and departure status. Rather than the usual list of flights, what he found appeared to be the Microsoft™ "blue screen of death."

He had learned of this likely outcome while en route to the airport. Fortunately, his airline's app was still able to provide status and schedule updates, and there were relatively small impacts on his flight. Other travelers were not so fortunate – other airlines cancelled thousands of flights in total, over several days, and had to scramble to recover their technology operations. They also had to communicate with delayed and disrupted passengers, track luggage, reschedule crews, and service and maintain aircraft, much as Southwest had to do during its earlier winter crisis (Witman et al., 2024). Many of those functions were dependent on Microsoft technology and were not immediately recovered.

Paul was fortunate that his travels were relatively unimpacted, the impact on the broader traveling public was highly visible turmoil, with passengers stuck in their departure or an intermediate airport, airlines unable to accurately predict "return to normal," and all of the fallout that comes from that.

Learning objectives
After working through this case study and its exercises, students should be able to:

- Define and describe system resilience and brittleness
- Identify potential causes of system brittleness and potential solutions or mitigations.
- Identify and analyze trade-offs of increasing resilience
- Identify and analyze costs and benefits of resilience testing

Case application, organization
This case has been used in introductory information systems courses for both graduate and undergraduate courses. Course applications focus on the role of system resilience in many and varied ways. Shocks to a business can come from technological failures, pricing changes, regulatory issues, and more (Rose, 2025). We have used technically-oriented case vignettes to illustrate various resiliency and brittleness issues, and have provided some broader follow-up research questions.

The remainder of this paper goes on to discuss background information on the technical concepts related to system resilience in Section 2, followed by a number of sample scenarios where resilience played a role in section 3. Each of those scenarios includes discussion questions for use by small groups or whole-class discussions or homework. Following the scenarios, Section 4 offers additional research questions for potential follow-on assignments, and a summary and conclusion follow in Section 5.

2. TECHNICAL CONCEPTS AND TERMS

There are a number of technical concepts and elements that apply to various aspects of these case scenarios, and we identify and describe them briefly here. Key concepts include system resilience, and its converse, system brittleness. Underlying those concepts are elements that affect systems, sometimes making them more or less resilient. These elements include but are not limited to system complexity, challenges that impede resilience testing, automatic updates, and pervasive client software.

System Resilience was defined by Google AI as the "ability to maintain its essential functions and capabilities when facing disturbances, disruptions, or adverse events." This definition clearly leaves a lot of room for flexibility, including a need to define which functions are "essential", and whether doing those functions in a degraded way (more slowly, at higher cost, etc.) might also be treated as a failure. Fundamentally, we believe that the concept of system resilience is one that must be defined for each system, and agreed to by each stakeholder in a system's functions.

System Brittleness (or Fragility) is to some extent the polar opposite of resilience. It speaks to a state of being vulnerable to one or more kinds of disruptions or adversity, both alone and simultaneously. Often, adversity and disruptions come in bursts - weather problems in the travel industry will force schedule changes, which will affect staffing, and will also cause bursts of customer service activity to re-plan travels (Witman et al., 2024).

Complexity

Baccarini (1996) defines complexity in projects as "consisting of many varied interrelated parts" and distinguishes between two critical types that directly impact system resilience. Organizational complexity relates to interconnections among organizational units, stakeholders, and procedures – the human and procedural elements that must coordinate for system success. Technological complexity concerns the number and diversity of technologies and their interdependencies. This dual framework remains highly relevant for understanding modern information systems, where both organizational factors and technical architectural decisions contribute to system vulnerability and resilience challenges.

The complexity challenge extends beyond Baccarini's framework, as modern systems operate within intricate webs of external dependencies that multiply complexity exponentially. Baskerville et al. (2018) note that complexity arises from "the number and diversity of components and the nature of interconnections among them, leading to emergent and often unpredictable behaviors." This complexity is amplified by dependencies on external providers outside the system owner's direct control: power grid operators, telecommunications providers, cloud computing platforms, and facilities maintenance contractors, among others. These effects are further exacerbated by process complexity, the intricate nature of workflows, handoffs, poor process management, and the like (Shi et al., 2024).

Each external dependency introduces its own complexity and creates additional interdependencies and potential failure points. This multi-layered complexity – combining internal organizational and technological factors with external provider dependencies – creates emergent behaviors that can be difficult to predict or control, fundamentally challenging traditional approaches to system resilience planning.

Challenges of testing resilience

Active, "invasive" testing on running systems presents significant operational and strategic challenges for organizations seeking to validate their resilience capabilities. Fault injection testing involves "the deliberate introduction of errors and faults to a system to validate and harden its stability and reliability" with the goal of improving system design for resiliency under intermittent failure conditions. However, such invasive testing on production systems carries inherent risks – testing that introduces various disruptions may induce behaviors that degrade or destroy system functionality, necessitating repair and recovery operations that can be costly and disruptive. Production fault injection should be considered one of many approaches used to gain confidence in the safety and resiliency of a system, similar to unit testing and code review, though it is limited in which surprising events it can prevent. To mitigate these risks, system owners often schedule "planned outages" to allow for controlled testing scenarios, but tolerance for even planned outages decreases significantly as system criticality increases.

Organizations like the University of Wisconsin Hospital spend weeks preparing and testing before planned instances of downtime to help things run as quickly and smoothly as possible. However, they recognize that the balance between resilience validation and operational continuity becomes increasingly delicate for mission-critical systems where downtime can have severe consequences for users, revenue, public safety, and more (Otkhozoria et al., 2025).

Cloud Computing and other shared services

Cloud Computing generally refers to contracting with a vendor to operate some portion of an organization's technological needs in a remote data center, sharing that infrastructure with other clients. The cloud provider handles infrastructure operations including power, cooling, monitoring, and security. Such providers are often selected based on their ability to operate at a higher level of quality than clients could otherwise achieve. While this upgrade in operational capability is usually real, it does not eliminate the risk inherent in running complex technological systems, and cloud providers can still experience failures in both primary and backup operations (Alozie et al., 2024).

Pervasive client software

"Pervasive client software" refers to software that must be installed across many or all computing devices in an organization. A healthcare industry CIO used this term to describe software deployed on everything from desktop computers to embedded systems like airport display controllers or ATM operating systems. The critical risk with pervasive software is that a single software failure can simultaneously disable multiple systems across the organization. This creates a "single point of failure" scenario, as demonstrated during the July 2024 CrowdStrike incident when a faulty update disrupted operations globally.

Remote and automated updates

Enterprise IT teams managing thousands of devices face a complex challenge balancing two competing risks: the risk of deploying a faulty update that could crash multiple systems simultaneously, versus the risk of delaying critical security patches that leave the organization vulnerable to cyberattacks. This creates a fundamental trade-off in system resilience (Usman & Asplund, 2025).

For example, Carol, the CIO of a large healthcare organization, worked with her Chief Information Security Officer to develop an update deployment policy. Most routine updates underwent thorough testing before organization-wide deployment. However, for a select group of highly trusted vendors, updates marked as "critical and urgent" were deployed immediately to the most at-risk systems. This policy aimed to minimize vulnerability windows but increased exposure to potential update failures.

Root Cause Analysis

Root cause analysis refers to the process of attempting to discern the most fundamental, or root, cause of a problem. It is often conducted by repeatedly asking "why" after identifying a partial cause. The goal is to focus on solving the fundamental cause rather than addressing symptoms (Rooney & Heuvel, 2004).

For example, analyzing system failures: Why did the system experience widespread service disruptions? Because an external shock overwhelmed the system's ability to manage operational complexity. Why? Because the system was not designed to deal with such disruption levels. Why? … A related Why? might be a question of why the organization allowed a system in production without sufficient resilience testing. Often, the answer to that is some combination of pressures to reduce costs and/or time to market (Rooney & Heuvel, 2004).

This analysis reveals the real cause: lack of investment in resilience-building measures left the system with dangerous brittleness and vulnerability to cascading failures.

Risk management

Risk refers to the probability that a certain type of event will occur, while impact is the magnitude of damage that might result. There are several fundamental approaches to managing risk (Ahmed, 2017).

- **Avoidance**: Avoiding activities that create unacceptable risk
- **Transfer**: Moving risk to another party (insurance, contracts)
- **Reduction**: Taking steps to reduce risk to acceptable levels
- **Acceptance**: Accepting remaining "residual risk" after mitigation

Sometimes, organizations take an approach that externalizes some part of their risk. This means that they (deliberately or inadvertently) transfer some of their risk to external parties - customers, vendors, employees, supply chain partners, etc. In brittle systems, organizational choices for resilience investment essentially transfer some of the impact of these risks to various other parties, also known as creating a trade-off.

3. SCENARIOS: SYSTEM RESILIENCE FAILURES

Scenario 1 CrowdStrike scenario completion

Getting through a US airport is always an engaging experience, and one guided by information made available to the travelers and to security and other staff. While going through the airport was relatively easy for Paul (his airline's systems were generally OK, and the airline’s app on his phone provided correct gate and flight information), other travelers were not as fortunate.

Security protocols limit the secure area of the airport to those with tickets for active flights. With many flights being canceled, it was difficult for security staff to be sure who they could properly admit. Customer service staff from all airlines, both at the airport and in call centers, were working hard to help passengers re-arrange their travel plans. However, the scope of the problems was so broad as to create significant queues of people that were impacted by the technology issues. These customers added greatly to the delay faced by customers dealing with “normal” travel challenges, forcing airlines to rapidly ramp up on-site and call-center staffing to help rearrange travel plans.

Overall, the airline industry alone suffered many billions of dollars of damages due to the CrowdStrike issue. One airline, Delta, reported its own costs at over USD500 million, with impacts on revenues, flight crews, manual rescheduling, and compensation to affected customers (Cerullo, 2024).

In brief, the underlying technical problem that caused the outage to occur was a faulty software update released by CrowdStrike for its cybersecurity software that caused 8.5 million Windows computers to crash. CrowdStrike is a cybersecurity company whose software runs on millions of computers worldwide to protect them from cyber threats. The faulty configuration update caused an error in the computer's memory, resulting in the infamous "Blue Screen of Death" that made affected computers unusable.

The impact was massive because CrowdStrike's software operates at a very deep level within computer systems, and when it crashed, it took the entire computer down with it. Airlines, healthcare systems, media companies, and organizations all over the world were unable to operate properly. The computers affected were unable to recover without direct manual intervention, meaning IT teams had to physically access each machine to fix the problem. While CrowdStrike's CEO confirmed this was not a cyberattack and deployed a fix, the incident highlighted how a single software error from one company could bring down critical infrastructure worldwide (Kerner, 2024).

Questions:

- Of the technical factors around the topic of system resilience, documented in Section 2, which seems most applicable to this situation?
- Given that strong competitors often become ubiquitous within many technology operations (e.g., iOS, Windows, CrowdStrike, etc.), what can an organization do to protect themselves from this sort of incident?
- Even if a company does protect itself from such risks, what can they do to protect their supply chain against risks among their vendors?
- What other questions or observations can you make about this scenario as it relates to resilience, risk management, information, and information systems?

Scenario 2 Ninety seconds of terror

Air traffic controllers have very high stress jobs. With little room for error, they simultaneously coordinate the movements of multiple aircrafts carrying thousands of people. They do so for extended periods of time, making critical decisions with often incomplete and/or rapidly changing information. They communicate via radio directly with pilots and they monitor aircraft via radar.

Picture an air traffic controller in a radio conversation with a pilot bringing a plane in for landing at one moment. And then, in the next moment, the controller loses radio contact with the pilot, and their radar screen goes blank.

This happened to the air traffic controllers at the Philadelphia Pennsylvania Terminal Radar Approach Control (TRACON) center, which is responsible for separating and sequencing planes in and out of Newark Liberty International Airport in New Jersey. The outage – no radio, no radar – lasted for ninety seconds.

Fortunately, there were no serious incidents during the blackout. Pilots have procedures they follow under circumstances like these, and once radio communications were restored, the air traffic controllers and pilots quickly recovered, despite the stress involved.

But the follow-on impact was profound. A number of the controllers on duty during the outage took time off due to the stress and trauma of the event. The Newark airport was shut down for two hours while the Federal Aviation Administration (FAA) worked on the issue. Over 65 flights were diverted to other airports, 160 flights were canceled and 424 delayed. Those problems continued into the next day with more than 370 flights delayed and more than 500 cancelled (Martínez & Dumas, 2025). Additional outages at Newark prompted the FAA to reduce the number of departures and arrivals per hour by 50% as compared to peak times in order to maintain safety and reduce flight delays.

Two issues were at the root of the problem:

- A telecommunication line that transmits data and audio to air traffic controllers failed, and
- A radar feed that transmits data from an FAA facility to the Philadelphia TRACON and finally to the Newark airport also failed.

These issues were very likely tied to a contentious move of Newark air traffic controllers from Westbury, N.Y. to Philadelphia in 2024. Many controllers quit rather than relocate. But with the move there were no redundant data connections built between the radar processing center in New York and the Philadelphia facility where the controllers were moved. Nor was a backup system put in place (Sherman, 2025).

On top of all that, the Newark air traffic controllers faced long-standing, FAA-wide challenges:

- Outdated technology:
    - Telecommunication links built on older

copper wire.
  - o Air traffic controllers using paper strips to track flights, manage traffic flow, and communicate between themselves.
  - o Floppy disks used to update software and transfer data.
  - o Computers running Windows 95.
- Understaffing
  - o The FAA says that nationwide they are around 3,500 air traffic controllers short of staffing targets (Federal Aviation Administration, 2025).
  - o The demanding nature of the air traffic control profession leads to early retirements and additional turnover. Filling those empty seats requires candidates with unique skills. According to the air traffic controllers' union, only 50% of trainees complete the training and achieve full certification.

In response to the outages, the FAA quickly outlined several measures to strengthen air traffic control at Newark Liberty International Airport. These included:

- o Installing three additional high-capacity telecommunications links to the Philadelphia TRACON to increase reliability, speed, and system redundancy.
- o Upgrading existing copper connections to fiber-optic lines, which support faster data transfer, greater bandwidth, longer transmission distances, and improved security.
- o Introducing a temporary backup system at the Philadelphia TRACON to ensure operational continuity (Sherman, 2025).

Furthermore, the FAA continues their decades-long effort to implement solutions for these problems across the U.S., but it remains slow-going. For instance, technology upgrades are challenging in circumstances where existing information systems must be up and running 24/7, which is paramount given the importance of the air traffic control system.

<u>Questions:</u>

- What does this scenario reveal about the resilience (or lack thereof) of the U.S. air traffic control system?
- What are the risks of relying on outdated technology in critical infrastructure like air traffic control?
- Why do you think only 50% of trainees successfully complete air traffic control certification? What might be done to improve that rate?
- If you were tasked with reimagining the air traffic control system for the next 5 years, what would be your top three priorities – and why?
- What other questions or observations can you make about this scenario as it relates to resilience, risk management, information, and information systems?

Scenario 3 Ransomware attack on Baltimore

One Tuesday morning in May 2019, Maria Santos arrived at her real estate office in Baltimore with excitement bubbling inside her. After months of searching, her clients, the Johnsons, had finally found their dream home – a charming rowhouse in Federal Hill. The closing was scheduled for that afternoon, and all that remained was the routine verification of city liens and water bills. But when Maria tried to access the city's online system to pull the necessary documents, she was met with an ominous message. The city's servers were down. What she did not know yet was that Baltimore had become the latest victim of a devastating ransomware attack – one where hackers had demanded $76,000 in ransom and would hold the city's computer systems hostage for over five weeks.

The attack had brought the real estate market to a grinding halt as property transfers could not be completed digitally, and the city was unable to issue lien certificates or generate water bills. Maria spent hours on the phone with increasingly frustrated city employees who could only tell her that their systems were compromised, and that they had no timeline for recovery. The Johnsons' closing was postponed indefinitely, along with hundreds of other real estate transactions across the city. For weeks, Baltimore's housing market essentially froze as buyers, sellers, and real estate professionals found themselves caught in digital limbo.

Meanwhile, across town, David Kim was dealing with his own crisis. As a city employee working in the health department, he arrived at work to find that not only could he not access his email, but the entire network was locked down. By day 36 of the attack, only 70 percent of city employees had regained access to their email accounts, with recovery efforts aiming to restore 95 percent of employee access by week's end. David's team was responsible for sending out health alerts and coordinating with local hospitals, but with their systems compromised, they were forced to resort to personal phones and handwritten notes. Critical public health communications that should have taken minutes to distribute now took hours, creating potential risks for the city's most

vulnerable residents (Mathews, 2025).
The attack's effects rippled through Baltimore's daily life for months. As Deputy Chief of Staff Sheryl Goldstein warned residents, "I do not expect June bills to go out," water bills for June were delayed indefinitely, creating a backlog of charges that would eventually hit residents all at once. Families who budgeted for monthly water payments suddenly faced the uncertainty of not knowing their usage or owing amounts. The city's inability to process online payments meant residents had to find alternative ways to pay fines and taxes, though parking tickets could eventually be looked up in person. The city ultimately invested more than $18 million in recovery efforts – far exceeding the hackers' original $76,000 ransom demand that Mayor Young had refused to pay. But the true cost – measured in disrupted lives, delayed dreams, and shaken confidence in digital infrastructure – was immeasurable. The attack served as a stark reminder of how deeply intertwined modern urban life had become with digital systems that could vanish in an instant (Gallagher, 2019).

Questions:

- What are the pros and cons of paying a ransom to a ransomware attacker?
- How can systems be protected from ransomware attacks?
- Baltimore operated most of the systems in this scenario in its own data centers. Could they have reduced their risk by running their systems in the cloud instead? Does moving to the cloud have risk trade-offs?
- What other questions or observations can you make about this scenario as it relates to resilience, risk management, information, and information systems?

Scenario 4 Change Healthcare Cyber Attack
A series of incidents related to patients, providers, and hospitals impacted stakeholders in Change Healthcare, a health insurance company based in Nashville, TN, USA.

Alan regularly went to his Naperville, Illinois, pharmacy to pick up medications for his congestive heart failure and diabetes, which are covered by Medicaid. At his most recent visit, his pharmacist told him that he would have to pay out of pocket, that Medicaid would not cover the cost. Unable to do so, Alan ended up in a hospital (De Mar, 2024).

For a month now, Margaret, a dermatologist in California, has been unable to submit insurance claims to get paid for the services her private practice has provided. She's been considering borrowing money to pay rent and her staff (Johnson & Ibarra, 2024).

A cancer clinic in Oregon has been unable to bill for its $500,000 to $1 million daily chemotherapy costs. Mel, the company's Chief Financial Officer, has managed so far out of cash flow, but with reserves running low, she worries they may not have the money to pay for labor and keep their doors open (Steenhuysen, 2024).

These three unfortunate circumstances, along with countless others, were due to the largest healthcare data breach in U.S. history.

Change Healthcare, a subsidiary of UnitedHealth Group (UHG), based in Nashville, Tennessee, provides billing and data services for the healthcare industry, processing 15 billion transactions a year. On February 21st, 2024, the company discovered that their information systems were under cyber-attack. In order to protect their partners and patients, they quickly disconnected their systems which were relied upon by hospitals, pharmacies, and doctors' offices.

The cyber-attack actually began on February 12th, when a hacker gained access to Change Healthcare's information system network by logging into a remote access service that lacked multi factor authentication using compromised credentials (see vulnerabilities and their remediations in Table 1, below). Undetected over a ten-day period, the hacker was able to create privileged administrative accounts, remove vast amounts of sensitive data, and encrypt files on Change Healthcare's systems via ransomware - malicious software designed to block access to a computer system until a ransom payment is made. The attack was only detected when system files were encrypted, preventing access.

Five days after Change Healthcare discovered the attack, a ransomware group named ALPHV/Blackcat claimed responsibility for the attack, said that 6TB of data was stolen, and demanded a $22 million ransom to prevent the publication of the stolen data. Change Healthcare ultimately paid the ransom, but their payment did not secure the stolen data due to a scam by the ALPHV/Blackcat group. As of June 2025, there have been partial leaks of the stolen data - screenshots and documents - but there have been no indications of complete data sets being disclosed.

Change Healthcare restored operations of their electronic payments platform and reinstated 99% of its pharmacy network services in mid-March (Hyperproof Team, 2024). And while many of their services recovered within a few months, Change Healthcare didn't declare that their clearinghouse services had been fully restored until November, nine months after the attack. Furthermore, in October, Change Healthcare estimated that the breach involved the data of approximately 100 million people, and amid ongoing analysis revised that number to 190 million in January of 2025 - eleven months after the attack began. The stolen data included medical records, insurance records, dental records, payments/claims information, and patients' Personally Identifiable Information (PII) - phone numbers, addresses, social security numbers, driver's license numbers, and email addresses (Gatlan, 2024).

The financial impacts of the attack were staggering. UnitedHealth Group estimated the overall cost of the Change Healthcare ransomware attack at $2.87 billion in 2024. According to the American Medical Association, the attack resulted in a $100 million impact to the healthcare industry per day, as claims were delayed and physicians were unable to get paid for their services (Hatton, 2024). UnitedHealth Group made loans totaling close to $9 billion to healthcare providers to help ease the financial strain caused by the extended outage.

Vulnerability	**Remediation**
Remote access service that lacked multi factor authentication.	All external facing systems have been or will be enabled with multi-factor authentication.
Hacker was able to disable both Change Healthcare's primary information systems as well as their backup because the two weren't isolated.	Company is rebuilding legacy systems, shifting on-premise systems to cloud-based systems designed with built-in security controls, including segmentation, isolation and enhanced backup strategies (Jones, 2024).

Table 1: Vulnerabilities and Remediations

UnitedHealth Group and Change Healthcare learned some hard lessons from the attack, providing opportunities to improve their cybersecurity resilience. The two primary vulnerabilities and their remediations are captured in Table 1.

Questions:

- How does the use of cloud architecture enhance organizational security and resilience?
- How can organizations discover and evaluate third-party risks in their information systems strategy?
- How does this incident raise the importance of cybersecurity to a board-of-directors-level concern?
- What Key Performance Indicators (for instance the number of security incidents detected over a period of time) would you track to assess the effectiveness of security in a healthcare IT system?
- What other questions or observations can you make about this scenario as it relates to resilience, risk management, information, and information systems?

4. ADDITIONAL RESEARCH OPPORTUNITIES

If one pays attention, it is possible to see new examples of both brittle and resilient systems in many settings and situations. Please feel free to use these prompts as you and your instructor see fit, to conduct additional research and analysis of this broadly visible topic.

Resilience of emerging technologies

Emerging technologies significantly impact system resilience, offering both opportunities and challenges. While they can enhance a system's ability to withstand and recover from disruptions, they also introduce new vulnerabilities and complexities that require careful management. Key areas of impact include cybersecurity, infrastructure resilience, and the ability to adapt to evolving threats. Examples might include autonomous vehicles, drone deliveries, augmented and virtual realities, artificial intelligence in all its forms, and many more.

In consultation with your instructor, pick one or more related emerging technologies, and perhaps a particular industry to apply them to. Research the resilience of that technology and then analyze the potential resilience impacts of that technology on your particular industry choice.

Heathrow airport power outage

On March 21, 2025, London Heathrow Airport was forced to shut down for nearly a full day due to a fire at the North Hyde electrical substation in Hayes, West London (Mathews, 2025). The fire began at 23:23 on March 20, 2025, and the airport remained closed until 23:59 on March 21. Marhea, a 74-year-old passenger booked on a Brussels Airlines flight to Liberia, described arriving to "darkness and confusion" with no staff available to explain what had happened. "They didn't let us into terminal five. People were standing at the door everywhere," she said. Her airline eventually rebooked her on a Saturday flight with a different carrier. Another passenger, Ellen, had her surprise 30th birthday trip to Venice cancelled, telling Al Jazeera: "We were supposed to fly to Venice this morning from Heathrow for a day trip for my 30th birthday present, it was a surprise booked by my cousin for the two of us". Perhaps even worse, "transit" passengers who were passing through London en route to another country, often did not have a valid UK entry visa, and thus were restricted to a small part of the airport - without power, heating, cooling, or food.

The subsequent Kelly Review found that while Heathrow made appropriate decisions during the crisis, it highlighted the need for better systemic resilience planning and recommended prioritizing investment in backup power systems for critical operations. The review also noted communication gaps between technical teams and other airport staff regarding potential vulnerabilities (Mathews, 2025).

In consultation with your instructor, identify some aspects of the Heathrow power outage that you can analyze to better understand what went wrong, why such a heavily redundant system can fail, and what were the root causes of the failure.

Power outage in San Francisco

During a widespread PG&E power outage that cut electricity to nearly one-third of San Francisco on a Saturday, Waymo's autonomous vehicle fleet faced an unprecedented challenge when hundreds of traffic signals went dark across the city, creating severe gridlock that required law enforcement to manually manage intersections. While Waymo's vehicles successfully navigated over 7,000 dark signals by treating them as four-way stops, the scale of the outage created a concentrated spike in confirmation requests to human operators, causing delays that contributed to congestion on already overwhelmed streets (The Waymo Team, 2025).

As city officials urged residents to stay home, Waymo temporarily paused its service and directed vehicles to pull over and park appropriately so they could be returned to depots in waves, avoiding further congestion or obstruction of emergency vehicles. The company is now implementing fleet-wide software updates to provide vehicles with more regional outage context for more decisive navigation, improving emergency response protocols, and coordinating more closely with city officials and first responders (The Waymo Team, 2025).

In consultation with your instructor, research this failure and Waymo's technical and organizational response to it. You might consider topics such as:

1. **Scalability of Safety Protocols:** How should autonomous systems balance conservative safety measures (like confirmation checks) with operational efficiency during large-scale infrastructure failures?
2. **Graceful Degradation:** How could autonomous vehicle systems detect when their safety protocols are creating system-wide bottlenecks? How might they gracefully degrade service rather than contributing to cascading failures?
3. **Emergency Response Integration:** Can and should autonomous vehicle fleet operators coordinate with road management agencies to help make good decisions about failure management?
4. **Human-in-the-Loop Design:** Sudden scaling of a need for human technicians is likely to be difficult. What strategies could improve "human-in-the-loop" scalability?
5. **Testing Limitations:** What are the challenges and tradeoffs in testing autonomous systems for rare but high-impact events like this one?

Philosophical Question: Is it really autonomous if it can't operate when the power is off?

5. CONCLUSIONS

In this modern world, where so much of daily life is dependent on systems of one form or another, it is important to have the systems designed, built, and tested for resilience. Doing so requires all stakeholders, including business and technology teams, to understand the concepts of system resilience, as well as the factors that influence it. We believe that these scenarios will illustrate a significant range of resilience failure modes, and substantial fodder for discussion of system resilience issues in the classroom, as well as writing and research assignments outside the classroom.

6. ACKNOWLEDGEMENTS

The authors appreciate the support and insights of various industry experts who shared their time and expertise to discuss this case. The authors also appreciate the productive feedback provided by their students, their peers, and by the reviewers and conference chairs.

7. REFERENCES

Ahmed, R. (2017). Risk Mitigation Strategies in Innovative Projects. In *Key Issues for Management of Innovative Projects*. IntechOpen. https://doi.org/10.5772/intechopen.69004

Alozie, C. E., Akerele, J. I., Kamau, E., & Myllynen, T. (2024). Disaster recovery in cloud computing: Site reliability engineering strategies for resilience and business continuity. *International Journal of Management and Organizational Research*, *3*(1), 36-48. https://doi.org/10.54660/IJMOR.2024.3.1.36-48

Baccarini, D. (1996). The concept of project complexity—a review. *International journal of project management*, *14*(4), 201-204. https://doi.org/10.1016/0263-7863(95)00093-3

Baskerville, R., Baiyere, A., Gregor, S., Hevner, A., & Rossi, M. (2018). Design science research contributions: Finding a balance between artifact and theory. *Journal of the Association for Information Systems*, *19*(5), 3.

Cerullo, M. (2024). Delta cancels hundreds more flights as fallout from CrowdStrike outage persists. *CBS News*. Retrieved 7/1/2025, from https://www.cbsnews.com/news/delta-crowdstrike-outage-flight-status/

De Mar, C. (2024). Hacking of health care company leaves Chicago area man stuck in hospital. https://www.cbsnews.com/chicago/news/hacking-health-care-company-chicago-area-man-stuck-in-hospital/

Federal Aviation Administration. (2025). *Update: Newark Liberty International Airport* (FAA General Statements, Issue. https://www.faa.gov/newsroom/statements/general-statements

Gallagher, S. (2019). Baltimore ransomware nightmare could last weeks more, with big consequences. *Harm City*. https://arstechnica.com/information-technology/2019/05/baltimore-ransomware-nightmare-could-last-weeks-more-with-big-consequences

Gatlan, S. (2024). Ransomware gang claims they stole 6TB of Change Healthcare data. *Bleeping Computer*. https://www.bleepingcomputer.com/news/security/ransomware-gang-claims-they-stole-6tb-of-change-healthcare-data/

Hatton, R. (2024). How has the Change Healthcare cyberattack affected physicians? *Becker's ASC Review*. Retrieved June 28, 2025, from https://www.beckersasc.com/asc-news/how-has-the-change-healthcare-cyberattack-affected-physicians/

Hyperproof Team. (2024). Understanding the Change Healthcare Breach and Its Impact on Security Compliance. https://hyperproof.io/resource/understanding-the-change-healthcare-breach

Johnson, K., & Ibarra, A. B. (2024). California doctors struggle to make payroll one month after ransomware attack. *News from the States*. Retrieved July 5, 2025, from https://www.newsfromthestates.com/article/california-doctors-struggle-make-payroll-one-month-after-ransomware-attack

Jones, S. (2024, July 3). What the Change Healthcare Cyber Attack Means for the US Healthcare Industry. *Hornet Security*. https://www.hornetsecurity.com/en/blog/change-healthcare-cyber-attack/

Kerner, S. M. (2024). CrowdStrike outage explained: What caused it and what's next. *Tech Accelerator*. https://www.techtarget.com/whatis/feature/Explaining-the-largest-IT-outage-in-history-and-whats-next

Martínez, A., & Dumas, N. (2025). How archaic tech, staff shortages and construction made a meltdown at Newark airport. *NPR National*. Retrieved 06/23/2025, from https://www.npr.org/2025/05/07/nx-s1-5388438/airlines-trade-group-vp-discusses-newark-airport-delays

Mathews, R. (2025). *The Kelly Review: Lessons from Heathrow's power outage*. https://www.thebci.org/news/the-kelly-review-lessons-from-heathrow-s-power-outage.html

Otkhozoria, N., Petriashvili, L., Zhvania, T., & Imerlishvili, A. (2025). Advancing information system testing: challenges, methods, and practical recommendations. *International*

Science Journal of Engineering & Agriculture, *4*(2), 203-214. https://doi.org/10.46299/j.isjea.20250402.13

Rooney, J. J., & Heuvel, L. N. V. (2004). Root Cause Analysis For Beginners. *Quality Progress*. https://servicelink.pinnacol.com/pinnacol_docs/lp/cdrom_web/safety/management/accident_investigation/Root_Cause.pdf

Rose, J. (2025). Disbelief, then fury: A Newark air traffic controller says they saw a crisis coming. *NPR*. Retrieved 05/29/2025, from https://www.npr.org/2025/05/22/g-s1-68333/newark-air-traffic-controller-atc

Sherman, T. (2025). Newark's air traffic nightmare continues as controllers lose contact with planes a 4th time. *NJ News*. Retrieved May 20, from https://www.nj.com/news/2025/05/newarks-air-traffic-nightmare-continues-as-controllers-lose-contact-with-planes-a-4th-time.html

Shi, X., Liu, W., & Lim, M. K. (2024). Supply chain resilience: new challenges and opportunities. *International Journal of Logistics Research and Applications*, *27*(12), 2485-2512. https://doi.org/10.1080/13675567.2023.2262396

Steenhuysen, J. (2024). Patients or Payroll? US Healthcare Hack Creates Hard Choices. *Reuters*. https://www.reuters.com/world/us/patients-or-payroll-us-healthcare-hack-creates-hard-choices-2024-03-06/

The Waymo Team. (2025, Jan 2). **Autonomously navigating the real world: lessons from the PG&E outage**. *Waypoint - the official Waymo Blog*. https://waymo.com/blog/2025/12/autonomously-navigating-the-real-world

Usman, A. B., & Asplund, M. (2025). Update at Your Own Risk: Analysis and Recommendations for Update-Related Vulnerabilities. IFIP International Conference on ICT Systems Security and Privacy Protection. https://doi.org/10.1007/978-3-031-92886-4_7

Witman, P. D., Prior, J., Nickl, T., & Mackelprang, S. (2024). The Southwest Airlines Winter Meltdown Case Studies on Risk, Technical Debt, Operations, Passengers, Regulators, Revenue, and Brand. *Information Systems Education Journal*, *22*(5), 59-71. https://doi.org/10.62273/EFWA2093

From Excel to Python to AI: A Connectivist Model for Introducing Coding and Prompt Engineering to First-Year IS Students

Mark Frydenberg
mfrydenberg@bentley.edu
Bentley University
Waltham, MA

Abstract

This study examines the use of connectivist learning principles to teach first-year students about coding with Python in a Fundamentals of Information Systems course. The instructional design integrates tools such as Microsoft Excel, Google Colab, and AI chatbots to support conceptual understanding, promote knowledge exchange among students, and develop problem solving skills. Grounded in a connectivist approach, the module considers the relationship between students, spreadsheet logic, Python coding, and generative AI tools as inter-connected nodes that influence how students construct, transfer, and apply knowledge. Students engaged in collaborative coding activities, progressing from designing and sharing spreadsheet-based solutions to translating logical requirements into Python programs through iterative prompt engineering.

The study addresses four research questions: (1) To what extent does prior experience with Excel support students' understanding of Python programming concepts? (2) How do digital tools and peer networks support student engagement and learning in coding? (3) How do students perceive the value of learning Python for academic and career development? and (4) To what extent are students motivated to continue learning coding independently? Validated survey results indicate that students find benefit in using networked collaboration and learning tools, recognize the value in learning Python, and favor further informal study. These findings also support the use of connectivist learning techniques as an effective framework for presenting coding instruction to first-year information systems students engaged in a learning scenario shaped by personal networks, technology, and AI tools.

Keywords: Connectivism, Spreadsheet logic, Python Pedagogy, Prompt Engineering, AI Pedagogy

Recommended Citation: Frydenberg, M., (2026). From Excel to Python to AI: A Connectivist Model for Introducing Coding and Prompt Engineering to First-Year IS Students. *Information Systems Education Journal,* v24(n2), pp 27-43. DOI# https://doi.org/10.62273/CACW209

From Excel to Python to AI: A Connectivist Model for Introducing Coding and Prompt Engineering to First-Year IS Students

Mark Frydenberg

1. INTRODUCTION

The ability to code, or at least understand code, has become an essential skill for many 21st century professionals (Kivunja, 2014). Approaches to teaching coding concepts to students in introductory information systems survey courses have varied from teaching the vocabulary of coding (sequence, selection, repetition, input, output, variables, functions, classes, objects, methods) by example (Parsons, 2023) to using visual block-based coding tools to implement basic algorithms and procedures (Andone & Frydenberg, 2021; Bozan & Taslidere, 2024). Emphasis is often on concepts rather than creating actual programs.

This paper presents an innovative approach to introducing first-year students in a Fundamentals of Information Systems course to coding through a three-session module. Building on their prior knowledge and proficiency with Microsoft Excel, students learn the basics of Python programming and then use generative AI to create Python code for more complex tasks such as data visualization.

Guided by principles of connectivist learning, which emphasizes knowledge construction through connections between people, tools, and ideas, this study investigates the following research questions:

- **RQ1.** To what extent does prior experience with Excel support students' understanding of coding concepts?
- **RQ2** How do digital tools and peer networks support student learning and engagement in an introductory information systems course?
-
  RQ3. How do students perceive the value of learning Python for their academic and professional goals?
- **RQ4.** To what extent are students motivated to continue learning coding after the course?

2. THEORETICAL FOUNDATIONS AND RELATED WORK

Connectivist learning theory, as introduced by Siemens and Downes (Downes, 2010; Siemens, 2005; Siemens & Downes, 2011) emphasizes these core principles:

- Learning happens by making connections, constructing and interacting with both human and technological networks.
- Learning is not only about understanding concepts, but also making connections with individuals, ideas, and technology.
- Learning from peers and being exposed to a variety of perspectives and solutions is central.
- Making decisions is an ongoing process in a rapidly changing world driven by information.
- Learning also relies on the use of technology to store, process, and generate information. Technology is essential to the connectivist classroom, enabling students to access, share, and create knowledge.

Literature suggests a growing interest in applying connectivist learning approaches in business education, with a few studies specifically focusing on learning to code. Connectivist learning empowers students to learn collaboratively, generate knowledge, interact with different tools and connect with different information sources (Gottipati et al., 2023; Utecht & Keller, 2019).

A second-year programming course at the University of Pretoria integrated connectivist strategies with scaffolding interventions from instructors and found that students valued this approach, engaging with online resources and peers, and as a result, felt more confident in taking on complex coding projects (Matthee & van Deventer, 2022).

In a recent Finnish study investigating the impact of a connectivist learning approach on teaching sustainable business in an online context, results showed that peer interactions and digital tools were seen as active nodes of learning, allowing students to make use of technology to help strengthen knowledge and learn from different perspectives. (Dziubaniuk et al., 2023) This suggests that a connectivist approach also can be applied to learning to code, especially when learners choose tools and networks that suit their needs and learning styles. While the coding environment (Google Colab) was standardized, students used the AI tools of their own choice refining prompts and debugging code across platforms. This process reflects key connectivist principles: navigating multiple knowledge nodes, applying real-time feedback, and developing personal strategies for problem solving.

Recent research supports a meaningful connection between spreadsheet fluency and learning to code. Lovászová and Hvorecký (2004) describe how spreadsheets can be used to explore foundational algorithmic structures such as sequence, selection, and repetition without writing formal code in a programming language. As they note, "Programming problems can be also solved using spreadsheet calculations. The stage is built around one of the spreadsheet solutions. It is selected and arranged in a way that exhibits and amplifies the considered properties of the algorithm. By experimenting with the spreadsheet solution, the students reveal the properties and extrapolate them to the field of programming" (Lovászová & Hvorecký, 2004, p. 46).

Csernoch and Biró (2019) found that first-year students who approached solving spreadsheet tasks algorithmically demonstrated stronger problem solving and computational thinking skills than those who relied on more basic functions. From a connectivist perspective, these findings suggest that activating prior spreadsheet knowledge can help students build new conceptual connections when they learn to code.

Recent research also highlights the role of generative AI enabled coding environments as important learning nodes within a connectivist framework. Zviel-Girshin (2024) found that students quickly incorporated AI tools into their everyday coding habits, especially for tasks like debugging, adding comments, and looking up information, and became more comfortable with them as the semester went on. The study also noted that students often leaned on these tools too much, which sometimes diminished their learning of problem solving and coding skills.

Gardella et al. (2024), examined how novice programmers performed when using GitHub Copilot to complete introductory programming tasks. Their findings showed that AI-assisted coding significantly increased programming efficiency and reduced mental workload, although gains in confidence emerged only with repeated practice. These results suggest that AI tools can support beginning coders by providing immediate feedback and reducing frustration, but their ability to develop foundational skills may be limited when students over-rely on them. Together, these studies align with connectivist principles by showing how learners can benefit from multiple technological and informational nodes as part of their learning process.

Figure 1 presents a conceptual model of the Excel / Python / AI progression framed by connectivist learning. Knowledge develops as students traverse technological, peer, and conceptual models while translating their fluency in translating spreadsheet logic to developing Python code and then leveraging AI tools for higher-order tasks. Excel knowledge provides familiar problem structures that students then implement in Python code, while AI supports developing code for more complex tasks such as data visualization and requires students to critically evaluate the output of generated code.

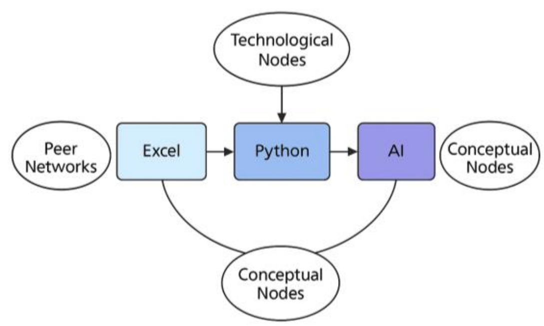


Figure 1. Conceptual model of the progression from Excel to Python to AI framed by connectivist learning.

The pedagogical transition explored in this research follows a clear evolution of digital literacy. As Prather et al. (2024) argue, in the AI era, students are not merely 'writing code'; they are performing high-level problem decomposition through 'Prompt Problems.' This aligns with the progression observed in this study, where students move from writing formulas in Excel to writing code in Python to writing prompts for AI**.**

The progression from Excel to Python to AI is deliberate, grounded in the idea that students learn coding more effectively when new concepts build on existing knowledge. Many first-year students arrive to college with some familiarity using spreadsheets (McCarron & Frydenberg, 2023). Excel provides a familiar entry point for exploring basic computational thinking principles such as calculations, data types, logical conditions and problem solving.

By starting with spreadsheet-based coding tasks and then translating those same steps into simple Python programs, students experience coding as an extension of the patterns they have already learned. It also aligns with connectivist learning principles by treating Excel and Python as two different knowledge nodes that students can learn to connect.

The transition from spreadsheets to coding is supported by the shared logical thinking that both tools require. Sarkar et al. (2020) found significant correlations between proficiency at writing formulas and programming expertise, suggesting that writing Excel formulas facilitates the progression to writing more abstract code in traditional programming languages such as Python or Java.

Introducing generative AI after students gain experience writing and running Python code serves a different pedagogical purpose. Once students can recognize and run code, AI becomes a tool for higher-order thinking. In this way, AI becomes a third node in the learning network, encouraging students to apply their Excel vocabulary to create prompts, critique outcomes, and iterate to refine the results.

As studied by Denny et al. (2023), the integration of generative AI into the programming curriculum provides a modern scaffolding mechanism that can reduce the initial cognitive load for novices, bridging the gap between basic data manipulation and complex software development.

The flow from Excel to Python to AI also reflects a workflow students are likely to see in their future academic and professional careers. Analysts build models in spreadsheets, transition to code as tasks become more complex, and use AI tools to generate insights into data. Introducing this progression into the course supports students in their learning and prepares them for their future roles in the digital workplace.

3. IMPLEMENTATION: FROM EXCEL TO PYTHON TO AI

One of the biggest challenges in learning to code is understanding new syntax and abstract concepts. "The learning of programming is difficult because it involves the simultaneous acquisition of three domains of knowledge: the syntax and semantics of a programming language, the notional machine (an abstract model of the execution process), and problem-solving strategies" (Robins et al., 2003, p. 138). This study describes a three-session module for teaching coding concepts using familiar examples based on students' prior knowledge of Microsoft Excel in CS 100 ("Solving Business Problems with information Technology"), an information systems fundamentals course required of all first-year students at a business-focused university in New England. Course topics include intermediate proficiency in Excel along with basic computing concepts (operating systems, organizing files and data, cybersecurity, the Internet and World Wide Web, security and privacy), and emerging technologies such as virtual reality and generative AI.

The course also includes a coding component which historically had three classes on web development. Students learned to create simple web pages by hand-coding basic HTML tags for headers, paragraphs, images, and links, along with simple formatting for fonts and colors. The original intent was that by writing HTML and viewing it in a browser, students would learn about coding. The Python curriculum described here replaces the HTML unit for those sections that adopted it as an alternative.

Comparing Excel and coding concepts is an effective way to scaffold student learning (Groner, 2023). "At its heart, connectivism is the thesis that knowledge is distributed across a network of connections, and therefore that learning consists of the ability to construct and traverse those networks" (Downes, 2010).

Students used the Google Colab environment for developing and submitting their Python programs; after creating their notebooks, they shared a link to it with their instructor who checked it for grading and completion. Figure 2 shows an example of a student's Google Colab notebook. Students add a hierarchy of headings to make their notebooks easy to navigate and are required to show the program output to facilitate grading. Text blocks describe code, and code blocks contain executable Python statements. Using an online environment eliminates the need to install an IDE and upload solutions to a course learning management system. After creating their notebooks, students share the links with their instructor by completing an online form.

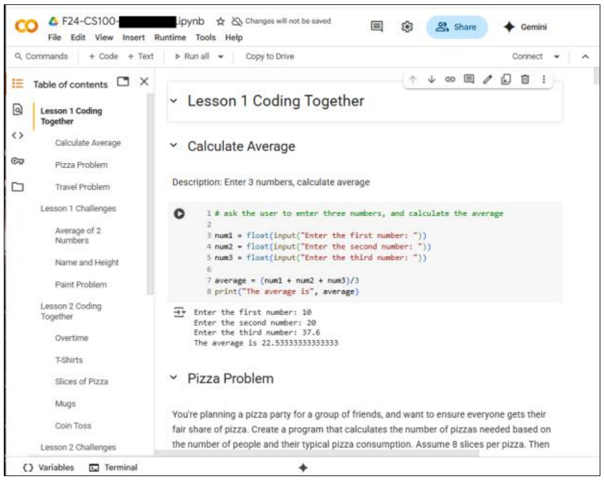


Figure 2. A Google Colab Notebook.

Google Colab can provide AI-generated code. Students were instructed to modify notebooks' settings to disable this feature for the first two lessons when they are learning coding fundamentals, and then to enable it for the third lesson when they will run and evaluate the results of AI-generated code to create charts and graphs.

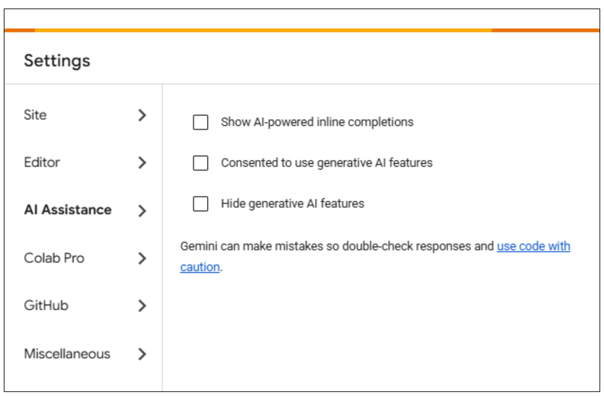


Figure 3. Disabling or Enabling AI Assistance in Google Colab.

Each day's lesson began with "coding together" problems where students solved a problem in small groups using Microsoft Excel. The instructor then showed them how to translate that solution into a Python program. See Figure 4 for a sample spreadsheet and coding solution for a problem of calculating the number of pizzas required and cost of a pizza party. Note that the spreadsheet solution (at the left of Figure 4) uses a named range for cell B3 as an example of how named ranges can be more descriptive than cell names, encouraging the use of descriptive variable names.

Each Coding Together problem had a similar Coding Challenge for students to work on, usually with a partner, for homework. Students presented their coding challenge solutions at the start of the next class.

This structure reflects connectivist learning by applying its core ideas: students learn through peer collaboration, make use of familiar tools and learn to use new ones, and they engage with technology as an active partner in their learning. This approach moves beyond memorizing concepts and typing code without understanding to develop students as networked learners, capable of adapting and learning through their connections with tools, peers, and knowledge systems. Each day's lesson introduces a coding concept by relating it to a familiar Excel concept. The focus is on what coding can do, as much as it is on writing the code itself. On the third day, by using AI to generate code, students learn that AI can be an effective problem-solving tool, where the solution that AI provides is not the end, but a means to accomplishing a greater task (in this case, visualizing and interpreting data).

Students still need to run the AI-generated code and make sure that it produces the desired results (or modify their prompts or possibly the code) if the results are not as expected. Students also learn that when the AI-generated code does not work (and it often may not), they should try a different AI tool to see if it generates different solutions. The exercise becomes a lesson in prompt engineering, as students must apply the vocabulary of charts and graphs (chart types, legends, titles, axis labels, major and minor axis, etc.) to specify the desired appearance of their charts and then verify that the results are correct.

	A	B
1	Let's have a Pizza Party!	
2		
3	SLICES PER PERSON	2
4	SLICES PER PIZZA	8
5	COST PER PIZZA	10
6		
7	How many people?	5
8		
9	Slices Needed	=B7*SLICES_PER_PERSON
10	Pizzas Needed	=CEILING(B9/B4,1)
11	Total Cost	=B10*B5

Pizza Problem

You're planning a pizza party for a group of friends, and want to ensure everyone gets their fair share of pizza. Create a program that calculates the number of pizzas needed based on the number of people and their typical pizza consumption. Assume 8 slices per pizza. Then enter the cost of a pizza to calculate the total cost of the pizza order. import math and use math.ceil() to round up!

[2]

```python
# pizza party
import math

SLICES_PER_PERSON = 2
SLICES_PER_PIZZA = 8
COST_PER_PIZZA = 10

print("Let's have a pizza party!")

num_people = int(input("How many people are attending the pizza party? "))
slices_needed = num_people * SLICES_PER_PERSON
pizzas_needed = math.ceil(slices_needed / SLICES_PER_PIZZA)
total_cost = COST_PER_PIZZA * pizzas_needed
print ("You will need", pizzas_needed, "pizzas. The cost is $", total_cost)
```

```
Let's have a pizza party!
How many people are attending the pizza party? 5
You will need 2 pizzas. The cost is $ 20
```

Figure 4. From Excel to Python (Spreadsheet and Python solutions)

Anecdotally, ChatGPT's code seemed more reliable than that generated within Colab for data/charts. Figure 5 shows part of a student's Colab notebook with AI-generated code and the resulting bar chart visualizing temperature data.

Table 1 summarizes the concepts of each lesson. Appendix B shows a summary of the Coding Together and Coding Challenge problems for each lesson.

Lesson	Python Concepts	Excel Concepts	Connections
1	variables, calculations, data types, input and print	Cells, formulas, data types, calculations, entering and displaying values	Calculations are the basic feature of spreadsheets and at the foundation of any algorithm. Topics include data types (int and float), simple math operations (round, ceiling, max, min, average) and their corresponding functions in Excel.
2	if statements, conditions for Loops	=IF(), =AND(), =OR() functions Copying values or formulas down a range of cells	Students are familiar with various forms of the if function in Excel, so writing if statements in Python is a natural extension. The process of copying values down a range of cells in a spreadsheet is similar to iterating over a list of items or values using a for loop.
3	using AI to generate Python code	Charts, graphs	Students are familiar with making pie, bar, line, and other charts in Excel. Coding these in Python by hand is difficult, but an AI tool can often generate the code if given a descriptive prompt.

Table 1. Excel to Python to AI Lesson Topics and Rationale

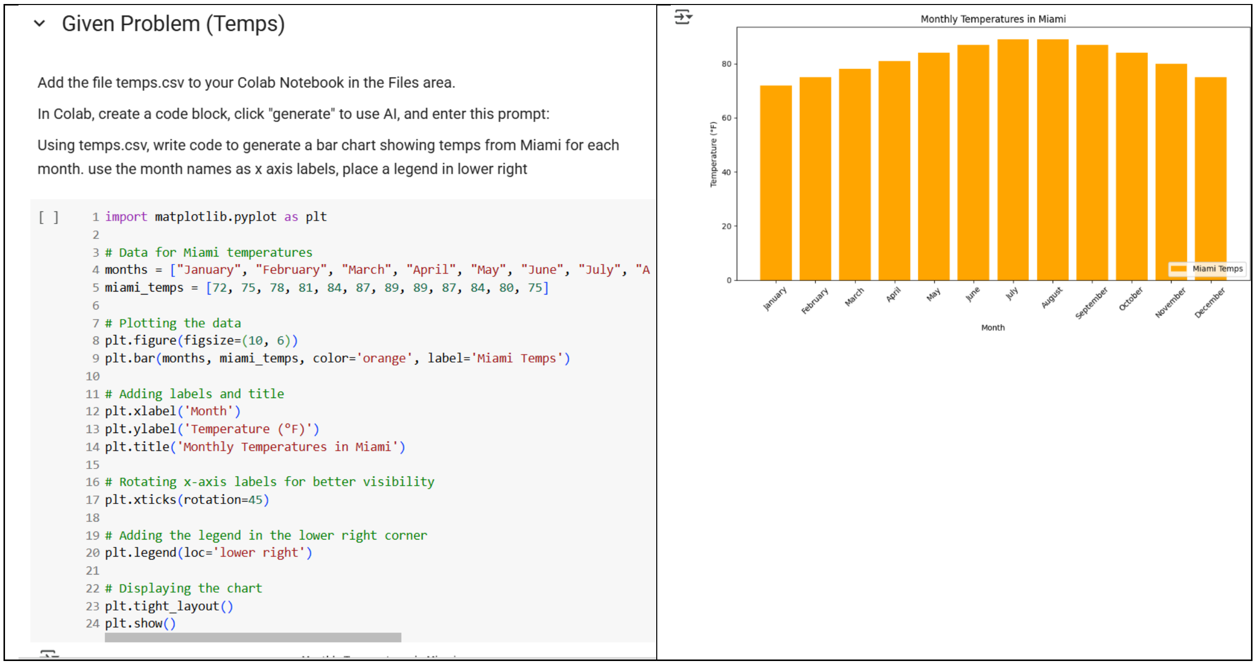


Figure 5. AI-Generated Code for Charting Weather data, shown in a Colab notebook.

4. METHODOLOGY

The author piloted the three-session module during the Spring 2024 semester with one section of 18 honors students. The following Fall, after making it available to other CS 100 instructors, one instructor adopted the lesson in place of HTML/Web Development in Fall 2024 for the two sections of approximately 35 students that he taught, and three instructors (five sections of approximately 35 students each) adopted the module for Spring 2025. All instructors who taught the Python module participated in two training sessions led by the author, who reviewed both the pedagogy and coding principles for these lessons. Students completed a short survey reflecting on their experience learning about coding after completing the last module.

Survey Data and Reliability

During these three semesters, 217 students (124 males, 92 females, 1 preferred not to say) completed the survey, usually during class time. Given maximum class sizes, at most 299 students would have been enrolled in sections teaching this curriculum, for a response rate of approximately 73%. The majority (60%) had not coded before; 30% had some Java or Python in high school, and the rest had varying degrees of coding experience (self-taught, knew HTML coded in Scratch, did "hour of code"). Most students (86%) were 18 or 19 years old, with 30 students 20 or older. When asked about their intended majors, there were only two that mentioned Computer Information Systems, while the majority were split between Accounting, Finance, and Corporate Finance and Accounting.

The survey includes 16 questions based on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree), as shown in Appendix A. For clarity of results, the author organized the questions into broad categories:

- Learning Experience (6 questions)
- Engagement and Interest (4 questions)
- Perceived Career Value (4 questions)
- Prior or Recommended Exposure to Python (2 questions)

These categories were developed to track the pedagogical transition from established spreadsheet logic (Excel) to algorithmic thinking (Python), while incorporating the emerging role of AI tools for those with little or no coding experience.

The internal consistency of the survey instrument was evaluated using Cronbach's alpha (α). This step ensures that the questions within each category reliably measure the same underlying concept. Values were reverse-coded for the item *Learning Python was challenging* to ensure that a higher score (5) consistently represented a more positive perception of the learning process while a lower score (1) represented a perceived barrier.

The items *'Using AI tools to generate Python code was new to me'* and *Other sections of CS 100 should incorporate Python as we* did were excluded from the internal consistency (Cronbach's alpha) testing. While the other items measured perception (learning, engagement, and perceived value) these measured prior technological experience or future course recommendations. The summary of all responses is shown in Fig. 6. The results, shown in Table 2, indicate that all categories exceed the standard academic threshold of 0.70, demonstrating high reliability(Nunnally & Bernstein, 1994).

Category	**# Items**	**Cronbach's α**
Learning and Understanding	6	0.788
Engagement and Interest	4	0.883
Perceived Career Value	4	0.847

Table 2. Cronbach's α by Category

Survey Results
Because the Python curriculum was entirely similar across sections, and because all instructors were trained to follow the pedagogy for this module, the results that follow aggregate all results from all sections over three semesters.

The charts in Figure 6 summarize the averages of all responses for each question in each category.

Examining *Student Engagement and Interest*, students found the coding activities to be engaging (3.79) and fun (3.46). They had a moderate interest in learning more about Python independently (3.24), and lower-rated the possibility of taking a college course in Python (2.81). While they see the value in learning Python, fewer are motivated to take a course in it. This is consistent with the selection of majors that students are interested in pursuing, most of which do not require a coding course. One student commented, "I really enjoyed Python, and I think basic coding is an essential skill for extracurricular projects; I have already needed to code for a case competition."

Figure 6. Survey Results

In the *Learning Experience* category, students overwhelmingly rated Google Colab as an easy-to-use tool for coding in Python (4.14). They felt that learning Python helped them understand programming concepts (3.88) and improve their problem-solving skills (3.58). While the examples were easy to understand (3.63), students were challenged to create their own code (3.32).

Said one student, "[writing code] definitely is something that you need to practice at and understand what you're doing, writing code is not always easy and you have to think about every action involved." Another student commented that they could connect learning Python with previous coding experiences: "I had some prior experience with Java so it was nice to learn a new language and apply some of my prior knowledge."

Unexpectedly, students rated "knowing Excel helped me to learn Python (3.24)" lowest in this category, suggesting mixed impact of prior Excel experience on learning about coding in Python. Upon sharing this finding with a CIS tutor who helped CS 100 students learning Python, the tutor reflected: "When I would help students in CS 100 with Python, I noticed there was a logic disconnect. Some students struggled with the multi-step process of solving a Python problem, whereas Excel tended to work in smaller, separate chunks." While formulas would often translate similarly, when writing Python programs, students also needed to think about accepting user input, converting values to numeric data types, printing results, adding comments to code, and navigating Python syntax issues (indentation, colons, etc.) not encountered when creating spreadsheets.

Considering the *Perceived Value* of learning Python, students overwhelmingly felt it was a valuable skill to have in today's job market (4.16) and saw it as valuable to their career (3.68), with practical applications beyond the course (3.87). They found the value in using Python as found Python to be useful for manipulating and visualizing data (4.00).

Under *Prior or Recommended Exposure,* using AI tools to generate Python code was new to most students (4.35), and recommended that other sections should teach it (3.98)

Overall, the data suggests that students found the coding environment and Colab to be helpful; they were engaged with the topic (especially using AI); and they clearly valued the relevance of knowing Python; but they prefer informal over formal instruction for future learning.

Sentiment Analysis

To better understand students' responses to the coding module, this study implemented a sentiment analysis using Python's TextBlob library (Loria, 2018). TextBlob evaluates sentiment along two dimensions: polarity (-1.0 for strongly negative to +1.0 for strongly positive), and subjectivity (0 for objective to 1.0 for highly personal) (Liu & Pang, 2025). The analysis shown here focuses on polarity, and classifies comments as:

- Positive (polarity > 0.2)
- Neutral (-0.2 <= polarity <= 0.2)
- Negative (polarity < -0.2)

These bounds are based on a commonly used heuristic (Loria, 2018). Although they provide a practical way to filter positive or negative sentiment, they are admittedly not perfect, especially for short comments, and the author acknowledges this as a limitation of this study.

Out of 78 responses, 47 (60%) were positive, 24 (31%) were neutral, and only 7 (9%) were negative. Neutral sentiments were mostly due to blank ("n/a" or "none") responses. Figure 7 displays a word cloud colored by sentiment polarity. Green words indicate positive sentiment, and red words indicate negative sentiment (See Figure 7).

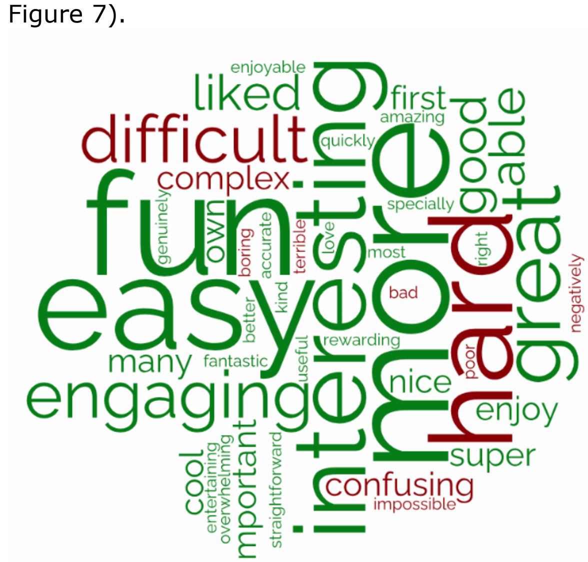


Figure 7. Sentiment Word Cloud (Generated with wordclouds.com).

Positive themes included student enjoyment when learning Python, recognition if its value for

Comment	Polarity	Sentiment
"Very interesting and enjoyable."	0.58	Positive
"I was never efficient with technological skills so my opinion is biased and against coding however this course was not terrible for someone like myself."	0.50	Positive
"I enjoyed learning about Python and felt that it was an easy concept to understand."	0.47	Positive
"It was a fun and engaging experience. I'm not much for coding, but it was nice to learn some basics nonetheless."	0.30	Positive
"I found the Excel integration a little challenging even though I already kind of know python. I also didn't completely understand its value."	0.25	Neutral
"It was fun but i don't see a ton of practical applications for it in my major."	0.18	Neutral
"At first I didn't understand it completely but by the time we had to do the project I felt ready and it was very simple to do because I knew what I was doing."	0.14	Neutral
"I did not enjoy using AI to code because it gave me error messages and I didn't know how to fix it. Everything else was fun."	0.05	Neutral
"The third python assignment was difficult. Adding the data into the platform didn't work."	-0.25	Negative
"Bad/boring."	-1.00	Negative

Table 3. Comments and Polarity Scores.

future careers or in other courses and appreciating the use of Google Colab and engaging projects. Selected examples are shown in Table 3 to provide additional context.While the results from the survey's Likert scale responses show that students valued the experience overall, this sentiment analysis confirms engagement and enjoyment, even among students who found coding difficult, and suggests their emotions (frustration, reward, curiosity) while completing the module.

6. CONCLUSIONS

This study demonstrates that a connectivist framework provides a powerful lens for understanding how first-year students can learn about basic coding concepts when supported by familiar tools, peers, and AI technologies. By intentionally following in the Excel /Python / AI sequence, students engage with multiple technologies and concepts and find relevant connections between them. Survey results show that while students drew some parallels between spreadsheets and programming, the transition from Excel's cell-based logic to Python's multi-step reasoning and algorithm development required more support and explanation than was originally anticipated.

Students confirmed strong engagement with digital tools and their peers and described positive emotions when learning about Python and using AI tools in a new way. Their responses, which range from struggling with technology to succeeding in using it, reflect the iterative and exploratory nature of constructivist learning environments, where learners construct meaning by navigating multiple information sources and actively exchanging ideas with peers through collaborative problem solving, shared coding challenges, and comparing Ai-generated results.

Many students said they would pursue coding informally rather than through additional course work; this self-directed interest also is aligned with connectivist principles. As digital technologies and AI tools continue to evolve, developing the ability to navigate, evaluate, and connect these tools will be critical to future success. This study shows that connectivist strategies can enhance student understanding and prepare them for the data-driven and AI-enabled tools and processes they will encounter in their future careers.

Limitations and Future Research

This study combines data from eight sections taught by five different instructors during a three-semester span. For all except the original instructor, none had taught this material before. Teaching Excel concepts knowing what the upcoming Python content would be, or having more familiarity with it, might also influence student perception of the value of knowing Excel prior to learning basic Python skills. Future studies could include a larger sample size, introducing this curriculum in a similar course at another university, or examining results based on instructor or students’ anticipated majors or minors to discover learning trends or biases.

Research Questions
Returning to the research questions:

RQ1. To what extent does prior experience with Excel support students’ understanding of coding concepts?

While the curriculum aimed to build on students’ prior experience with Excel, the impact of that prior knowledge on learning Python was less significant than expected. The survey results suggest this connection was moderate. Students gave lower ratings to the statement “Applying my Excel knowledge to Python made it easier…” (mean = 3.24), indicating mixed perceptions about the strength of that knowledge transfer. From a connectivist learning perspective, not all students were able to make those connections effectively without additional guidance. While the instructor designed the module to create the bridge between Excel and Python, it may require additional scaffolding to make those connections clearer to more students.

RQ2. How do digital tools and peer networks support student learning and engagement in an introductory information systems course, through a connectivist lens?

Web-based tools such as Google Colab assist with student learning, but surprisingly, prior Excel knowledge was less of a benefit in learning Python than expected. Students made connections between spreadsheet and Python concepts, but found the coding exercises to be more challenging, and using AI to generate complex Python code to be most interesting.

RQ3. How do students perceive the value of learning Python for their academic and professional goals?

Students' high ratings of Python’s usefulness and relevance suggest that they clearly see the benefit of learning about coding for both academic projects and future employment.

RQ4. To what extent are students motivated to continue learning coding after the course?

Students recognized the value of learning Python, particularly in their continued studies and in their careers. Their interest in pursuing future course work, however, was less than expected. This aligns with connectivist principles, however, which stresses learning as a self-directed process. One student noted “I really enjoyed Python, and I think basic coding is an essential skill for extracurricular projects; I have already needed to code for a case competition.”

Discussion
These results collectively support the use of connectivist learning as a lens through which to evaluate this instructional approach. Students engaged meaningfully with tools such as Google Colab, Excel, and AI chatbots, while also learning through interactions and collaborative problem solving with their peers. Sharing solutions in class reinforced social learning and using technology tools enabled students to draw connections between familiar and new concepts. Overall, the findings suggest that coding education grounded in connectivist principles can result in greater engagement and deepen understanding in learning environments that thrive on technology, collaboration, and real-world applications.

While this study focused on the transition between spreadsheet logic, coding, and generative AI, emerging trends suggest a fundamental shift is taking place in programming pedagogy, one that prioritizes intention over syntax. The direct transition from Excel to AI has been accelerated by the emergence of vibe coding, a practice where “you fully give in to the vibes… and forget that the code even exists… It’s not really coding – I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works” (Karpathy, 2025).

Recent shifts in computer science education suggest that the “vibe coding” approach, skipping the coding language entirely in place of natural language prompts that lead to an AI-generated result, shifts the focus from syntax to solution (Pradhan et al., 2025; Sarkar & Drosos, 2025). The world has changed, and vibe coding shifts the student’s role from coder to code-reviewer or system architect, allowing especially non-technical users to build technology solutions while leaving the implementation details to an AI large language model. Future research could investigate the implications and possibilities of vibe coding on learning to code, computational thinking, and empowering students to leverage AI for social impact and ethical data-driven decision making.

7. REFERENCES

Andone, D., & Frydenberg, M. (2021). Co-creating with TalkTech: Developing Attributes through International Digital Collaborative Projects. *2021 IEEE Global*

Engineering Education Conference (EDUCON), 1073–1077. https://doi.org/10.1109/EDUCON46332.2021.9454119

Bozan, I., & Taslidere, E. (2024). The Effect of Digital Game Design-Supported Coding Education on Gifted Students' Scratch Achievement and Self-Efficacy. *International Journal of Contemporary Educational Research*, *11*(1), 20–28. https://doi.org/10.52380/ijcer.2024.11.1.531

Csernoch, M., & Biró, P. (2019). *Are digital natives spreadsheet natives?* (arXiv:1909.00865). arXiv. https://doi.org/10.48550/arXiv.1909.00865

Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B. N., Santos, E. A., & Sarsa, S. (2023). *Computing Education in the Era of Generative AI* (arXiv:2306.02608). arXiv. https://doi.org/10.48550/arXiv.2306.02608

Downes, S. (2010). New Technology Supporting Informal Learning. *Journal of Emerging Technologies in Web Intelligence*, *2*(1). https://doi.org/10.4304/jetwi.2.1.27-33

Dziubaniuk, O., Ivanova-Gongne, M., & Nyholm, M. (2023). Learning and teaching sustainable business in the digital era: A connectivism theory approach. *International Journal of Educational Technology in Higher Education*, *20*(1), 20. https://doi.org/10.1186/s41239-023-00390-w

Gardella, N., Pettit, R., & Riggs, S. L. (2024). Performance, Workload, Emotion, and Self-Efficacy of Novice Programmers Using AI Code Generation. *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, 290–296. https://doi.org/10.1145/3649217.3653615

Groner, D. (2023). *Python for Data Analytics: A Business Oriented Approach*. Prospect Press. https://www.prospectpressvt.com/textbooks/groner-python

Karpathy, A. (2025, February 2). *There's a new kind of coding I call "vibe coding"* [Letter]. https://x.com/karpathy/status/1886192184808149383

Kivunja, C. (2014). Do You Want Your Students to Be Job-Ready with 21st Century Skills? Change Pedagogies: A Pedagogical Paradigm Shift from Vygotskyian Social Constructivism to Critical Thinking, Problem Solving and Siemens' Digital Connectivism. *International Journal of Higher Education*, *3*(3), 81–91. http://dx.doi.org/10.5430/ijhe.v3n3p81

Liu, C., & Pang, S. (2025). *Empty Vessels Make the Most Noise: Analyst Self-Promotion Behavior and Market Outcomes* (SSRN Scholarly Paper No. 5292117). Social Science Research Network. https://doi.org/10.2139/ssrn.5292117

Loria, S. (2018). *TextBlob: Simplified Text Processing—TextBlob 0.19.0 documentation*. https://textblob.readthedocs.io/en/dev/

Lovászová, G., & Hvorecký, J. (2004). On Programming and Spreadsheet Calculations. *Spreadsheets in Education*, *1*(1). https://sie.scholasticahq.com/article/4510-on-programming-and-spreadsheet-calculations, https://sie.scholasticahq.com/article/4510-on-programming-and-spreadsheet-calculations

Matthee, M., & van Deventer, P. (2022). Preparing IS Programming Students for Work by Enhancing a Connectivist Teaching Approach by Scaffolding Practices. *Proceedings of the 2022 AIS SIGED International Conference on Information Systems Education and Research*. https://aisel.aisnet.org/siged2022/11

McCarron, E., & Frydenberg, M. (2023). *Digitally Prepared for Success? Technology Skills of Incoming First-Year College Students*. *21*(3), 70–90. http://www.isedj.org/2023-21/n3/ISEDJv21n3p70.html

Nunnally, J. . C., & Bernstein, I. H. (1994). *Psychometric Theory* (3rd Edition). McGraw-Hill. https://archive.org/details/dli.scoerat.1556psychometrictheorysecondedition

Parsons, J. (2023). Coding with Python. In *New Perspectives Concepts 2021 and Microsoft Office 2021* (21st ed.). Cengage.

Pradhan, A., Henriques, J., Pan, S., & Engineer, R. (2025). EXPLORING VIBE CODING PRACTICES IN COMPUTER SCIENCE EDUCATION. *ICERI2025 Proceedings*, 5153–5162. 18th annual International Conference of Education, Research and Innovation. https://doi.org/10.21125/iceri.2025.1451

Prather, J., Denny, P., Leinonen, J., Smith, D. H., Reeves, B. N., MacNeil, S., Becker, B. A., Luxton-Reilly, A., Amarouche, T., & Kimmel,

B. (2024). *Interactions with Prompt Problems: A New Way to Teach Programming with Large Language Models* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2401.10759

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, *13*(2), 137–172. https://doi.org/10.1076/csed.13.2.137.14200

Sarkar, A., Borghouts, J. W., Iyer, A., Khullar, S., Canton, C., Hermans, F., Gordon, A. D., & Williams, J. (2020). Spreadsheet Use and Programming Experience: An Exploratory Survey. *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, CHI EA '20*, 1–9. https://doi.org/10.1145/3334480.3382807

Sarkar, A., & Drosos, I. (2025). *Vibe coding: Programming through conversation with artificial intelligence* (arXiv:2506.23253). arXiv. https://doi.org/10.48550/arXiv.2506.23253

Siemens, G. (2005). Connectivism: A Learning Theory for the Digital Age. *International Journal of Instructional Technology and Distance Learning*, *2*(1), 3–10.

Siemens, G., & Downes, S. (2011). *Connectivism and Connective Knowledge 2011*. https://cck11.mooc.ca/

Zviel-Girshin, R. (2024). The Good and Bad of AI Tools in Novice Programming Education. *Education Sciences*, *14*(10), 1089. https://doi.org/10.3390/educsci14101089

APPENDIX A

Survey Questions

Demographics
- Your age
- Gender
- Have you decided what you might major in? If so, what?
- Which section of CS 100 are you in?
- Prior to learning Python in CS 100, what was your coding experience?

Learning Experience*
- The Python programming concepts introduced were easy to understand
- Applying my Excel knowledge to Python concepts made it easier when learning to code in Python
- I found the activities we completed using Python to be challenging
- Using the Google Colab environment made it easy to code in Python
- Working on Python problems improved my problem-solving skills
- Learning to code in Python helped me understand fundamental programming principles

Engagement and Interest*
- I found the activities we completed using Python to be engaging
- I found the activities we completed using Python to be fun
- After completing the Python programming activities in CS 100, I am interested in learning more about Python on my own
- After completing the Python programming activities in CS 100, I am interested in taking a Python course in college

Perceived Career Value
- Python is a useful language for data manipulation and visualization
- Knowing some coding will be a valuable skill in my career
- Having some coding skills are valuable in today's job market
- I can see practical applications for Python beyond this course

Prior or Recommended Exposure
- Using AI tools to generate Python code was new to me
- I would recommend that other sections of CS 100 incorporate learning Python as we did

Open-Ended Feedback
- Other comments about your experience learning Python in CS 100?

***Likert Scale (1 = Strongly Disagree, 5 = Strongly Agree)**

APPENDIX B.

Sample Lessons and Activities

Lesson 1: Calculations and Data Types

Coding Together

- Write a program to calculate and print the sum and product of two numbers that the user enters.
- You're planning a pizza party for a group of friends and want to ensure everyone gets their fair share of pizza. Create a program that calculates the number of pizzas needed based on the number of people and their typical pizza consumption. Assume 8 slices per pizza. Then enter the cost of a pizza to calculate the total cost of the pizza order. import math and use math.ceil() to round up!
- You're going on a road trip and want to figure out the cost of travel based on user inputs. Write a program to ask the user for the total distance of the trip in miles, the average miles per gallon (MPG) of the vehicle you're driving, the expected price of gasoline per gallon, and the expected average speed of the trip in miles per hour (MPH). Your program should calculate the total amount of fuel needed for the trip, the total cost of fuel, and the estimated travel time. Finally, it should output the total cost of the road trip.

Challenges

- Write a program that takes three numbers as input from the user, calculates their average, and prints the result.
- Write a program to ask the user to enter their name and their height in inches. Print their name followed by their height in feet and inches. Hint: use the // operator for integer division and the % operator for remainder.
- You want to paint four walls in a rectangular room (don't worry about doors and windows!) Assume a gallon of paint covers 350 square feet, and a painter can paint 150 square feet in an hour. Ask the user to enter the length, height, and width of the room in feet. Also enter the painter's hourly rate, and the cost per gallon of paint. Calculate the number of gallons you'll need, the total square feet needed for coverage, the cost of paint, the cost of labor, and the total cost to paint the room.
  Run your program for a 9 x 12 room with 8-foot ceilings, and a painter who makes $50/hour. Paint costs $25/gallon.

Lesson 2. If Statements and For Loops

Coding Together

- Enter an employee's name and the number of hours worked. Assume an hourly rate of $15, with time and a half for all hours over 40.
- A store sells a case of 6 books at a time. Each book costs $10. Write a program to calculate the cost of 6, 12, ..., 48 books.
- A user enters their t-shirt size (S/M/L) and the program converts that size to words (small, medium, large).
- Write a program to simulate flipping 100 coins, count the number of heads and tails. Use these statements to generate random numbers.

```
import random
number = random.randint(0,1) # 0 for heads, 1 for tails
```

Challenges

- A user enters an integer, the program determines whether the number is odd or even. Hint: If number % 2 is 0, the number is even. Otherwise it's odd.
- Print a table of Fahrenheit to Celsius temperature conversions for C values between 0 and 100 in multiples of 10 (0, 10, 20, ... 100). Use the formula F = (C * 9/5) + 32 to convert from C to F.

- Write a program of your own choosing that shows your understanding of several concepts from today's lesson. Be sure to include a comment at the top of your program that describes what your program is trying to compute.

Lesson 3. Charts, Graphs, and Coding with AI

Coding Together
Add the file temps.csv to your Colab Notebook in the Files area.

Temps.csv

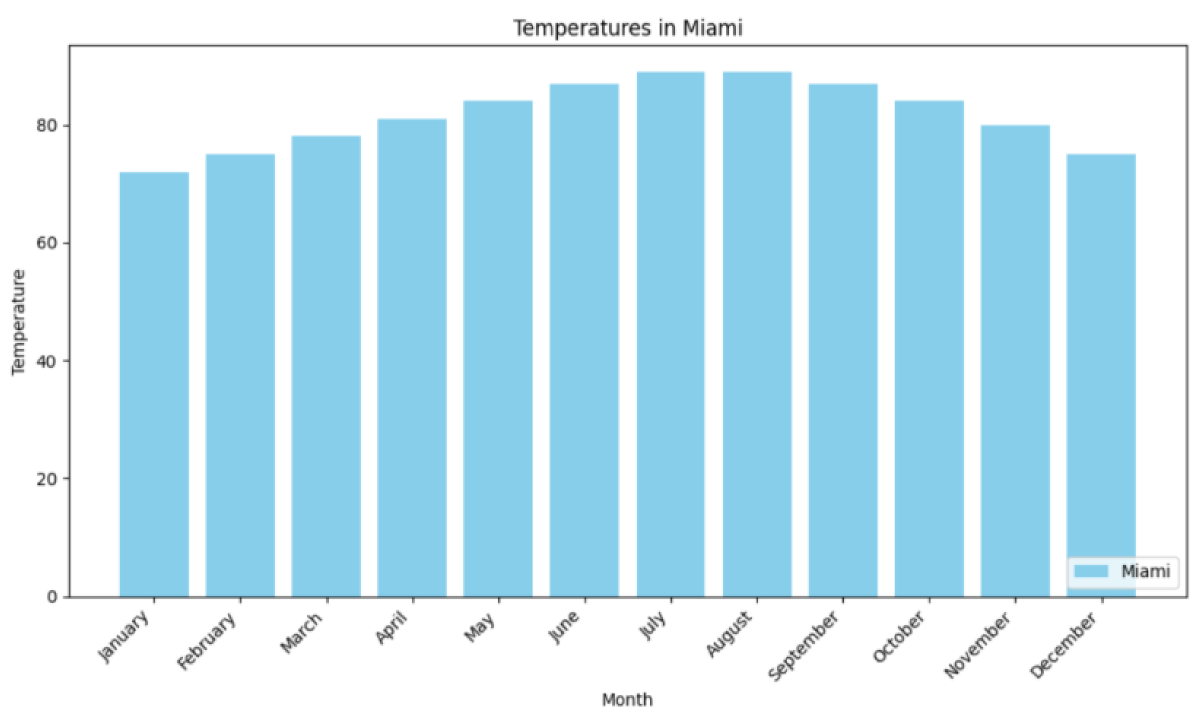
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	City	January	February	March	April	May	June	July	August	Septembe	October	Novembe	December
2	New York City	32	41	51	61	71	81	88	86	78	67	55	42
3	Los Angeles	57	59	62	65	69	73	77	79	77	73	67	59
4	Chicago	26	34	45	56	67	77	81	80	72	61	48	35
5	Houston	55	61	69	77	85	92	97	96	89	80	70	60
6	Miami	72	75	78	81	84	87	89	89	87	84	80	75

```
City,January,February,March,April,May,June,July,August,September,October,November,December
New York City,32,41,51,61,71,81,88,86,78,67,55,42
Los Angeles,57,59,62,65,69,73,77,79,77,73,67,59
Chicago,26,34,45,56,67,77,81,80,72,61,48,35
Houston,55,61,69,77,85,92,97,96,89,80,70,60
Miami,72,75,78,81,84,87,89,89,87,84,80,75
```

In Colab, create a code block, click "generate" to use AI, and enter this prompt:

> *Using temps.csv, write code to generate a bar chart showing temps from Miami for each month. use the month names as x axis labels, place a legend in lower right*

You should see a chart that looks like this when you run the Python code:



Now, create a new text block and this time use the prompt:

> *Using temps.csv generate a bar chart showing average temperature for each city make each bar a different color*

Run the code. If you get an error, it may be because Colab is assuming what the data looks like. In fact there is no "AvgTemperature" column in the data. Try to refine the prompt to give Colab more information about the data:

Challenges
Download the data file: sales.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Product	January	February	March	April	May	June	July	August	Septembe	October	November	December
2	Laptops	245	112	331	315	451	573	561	812	553	884	590	965
3	Tablets	904	317	790	747	402	720	867	733	965	267	418	955
4	Phones	854	221	769	386	396	298	546	198	377	497	313	836

```
Product,January,February,March,April,May,June,July,August,September,October,November,December
Laptops,245,112,331,315,451,573,561,812,553,884,590,965
Tablets,904,317,790,747,402,720,867,733,965,267,418,955
Phones,854,221,769,386,396,298,546,198,377,497,313,836
```

Your challenge is to ask Colab (or ChatGPT, Gemini, CoPilot, or another tool) to create four charts that you design, that show many different features of charts that you learned about in Excel. Try making pie charts, bar charts, horizontal bar charts, or line charts. Use what you know about Charts in Excel to make your prompts as detailed as possible.

If the code that Colab generates doesn't work (as sometimes happened for me), you'll see some Python code with error messages. When that happens, try another tool. Upload the data file, and re-enter your prompt. If that tool doesn't run the code, copy and paste it back into a code block in Google Colab and run it there. See if the results are any better. You may need to edit the line of code df = pd.read_csv('temps.csv') to use the name of your data file if that line shows a different file name other than the one you are using.

Ask Colab, CoPilot, Gemini, or ChatGPT to generate code to display this data as a line chart, pie chart or bar chart. If the chart does not appear when you run the code in Colab, you may need to add a line such as chart.show() at the bottom of the code to instruct Python to display the chart.

How specific do your prompts have to be to get the charts you desire? Some AI tools work better than others with this data when generating code, so if you get error messages with one tool, try using a different one. What problems do you encounter?

Exploring Student Experiences with ChatGPT in Data Analytics Education: Gender, Academic Level, and Structural Model Evidence

Mandy Yan Dang
Mandy.Dang@nau.edu

Yulei Gavin Zhang
Gavin.Zhang@nau.edu

Yiyan Stella Li
Yiyan.Li@nau.edu

Susan Williams
Susan.Williams@nau.edu

Howard Qi
Hao.Qi@nau.edu

The W. A. Franke College of Business
Northern Arizona University
Flagstaff, AZ 86011, USA

Xihui "Paul" Zhang
xzhang6@una.edu
Sanders College of Business and Technology
University of North Alabama
Florence, AL 35632, USA

Abstract

The rapid rise of generative AI technologies, particularly large language models such as ChatGPT, has introduced new opportunities for supporting student learning in data analytics education. This study investigates the use of ChatGPT as a complementary aid in a business data analytics course that teaches data mining and machine learning techniques. To accommodate students with differing levels of programming experience, three ChatGPT-assisted Python labs were incorporated alongside traditional learning tools. A structural model was developed and tested using survey responses from 260 students. The results indicate that effort expectancy, task–technology fit, and difficulty management significantly influenced learning satisfaction, and that difficulty management also significantly affected perceived learning performance. In addition, subgroup comparisons across academic levels revealed limited differences, with graduate students reporting higher task–technology fit, whereas undergraduates reporting higher perceived learning performance. Gender-based differences were more evident among undergraduates than graduates. Overall, the findings suggest that ChatGPT was positively received, although its perceived benefits varied across student subgroups. Students' perceptions of ease of use, alignment between AI assistance and analytical tasks, and their ability to manage task difficulty played central roles in shaping their learning experiences. This study provides empirical insights into how generative AI tools can be effectively integrated into data analytics education to complement existing instructional approaches.

Keywords: Generative AI, ChatGPT, Data analytics education, Data mining, Python programming

Recommended Citation: Dang, M.Y., Zhang, Y.G., Li, Y.S., Williams, S., Qi, H., Zhang, X., (2026). Exploring Student Experiences with ChatGPT in Data Analytics Education: Gender, Academic Level, and Structural Model Evidence. *Information Systems Education Journal*. v24(n2) pp 44-58. DOI# https://doi.org/10.62273/HVBV9700

Exploring Student Experiences with ChatGPT in Data Analytics Education: Gender, Academic Level, and Structural Model Evidence

Mandy Yan Dang, Yulei Gavin Zhang, Yiyan Stella Li, Susan Williams, Howard Qi and Xihui "Paul" Zhang

1. INTRODUCTION

In recent years, generative artificial intelligence (GenAI) (Corchado et al., 2023) has experienced unprecedented growth, emerging as a transformative force across multiple sectors. The rapid advancements in large language models (LLMs) (Corchado et al., 2023; Myers et al., 2024) and multimodal systems have enabled GenAI to be widely deployed in various domains, such as healthcare, finance, law, and education (Yang et al., 2024). For instance, in the financial domain, GenAI has been developed and adopted in enhancing tasks such as financial modeling, risk assessment, fraud detection, and customer service (Remolina, 2024). In addition, GenAI has also been utilized in the law sector to streamline legal research, automate document analysis, and improve case management, leading to increased efficiency and productivity (Wilkins, 2024). In education, GenAI has facilitated personalized learning experiences and adaptive teaching methods, contributing to improved student engagement and learning outcomes (Denny et al., 2024).

Alongside advancements in GenAI, the field of data analytics has also experienced significant growth in recent years, becoming an essential component of modern business operations. Organizations across various industries increasingly rely on data-driven decision-making to enhance efficiency, competitiveness, and innovation. This surge in demand has led to a considerable expansion of data analytics programs in higher education. Many business schools across the US have responded by developing interdisciplinary curricula that combine critical technical skills with essential business knowledge, preparing graduates to meet the evolving needs of the industry (Olsen et al., 2022).

Building on both perspectives, this study explores the synergy between GenAI and data analytics education. While a substantial body of empirical research has recently emerged analyzing the impact of GenAI tools, primarily in the domains of foreign language education and programming, relatively fewer studies have focused on data analytics. Therefore, this study specifically aims to examine how GenAI tools, such as ChatGPT (Corchado et al., 2023; OpenAI, 2022), can be leveraged to assist students in overcoming technical challenges associated with data analytics tasks, particularly within the domain of data mining. By enabling natural language-driven programming support, GenAI has the potential to lower barriers for students with limited coding experience, thereby enhancing their engagement with complex analytical methods. Accordingly, this study presents and discusses the integration of ChatGPT in a data analytics course to support student learning of data mining algorithms using Python. In addition, we developed and tested a structural model to examine how key learning factors (i.e., learning effort expectancy, task–technology fit, and difficulty management) influence students' learning satisfaction and perceived learning performance in this context. We also conducted comparative analyses of student perceptions across gender and academic levels.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on the use of GenAI in education. Section 3 describes the design of the ChatGPT-assisted learning tasks in the data analytics course. Section 4 presents the research model and hypothesis development. Section 5 first reports the structural model testing results and then presents detailed comparative analyses across gender and academic levels. Finally, Section 6 discusses the research contributions and implications, and the study's limitations and suggestions for future research.

2. RELATED LITERATURE

In recent years, a growing body of research has emerged examining the influence of GenAI tools on higher education (Xia et al., 2024). As the adoption of large language models and related technologies accelerates, scholars have increasingly investigated both the positive and negative impacts of these tools on teaching, learning, and academic integrity (Denny et al., 2024; Kakhki et al., 2024). Some studies focus on broader discussions of how GenAI alters educational practices and student engagement, highlighting opportunities for personalized learning and concerns about misuse and overreliance (Gill et al., 2024; Kakhki et al., 2024). Others take a more focused approach by exploring the practical application of specific GenAI technologies, such as ChatGPT and other language models, in supporting teaching tasks, improving student writing, and assisting with coding and data analysis (Kasneci et al., 2023).

For instance, as a comprehensive guidance study, Kakhki et al. (2024) investigated the affordances of ChatGPT in higher education and examined how AI technologies could reshape learning functions within academic institutions. Using a grounded theory approach, the study analyzed academic panel discussions to examine perceptions of ChatGPT and related AI tools in higher education. The study presents its findings through a framework consisting of four categories of affordances: (1) mitigating challenges in traditional learning environments, (2) enhancing effective educational practices, (3) transforming traditional learning approaches, and (4) negatively impacting current effective educational practices.

As another guidance paper, Gill et al. (2024) discussed the wide range of benefits that ChatGPT could bring to the education sector, as well as its potential risks. The authors categorized the transformative effects of ChatGPT on modern education into six dimensions: (1) educating with ChatGPT, (2) important ethics, (3) transforming online education, (4) higher education risks, (5) treatment required immediately, and (6) potential challenges.

In addition to examining the overall impact of ChatGPT in higher education, a recent body of research has emerged investigating how ChatGPT can assist student learning across various domains, such as foreign language acquisition (Koraishi, 2023; Warschauer et al., 2023), computer programming (Kosar et al., 2024; Yilmaz & Yilmaz, 2023), mathematics (Sánchez-Ruiz et al., 2023), business writing (Kétyi et al., 2025), data science (Zheng, 2023), and data analytics (Zhong & Kim, 2024).

For instance, Warschauer et al. (2023) investigated the use of AI-generated text, including ChatGPT, in supporting second language (L2) English writing. Through surveys and interviews with students and instructors, the study identified key benefits such as reducing writing anxiety, providing immediate feedback, and offering model texts to enhance L2 learning. However, it also noted risks, including overreliance on AI outputs, inaccuracies, and diminished critical thinking and independent writing skills. The authors emphasize the need for structured pedagogical strategies to ensure that AI tools are used as a complement, not a replacement, to traditional L2 writing instruction.

Similarly, Koraishi (2023) investigated the application of ChatGPT in English as a Foreign Language (EFL) education, with an emphasis on material development and assessment. The study demonstrated how ChatGPT can aid teachers by generating customized reading texts, adapting materials across different proficiency levels, integrating targeted vocabulary, and creating assessments and lesson plans. These capabilities have been shown to reduce teacher workload and provide more personalized learning experiences. However, the author also emphasizes the need for teacher supervision, as the tool occasionally produces inaccuracies or content requiring refinement.

In another study examining ChatGPT's role in computer science education, Kosar et al. (2024) investigated the impact of ChatGPT on novice programmers in an undergraduate object-oriented programming course. The study involved 182 students, divided into a control group and an experimental group using ChatGPT for programming assignments. Results showed no significant performance differences between the two groups, suggesting ChatGPT, when properly guided, does not compromise or improve learning performance. The authors note that ChatGPT can reduce frustration and provide immediate assistance, but emphasize that structured instructional strategies are essential to prevent overreliance and ensure the development of critical coding skills.

In another study, Yilmaz and Yilmaz (2023) explored student perceptions of ChatGPT as a learning tool in an Object-Oriented Programming II course involving 41 undergraduates over eight weeks. Students highlighted benefits such as rapid responses, assistance with debugging, improved problem-solving skills, and increased confidence. However, some noted risks of overreliance, occasional inaccuracies, and reduced independent thinking. The authors conclude that, with proper instructional design and supervision, ChatGPT can be a valuable supplement to programming education.

In mathematics education, Sánchez-Ruiz et al.

(2023) evaluated the impact of ChatGPT in a blended learning mathematics course for engineering students. Students reported that ChatGPT supported their understanding of concepts and problem-solving, although trust was higher for theoretical explanations than for numerical accuracy. In business writing education, Kétyi et al. (2025) explored ChatGPT's use for drafting business letters and dialogues, with students reporting improved writing efficiency and confidence alongside concerns about overreliance and content accuracy. As to data science education, Zheng (2023) examined ChatGPT's use in a graduate-level course and found that it aided learning in areas such as coding explanations and tool recommendations, while limitations remained in fostering critical thinking and problem-solving without human guidance.

However, to the best of our knowledge, relatively few studies have specifically investigated the role of GenAI tools, such as ChatGPT, in supporting student learning in data analytics, particularly in relation to data mining algorithms.

Through an extensive review of the literature, we identified one relevant study conducted by Zhong and Kim (2024). Their study examined the use of ChatGPT to support business students in learning logistic regression in R. By guiding students through data preparation, model building, and evaluation using AI-generated code within the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework, the study aimed to help lower technical barriers and enhance students' data analytics and problem-solving skills.

Our study differs from theirs in three key ways: (1) we focus on multiple data mining algorithms rather than solely on logistic regression; (2) our course uses Python instead of R, both of which are widely adopted programming languages in business analytics education; and (3) while their study presents a teaching case with detailed assignment information, our study also provides empirical results to demonstrate the effectiveness of the proposed instructional design.

3. THE DATA ANALYTICS COURSE AND TASK DESIGN

The Data Analytics Course

Our data analytics course is a technical course focused on teaching students data mining and machine learning techniques, as well as their applications in solving business problems. The major data mining algorithms covered include linear regression, logistic regression, association analysis, k-nearest neighbors (k-NN), decision trees, artificial neural networks, and clustering. The course is offered to both senior undergraduate and graduate students. Specifically, it is a required course for all business analytics, information systems, and marketing undergraduate majors, and an elective for other majors within the College of Business. At the master's level, it is a required course for business analytics students and an elective available to students from other colleges, such as engineering, including those majoring in computer science, information technology, electrical engineering, and others. Both undergraduate and graduate students use the same learning materials (textbooks and lectures) and are assessed using the same tools (weekly quizzes and lab projects). However, master's students are also required to complete a more comprehensive, term-long project, which is not required for undergraduates.

Given the diverse academic backgrounds of students enrolled in the course, not all students possess prior programming experience. To accommodate this variability and ensure equitable learning outcomes, the course has traditionally incorporated RapidMiner, a user-friendly data analysis software platform (https://academy.rapidminer.com/), for regular lab projects. RapidMiner's intuitive drag-and-drop interface allows students to focus on understanding data mining concepts and workflows without the steep learning curve associated with coding. However, we also recognize the value of providing students with exposure to data analytics using programming languages, such as Python, to broaden their skill set and prepare them for industry expectations. While software tools like RapidMiner offer accessibility and ease of use, they can limit students' flexibility in handling complex data tasks or customizing algorithms. Conversely, programming approaches offer greater control, scalability, and problem-solving opportunities, but they are often associated with a higher initial learning barrier. Therefore, in our course, we aim to combine both perspectives to leverage RapidMiner's strengths in helping students quickly grasp the fundamental concepts of data mining algorithms, while also giving them experience in performing data analytics using Python.

To implement this idea, we decided to leverage the capabilities of a GenAI tool, specifically ChatGPT, to support student learning and engagement with programming-based data analytics. We designed a series of three targeted lab sessions, distributed throughout the semester, to provide students with progressive exposure to data analytics using Python. These labs were strategically placed at the beginning, middle, and end of the course to gradually introduce students to Python-based data analytics while reinforcing key data mining concepts. The first ChatGPT-assisted Python data analytics lab was conducted early in the semester and focused on multiple linear regression. The second lab took place in the middle of the semester and addressed association analysis, while the last lab, conducted towards the end of the semester, focused on artificial neural networks. We selected these three algorithms because they were introduced at different points in the course and, more importantly, they represent distinct categories of data mining algorithms.

To design the three ChatGPT-assisted Python labs, we utilized Google Colab, a cloud-based programming environment that allows users to write and execute Python code. Following the approach of Zhong and Kim (2024), we provided students with structured ChatGPT prompts to guide them in generating Python code for data analytics tasks step by step — from data understanding and exploration to model development and results interpretation. Students were instructed to use two web browsers during each lab session: one to access ChatGPT and input the prompts to obtain relevant code, and the other to open Google Colab, where they would paste, run, and explore the code and interpret the results. In addition to running the code, students were required to read and understand the detailed explanations of the logic and functions that ChatGPT provided alongside the generated Python code. They were also encouraged to make necessary adjustments and minor modifications to the prompts to ensure the successful completion of the lab tasks.

Due to space limitations, we present only the final ChatGPT-assisted Python data analytics lab below, with the hope that it offers inspirational insights for educators in data analytics.

ChatGPT-Assisted Python Data Analytics Lab on Artificial Neural Networks

The dataset used in this lab is the Diagnostic Wisconsin Breast Cancer Database, which contains 30 features and 569 instances. The features are derived from a digitized image of a fine needle aspirate (FNA) of a breast mass and describe various characteristics of the cell nuclei present in the image.

Part 1: Data Understanding and Exploration

1) Import the data file onto Google Colab.
2) Import the necessary Python libraries for data analysis and visualization. Prompt: ***Import the libraries for data analysis and visualization.***
3) Read data from the data file. Prompt: ***Read data from the data file.***
4) After loading the data, describe the dataset step by step as follows:
   a. Prompt: ***Display the number of records and the number of attributes in the dataset.***
   b. Prompt: ***Display the names of the attributes in the dataset.***
   c. Prompt: ***Calculate the minimum, maximum, average, and standard deviation of each attribute in the dataset.***
5) Check for missing values in the dataset. Prompt: ***Check if there is any missing value in each column of the dataset.***
6) Explore the distribution of the dependent variable on a histogram. Prompt: ***Visualize the dependent variable on a histogram.***
7) Visualize the data using a parallel plot based on all features, excluding the dependent variable. Use color to differentiate between the two categories, based on the dependent variable. Prompt: ***Visualize the data using a parallel coordinates plot for all features, excluding the dependent variable. Use color to differentiate between the two categories based on the dependent variable.***
8) Calculate the correlation values between each feature and the dependent variable, and sort the results in descending order. Prompt: ***Calculate correlation values between each feature and the dependent variable, and sort the results in descending order.***
9) To further explore the relationships across variables, visualize the correlation matrix on a heat map. Prompt: ***Create a heat map based on the correlation matrix.***

Part 2: Model Development and Evaluation

Part 2 focuses on creating the artificial neural network model. To do this, students first need to split the entire dataset into a training dataset (containing 75% of the data points) and a testing dataset (containing the remaining 25%). They will then use the training dataset to build the model and the testing dataset to evaluate the model's performance.

1) Split the entire dataset into a training

dataset (containing 75% of the data points) and a testing dataset (containing the remaining 25%). Prompt: ***Split the entire dataset into a training dataset (containing 75% of the data points) and a testing dataset (containing the remaining 25%).***

2) Use the training dataset to build an artificial neural network model with a total of 3 layers: an input layer, a hidden layer, and an output layer. The input layer should have 30 nodes, and the output layer should have 1 node. The hidden layer is set to have 16 nodes in this example. Prompt: ***Use the training dataset to build an artificial neural network model with a total of 3 layers: an input layer, a hidden layer, and an output layer. The input layer should have 30 nodes, and the output layer should have 1 node. Make the hidden layer with 16 nodes.***
3) Use the testing dataset to evaluate the model's performance. Calculate the confusion matrix and accuracy. Prompt: ***Use the testing dataset to evaluate the model's performance. Calculate the confusion matrix and accuracy.***

Part 3: Further Exploration With Model Comparisons

To gain an in-depth understanding through further exploration, students are asked to use the same training and testing datasets to create another classifier using the decision tree algorithm and evaluate its performance. They are then asked to compare which algorithm, the artificial neural network or the decision tree, yields a higher accuracy. After that, they are asked to further explore a third model — a 4-layer artificial neural network model — to once again compare the performance across the three models.

1) Create a decision tree model using the same training dataset, and then evaluate the model's performance using the same testing dataset. Prompt: ***Create a decision tree model using the same training dataset, and then evaluate the model's performance using the same testing dataset. Calculate the confusion matrix and accuracy.***
2) **Discussion question:** Based on the results from the two models, which one has a higher accuracy on this dataset: the artificial neural network or the decision tree model?
3) Using the same training dataset, build another artificial neural network model with a total of 4 layers: an input layer, two hidden layers, and an output layer. The input layer should have 30 nodes, and the output layer should have 1 node. In this example, use 20 nodes for the first hidden layer and 16 nodes for the second hidden layer. Then, use the same testing dataset to evaluate its performance. Prompt: ***Use the same training dataset to build another artificial neural network model with a total of 4 layers: an input layer, two hidden layers, and an output layer. The input layer should have 30 nodes, and the output layer should have 1 node. Make the first hidden layer with 20 nodes and the second one with 16 nodes. Then, evaluate the model's performance using the same testing dataset. Calculate the confusion matrix and accuracy.***
4) **Discussion question**: So far, we have created 3 models: a 3-layer artificial neural network, a decision tree, and a 4-layer artificial neural network. Which one has the highest accuracy on this dataset, and what is that accuracy value?
5) **Discussion question**: What insights have you gained from this lab and the comparisons among the performances of models?

4. RESEARCH MODEL AND HYPOTHESIS DEVELOPMENT

To assess the effectiveness of the task design and its impact on student learning, we draw on perspectives from technology adoption, task–technology fit, and learning outcome research. Specifically, we include effort expectancy, grounded in technology acceptance theory (Venkatesh et al., 2003), to capture students' perceptions of ease of use when interacting with AI-supported tools. Task–technology fit (Goodhue & Thompson, 1995), derived from cognitive fit theory, reflects the alignment between AI assistance and analytical task requirements. Difficulty management (Wall & Knapp, 2014) is incorporated to represent the inherent complexity and learning curve associated with programming-based data analytics tasks. Finally, learning satisfaction and perceived learning performance (Bere, 2018) serve as commonly used outcome measures in IS education research.

Although these constructs do not form an exhaustive set of influencing factors, together they form a theoretically informed nomological network that enables us to examine how perceptions of tool usability, task alignment, and

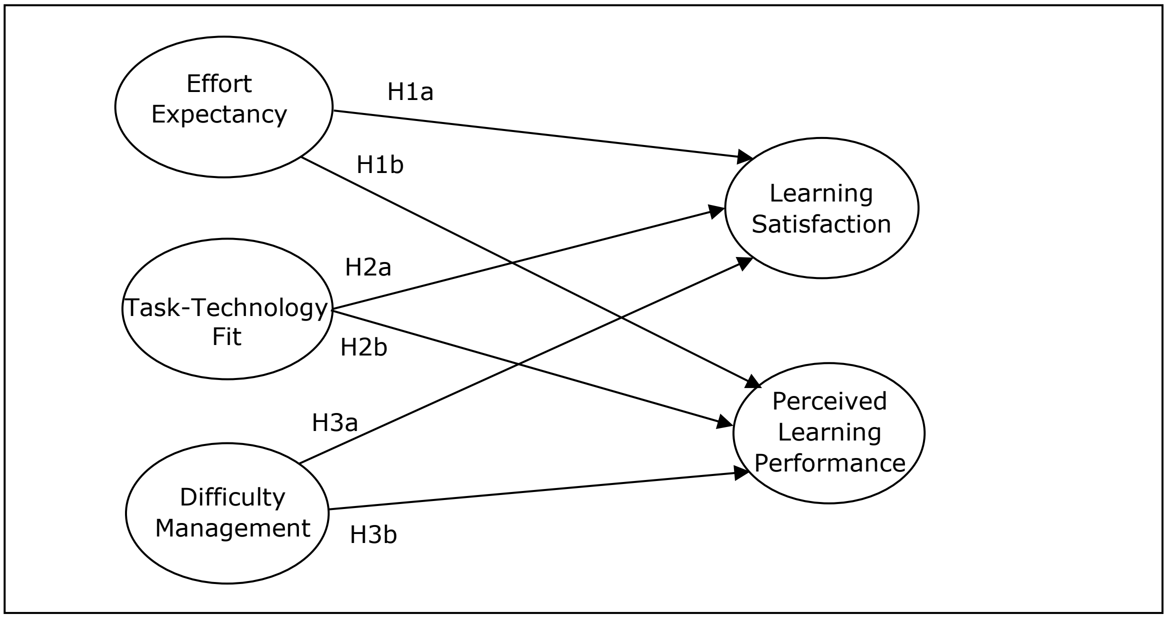


Figure 1: Research Model

difficulty management shape students' learning experiences in this context.

Effort expectancy, a core construct of the UTAUT (Unified Theory of Acceptance and Use of Technology) model (Venkatesh et al., 2003), refers to the degree of ease associated with using a particular system. Prior research in computing and online education has shown that effort expectancy could significantly shape students' adoption and use of learning support systems (Abdullahi et al., 2003; Suleiman et al., 2018). More recently, Sabeh (2024) found that effort expectancy played a significant role in IT students' intention to adopt ChatGPT for academic purposes. In our context, ChatGPT provides step-by-step assistance that enables students, especially those with limited Python experience and knowledge, to complete data analytics tasks that might otherwise be too challenging. By reducing the perceived effort required to learn and apply Python, ChatGPT may enhance students' satisfaction with the learning process and improve their perceived performance. Therefore, we propose the following hypotheses:

> H1a: Effort expectancy has a positive impact on learning satisfaction when performing AI-assisted Python learning for data analytics.
> H1b: Effort expectancy has a positive impact on perceived learning performance when performing AI-assisted Python learning for data analytics.

Task–technology fit (TTF) suggests that an information technology tool is more likely to enhance an individual's task performance when its capabilities align closely with the requirements of the task being performed (Goodhue & Thompson, 1995). Prior research consistently shows that higher levels of TTF can strengthen student learning outcomes, including learning satisfaction and continued use of learning support systems (Chen & Roldan, 2021; Dang & Zhang, 2022; Lin, 2012), as well as perceived performance (Bere, 2018).

In the context of our study, ChatGPT offers meaningful support by helping students understand and apply Python for data analytics, particularly for those with limited prior programming experience. This alignment between the assistance provided by ChatGPT and the demands of the learning tasks could form a strong level of task–technology fit. This may further facilitate smoother task completion, reduce friction in the learning process, and ultimately enhance both satisfaction and perceived performance. Therefore, we posit the following hypotheses:

> H2a: Task-technology fit has a positive impact on learning satisfaction when performing AI-assisted Python learning for data analytics.
> H2b: Task-technology fit has a positive impact on perceived learning performance when performing AI-assisted Python learning for data analytics.

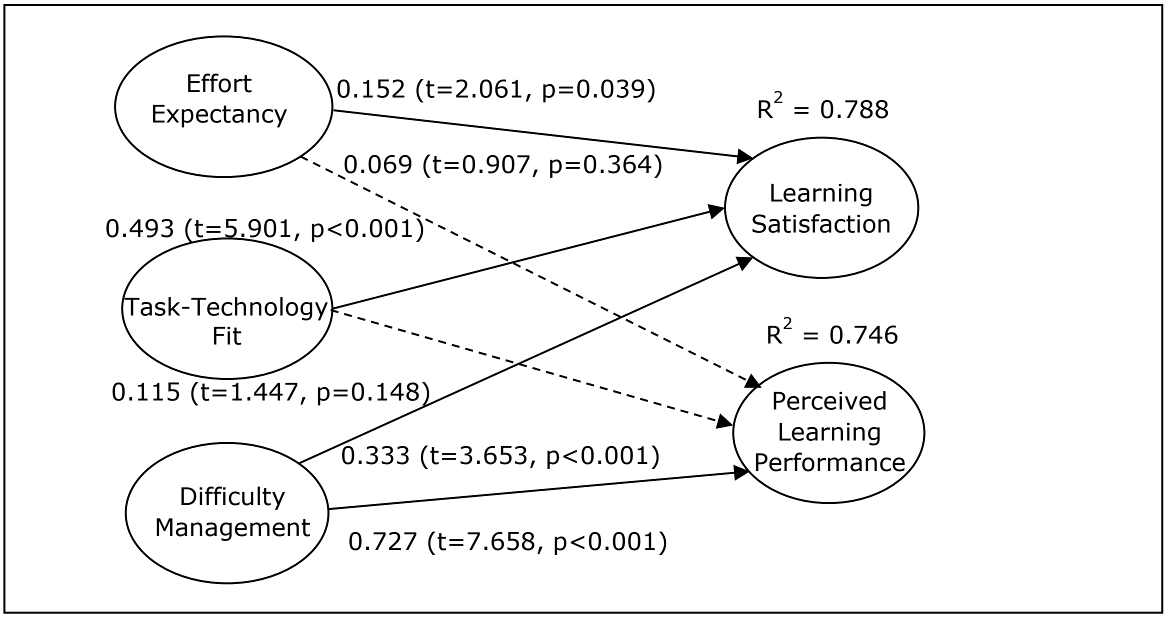


Figure 2: Research Model Test Results

Difficulty management refers to students' ability to proactively handle the perceived difficulty associated with a learning task (Wall & Knapp, 2014). In their study on difficulty management in information systems courses, Wall and Knapp (2014) found that students who managed course difficulty more effectively reported higher perceived learning outcomes. Similarly, other research has shown that the ability to cope with course difficulty is positively associated with overall learning outcomes (Dang et al., 2023).

In our context, both Python programming and machine learning–based data analytics can be challenging subjects for many students. Effective difficulty management may therefore play an essential role in helping students navigate the learning process. When students feel capable of handling the difficulty of the material with the support of ChatGPT, they may experience greater satisfaction and report higher perceived performance. Therefore, we hypothesize that:

H3a: Difficulty management has a positive impact on learning satisfaction when performing AI-assisted Python learning for data analytics.

H3b: Difficulty management has a positive impact on perceived learning performance when performing AI-assisted Python learning for data analytics.

Figure 1 shows the research model with hypotheses.

5. EMPIRICAL RESULTS

Structural Model Results

In this sub-section, we examine how effort expectancy, task-technology fit and difficulty management influence students' learning satisfaction and perceived learning performance.

To do that, a survey was conducted. After obtaining the IRB (Institutional Review Board) approval, a survey invitation was sent to all students enrolled in various sections of the course approximately one week before the end of the semester, following the completion of the final lab. Participation was voluntary and anonymous, and a small amount of extra credit (approximately 1.5% of the total possible course points) was offered as an incentive for completing the survey. In total, the survey invitation was sent to 410 students; 264 students participated, and 260 completed the survey, yielding a response rate of 64.39% and a completion rate of 63.41%. The average age of participants who completed the survey was 23.45 years.

The senior undergraduate and graduate students were enrolled in separate classroom sections. Although the courses have different prefixes, they share the same learning materials and assessment methods, with the exception that the master's-level course includes a comprehensive semester-long project. In the year the study was conducted, there were 10 sections in total, five undergraduate and five graduate sections, taught

	Composite Reliability	AVE	DM	EE	PLP	SAT	TTF
DM	0.937	0.887	**0.942**				
EE	0.929	0.812	0.662	**0.901**			
PLP	0.922	0.852	0.824	0.644	**0.923**		
SAT	0.948	0.863	0.846	0.738	0.806	**0.929**	
TTF	0.907	0.835	0.819	0.743	0.777	0.814	**0.914**

Note: Diagonal values (in bold) represent the square root of the average variance extracted (AVE). Off-diagonal values indicate the correlations among constructs.

Table 2: Internal Consistency and Validity Test Results

by two instructors, each teaching a combination of undergraduate and graduate sections. All other lab sessions followed the regular course format using RapidMiner, and no Python-related content had been introduced prior to the three dedicated Python lab sessions. The instructors closely coordinated course delivery to ensure consistency across sections.

For four out of the five constructs, we adopted measurement items from existing literature with modifications to fit the context of our study. Specifically, the measure for learning effort expectancy was adapted from Sabeh (2024); the measure for task-technology fit was adapted from Chen et al. (2023). To assess difficulty management, we drew on the conceptual framework of Wall and Knapp (2014) and developed our own items. The measure for learning satisfaction was adopted from Almulla (2024), and the measure for perceived learning performance was adapted from Islam (2013). All questionnaire items were rated on a 7-point Likert scale, ranging from 1 for "strongly disagree" to 7 for "strongly agree." Appendix A lists all the measurement items.

To test the proposed research model and examine the associated hypotheses, we applied partial least squares structural equation modeling (PLS-SEM) using SmartPLS 4.0. Tables 1 and 2 present the reliability and validity assessment results. Reliability was evaluated using item loadings and Cronbach's alpha, while validity was examined through the square root of AVE, and interconstruct correlations. Internal consistency was assessed using composite reliability.

As shown in Table 1, the Cronbach's alpha values for all constructs exceed the recommended threshold of 0.70, demonstrating strong internal reliability. In addition, all item loadings were statistically significant at the $p < 0.0001$ level, further confirming the reliability of the measurement model.

As shown in Table 2, all composite reliability values exceed the recommended threshold of 0.70, indicating strong internal consistency for each construct. Likewise, all average variance extracted (AVE) values are above the suggested minimum of 0.50, consistent with the implied square-root-of-AVE guideline of 0.707. These results demonstrate solid convergent validity. In addition, the square root of the AVE for each construct is greater than its correlations with other constructs, providing evidence of satisfactory discriminant validity.

	Cronbach's Alpha	Item	Loading	t-stats/p-value
DM	0.936	DM1	0.924	<0.0001
		DM2	0.954	<0.0001
		DM3	0.948	<0.0001
EE	0.922	EE1	0.843	<0.0001
		EE2	0.925	<0.0001
		EE3	0.917	<0.0001
		EE4	0.917	<0.0001
PLP	0.947	PLP1	0.938	<0.0001
		PLP2	0.962	<0.0001
		PLP3	0.867	<0.0001
SAT	0.913	SAT1	0.914	<0.0001
		SAT2	0.945	<0.0001
		SAT3	0.932	<0.0001
		SAT4	0.924	<0.0001
TTF	0.901	TTF1	0.929	<0.0001
		TTF2	0.885	<0.0001
		TTF3	0.926	<0.0001

Table 1: Reliability Test Results

The inner model testing results are illustrated in Figure 2. The analysis shows that all three learning factors (i.e., effort expectancy, task–technology fit, and difficulty management) significantly influence learning satisfaction, supporting H1a, H2a, and H3a. In addition, difficulty management exerts a significant positive effect on perceived learning performance, providing support for H3b.

These results suggest that students' satisfaction with AI-assisted Python learning is shaped by both the perceived ease of using ChatGPT and the degree to which the tool aligns with their analytical tasks, as well as their confidence in handling the difficulty of the material. Moreover, students who feel more capable of managing the technical challenges of Python and machine learning are more likely to view themselves as performing well, with relatively little influence from effort expectancy or task–technology fit. This highlights the central role of difficulty coping in shaping students' perceptions of their learning outcomes in rigorous, AI-supported analytics environments.

Comparative Analyses Across Gender and Academic Levels

This subsection presents comparative analyses across gender and academic levels. Among all 260 participants, 134 identified as male, 121 as female, 3 as non-binary, and 2 preferred not to specify their gender. Additionally, 117 were undergraduate students and 143 were graduate students.

Table 3 summarizes the descriptive statistics of student responses across the five measured constructs. Overall, the results indicate a positive evaluation of the ChatGPT-assisted Python data analytics labs. The mean scores for all assessment dimensions exceeded the midpoint value of 4 on the 7-point Likert scale, suggesting generally favorable student perceptions. Specifically, perceived learning performance (M = 5.578, SD = 1.267) and learning effort expectancy (M = 5.507, SD = 1.251) received the highest ratings, indicating that students felt ChatGPT meaningfully supported their learning process and reduced the effort needed to conduct data analytics tasks using Python. The other dimensions also demonstrated strong positive responses, with means ranging from 5.341 to 5.501 across task-technology fit, difficulty management, and learning satisfaction. These findings collectively indicate the overall effectiveness of integrating ChatGPT as a supplementary instructional tool in data analytics education.

Assessment	Mean/Std dev
Difficulty Management (DM)	5.501/1.261
Learning Effort Expectancy (EE)	5.507/1.251
Perceived Learning Performance (PLP)	5.578/1.267
Learning Satisfaction (SAT)	5.479/1.323
Task-Technology Fit (TTF)	5.341/1.342

Table 3: Overall Statistics (N=260)

To further assess the effectiveness of our instructional design, we also compared the results between undergraduate and graduate students. Within each group, we then conducted additional comparisons between male and female students. Table 4 presents the results of the comparison between undergraduate and graduate students across the five measured dimensions.

As shown in Table 4, overall, both groups reported similarly positive perceptions of the ChatGPT-assisted labs, with all mean scores exceeding the midpoint of the 7-point Likert scale. Statistically significant differences were found in task-technology fit (TTF) and perceived learning performance (PLP). Graduate students rated TTF significantly higher than undergraduates (M = 5.422 vs. M = 5.242; p = 0.031), indicating a greater perceived alignment between ChatGPT and their learning tasks. Conversely, undergraduates reported significantly higher perceived learning performance (M = 5.664 vs. M = 5.508; p = 0.044). These contrasting results may reflect differences in academic experience and learning needs. Graduate students, often possessing more familiarity with complex tools and independent learning strategies, may have found ChatGPT to align more effectively with their task requirements, resulting in higher task-technology fit scores. In contrast, undergraduate students, who generally have less prior experience with data mining and programming, may have perceived greater learning gains from the structured support provided by ChatGPT, contributing to higher perceived learning performance. No significant differences were observed for learning effort expectancy, difficulty management, or learning satisfaction between the two groups.

	Under (N=117) M/SD	**Master (N=143) M/SD**	**p-value**
DM	5.507/1.185	5.497/1.321	0.453
EE	5.498/1.092	5.514/1.368	0.418
PLP	5.664/1.134	5.508/1.363	0.044*
SAT	5.513/1.276	5.451/1.360	0.454
TTF	5.242/1.338	5.422/1.342	0.031*

Note: * indicates statistically significant at 0.05.

Table 4: Undergraduate vs. Graduate Students

We further examined gender differences within both the undergraduate and graduate student groups. Tables 5 and 6 present the results for each group, respectively.

As illustrated in Table 5, female undergraduate students reported significantly higher scores than male undergraduate students on all dimensions, with all differences reaching statistical significance ($p < 0.05$). Specifically, female undergraduate students rated learning effort expectancy (EE), task-technology fit (TTF), difficulty management (DM), learning satisfaction

(SAT), and perceived learning performance (PLP) more positively than male undergraduate students. These findings suggest that female undergraduate students perceived greater benefits and support from the ChatGPT-assisted labs compared to their male peers. One possible explanation is that female undergraduate students, who may have had lower initial confidence or less prior exposure to programming tasks, found the structured and supportive nature of the ChatGPT-assisted approach particularly valuable. The AI-driven guidance may have helped reduce anxiety and increase engagement, contributing to their overall positive evaluation of the learning experience.

	Under-male (N=59) M/SD	**Under-female (N=55) M/SD**	**p-value**
DM	5.181/1.328	5.842/0.930	<0.001*
EE	5.364/1.190	5.623/0.983	0.006*
PLP	5.463/1.206	5.867/1.039	0.001*
SAT	5.233/1.429	5.800/1.032	<0.001*
TTF	4.949/1.505	5.539/1.068	<0.001*

Note: The 3 non-binary students were excluded in this table. * indicates statistically significant at 0.05.

Table 5: Undergraduate Male vs. Female Students

	Master-male (N=75) M/SD	**Master-female (N=66) M/SD**	**p-value**
DM	5.293/1.571	5.611/1.190	0.041*
EE	5.362/1.603	5.527/1.328	0.480
PLP	5.361/1.579	5.586/1.279	0.191
SAT	5.194/1.671	5.587/1.183	0.023*
TTF	5.265/1.623	5.535/1.212	0.086

Note: The 2 students who preferred not to specify their gender were excluded in this table. * indicates statistically significant at 0.05.

Table 6: Graduate Male vs. Female Students

As indicated in Table 6, while female graduate students reported higher mean scores on all dimensions, statistically significant differences were observed only for difficulty management (DM; p = 0.041) and learning satisfaction (SAT; p = 0.023). Female graduate students rated DM and SAT more positively, suggesting they experienced greater confidence in managing the technical demands of the labs and were more satisfied with the learning experience. No significant differences were found for learning effort expectancy, task-technology fit, or perceived learning performance. These results may indicate that at the graduate level, both male and female students possess relatively high baseline skills and comfort with learning technologies, minimizing gender disparities. However, female students may still have particularly benefited from the structured support and clear guidance provided by the ChatGPT-assisted labs, which could explain their higher satisfaction and better perceived ability to manage lab challenges.

6. CONCLUSIONS

Research Contributions and Implications

This study makes several key contributions to the literature on data analytics education and the pedagogical application of GenAI tools. First, it advances our understanding of how GenAI tools can be systematically leveraged to support students with diverse academic and technical backgrounds in learning highly technical content such as data mining and machine learning using Python. Data analytics courses often present a significant learning barrier for students with limited programming experience. Our approach demonstrates that AI-assisted coding guidance, integrated within a supportive instructional framework, can reduce these barriers and provide an equitable learning experience. The positive student perceptions across learning effort, satisfaction, and perceived learning performance highlight the potential of GenAI to enhance access to complex analytics education.

Second, the study contributes a novel instructional design that differs from prior research. Building on earlier exploratory efforts, we implemented a staged design featuring three distinct ChatGPT-assisted labs spaced strategically throughout the semester. This design exposed students to progressively complex data mining algorithms (including multiple linear regression, association analysis, and artificial neural networks) rather than focusing on a single technique as in some earlier studies (e.g., Zhong & Kim, 2024). The inclusion of both traditional data mining tools (RapidMiner) and AI-supported Python coding offered students a dual learning pathway that accommodates varying skill levels while promoting deeper understanding of data analytics concepts and methodologies. This may serve as a practical framework that can be adapted to other technical and business education contexts.

Third, this study contributes theoretically by developing and testing a structural model that explains how key learning factors shape students' overall learning experience in an AI-assisted analytics environment. The results demonstrate that effort expectancy, task–technology fit, and difficulty management can significantly influence learning satisfaction, while difficulty management also significantly impacts perceived learning performance. These findings extend existing research on AI-assisted learning by identifying the specific factors via which GenAI tools support student outcomes in data analytics education. By integrating structural model analysis with instructional design and empirical comparison across student subgroups, this study provides a more comprehensive understanding of how GenAI tools can be effectively incorporated into technical coursework and lays the groundwork for future theory-driven investigations in this field.

Finally, our empirical findings offer actionable insights for data analytics educators. The survey results provide evidence that most students responded positively to the integrated instructional approach, with overall ratings above the scale midpoint across five learning dimensions. Additionally, subgroup analysis revealed meaningful differences between student groups, such as stronger positive perceptions among female undergraduate students. These insights suggest the importance of considering learner diversity when integrating AI tools into data analytics curricula. For example, instructors may consider providing differentiated levels of scaffolding, offering more structured guidance and example prompts for novice programmers, while encouraging more advanced students to engage in exploratory and independent coding tasks to maximize learning outcomes for all groups.

Beyond the empirical findings, several instructional insights emerged from implementing the AI-facilitated labs. First, structured prompt scaffolding appeared to support students' engagement with Python syntax and debugging processes, as students were guided to interpret and refine AI-generated code rather than simply execute it. Second, guided AI interaction encouraged iterative problem-solving, with students testing, modifying, and comparing outputs instead of passively copying responses.
For future iterations of the course, we plan to incorporate more direct assessments of independent coding competency, such as brief in-class coding tasks without AI assistance, to better evaluate students' actual learning outcomes.

Limitations and Future Research Directions
This study has several limitations. First, it was conducted at a single institution, which may limit the generalizability of the findings. Second, the research relied on self-reported survey data, which may be subject to bias. Third, while the study focused on students' perceptions, actual learning performance and long-term skill retention were not measured. Future research should explore longitudinal impacts of GenAI-assisted learning and examine outcomes across diverse institutional and cultural contexts. Experimental studies comparing AI-assisted instruction with traditional methods could also provide deeper insights into the pedagogical value and limitations of GenAI in data analytics education. Additionally, this study compared undergraduate and graduate students; future research may further explore comparisons based on prior coding experience and prior GenAI exposure. Future research may also conduct in-depth interviews to gain deeper insights into students' learning experiences. Finally, although this study introduced and tested a structural model explaining how key learning factors shape student satisfaction and perceived performance, the model included only a subset of potential determinants relevant to AI-assisted learning. Future research could expand the model by incorporating additional constructs, such as AI literacy or GenAI self-efficacy. Testing the model across multiple courses, instructional formats, and cultures would also strengthen external validity and help refine theory building in GenAI-supported education. Also, in the current study, we conducted subgroup analyses to explore differences across gender and academic level. As we continue to expand this research with additional data and potentially adopt a longitudinal design, we will consider integrating demographic characteristics more explicitly into the theoretical framework. Last but not least, the findings are based on students' self-reported perceptions collected after the AI-facilitated lab sessions. As such, we cannot make definitive claims about changes in students' objective Python competency or long-term reliance on AI tools. Future research could incorporate objective performance measures and longitudinal designs to better assess learning outcomes and patterns of AI use over time.

7. REFERENCES

Abdullahi, H. O., Mohamud, A. H., Ali, A. F., & Abi Hassan, A. (2023). Determinants of the intention to use information system: A case of SIMAD University in Mogadishu, Somalia. *International Journal of Advanced and Applied Sciences, 10*(4), 188-196. https://doi.org/10.21833/ijaas.2023.04.023

Almulla, M. A. (2024). Investigating influencing factors of learning satisfaction in AI ChatGPT for research: University students' perspective. *Heliyon, 10*, e32220. https://doi.org/10.1016/j.heliyon.2024.e32220

Bere, A. (2018). Applying an extended task–technology fit for establishing determinants of mobile learning: An instant messaging initiative. *Journal of Information Systems Education*, *29*(4), 239–252. https://jise.org/Volume29/n4/JISEv29n4p239.html

Chen, Y., & Roldan, M. (2021). Digital innovation during COVID-19: Transforming challenges to opportunities. *Communications of the Association for Information Systems, 48*, Article 3. https://doi.org/10.17705/1CAIS.04803

Chen, J., Zhuo, Z., & Lin, J. (2023). Does ChatGPT play a double-edged sword role in the field of higher education? An in-depth exploration of the factors affecting student performance. *Sustainability, 15*(24), 16928. https://doi.org/10.3390/su152416928

Corchado, J. M., F, S. L., V, J. M. N., S., R. G., & Chamoso, P. (2023). Generative artificial intelligence: Fundamentals. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 12*(1), e31704. https://doi.org/10.14201/adcaij.31704

Dang, M. Y., & Zhang, Y. (2022). The impact of the coronavirus (COVID-19) pandemic on education: A model toward technology-supported synchronous remote learning. *International Journal of Information and Communication Technology Education, 18*(1), 1–20. https://doi.org/10.4018/IJICTE.292481

Dang, M. Y., Zhang, Y., & Albritton, M. D. (2023). Impact of course learning factors on student interest in business analytics careers. *International Journal of Information and Communication Technology Education, 19*(1), Article 60, 1-19. https://doi.org/10.4018/IJICTE.324160

Denny, P., Gulwani, S., Heffernan, N. T., Käser, T., Moore, S., Rafferty, A. N., & Singla, A. (2024). Generative AI for education (GAIED): Advances, opportunities, and challenges. *arXiv*. https://arxiv.org/abs/2402.01580

Gill, S. S., Xu, M., Patros, P., Wu, H., Kaur, R., Kaur, K., . . . Buyya, R. (2024). Transformative effects of ChatGPT on modern education: Emerging era of AI chatbots. *Internet of Things and Cyber-Physical Systems, 4*, 19–23. https://doi.org/10.1016/j.iotcps.2023.06.002

Goodhue, D. L., & Thompson, R. L. (1995). Task–technology fit and individual performance. *MIS Quarterly, 19*(2), 213–236. https://doi.org/10.2307/249689

Islam, A. K. M. N. (2013). Investigating e-learning system usage outcomes in the university context. *Computers & Education, 69*, 387-399. https://doi.org/10.1016/j.compedu.2013.07.037

Kakhki, M. D., Oguz, A., & Gendron, M. (2024). Exploring the affordances of chatbots in higher education: A framework for understanding and utilizing ChatGPT. *Journal of Information Systems Education, 35*(3), 284-302. https://doi.org/10.62273/UIRX9922

Kasneci, E., Sessler, K., Kühn, S., Kasneci, G., & Bannert, M. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences, 103*, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Kétyi, A., Géring, Z., & Dén-Nagy, I. (2025). ChatGPT from the students' point of view – Lessons from a pilot study using ChatGPT in business higher education. *Society and Economy, 47*(1), 1–21. https://doi.org/10.1556/204.2024.00007

Koraishi, O. (2023). Teaching English in the age of AI: Embracing ChatGPT to optimize EFL materials and assessment. *Language Education & Technology, 3*(1), 55-72. https://langedutech.com/letjournal/index.php/let/article/view/48

Kosar, T., Ostojić, D., Liu, Y. D., & Mernik, M. (2024). Computer science education in ChatGPT era: Experiences from an experiment in a programming course for novice programmers. *Mathematics, 12*(5), 629. https://doi.org/10.3390/math12050629

Lin, W.-S. (2012). Perceived fit and satisfaction on web learning performance: IS continuance intention and task-technology fit perspectives. *International Journal of Human-Computer Studies, 70*(7), 498–507. https://doi.org/10.1016/j.ijhcs.2012.01.006

Myers, D., Mohawesh, R., Chellaboina, V. I., Sathvik, A. L., Venkatesh, P., Ho, Y.-H., . . . Jararweh, Y. (2024). Foundation and large language models: Fundamentals, challenges, opportunities, and social impacts. *Cluster Computing, 27*, 1–26.

https://doi.org/10.1007/s10586-023-04203-7

Olsen, T., Chudoba, K. M., & Dupin-Bryant, P. A. (2022). Examining trends in business analytics education from 2011 to 2021. *Journal of Information Systems Education, 33*(3), 232–244. https://jise.org/Volume33/n3/JISE2022v33n3pp232-244.pdf

OpenAI. (2022). ChatGPT: Optimizing language models for dialogue. *OpenAI Blog*. https://openai.com/blog/chatgpt

Remolina, N. (2024). Generative AI in finance: Risks and potential solutions. *Law Ethics & Technology Review, 2*(1), 1-25. https://doi.org/10.55092/let20240002

Sabeh, H. N. (2024). What drives IT students toward ChatGPT? Analyzing the factors influencing students' intention to use ChatGPT for educational purposes. Paper presented at *the 21st International Multi-Conference on Systems, Signals & Devices* (SSD) (pp. 533–539). IEEE. https://doi.org/10.1109/SSD61670.2024.10548826

Sánchez-Ruiz, L. M., Moll-López, S., Nuñez-Pérez, A., Moraño-Fernández, J. A., & Vega-Fleitas, E. (2023). ChatGPT challenges blended learning methodologies in engineering education: A case study in mathematics. *Applied Sciences, 13*(10), 6039. https://doi.org/10.3390/app13106039

Suleiman, M. S., Usman, U. M. Z., & Yahaya, M. (2018). E-learning adoption based on technology adoption theory in Nigeria. *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, 6(10), 9–14. https://doi.org/10.17148/IJIREEICE.2018.6102

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). *User acceptance of information technology: Toward a unified view. MIS Quarterly, 27*(3), 425–478. https://doi.org/10.2307/30036540

Wall, J. D., & Knapp, J. (2014). Learning computing topics in undergraduate information systems courses: Managing perceived difficulty. *Journal of Information Systems Education, 25*(3), 245-259. https://aisel.aisnet.org/jise/vol225/iss243/248.

Warschauer, M., Tseng, W., Yim, S., Webster, T., Jacob, S., Du, Q., & Tate, T. (2023). The affordances and contradictions of AI-generated text for second language writers of English as a second or foreign language. *Journal of Second Language Writing, 62*, 101071. https://doi.org/10.1016/j.jslw.2023.101071

Wilkins, D. (2024). The legal profession in 2024: AI. *Harvard Law School*. https://hls.harvard.edu/today/harvard-law-expert-explains-how-ai-may-transform-the-legal-profession-in-2024/

Xia, Q., Weng, X., Ouyang, F., & Lin, T. J. (2024). A scoping review on how generative artificial intelligence transforms assessment in higher education. *International Journal of Educational Technology in Higher Education, 21*(1), 1-20. https://doi.org/10.1186/s41239-024-00468-z

Yang, H., Xu, J., Zhao, L., & Wang, X. (2024). The rapid evolution and deployment of generative AI across industries: Challenges and opportunities. *arXiv*. https://arxiv.org/abs/2405.11029

Yilmaz, R., & Yilmaz, F. G. K. (2023). Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. *Computers in Human Behavior: Artificial Humans, 1*, 100005. https://doi.org/10.1016/j.chbah.2023.100005

Zheng, Y. (2023). ChatGPT for teaching and learning: An experience from data science education. Paper presented at *the 24th Annual Conference on Information Technology Education* (SIGITE '23). https://doi.org/10.1145/3585059.3611431

Zhong, C., & Kim, J. B. (2024). Teaching case: Teaching business students logistic regression in R with the aid of ChatGPT. *Journal of Information Systems Education, 35*(2), 138–143. https://doi.org/10.62273/DYLI2468

APPENDIX A

Measurement Items

Learning Effort Expectancy
1. Using ChatGPT to learn and understand data analytics in Python is easy for me.
2. I would describe my interaction with the ChatGPT interface for learning/understanding data analytics in Python as being clear and understandable.
3. I find ChatGPT easy to use for data analytics in Python.
4. It has been easy for me to become skilled at using ChatGPT for data analytics in Python.

Task-Technology Fit
1. ChatGPT is well-suited to help me with my learning tasks on data analytics in Python.
2. ChatGPT is a necessary support for my learning and understanding of data analytics in Python.
3. ChatGPT is an appropriate fit for my learning needs.

Difficulty Management
1. ChatGPT has helped make the difficulty of learning data analytics in Python manageable for me.
2. ChatGPT helps me manage the difficulty of learning data analytics in Python.
3. ChatGPT helps me effectively cope with the difficulty of learning data analytics in Python.

Learning Satisfaction
1. Overall, I am satisfied with my ChatGPT-facilitated learning experience.
2. Using ChatGPT enhances my level of satisfaction with learning data analytics in Python.
3. ChatGPT contributes significantly to my overall satisfaction with learning data analytics in Python.
4. I am satisfied with my learning outcomes and experiences using ChatGPT.

Perceived Learning Performance
1. With the support of ChatGPT, I can effectively accomplish my learning tasks on data analytics in Python.
2. With the support of ChatGPT, I can efficiently accomplish my learning tasks on data analytics in Python.
3. Overall, I am satisfied with my performance on my data analytics in Python lab projects.

Teaching Case:

Code Together, Apart: Teaching Asynchronous Team Development Workflows with GitHub

Kareem Dana
kdana@wtamu.edu

Abraham Abby Sen
aabbysen@wtamu.edu

West Texas A&M University
Canyon, TX 79016, USA

Jeen Mariam Joy
joyj2@vcu.edu
Virginia Commonwealth University
Richmond, VA 23222, USA

Hook

When technology and industry expectations evolve faster than your syllabus, what do you do? Are you tired of student drama in group projects? With this teaching case, we present a novel group assignment that minimizes frustration and better prepares students for their IS careers.

Abstract

Software development workflows in the information systems industry change quickly and are often different than what students experience in the classroom. This gap leaves many students underprepared for the engineering, analysis, or management jobs they seek. Asynchronous collaboration and tools like GitHub are increasingly prevalent in development teams. Traditional group projects, while beneficial, present persistent challenges, especially in online courses. This teaching case offers a practical assignment that bridges the classroom-industry gap and improves group work experience. Drawing on established pedagogical foundations, the assignment emphasizes cooperation—where students work independently toward a shared outcome—over collaboration, introduces students to real-world developer workflows with GitHub, and uses a narrative story to improve engagement. Students complete individual yet interdependent tasks within an existing codebase through GitHub. We share evidence showing that our assignment increased student confidence, self-efficacy, and satisfaction, particularly compared to traditional group projects. We conclude with several practical recommendations instructors can use to reduce group work frustration and build job-ready skills.

Keywords: Asynchronous learning, Git, Group Projects, Pedagogy, Collaboration, Cooperative learning

Recommended Citation: Dana, K., Abby Sen, A., Joy, J., (2026). Code Together, Apart: Teaching Asynchronous Team Development Workflows with GitHub. *Information Systems Education Journal*. v24(n2) pp 59-74. DOI# https://doi.org/10.62273/ZRXR5679

Code Together, Apart: Teaching Asynchronous Team Development Workflows with GitHub

Kareem Dana, Abraham Abby Sen, Jeen Mariam Joy

1. INTRODUCTION

More and more software development occurs remotely through geographically distributed teams collaborating asynchronously. This increase is driven in part by the rise of remote and hybrid work and the desire by companies to hire the best talent regardless of location (Kumar et al., 2024; Sharma, 2023). These teams are increasingly using collaborative tools to manage their work and improve productivity. Among these tools, Git and GitHub have become industry standards (Dabbish et al., 2012; Hu et al., 2016).

Git is a widely used version control system, while GitHub is a web-based platform built on top of Git that hosts code repositories. GitHub has features such as issue tracking, pull requests, and code reviews that naturally support asynchronous, collaborative workflows needed by distributed software teams. As of 2023, GitHub hosts over 420 million repositories and is used by over 100 million developers worldwide, including many large corporations such as Microsoft, Google, and IBM (Dohmke, 2023).

This widespread use of GitHub and the uptick in asynchronous software development suggests that mastery of this tool and workflow is now a core expectation for new graduates entering the workforce. This expectation is not just for computer scientists or software developers. Experience with GitHub and asynchronous collaboration are growing in importance across a variety of Information Systems (IS) jobs. Product owners, managers, and systems analysts all use GitHub to track issues, requirements, and communicate with their teams (Wagner & Thurner, 2025).

Additional research indicates that IS students are frequently unfamiliar with the workflows and tools that they will encounter in the workplace, leaving them underprepared to start their careers (Craig et al., 2018; Liebenberg et al., 2015; Moreno et al., 2012). Consequently, IS educators are tasked with addressing this challenge. Traditional classroom settings, or online courses relying solely on discussion forums, may not sufficiently prepare students for their careers.

Group projects, on the other hand, can prepare students well for collaboration and are frequently used in IS classes, including online. However, implementing effective group projects remains difficult, especially asynchronously (Hafner & Ellis, 2004). They are often met by student frustration. Students highlight issues such as unequal participation, communication challenges, and a lack of accountability. These frustrations are often exacerbated in asynchronous environments (Awuor et al., 2022; Bakir et al., 2020; Roberts & McInnerney, 2007).

In this paper, we present a teaching case that addresses both challenges. This learning activity simulates real-world development workflows while introducing students to asynchronous collaboration using GitHub. It also follows best practices to avoid many group work pitfalls.

Our main contributions are:

1. The description and development of the learning activity, backed by a strong pedagogical foundation outlined in sections two and three.
2. Evidence of the assignment's effectiveness, presented in section four through statistical analysis of pre- and post-assignment survey results along with qualitative feedback from students.
3. Practical recommendations, discussed in section five, for IS educators to implement and adapt to different objectives and instructional contexts.

2. MOTIVATION AND BACKGROUND

We taught this assignment in our IS department's second programming course for undergraduates. We developed it after discussions with our capstone instructor that centered around the question, "How can we introduce advanced IS topics used in the capstone course earlier in the curriculum?" This led to further discussions about student preparedness for real-world projects and challenges with group work.

Preparing Students for IS Workflows

Past research highlights a well-acknowledged gap between industry expectations and graduating student skill levels (Aasheim et al., 2012; Craig et al., 2018; Moreno et al., 2012; Oguz & Oguz, 2019; Tuzun et al., 2018). Craig (2018) states that soon after graduation, students notice real life projects are of a different breed from the ones they have handled during their education. School assignments usually involve small stand-alone programs, while companies rely on large, existing codebases with a variety of dependencies and libraries. Companies often use tools such as Git and GitHub to manage those codebases. To address this skills gap, we drew on prior literature and pedagogical theories.

First, we provide students with an existing codebase, written by the instructor, to simulate the large codebases seen in IS jobs. Second, to help students avoid feeling overwhelmed by unfamiliar code, we require them to follow a structured GitHub workflow that includes creating issues, branching, merging, and resolving merge conflicts. This workflow guides their initial interactions with the codebase through clearly defined, step-by-step tasks allowing them to build familiarity and gain confidence as they progress. This design builds on the work of Kertész (2015), Luce (2021), and Glassey (2019), who all emphasize the pedagogical value of incorporating GitHub in the classroom. In addition, Agrawal (2024) and Xia (2017) highlight the importance of introducing students to unfamiliar codebases.

Lastly, our assignment provides a platform to introduce advanced topics and workflows earlier in the curriculum at lower levels (remember and understand) of Bloom's taxonomy (Anderson et al., 2001), with reinforcement occurring in later courses. This design also supports the introduce-reinforce-emphasize curriculum model advocated by the accreditation board, ABET, where topics are introduced in foundational classes, reinforced in later classes, and emphasized or mastered in a capstone class (Almuhaideb & Saeed, 2020; Calderón, et al., 2016). In our curriculum, these topics include C# class libraries and unit testing, both of which are part of that skills gap (Craig et al., 2018).

To help introduce students to these topics, we designed the assignment to follow a task-based pedagogical model, wherein students complete well-defined tasks with step-by-step instructions. This is a beneficial way to successfully introduce new, challenging, or advanced topics without cognitively overloading students (Kulesza et al., 2011; Warrick, 2021).

Another key innovation of our approach was adding a narrative-based story. Narrative pedagogy uses storytelling for learning complex topics and has been shown to increase learning and improve engagement (Humpherys & Babb, 2020; Lee et al., 2006).

Group Work Challenges

Group projects are another excellent pedagogical tool to prepare students for real-world projects and are used often in IS classes. However, prior research suggests that many students dislike group work (Lowes, 2014). Roberts & McInnerney (2007) and Ekblaw (2016) highlight seven major challenges students face with group work, while Bakir, et al. (2020) discusses group work challenges within a Management Information Systems course.

Students often express apathy towards group work. They may not be motivated or do not understand the benefits (Ekblaw, 2016; Roberts & McInnerney, 2007). Students are most motivated by their grade. Fairly assessing individual performance can significantly reduce this challenge (Favor & Harvey, 2016; Roberts & McInnerney, 2007). We accomplish that by tracking student contributions through GitHub commit logs and ultimately grading each student individually based solely on their contributions.

Lack of communication is another major challenge in group work. To solve that challenge, we structured our assignment so that students work on their parts asynchronously and independently. This technique is recommended by Lowes (2014) and Ekblaw (2016). We further make a distinction between collaboration and cooperation. Cooperation is where individuals are working towards a shared goal, but each works independently. Collaboration involves interactions among all group members to achieve the goal (Bruffee, 1995; Panitz, 1999; Paulus, 2005). In our assignment, students cooperate but do not need to collaborate.

We are not suggesting that collaboration is unimportant. On the contrary, collaboration and communication skills are critical for IS graduates. Rather, this teaching case enables students to focus first on mastering professional workflows and individual accountability before engaging in more collaborative projects later in the curriculum. In this way, the case offers a complementary group work technique that can be particularly effective in asynchronous learning environments.

On that note, lack of accountability or "free riding" is another challenge in group work. That is when one group member slacks off or does not perform. We approach this problem with the

same techniques as described above. Cooperation, as defined above, and fairly assessing individual performance through GitHub commit logs mitigates free riding.
Finally, students complain about poor distribution of responsibilities. Our assignment clearly defines which group member will do which task ahead of time to eliminate this common problem.

3. THE ASSIGNMENT

In this section, we describe how we developed the assignment, provide the steps needed for instructors to deploy it in their classroom, outline how a typical student would complete the assignment, and share grading guidelines.

Instructor Guidelines

Our programming course is taught online and asynchronously at a regional business school in the southwestern United States. All students are computer information systems (CIS) majors. Most are sophomores or juniors and have already taken an introductory programming course. They have basic C# skills and experience with Visual Studio (VS) Code. While students also had one semester of experience with git and GitHub, no prior experience is required.

First, to build a narrative story, we created ThoughtTronix, a morally ambiguous technology company that develops artificial intelligence devices. The story of ThoughtTronix starts early in the semester and is woven throughout the semester to keep students interested, engaged, and motivated. As described earlier, this is supported by narrative pedagogy theory (Humpherys & Babb, 2020; Lee et al., 2006).

Through successive assignments, students take on the role of junior developers contributing to ThoughtTronix's ethically questionable products, such as MindSync, a brain implant, and SoulSear, an AI weapon. In this assignment, students are required to add features to an existing web application that processes orders for those products. The features are to calculate (1) sales tax and (2) shipping charges. Most students are familiar with ordering products online, so these features will feel familiar to them. This also mimics real-world development where many software developers fix bugs, add features, or maintain existing applications instead of creating new ones (Craig et al., 2018).

Second, to support this existing codebase, we created a template repository in GitHub that contains the starter code – a C# ASP.NET Core web application for order processing with the sales tax and shipping cost features not implemented. A template repository allows the instructor to create new repositories for each student team with the same structure and starter code. The instructor should have some prior knowledge of Git and GitHub, while students do not need any prior experience. Appendix C provides a quick primer on the GitHub concepts and workflows used in this case, including issues, branching, pull requests, and merging code.

We wrote the starter code in a deliberate manner such that students can complete their parts independently and then merge them to form the complete, working application. We chose to separate each feature into a class library and have students implement their class library. This gave us the benefit of introducing students to class libraries, which are used in our capstone course.

Third, we placed students in teams of two chosen randomly and created a GitHub repository for each team. The instructor was the repository administrator and added each student as a collaborator.

Finally, the instructor emailed both students. In the email, we assigned each their feature – either to calculate sales tax or calculate shipping charges. Both tasks achieve the same learning objectives and can be completed in any order. The email also included links to the GitHub repository, assignment instructions, and the grading rubric - all the details needed for students to start working on the assignment.

Learning Objectives

The assignment has five learning objectives, designed to align with industry skills, particularly those associated with the skills gap discussed earlier:

1. Gain experience in asynchronous software development teams. This is the main objective of the assignment. Students gain hands-on experience with GitHub in a team setting, reflecting the distributed collaboration workflows commonly used in the IS industry.
2. Create Git branches. Students create branches to facilitate development of their software features, reflecting common industry practices for managing parallel work and isolating code changes.
3. Implement class libraries. Students are introduced to class libraries, create their own class library, and connect it to the existing code. This models the modular, reusable code design expected in

professional software development. This objective can be replaced by another as needed by the instructor.

4. Run unit tests. Students were given existing unit tests and were required to test their code. This introduces students to test-driven development and automated testing practices that are widely used in real-world software development teams to ensure code reliability and maintainability. This learning objective introduces unit testing at the lower levels of Bloom's taxonomy (remembering and applying). Reinforcement and emphasis occur in the capstone course. This is the other learning objective that can be replaced to match other instructors' goals.
5. Merge Git branches and resolve merge conflicts. Students learn how to merge their branches back into the main branch and cooperate to resolve merge conflicts, closely simulating version control challenges and code integration issues developers face in industry but rarely see in the classroom.

Student Experience

There is no prior lecture content. We designed the assignment to introduce students to each topic through a learn-by-doing approach (Reese, 2011). Students did not have any prior knowledge of GitHub issues, branching, class libraries, unit testing, or git merging. Instructors can supplement this assignment with additional content to fill in any gaps in theory if desired. However, our evidence suggests that students were successfully introduced to all these topics through this learning activity.

The students complete the assignment through the following steps:

Step 1: Create an issue in GitHub. An Issue is a way to track and manage tasks. It is a key collaborative feature in GitHub. Through this step, the student learns to clearly articulate what feature they are implementing and how to implement it. They are instructed to list clear steps on how they plan to complete the assignment. This helps students prepare and improves the chance of success.

Step 2: Create a branch. Students then create a Git branch tied to their issue. This introduces them to branching, a foundational part of most team-based workflows.

Step 3: Checkout their branch. After creating a branch, the student will clone the repository and checkout their branch. Students complete this step through VS Code or the command line with a "git checkout" command.

Step 4: Write the code. Students implement their assigned feature following the step-by-step instructions. While this may seem like the most important step, it primarily serves as a means for engaging the students in the broader learning objectives.

Step 5: Test their code. Students run pre-written unit tests to validate what they wrote in step 4. This introduces them to test-driven development and provides immediate feedback, allowing iterative improvement until all tests pass.

Step 6: Commit and push changes. Students commit their changes and push them to GitHub using VS Code or git at the terminal. We instruct students to visit GitHub to verify that their changes have been successfully pushed.

Step 7: Merge branches and close the issue. Students merge their branch back into the main branch and then close their issue from step one. This closes the loop, demonstrating the full development cycle from planning to delivery – albeit of a small feature.

Each student completes these steps independently. When both team members succeed, the application is functionally complete. This design fosters cooperative engagement without requiring direct collaboration, minimizing a common pitfall of traditional group projects.

Grading Guidelines

We developed a grading rubric for the assignment (see Appendix A), which was also shared with students in advance. After the due date, we reviewed each team's GitHub repository and assessed each student individually based on their contributions. GitHub's commit history provided an objective record, allowing us to verify what each student completed and when.

We then provided detailed feedback through our university's learning management system with references to specific rubric items (e.g. "Missed Task 10") and, as needed, further explanation of why and how to correct mistakes.

4. RESULTS AND STUDENT FEEDBACK

We administered pre- and post-assignment surveys to students across two semesters: Fall 2024 (n = 24) and Spring 2025 (n = 19). The

	Pre-Assignment Survey (N = 43)		Post-Assignment Survey (N = 43)				
How comfortable are you with the following:	Mean	SD	Mean	SD	*p*-Value	t(df)	Cohen's d
GitHub Issues	2.91	1.31	4.56	0.63	< .001	-8.93 (42)	1.21
Creating Branches	3.33	1.17	4.56	0.63	< .001	-7.93 (42)	1.02
Merging Branches	2.91	1.27	4.40	0.69	< .001	-7.73 (42)	1.26
Class Libraries	3.02	1.23	4.09	0.75	< .001	-6.65 (42)	1.06
Unit Tests	2.86	1.21	3.88	0.79	< .001	-5.02 (42)	1.34
Job Preparedness	2.72	1.05	3.70	0.99	< .001	-6.48 (42)	0.99

Table 1: Paired Sample T-Test Results

surveys asked students to rank their comfort or familiarity with the assignment's five learning objectives: GitHub issues, creating branches, merging branches, class libraries, and unit tests using a five-point Likert-scale (1 = not comfortable, 5 = very comfortable) along with one question about how prepared students feel for a job in a software development team. The post-survey added questions about students' satisfaction with the assignment and open-ended questions for qualitative feedback. See Appendix B for the complete survey questions. We identified three key themes in the survey results.

Improved Learning

We combined both semesters of survey data together and used a paired samples t-test measuring improvements between the pre- and post-assignment surveys for each of the five learning objectives. We observed statistically significant self-reported improvement in all learning objectives and present them in Table 1.

Students self-reported these results, but through grading the assignment, we confirm students demonstrated sufficient knowledge of the learning objectives.

Improved Job Preparedness

There was a statistically significant increase in self-reported job preparedness when asked "How prepared do you feel for a job in a software development team?" as shown on the final row of Table 1. Job preparedness is a multi-faceted challenge, and we recognize that a single assignment has limited impact. Nonetheless, the results suggest that students felt at least somewhat more prepared to work in a professional software development team. A valuable extension of this analysis would be a longitudinal study to track whether these perceptions persist over time.

Positive Assignment Satisfaction

On the post-assignment survey, students expressed increased satisfaction with this assignment compared to prior group work experiences. Of the forty-three respondents, 36 (84%) indicated a preference to asynchronous group work like this assignment, while 7 (16%) stated no preference. Zero students preferred their prior group work experiences. We recognize that we had no control over nor any familiarity with their prior group experiences.

Student Feedback

In addition, students left open-ended feedback expressing the same preference:

- "One of the benefits of this type of asynchronous group assignment was that there were no scheduling conflicts as we could finish our part of the assignment when we had the time. The instructions were clear and easy to follow as well."
- "Yes, not having to depend on others and getting graded individually."
- "Not being responsible for my teammates work."

Several students also noted the storytelling aspect of the assignment as a benefit:

- "I really enjoyed this class. The themed homework assignments with the ominous defence company and their products was a really nice touch and I enjoyed the narrative that it sort of wove through the homework. I learned a lot from this class!"
- "Teaching methods were clear, well-organized, and engaging, using interesting and funny examples to make complex concepts relatable and applications more enjoyable."

5. DISCUSSION AND TEACHING SUGGESTIONS

In this section we discuss challenges encountered, modifications instructors can make to the assignment, and practical teaching suggestions based on our experiences.

Challenges

We experienced some challenges with this assignment. It can be onerous to manually create GitHub repositories for all teams and invite students individually, especially in a large class. Moreover, GitHub invitations expire after seven days and several students forgot to accept the invitation, requiring the instructor to re-send it. While we had developed the ThoughtTronix narrative throughout the semester, it may be time-consuming or challenging for another instructor to use the same narrative or develop their own.

Students expressed their own challenges as well. Some stated learning the inner workings of GitHub was challenging. Several other students expressed confusion with merging and merge conflicts, as well as challenges related to software incompatibilities with git, .NET, VS Code, and the students' own personal computers.

Finally, during grading, we identified that several teams had one student who did not participate. However, the free rider did not negatively impact their teammate, and no student expressed frustration with free riding.

Modifications

A benefit of this assignment is that it can be modified in a myriad of ways. Below we outline a few of those ways:

Change Learning Objectives. Our course specific learning objectives were class libraries and unit tests. Those can be swapped out and still maintain the core benefits of the assignment. For example, advanced students could be asked to implement a full authentication-based library, while beginner students might implement small logic changes and both cohorts of students would benefit from utilizing GitHub and gain valuable experience following a professional workflow in an asynchronous setting.

Change Programming Language. This assignment is written in C# to fit the needs of our curriculum. It could be modified for another language while still utilizing GitHub, issues, branches, and asynchronous teamwork.

Change Group Size. This assignment is designed for groups of two. There are two features to implement, and each student implements one. We understand common groups tend to be 3 or 4 students. Supporting larger groups requires additional features to be brought into the assignment. Perhaps a third student could implement a rewards discount. Ultimately, changing the group size will require an additional feature for each extra student to implement.

Change Course. We believe this assignment can be adapted for an introductory programming course with only a few modifications. First, we recommend assigning it near the end of the term once students have learned basic programming skills. Second, instructors can add preparatory materials and simplify or change learning objectives 3 and 4 to better match their course goals.

The assignment can also be extended to non-programming courses or to students from non-technical majors. For example, students could create UML diagrams instead of class libraries. Instructors teaching non-technical students might emphasize issue tracking, GitHub collaboration, and workflow planning rather than programming. These adjustments maintain the assignment's structure and benefits while meeting students at their skill level.

Teaching Suggestions

Through developing this assignment, student feedback, and ongoing conversations with colleagues we present several teaching suggestions below. Whether you implement this assignment or not, these are practical strategies for any IS educator to utilize.

Teaching Suggestion 1: Offer an asynchronous assignment like this one as an alternative to traditional group work. While this assignment does not replace group projects with full collaboration, it can be a valuable tool and provide students with a positive experience. Our students reported, overwhelmingly, preferring this assignment over traditional groupwork. Moreover, it can serve as a foundation that prepares students for more complex collaboration later in the curriculum. Once students gain confidence with developer workflows and individual accountability, instructors can transition to fully collaborative projects.

We showed that this learning activity mitigates some of the major challenges students face with group work. Additionally, providing multiple means of engagement to cater to a diverse set of learners can help all students succeed (Gadsden & Goegan, 2023; Kolb, 2014; Saunders & Wong, 2023). This assignment offers another means of engagement through a hands-on alternative to lecture and a different group project dynamic. This suggestion highlights the importance of finding a balance between the benefits of group work and the frustrations.

Teaching Suggestion 2: In group projects, we recommend grading students individually based on their own contributions. Instructors should actively track these contributions to promote accountability and clearly communicate to students that individual efforts will be recognized. This approach helps address common concerns about free riding. Even when a groupmate contributes little or no work, individual grading based on tracked contributions ensures that a diligent student can still earn full credit. Our students expressed high satisfaction with this approach, as one student remarked on the survey, "*I was able to independently get my portion of the work done in my own time and at my own pace which greatly reduced stress and frustration on my end. I also didn't feel like there was a power dynamic where one group member tries to take everything or the opposite, does nothing, I really liked this way of doing things.*"

Teaching Suggestion 3: Introduce advanced topics earlier in the curriculum. We introduced class libraries and unit testing earlier than usual, giving students exposure to key concepts before they encounter them in more advanced courses. Instructors can choose any topic, and we believe students will benefit from this early introduction. This strategy aligns with the pedagogical foundation of Bloom's taxonomy, which provides a framework for introducing concepts at lower cognitive levels and reinforcing them in later classes (Anderson et al., 2001). It is also consistent with ABET's introduce-reinforce-emphasize model (Almuhaideb & Saeed, 2020; Calderón, et al., 2016).

Teaching Suggestion 4: Use a narrative story in your assignments, especially when introducing complex topics. Research supports this approach, showing that narrative-based learning can be an effective instructional tool (Humpherys & Babb, 2020; Lee et al., 2006; Zazkis, 2009). Our findings support this, as our students demonstrated positive outcomes and shared feedback showing satisfaction with the narrative story. When possible, consider using a narrative approach throughout the course instead of for one assignment.

Teaching Suggestion 5: Utilize real-world tools to simulate professional development environments. Integrating tools like GitHub into assignments does more than just support asynchronous work; it exposes students to industry-standard workflows they are likely to encounter after graduation. We familiarized our students with core professional practices, such as using GitHub Issues to define tasks, creating feature branches, resolving merge conflicts, and modifying an existing codebase. Students reported feeling more prepared in their ability to contribute to a development team after completing this assignment.

This suggestion is consistent with two well-established pedagogical theories. The first, Experiential Learning Theory (Kolb, 2014), emphasizes learning as a process where knowledge is created through direct experience and reflection. According to Kolb, students learn most effectively when they engage in a full cycle of doing, reflecting, conceptualizing, and applying. The second, Learning-by-Doing (Reese, 2011), asserts that students gain deeper understanding when they are actively engaged in meaningful tasks, rather than passively receiving information. Reese highlights that active participation and direct engagement leads to better motivation and skill development when compared to methods like lectures. By immersing students in real-world development workflows, our assignment puts these principles into practice.

6. CONCLUSIONS

The purpose of this teaching case is to provide IS educators with a practical model for integrating real-world programming workflows into a unique, asynchronous group assignment, ensuring that students are better equipped for an IS career.

Our strategies of mitigating group work frustrations through cooperative (rather than collaborative) work and individual accountability, integrating a narrative story, and using real-world workflows are backed by strong pedagogical theories. By combining these elements into a cohesive, practical assignment, we provide a model that not only engages students but also prepares them for the expectations of industry.

Evidence from student feedback shows increased confidence and self-efficacy with GitHub and associated developer workflows, as well as a strong preference for this type of group assignment. This case can also be adapted by instructors to fit a range of learning objectives and instructional contexts.

Finally, all the teaching materials including the assignment instructions, starter code, and rubric are available through a GitHub repository. Given their size and code-heavy nature, they are not included in the appendix. The solution is also available from the authors or the editor upon request. The GitHub repository link is: https://github.com/kareemy/CodeTogether_TeachingMaterials

7. REFERENCES

Aasheim, C., Shropshire, J., Li, L., & Kadlec, C. (2012). Knowledge and skill requirements for entry-level IT workers: A longitudinal study. *Journal of Information Systems Education*, *23*(2), 193-204.

Agrawal, E., Alam, O., Goenka, C., Iyer, M., Moise, I., Pandian, A., & Paul, B. (2024). Code compass: A study on the challenges of navigating unfamiliar codebases. *arXiv*. https://doi.org/10.48550/arXiv.2405.06271

Almuhaideb, A. M., & Saeed, S. (2020). Fostering sustainable quality assurance practices in outcome-based education: Lessons learned from ABET accreditation process of computing programs. *Sustainability*, 12(20), 8380. https://doi.org/10.3390/su12208380

Anderson, L. W., Krathwohl, D. R., & Bloom, B. S. (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives: complete edition.* Addison Wesley Longman.

Awuor, N. O., Weng, C., & Militar, R. (2022). Teamwork competency and satisfaction in online group project-based engineering course: The cross-level moderating effect of collective efficacy and flipped instruction. *Computers & Education*, 176, 104357. https://doi.org/10.1016/j.compedu.2021.104357

Bakir, N., Humpherys, S., & Dana, K. (2020). Students' Perceptions of Challenges and Solutions to Face-to-Face and Online Group Work. *Information Systems Education Journal*, 18(5), 75-88. https://isedj.org/2020-18/n5/ISEDJv18n5p75.pdf

Bruffee, K. A. (1995). Sharing our toys: Cooperative learning versus collaborative learning. *Change: The Magazine of Higher Learning*, *27*(1), 12-18. https://doi.org/10.1080/00091383.1995.9937722

Calderón, H. E., Vásquez, R. E., Aponte, D. A., & Del Valle, M. (2016). Successful Assessment Strategies for ABET Accreditation of Engineering Programs Offered at Different Campuses. In *Proceedings of the 14th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Engineering Innovations for Global Sustainability"*, San José, Costa Rica (pp. 20-22). https://doi.org/10.18687/LACCEI2016.1.1.226

Craig, M., Conrad, P., Lynch, D., Lee, N., & Anthony, L. (2018). Listening to early career software developers. *Journal of Computing Sciences in Colleges*, 33(4), 138-149.

Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on computer supported cooperative work* (pp. 1277-1286). https://doi.org/10.1145/2145204.2145396

Dohmke, T. (2023). 100 Million Developers and Counting. https://github.blog/2023-01-25-100-million-developers-and-counting/

Ekblaw, R. (2016). Effective Use Of Group Projects In Online Learning. *Contemporary Issues in Education Research (CIER)*, 9(3), 121–128. https://doi.org/10.19030/cier.v9i3.9707

Favor, J. K., & Harvey, M. (2016). We Shall Not be Moved: Adult Learners' Intransigent Attitudes About Group Projects. *Adult Education Research Conference*. Retrieved June 6, 2025 from https://newprairiepress.org/aerc/2016/papers/18/

Gadsden, A. D., & Goegan, L. D. (2023). Informing Inclusive Practice in Post-Secondary Environments: Perspectives of Post-Secondary Instructors with Learning Disabilities. *The Canadian Journal for the Scholarship of Teaching and Learning*, 14(2). https://doi.org/10.5206/cjsotlrcacea.2023.2.8020

Glassey, R. (2019, May). Adopting git/github within teaching: A survey of tool support. In *Proceedings of the ACM Conference on Global Computing Education* (pp. 143-149). https://doi.org/10.1145/3300115.3309518

Hafner, W., & Ellis, T. J. (2004). Asynchronous collaborative learning using project-based assignments. In *34th Annual Frontiers in*

Education, 2004. FIE 2004. (pp. F2F-6). IEEE. https://doi.org/10.1109/fie.2004.1408607

Hu, Y., Zhang, J., Bai, X., Yu, S., & Yang, Z. (2016). Influence analysis of GitHub repositories. *SpringerPlus*, 5, 1-19. https://doi.org/10.1186/s40064-016-2897-7

Humpherys, S. L., & Babb, J. (2020). Using Folklore, Fables, and Storytelling as a Pedagogical Tool in Assessment Exams. *Information Systems Education Journal*, 18(5), 34-53. https://isedj.org/2020-18/n5/ISEDJv18n5p34.pdf

Kertész, C. Z. (2015). Using GitHub in the classroom-a collaborative learning experience. In *2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME)* (pp. 381-386). IEEE. https://doi.org/10.1109/siitme.2015.7342358

Kolb, D. A. (2014). Experiential learning: Experience as the source of learning and development. FT press.

Kulesza, J., DeHondt, G., & Nezlek, G. (2011). More Technology, Less Learning?. *Information Systems Education Journal*, *9*(7), 4-13. https://isedj.org/2011-9/N7/ISEDJv9n7p4.pdf

Kumar, A., Priyadarshi, P., & Garg, N. (2024). Bibliometric Analysis of Remote Working: 20-year Literature Review. *Human Resource Development Review*. https://doi.org/10.1177/15344843241305920

Lee, J. H. M., Lee, F. L., & Lau, T. S. (2006). Folklore-based learning on the web—pedagogy, case study, and evaluation. *Journal of Educational Computing Research*, 34(1), 1-27. https://doi.org/10.2190/3hfm-d9nq-g7y7-qc1g

Liebenberg, J., Huisman, M., & Mentz, E. (2015). The relevance of software development education for students. *IEEE Transactions on Education*, 58(4), 242-248. https://doi.org/10.1109/TE.2014.2381599

Lowes, S. (2014). How Much "Group" Is There in Online Group Work? *Journal of Asynchronous Learning Networks*, 18(1), n1. https://doi.org/10.24059/olj.v18i1.373

Luce, T. (2021). Distributed Project Teams and Software Development: An Introduction to the Use of Git and GitHub for ASP. NET MVC Development. *Information Systems Education Journal*, *19*(5), 45-57. https://isedj.org/2021-19/n5/ISEDJv19n5p45.pdf

Moreno, A. M., Sanchez-Segura, M. I., Medina-Dominguez, F., & Carvajal, L. (2012). Balancing software engineering education and industrial needs. *Journal of systems and software*, 85(7), 1607-1620. https://doi.org/10.1016/j.jss.2012.01.060

Oguz, D., & Oguz, K. (2019). Perspectives on the gap between the software industry and the software engineering education. *IEEE Access*, 7, 117527-117543. https://doi.org/10.1109/ACCESS.2019.2936660

Panitz, T. (1999). Collaborative versus cooperative learning: A comparison of the two concepts which will help us understand the underlying nature of interactive learning. Retrieved June 6, 2025 from https://files.eric.ed.gov/fulltext/ED448443.pdf

Paulus, T. M. (2005). Collaborative and cooperative approaches to online group work: The impact of task type. *Distance education*, 26(1), 111-125. https://doi.org/10.1080/01587910500081343

Reese, H. W. (2011). The Learning-by-Doing Principle. *Behavioral Development Bulletin*, 17(1), 1-19. https://doi.org/10.1037/H0100597

Roberts, T. S., & McInnerney, J. M. (2007). Seven problems of online group learning (and their solutions). *Educational Technology & Society*, 10(4), 257–268.

Saunders, L., & Wong, M. (2023). Multiple Means of Engagement: Connecting with Students Across Modalities through Choice, Flexibility and Authentic Assessment. In *Proceedings of the ALISE Annual Conference*. http://dx.doi.org/10.21900/j.alise.2023.1264

Sharma, T. K. (2023). Hybrid Working: The Future of Organizations. In Reshaping the Business World Post-COVID-19. Apple Academic Press. https://www.doi.org/10.1201/9781003372424-2

Tuzun, E., Erdogmus, H., & Ozbilgin, I. G. (2018, May). Are computer science and engineering graduates ready for the software industry? Experiences from an industrial student training program. In *Proceedings of the 40th*

International Conference on Software Engineering: Software Engineering Education and Training (pp. 68-77). https://doi.org/10.1145/3183377.3185754

Wagner, G., & Thurner, L. (2025). Teaching Tip: Rethinking How We Teach Git: Pedagogical Recommendations and Practical Strategies for the Information Systems Curriculum. *Journal of Information Systems Education*, 36(1), 1-12. https://doi.org/10.62273/BTKM5634

Warrick, A. (2021). Strategies for reducing cognitive overload in the online language learning classroom. *International Journal of Second and Foreign Language Education*, *1*(2), 25-37. https://doi.org/10.33422/ijsfle.v1i2.124

Xia, X., Bao, L., Lo, D., Xing, Z., Hassan, A. E., & Li, S. (2017). Measuring program comprehension: A large-scale field study with professionals. *IEEE Transactions on Software Engineering*, *44*(10), 951-976. https://doi.org/10.1109/tse.2017.2734091

Zazkis, R., & Liljedahl, P. (2019). *Teaching mathematics as storytelling*. Brill. https://doi.org/10.1163/9789087907358

APPENDIX A
Grading Rubric

Below is the grading rubric used for this teaching case assignment. All the teaching materials including the assignment instructions, starter code, rubric, and solution are available from the authors or the editor upon request and at this GitHub link:
https://github.com/kareemy/CodeTogether_TeachingMaterials

Task	Points
Create an issue on GitHub	5
Assign the issue to yourself	5
Create a new branch on GitHub	10
Checkout your branch	5
Write all your code in your branch (Do not code directly in the main branch)	5
Create a new class library	5
Add your class library to the main solution file	5
Add a reference to your class library in OrderApp	5
Write your code in the class library	10
Uncomment the correct using directive in ReviewOrder.cshtml.cs	5
Uncomment the correct line of code in ReviewOrder.cshtml.cs	5
Test your code with dotnet test	5
Push your changes back to GitHub in the correct branch	5
Create a pull request on GitHub	10
Merge pull request and recognize "Can't automatically merge" message	10
Resolve merge conflicts if applicable	5
Total:	**100**

APPENDIX B
Survey Questions

The purpose of this survey is to better understand your learning experiences with group projects in general and your understanding of software development project workflows.
Please take the time to answer these questions.
Thank You.

Pre-Assignment Survey Questions:

1. How comfortable are you with using the Issues feature on GitHub?

	1	2	3	4	5	
Not comfortable	○	○	○	○	○	Very comfortable

2. How comfortable are you with creating branches on GitHub?

	1	2	3	4	5	
Not comfortable	○	○	○	○	○	Very comfortable

3. How comfortable are you with merging code changes on GitHub?

	1	2	3	4	5	
Not comfortable	○	○	○	○	○	Very comfortable

4. How familiar are you with the computer programming concept of class libraries?

	1	2	3	4	5	
Not familiar	○	○	○	○	○	Very familiar

5. How familiar are you with the concept of unit tests?

	1	2	3	4	5	
Not familiar	○	○	○	○	○	Very familiar

6. How prepared do you feel for a job in a software development team?

	1	2	3	4	5	
Not prepared	○	○	○	○	○	Very prepared

Post-Assignment Survey Questions:

1. As you respond, consider only prior online group experiences. How would you rate your overall experience with PAST online group projects?

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

2. Describe the main challenges you faced with PAST online group projects. [Open-Ended]
3. Consider your CURRENT experience with this asynchronous group assignment. How would you rate your overall experience?

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

4. Describe the main challenges you faced with THIS assignment. [Open-Ended]
5. Did this type of assignment have any benefits over your prior group experience? If so, please describe them. [Open-Ended]
6. After experiencing this asynchronous group assignment and considering your prior experience with online group projects, which style of assignment do you prefer?
   - ○ I prefer group assignments that are asynchronous like this assignment.
   - ○ I prefer my prior group projects.
   - ○ I have no preference.
7. How comfortable are you with using the Issues feature on GitHub?

	1	2	3	4	5	
Not comfortable	○	○	○	○	○	Very comfortable

8. How comfortable are you with creating branches on GitHub?

	1	2	3	4	5	
Not comfortable	○	○	○	○	○	Very comfortable

9. How comfortable are you with merging code changes on GitHub?

	1	2	3	4	5	
Not comfortable	○	○	○	○	○	Very comfortable

10. How familiar are you with the computer programming concept of class libraries?

	1	2	3	4	5	
Not familiar	○	○	○	○	○	Very familiar

©2026 ISCAP (Information Systems and Computing Academic Professionals)
https://isedj.org/; https://iscap.us/org

11. How familiar are you with the concept of unit tests?

	1	2	3	4	5	
Not familiar	○	○	○	○	○	Very familiar

12. How prepared do you feel for a job in a software development team?

	1	2	3	4	5	
Not prepared	○	○	○	○	○	Very prepared

APPENDIX C
Quick GitHub Primer

This appendix provides a brief overview of Git and GitHub concepts necessary to implement the teaching case.

Git vs. GitHub
Git is a version control system that tracks changes in source code over time. It allows multiple developers to work on a codebase simultaneously while maintaining a history of changes.

GitHub is a web-based platform built on Git that hosts repositories and provides collaboration features such as issue tracking, pull requests, and project management tools. While Git operates locally on a developer's machine, GitHub enables remote collaboration and coordination.

Core Concepts Used in This Assignment

Repository (Repo)
A repository is a project folder tracked by Git. It contains all source code, configuration files, and version history.

Issue
An issue is a task created within a GitHub repository. Issues are commonly used in industry to define features, bugs, or enhancements. In this assignment, students create an issue before beginning work to simulate a common professional workflow.

Branch
A branch is an independent line of development within a repository. Branches allow developers to work on features without affecting the main codebase. Students create a feature branch tied to their issue.

Commit
A commit is a saved snapshot of changes made to the codebase. Each commit includes a message describing the change, creating an auditable history of work.

Push
Pushing sends local commits to the remote repository hosted on GitHub, making changes visible to collaborators.

Pull Request (PR)
A pull request is a formal request to merge changes from one branch into another (typically into the main branch). It provides an opportunity to review changes before integration.

Merge Conflict
A merge conflict occurs when Git cannot automatically reconcile differences between branches. Resolving conflicts requires selecting which changes to keep. This simulates real-world integration challenges in team development.

Technical Setup

To implement this assignment, students need:
- A GitHub account (Available at https://github.com)
- Git installed locally (Available at https://git-scm.com)
- An integrated development environment (e.g., VS Code)
- The appropriate programming language runtime (e.g., .NET SDK for this case)

Individuals unfamiliar with command-line Git may use graphical interfaces such as GitHub Desktop or built-in IDE Git tools.

K-12 AI Curriculum Design: A Review of Frameworks, Approaches, and Evaluation

Ni Lei
nlei1@students.kennesaw.edu

Zhe Zhao
zzhao1@students.kennesaw.edu

Zhigang Li
zli8@kennesaw.edu

Xin Tian
xtian2@kennesaw.edu

College of Computing and Software Engineering
Kennesaw State University
Marietta, GA 30060, USA

Abstract

An effective K–12 AI curriculum must extend beyond technical skills to cultivate computational thinking, creativity, problem-solving, and ethical awareness. This article reviews the growth of K–12 AI curriculum research from 2019 to 2024, examining the theories, frameworks, and models that guide curriculum content and design. It analyzes development approaches ranging from individual efforts to co-design and explores collaborative partnerships among universities, K–12 educators, industry, and government. The review also evaluates qualitative, quantitative, and mixed-method approaches used for curriculum assessment. Despite notable progress, significant challenges remain in both design and implementation. This study argues that effective AI curricula require stronger theoretical grounding, sound instructional design, and rigorous, evidence-based evaluation. The review highlights current inconsistencies and recommends more systematic practices to better align curriculum development with educational goals and societal needs, while strengthening validation processes to support long-term impact.

Keywords: K-12 curriculum design, Artificial Intelligence Education, curriculum design framework, theoretical guidance, curriculum design approach, curriculum evaluation.

Recommended Citation: Lei, N., Zhao, Z., Li, Z., Tian, X., (2026). K-12 AI Curriculum Design: A Review of Frameworks, Approaches, and Evaluation. *Information Systems Education Journal*, v24(n2) pp 75-86. DOI# https://doi.org/10.62273/GIII1608

K-12 AI Curriculum Design: A Review of Frameworks, Approaches, and Evaluation

Ni Lei, Zhe Zhao, Zhigang Li, and Xin Tian

1. INTRODUCTION

As artificial intelligence (AI) is rapidly growing and profoundly impacting our lives, AI education has been widely promoted and popularized. Research on AI education could be traced back to 1970 when Carbonnell (1970) introduced a new computer-assisted instruction system (CAI) to education, i.e., information-structure-oriented (ISO), where artificial intelligence techniques are extensively applied via a program called "Scholar" to improve students' learning experience. Considerable research has been done since then, with scopes covering kindergarten to higher education, and topics range from implementing AI technologies in education to teaching AI literacy to students.

With the emergence of machine learning, robotics, and block-based programming environments allowing students to create programs by dragging and connecting visual code blocks, making programming more accessible for beginners and young learners, AI education in K-12 has drawn increased attention of governments, educators, educational institutions, and researchers. Touretzky et al. (2019b) point out that the young generation is growing up with AI, and they likely interacted with Siri or Alexa before starting kindergarten. They (2019a) suggest that children should be equipped with AI literacy from an early age and that AI education should be integrated throughout the K–12 years to foster students' early awareness of AI-related careers. Efforts have been made to explore teaching AI to K-12 students. The previous and ongoing research themes on this topic include curriculum design, AI literacy, AI ethical and societal issues, teaching tools, pedagogies, and the integration of AI learning into other subjects (Grover, 2024). More recently, the spread of chatbot-based AI tools, especially those powered by large language models, has made AI more visible in students' everyday learning experiences. This trend further underscores the importance of K-12 curricula that support students in understanding and responsibly engaging with AI.

A well-designed AI curriculum can help students master core AI technology efficiently and cultivate their computational thinking, problem-solving skills, ethical awareness, and creativity. Therefore, this review focuses on the curriculum design of K-12 AI education, to synthesize the framework guidance or theoretical foundations of K-12 AI curriculum design from 2019 to 2024 and analyze the adopted design approaches and evaluation methods. Three questions are addressed:

- What are the theoretical guidelines, and how are they implemented in the curriculum design?
- What are the approaches utilized in the curriculum design?
- What methods are used to evaluate the efficiency of the curriculum design?

The goal of this review is to highlight current research trends in designing the K-12 AI curriculum, identify existing limitations and gaps, and provide valuable insights for educators and researchers on future directions. Thus, this review will contribute to the future design of the K-12 AI curriculum and improve the teaching and learning experience in K-12 AI education.

2. METHODOLOGY

This study adopts a systematic literature review approach to explore the above-mentioned questions. A transparent and replicable process was used to ensure the credibility and comprehensiveness of the review.

The review followed the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) framework (Page et al., 2021) to ensure transparency in article identification, screening, and inclusion (see Figure 1).

Databases such as IEEE Xplore, ACM Digital Library, and Springer, as well as the academic search engine Google Scholar, are used to gather relevant studies. A comprehensive search is conducted using a combination of keywords, including "K-12", "Artificial Intelligence", "AI", "curriculum design framework", "curriculum design approach", and "curriculum design evaluation", and the papers are selected from 2019 to 2024. To minimize selection bias, studies were required to be peer-reviewed journal articles or conference papers focused on K-12 AI curriculum design, including theoretical frameworks, design approaches, and evaluation methods. Only studies with clear theoretical foundations and research methodologies, published in English after 2019 and available in full text, were included.

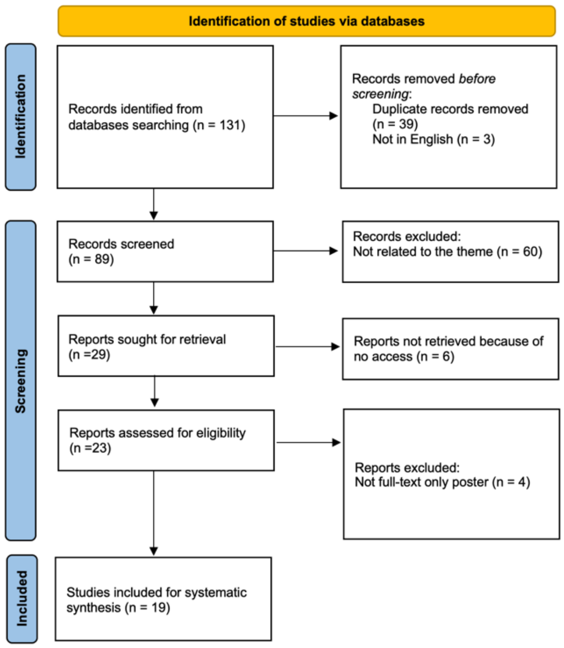


Figure 1: PRISMA Diagram

All studies identified through the search process were independently screened by two trained reviewers with relevant experience in AI curriculum design and IT backgrounds. Initial screening is based on titles and abstracts, followed by a full-text review to determine eligibility according to predefined inclusion criteria. Data from each eligible study is then extracted by the reviewers, capturing information such as theoretical foundations, framework, design approaches, and curriculum evaluation strategies. Any disagreements in data extraction or study eligibility assessment were resolved through discussion, and when needed, a third reviewer was consulted.

The data from the selected studies were synthesized using a qualitative thematic analysis approach to identify recurring patterns and key concepts across different research outputs. An iterative coding process was applied to categorize findings into broader themes related to theoretical foundations, curriculum design approaches, and evaluation strategies. All coding was conducted by each reviewer to ensure consistency, with differences resolved through discussion.

3. RESULT

This section presents the key findings of the study, focusing on three main aspects of K-12 AI curriculum design: theoretical guidance, content design approaches, and content evaluation methods.

3.1 Content Theoretical Guidance

Curriculum design needs to be guided by scientific theories and frameworks to ensure its systematicity, effectiveness, and adaptability (Tyler, 2013). The curriculum design framework provides structured guidance for learning objectives, instructional methods, and assessment strategies, ensuring that the curriculum meets the needs of diverse learners and adapts to the rapidly changing social and technological environment (Wiggins & McTighe, 2005). This highlights the global recognition of the importance of structured AI curriculum design in K-12 education, leading to efforts at the national level to establish standardized frameworks and curricula. Therefore, many national and international representative institutions have dedicated themselves to providing guiding frameworks.

At the K-12 level, there was little guidance for teaching about AI, compared with the broader subject of computing before 2018 (Touretzky, Gardner-McCune, et al., 2019a). Therefore, in May 2018, the Association for the Advancement of Artificial Intelligence (AAAI) and the Computer Science Teachers Association (CSTA) launched a collaborative initiative to establish national guidelines for K-12 AI education.

Additionally, organizations such as AI4All and the International Society for Technology in Education (ISTE) have acknowledged these needs and have started taking steps to address them (Touretzky, Gardner-McCune, et al., 2019b). These initiatives are laying the groundwork for AI education in K-12. In 2019, AI4K12 released the "Five Big Ideas in AI" poster and graphic, which includes Perception (Computers perceive the world using sensors), Representation and Reasoning (Agents maintain representations of the world and use them for reasoning), Learning (Computers can learn from data), Natural Interaction (Making agents interact comfortably with humans is a substantial challenge for AI developers), and Societal Impact (AI applications can impact society in both positive and negative ways), and these five concepts encompass the scope of the field and make it more accessible to teachers. Besides, this "Five Big Ideas" was further broken down into four grade bands: K-2, 3-5, 6-8, and

9-12, and this allows curriculum designers to create AI courses according to students' cognitive development at different age levels, ensuring a progressive development in AI education (Touretzky, Gardner-McCune, et al., 2019b). This framework provides definitions and examples for each idea, along with detailed guidance on what K-12 students are expected to accomplish in each area.

The United Nations Educational, Scientific, and Cultural Organization (UNESCO) has also made important efforts in promoting the standardization of AI. The UNESCO (2022) report, "K-12 AI Curricula — A Mapping of Government-Endorsed AI Curricula", aims to support the development of tools and frameworks to create a comprehensive guiding framework for AI competencies. For example, there are three frameworks, the AI Literacy Competency Framework, AI4K12: Five Big Ideas, and the Machine Learning Education Framework, developed to inform the development of AI curricula. By analyzing these AI curriculum guidelines from various countries, UNESCO has proposed recommendations on curriculum content, learning objectives, and implementation strategies to guide the K-12 AI curriculum development.

Through research, we have found that scholars adopted different strategies when selecting guiding frameworks or theories in curriculum content design, and these approaches are mainly divided into the following categories (see Table 1).

3.1.1 Integrating Established Frameworks and Theories in Curriculum Design

Different scholars have modified these established frameworks based on their contexts and have integrated other educational theories into the curriculum design.

Sabuncuoglu (2020) developed a 36-week open-source AI curriculum for middle school education, using "Five Big Ideas on AI Education" as its foundational framework when designing the modules. The author presented the curriculum design process and design details about the lecture structures in the article. There are three modules in the curriculum, and each module covers the Five Big Ideas. Additionally, the author used Akker's spider web curriculum development framework (Jan van den Akker, 2010) when designing a new curriculum. There are ten points, which are rationale, objectives, content, learning activities, teacher role, materials and resources, grouping, location, time, and assessment in the design journey to keep the balance of the course, and these ten elements are interconnected within the framework, and the web metaphor highlights both the delicate and interdependent nature of designing a new curriculum.

Kim et al. (2021) designed a curriculum that primarily focused on parts of the Five Big Ideas. They also constructed the curriculum's content by surveying Computer Science Curricula 2013 (CS2013) (ACM Computing Curricula Task Force, 2013) with an adjusted difficulty, which is suitable for elementary school students by adopting a backward curriculum design approach, which means that they identified the desired learning outcomes first and then designed those related learning activities with instructional strategies to ensure that the course content aligns with the CS2013 computer science education framework. The authors identified three core competencies essential for achieving AI literacy: AI Knowledge, AI Skill, and AI Attitude.

Su & Yang (2024) primarily refined AI4K12's Five Big Ideas (AI4K12, 2025) by employing a conceptual analysis approach to make them more suitable for early learners. Combining with robotics, this study proposes an AI curriculum framework suitable for young children, providing valuable theoretical guidance for future early childhood AI education.

It can be observed that while some scholars adopted the "Five Big Ideas" framework, they modified it by integrating additional theories or selectively omitting certain components to better align with their specific curriculum design needs. Many scholars have also designed curricula by integrating their own teaching experiences with other educational frameworks or theories. To design an AI curriculum for kindergarten children, Su and Zhong (2022) applied the framework developed by Scott, which includes four key dimensions: (1) aims, goals, objectives, or declarations of outcome, (2) subject matter, domains, or content, (3) methods or procedure, (4) evaluation and assessment. They also adopted points from Kim et al. (2021), who summarized three competencies to achieve AI literacy: AI Knowledge, AI Skill, and AI Attitude. With the guidance of these frameworks and theories, the authors modified the curriculum content suited for kindergarten children.

Building on the curriculum recommendations of Kim et al. (2021) and Su and Zhong (2022), Zhao et al. (2024) developed an AI course that includes 10 units, covering a wide range of content from basic AI knowledge to specific applications for grades three and four, and the authors examined the changes in students after using a mixed-methods assessment approach.

While using theory to guide course design, some scholars have specifically developed curricula tailored for specific students, like female high school students. Alvarez et al. (2022) integrated social media bots, the sentiment of natural language in different media, and the role of AI in criminal justice to spark their interest and career identity in computer science. In the curriculum design, the authors referred to the AI4K12 Big Idea 3 progression chart because it included high school-level objectives specifically related to machine learning algorithms, neural networks, and datasets.

AI is a complex topic that integrates extensive information from various disciplines, ranging from mathematical algorithms to ethical considerations, making comprehensive knowledge transfer challenging (Sabuncuoglu, 2020). Therefore, some scholars have also incorporated interdisciplinary concepts during the design to help students better understand AI. Sabuncuoglu (2020) designed the course to help students better understand how science and innovation work in conjunction by combining artificial intelligence courses with Biology, Physics, and Sociology. In the curriculum, under the guidance of "Five Big Ideas", each module begins with a relevant physical explanation to help students understand how computers perceive the world — including how they see, hear, and even simulate the sense of taste. To understand how to build a human-like computational device, students first need to know how human organisms work, and this is exactly where the interdisciplinary course design of AI connects. Monteith et al. (2022) combined AI concepts with various artistic disciplines, such as art, music, and poetry, to design a 20-hour high school course.

3.1.2 Developing Self-Designed Frameworks for Curriculum Design

Recent research indicates that many scholars are developing their own self-designed frameworks for curriculum design in addition to relying on established theoretical models.

Chiu et al. (2022) introduced the AKIEE framework, a modular, level-based curriculum co-created with multiple stakeholders. It organizes content into five modules, which are respectively Awareness, Knowledge, Interaction, Empowerment, and Ethics (AKIEE) across varying difficulty levels to support differentiated instruction. Chiu (2021) also developed a framework based on four dimensions: content, product, process, and praxis, using teacher interviews to align instructional strategies with curricular goals.

Framework Construction Strategies	Brief Description	Sample Studies
Integrating Established Frameworks and Theories	Full use or part use of "Five Big Ideas on AI Education"	Sabuncuoglu (2020); Kim et al. (2021); Su & Yang (2024); Alvarez et al. (2022)
	Computer Science Curricula 2013 as the guidelines.	Kim et al. (2021)
	Akker's spider web curriculum development framework.	Sabuncuoglu (2020)
	Scott framework	Su and Zhong (2022)
	Curriculum recommendations from Kim et al. and Su and Zhong	Zhao et al. (2024)
Developing Self-Designed Frameworks	AKIEE curriculum framework	Chiu et al. (2022)
	A curriculum framework combining four aspects: content, produce, process, and praxis.	Chiu (2021)
	SAC Model (Student-AI Collaboration Model)	J. Kim et al. (2022)
	A comprehensive four-dimensional AI Literacy Framework (AILF)	Kong et al. (2024)
	The "Why-What-How" early childhood AI curriculum design framework	Yang (2022)

Table 1: Framework Construction Strategies: Descriptions and Representative References

Other frameworks draw on established theories. J. Kim et al. (2022) proposed the Student-AI Collaboration (SAC) Model, grounded in Distributed Cognition theory, which emphasizes the interplay among students, AI tools, and the learning environment. Kong et al. (2024) presented the AI Literacy Framework (AILF), which includes cognitive, metacognitive, affective, and social dimensions, providing a comprehensive structure for developing AI literacy.

For younger learners, Yang’s team (2022) proposed the "Why-What-How" early childhood AI curriculum design framework and designed a course, “AI for Kids,” with course objectives, core knowledge areas, and specific teaching methods under the guidance of constructivist learning theory, sociocultural learning theory, and AI education framework. Additionally, the authors also emphasized interdisciplinary integration and cultural relevance.

3.2 Content Design Approaches

Designing curricula for K-12 education is more complex than for higher education, as it requires careful consideration of how new initiatives are put into practice, with significant variation in implementation anticipated across different schools (Chiu & Chai, 2020), it also involves rapidly evolving technological advancements, different cognitive backgrounds, and the diverse needs of learners. Therefore, more relevant stakeholders should be involved in the initial stages of curriculum design to minimize inconsistencies in implementation and ensure that the curriculum aligns with diverse needs. Through the analysis of these papers, we divided the methods of curriculum design into independent design and co-design from the perspective of collaboration level in curriculum design (see Table 2).

Independent design and Co-design

In the literature on K-12 AI education, curriculum design approaches vary, with some authors developing curricula independently and others adopting collaborative co-design methods. Several studies report curricula created solely by individual instructors or researchers (see Table 2). In contrast, many others highlight the importance of co-design, involving multiple stakeholders in the curriculum development process. These collaborative efforts include partnerships among researchers, K-12 teachers (with and without subject-matter expertise), university faculty, industry professionals, and government agencies.

For example, Chiu et al. (2022) described a cross-sector initiative that brought together five key groups: university educators with expertise in education, AI specialists, K-12 teachers, industry representatives, and government partners. This project developed the AI4Future co-creation framework to guide the collaborative curriculum design process, offering a valuable model for engaging diverse stakeholders in AI education.

Design Approaches	Brief Description	Sample Studies
Independent design	The authors designed the curriculum independently, without collaboration with external stakeholders.	Alvarez et al.(2022); Chiu (2021); Kim et al. (2021); Kong et al. (2024); Monteith et al. (2022); Su and Yang (2024); Yang (2022); Zhao et al. (2024)
Co-design	Authors co-designed the curriculum with university specialists, industry professionals, government agencies, and K-12 teachers, with collaboration formats varying across studies.	Chiu et al. (2022); Sabuncuoglu (2020); Chiu (2021); Lin & Van Brummelen (2021) Gardner-McCune et al. (2022) Xie et al. (2024)

Table 2: Curriculum Design Approaches

Other studies have emphasized the role of teacher input in curriculum development. Sabuncuoglu (2020) used semi-structured interviews with 17 ICT teachers to gather feedback that ultimately led to revising the course structure from interconnected modules to more independent ones. Chiu (2021) also incorporated semi-structured interviews, working with three experienced research assistants and educators to identify key curriculum components through a hybrid approach of inductive and deductive thematic analysis. This collaborative process informed both the content and implementation strategy of the curriculum.

Co-design not only supports curriculum relevance but also fosters mutual learning. Gardner-McCune et al. (2022) collaborated with middle school teachers and university researchers to ensure that the resulting curriculum was both pedagogically sound and engaging for students. Recognizing that teachers without computer science backgrounds may face challenges teaching AI. Lin and Van Brummelen (2021) emphasized the need to tailor curriculum development to teachers' expertise. They proposed a teacher-led workshop involving 15 K-12 teachers and researchers, where participants collaboratively developed AI modules that integrated with core school subjects. In this model, teachers contextualized AI within familiar content areas while researchers provided technical guidance. Similarly, Xie et al. (2024) described a co-design process involving participants from diverse disciplines to create a cross-curricular AI program.

3.3 Content Evaluation Methods

A well-designed curriculum ensures that course objectives align with learners' needs and plays a critical role in maintaining high-quality teaching and learning outcomes. As such, evaluating curriculum content is essential. However, research indicates that not all curriculum designers conduct a formal evaluation of their content. Among those who do, various evaluation methods are employed, including qualitative, quantitative, and mixed methods approaches (see Table 3).

3.3.1 Qualitative Research

Qualitative evaluation methods are commonly used, such as self-assessment (Su & Zhong, 2022), thematic analysis (Chiu & Chai, 2020), semi-structured interviews (Chiu, 2021), classroom observation, feedback (Sabuncuoglu, 2020) and teacher reflections (Gardner-McCune et al., 2022). For instance, Sabuncuoglu (2020) gathered feedback from 60 students regarding their learning experiences, enjoyable moments, areas of confusion, and their interest in further AI learning. Similarly, Chiu (2021) conducted semi-structured interviews with 24 middle school teachers to explore their perspectives on AI curriculum design.

3.3.2 Quantitative Research

Some scholars adopted quantitative methods to assess the curriculum's impact. Kim et al. (2021), for example, employed statistical analyses (t-tests) on data from 60 elementary students to measure improvements in AI literacy across three competencies, using a five-point Likert scale. Their results indicated significant gains after the curriculum intervention. Likewise, Kit Ng et al. (2023) developed and validated an AI literacy questionnaire for secondary students, administering it to 363 school students from two different schools in Hong Kong. The validity and reliability of the tool were confirmed through Confirmatory Factor Analysis (CFA).

3.3.3 Mixed Methods Research

Several studies utilized a mixed-methods approach to achieve a more comprehensive evaluation. In Kong's (2024) quantitative research, 128 high school students participated in the course and completed an AI concept test and a survey. The authors then conducted a statistical analysis using the Wilcoxon signed-rank test. For the qualitative research, self-reflection reports and group interviews were adopted. Finally, the authors employed thematic analysis to examine students' feedback on their AI learning experiences. Zhao et al. (2024) used a mixed-methods approach to evaluate the AI curriculum, combining quantitative analyses (e.g., paired and one-sample t-tests on knowledge tests and attitude questionnaires) with qualitative methods, including expert consensus assessment and semi-structured interviews. Results showed significant improvements in students' knowledge, skills, creativity, and attitudes toward technology.

Similarly, Chiu et al. (2022) quantified the impact of the course on students' AI learning ability, attitudes, and motivation through pre-and post-tests, paired t-tests, and ANCOVA, demonstrating statistically significant results. Additionally, they conducted qualitative research using semi-structured interviews. Su and Zhong (2022) combined pre/post testing, classroom observations, teacher interviews, and student self-assessment to evaluate curriculum effectiveness.

©2026 ISCAP (Information Systems and Computing Academic Professionals)
https://isedj.org/; https://iscap.us/org

Research Methods	Tools	Sample Studies
Qualitative	Self-assessment Semi-structured interviews Feedback Classroom observation Teacher reflections	Sabuncuoglu (2020); Chiu (2021); Su & Zhong (2022); Chiu & Chai (2020); Gardner-McCune et al. (2022)
Quantitative	T-test A 5-point Likert scale	Kim et al. (2021); Kit Ng et al. (2023)
Mixed Methods	A statistical analysis using the Wilcoxon signed-rank test Self-reflection reports Group interviews	Kong et al. (2024)
	Different types of T-tests Self-Assessment Questionnaire PATT Questionnaire Expert consensus assessment Semi-structured interviews	Zhao et al. (2024)
	Pre-and post-tests, paired t-tests, and ANCOVA Semi-structured interviews	Chiu et al. (2022)
	A pre-test and post-test Observing Interviews Self-assessment questionnaire	Su and Zhong (2022)

Table 3. Content Evaluation Methods

4. DISCUSSION

Research has shown that scholars have chosen different theories or frameworks to guide curriculum content design by adopting either independent or collaborative design methods, and scholars have evaluated the effectiveness of the content by using qualitative (interviews, observations, and questionnaires, etc.), quantitative (t-test, analysis of covariance, etc.), or mixed research methods. Overall, the design process is complete; however, at each step, we have identified deficiencies that require further refinement.

4.1 Content Theoretical Guidance

Through research, we have found that multiple frameworks or theories have been selected to guide K-12 AI curriculum design, among which the "Five Big Ideas" was widely used, and we have also found that when scholars design the curriculum, they mainly choose theories or frameworks in the following ways: some referred to frameworks provided by representative institutions and made only minor modifications to them; some integrated frameworks with other theories; some developed new instructional frameworks or models. The diversity of educational theories, frameworks, and design models presented in these studies reflects the inherent complexity of establishing theoretical foundations in educational research.

However, we have found that some scholars offered vague explanations when it comes to selecting or applying theoretical frameworks. First, some scholars briefly mention the theories they reference without introducing them or the frameworks. Second, some papers do not explain how the authors integrate theories and frameworks into specific curriculum design content. Third, in some cases, scholars do not justify their choice of a particular theory or framework as a reference standard for curriculum design. Lastly, it is important to highlight that some of the reviewed studies did not clearly identify an educational theory, design model, or AI framework underpinning their AI curriculum design.

In addition to these issues, poor or unbalanced curriculum design is also reflected in some papers. For instance, Kong et al. (2024) observed that traditional AI literacy programs often emphasize technical content while underemphasizing ethical reasoning during their study, which can affect students' holistic understanding of AI. The findings suggest the importance of balanced exposure across conceptual, metacognitive, affective, and social dimensions. This is the pressing challenge faced by educators and learners in real-world AI-integrated learning environments. Besides, with the widespread adoption of artificial intelligence in education and everyday life, the emergence of AI-generated misinformation has become an increasingly pressing issue. This phenomenon not only poses risks to information integrity but also challenges the ability of young learners to critically evaluate the content they encounter online.

As a result, there is a growing need to design curriculum components specifically aimed at addressing these challenges. Such curricula should equip K-12 students with the skills to identify, analyze, and question the accuracy and reliability of AI-generated content. By fostering digital literacy and critical thinking, these educational interventions can better prepare students to navigate an AI-mediated information landscape responsibly and thoughtfully. Therefore, curriculum designers should promptly revise and enrich the content in response to emerging issues associated with AI, ensuring that the curriculum remains relevant and capable of addressing evolving educational challenges.

4.2 Content Design Approaches

Independent curriculum design offers scholars significant flexibility to develop instructional content tailored to their vision. However, co-design approaches also present substantial advantages. Through the collaborative design process, teachers who lacked formal AI training were able to acquire essential AI knowledge necessary for curriculum development, thereby increasing their perceived competence and confidence in delivering AI education. Furthermore, co-design fosters teachers' professional autonomy, enabling them to design classroom activities and school-based curricula more effectively. This autonomy enhances their ability to lead, support, and inspire students, ultimately contributing to more personalized and impactful learning experiences.

However, we also observed that co-design comes with various challenges. Some authors have noted that even when teachers were involved in the early stages of curriculum design, their suggestions were ultimately not adopted due to factors such as time constraints, curriculum standards, policy requirements, etc. This highlights the complexity and unpredictability of the process, from initial conceptualization to final curriculum implementation. A complete curriculum design cycle requires close collaboration among educational administrators, curriculum designers, teachers, and policymakers to ensure that teachers' professional insights are effectively incorporated into the curriculum. Therefore, the content aligns with actual teaching needs, which is necessary to enhance the feasibility, sustainability, and overall effectiveness of the curriculum.

4.3 Content Evaluation Methods

Although the authors mentioned using relevant research methods to evaluate the curriculum in various papers, we still identified some issues. Firstly, some articles (Su & Zhong, 2022) reported using a pre-test and post-test methodology to evaluate the effectiveness of the course, but they do not seem to provide specific statistical results or numerical data from these tests. Additionally, some scholars have evaluated the effectiveness of only certain parts of the designed curriculum for other reasons (S. Kim et al., 2021; Chiu et al., 2022). Kim et al. (2021) conducted a trial run by implementing only the first module of the curriculum as a pilot test, given that the experimental subjects had no prior learning experience with AI. Moreover, due to other factors, such as the tight schedule of the school program, the author had to change the three-week study to a three-hour "Introduction to Computer Vision" workshop, making it hard to see holistic results, but it did show that students are interested in learning.

Additionally, there are other problems, such as some studies having unrepresentative samples, as they focus only on regional participants and lack cross-cultural validation; the sample size is too small; and the questionnaire content should be demonstrated to provide clear guidance. Lastly, we noticed that factors that may cause bias in self-report questionnaire surveys, such as teachers' instructional methods and students' preferences for their teachers, have not been fully considered or mentioned in many studies. Perhaps in the future, AI experts could design more rigorous experiments to validate the effectiveness of curriculum content to conduct more precise scientific measurements, so maybe co-designing with education scholars is a good choice.

All in all, some of the above issues are from the designers themselves, while others are caused by external objective factors. These limitations and feedback may provide insights for educational administrators and scholars, helping refine future AI curriculum design and improve its effectiveness, accessibility, and adaptability across diverse educational contexts.

Based on the findings of this review, several practical recommendations can be proposed for future K-12 AI curriculum design. First, curriculum developers should clearly state the theoretical or framework basis of the curriculum and explain how it informs learning objectives, instructional activities, and assessment tasks. Second, curriculum content should balance technical knowledge with ethical reasoning, critical thinking, and AI-related misinformation awareness, so that students can develop a more comprehensive understanding of AI. Third, co-design with teachers, AI experts, and, where possible, school administrators should be encouraged to improve curriculum feasibility and classroom applicability. Finally, curriculum evaluation should move beyond self-report data by combining pre- and post-tests, interviews, classroom observations, and, where possible, longitudinal follow-up, to provide stronger evidence of curriculum effectiveness.

5. CONCLUSION

In summary, over the past five years, research on K-12 AI curriculum design has grown steadily. A review of research shows that scholars have selected different theories, frameworks, or models to guide curriculum content. In terms of design approaches, they have adopted either individual design or co-design. Regarding collaboration choices, some have partnered with university teachers and institutions, while others have collaborated with K-12 teachers, students, or a combination of experts, including educators specializing in education, AI professionals, K-12 teachers, government representatives, and AI industry stakeholders. This diversity highlights the various forms of collaborative design. For curriculum evaluation, scholars have employed qualitative, quantitative, or mixed research methods. However, various issues still arise in different stages of the process. A well-designed curriculum requires guidance from a comprehensive, widely recognized theoretical framework, the selection of appropriate instructional design methods, and a rigorous, evidence-based evaluation of its effectiveness to ensure its educational impact and the achievement of learning objectives. In the future, scholars may consider improving these aspects to better align curriculum design with societal needs and to ensure a more rigorous evaluation and validation process.

Despite its broad coverage, this review has several limitations. First, only English-language publications from 2019 to 2024 were included, which may have excluded relevant studies and curriculum practices reported in other languages or published outside this time range. Second, the review relied primarily on qualitative thematic analysis, and the categorization of frameworks, design approaches, and evaluation methods inevitably involved interpretive judgment. Moreover, the included studies differed in student age groups, curriculum duration, educational settings, and evaluation tools, making direct comparisons across studies more difficult. Finally, although various frameworks and theories were discussed, this review did not thoroughly compare their advantages and disadvantages. A more detailed comparison of these frameworks and theories will provide clearer guidance for educators and researchers in future studies.

6. REFERENCES

ACM Computing Curricula Task Force (Ed.). (2013). Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM, Inc. https://doi.org/10.1145/2534860

AI4K12. (2025). Artificial Intelligence (AI) for K-12 initiative (AI4K12). https://ai4k12.org

Alvarez, L., Gransbury, I., Cateté, V., Barnes, T., Ledéczi, Á., & Grover, S. (2022). A Socially Relevant Focused AI Curriculum Designed for Female High School Students. Proceedings of the AAAI Conference on Artificial Intelligence, 36(11), 12698–12705. https://doi.org/10.1609/aaai.v36i11.21546

Carbonell, J. (1970). AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction. IEEE Transactions on Man Machine Systems, 11(4), 190–202. https://doi.org/10.1109/TMMS.1970.299942

Chiu, T. K. F. (2021). A Holistic Approach to the Design of Artificial Intelligence (AI) Education for K-12 Schools. TechTrends, 65(5), 796–807. https://doi.org/10.1007/s11528-021-00637-1

Chiu, T. K. F., & Chai, C. (2020). Sustainable Curriculum Planning for Artificial Intelligence Education: A Self-Determination Theory Perspective. Sustainability, 12(14), 5568. https://doi.org/10.3390/su12145568

Chiu, T. K. F., Meng, H., Chai, C.-S., King, I., Wong, S., & Yam, Y. (2022). Creation and Evaluation of a Pretertiary Artificial Intelligence (AI) Curriculum. IEEE Transactions on Education, 65(1), 30–39. https://doi.org/10.1109/TE.2021.3085878

Gardner-McCune, C., Touretzky, D., Cox, B., Uchidiuno, J., Jimenez, Y., Bentley, B., Hanna, W., & Jones, A. (2022). Co-Designing an AI Curriculum with University Researchers and Middle School Teachers. Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2, 1306–1306. https://doi.org/10.1145/3545947.3576253

Grover, S. (2024). Teaching AI to K-12 Learners: Lessons, Issues, and Guidance. Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1, 422–428. https://doi.org/10.1145/3626252.3630937

Jan van den Akker. (2010). In Beyond Lisbon 2010: Perspectives from research and development for educational policy in Europe (Sheila M. Stoney, pp. 175–195). CIDREE.

Kim, J., Lee, H., & Cho, Y. H. (2022). Learning design to support student-AI collaboration: Perspectives of leading teachers for AI in education. Education and Information Technologies, 27(5), 6069–6104. https://doi.org/10.1007/s10639-021-10831-6

Kim, S., Jang, Y., Kim, W., Choi, S., Jung, H., Kim, S., & Kim, H. (2021). Why and What to Teach: AI Curriculum for Elementary School. Proceedings of the AAAI Conference on Artificial Intelligence, 35(17), 15569–15576. https://doi.org/10.1609/aaai.v35i17.17833

Kit Ng, D. T., Wu, W., Lok Leung, J. K., & Wah Chu, S. K. (2023). Artificial Intelligence (AI) Literacy Questionnaire with Confirmatory Factor Analysis. 2023 IEEE International Conference on Advanced Learning Technologies (ICALT), 233–235. https://doi.org/10.1109/ICALT58122.2023.00074

Kong, S.-C., Cheung, M.-Y. W., & Tsang, O. (2024). Developing an artificial intelligence literacy framework: Evaluation of a literacy course for senior secondary students using a project-based learning approach. Computers and Education: Artificial Intelligence, 6, 100214. https://doi.org/10.1016/j.caeai.2024.100214

Lin, P., & Van Brummelen, J. (2021). Engaging Teachers to Co-Design Integrated AI Curriculum for K-12 Classrooms. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 1–12. https://doi.org/10.1145/3411764.3445377

Monteith, B., Noyce, P., & Zhang, P. (2022). Teaching Artificial Intelligence Through the Arts in Beijing. The Science Teacher, 89(5), 42–49. https://doi.org/10.1080/00368555.2022.12293698

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. Systematic Reviews, 10(1), 89. https://doi.org/10.1186/s13643-021-01626-4

Sabuncuoglu, A. (2020). Designing One Year Curriculum to Teach Artificial Intelligence for Middle School. Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education, 96–102. https://doi.org/10.1145/3341525.3387364

Su, J., & Yang, W. (2024). AI literacy curriculum and its relation to children's perceptions of robots and attitudes towards engineering and science: An intervention study in early childhood education. Journal of Computer Assisted Learning, 40(1), 241–253. https://doi.org/10.1111/jcal.12867

Su, J., & Zhong, Y. (2022). Artificial Intelligence (AI) in early childhood education: Curriculum design and future directions. Computers and Education: Artificial Intelligence, 3, 100072. https://doi.org/10.1016/j.caeai.2022.100072

Touretzky, D., Gardner-McCune, C., Breazeal, C., Martin, F., & Seehorn, D. (2019a). A Year in K–12 AI Education. AI Magazine, 40(4), 88–90. https://doi.org/10.1609/aimag.v40i4.5289

Touretzky, D., Gardner-McCune, C., Martin, F., & Seehorn, D. (2019b). Envisioning AI for K-12: What Should Every Child Know about AI? Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 9795–9799. https://doi.org/10.1609/aaai.v33i01.33019795

Tyler, R. (2013). Basic Principles of Curriculum and Instruction. University of Chicago Press.

UNESCO. (2022). K-12 AI curricula: A mapping of government-endorsed AI curricula. UNESCO. https://doi.org/10.54675/ELYF6010

Wiggins, G., & McTighe, J. (2005). Understanding by Design. ASCD.

Xie, B., Sarin, P., Wolf, J., Garcia, R. C. C., Delaney, V., Sieh, I., Fuloria, A., Varuvel Dennison, D., Bywater, C., & Lee, V. R. (2024). Co-designing AI Education Curriculum with Cross-Disciplinary High School Teachers. Proceedings of the AAAI Conference on Artificial Intelligence, 38(21), 23146–23154. https://doi.org/10.1609/aaai.v38i21.30360

Yang, W. (2022). Artificial Intelligence education for young children: Why, what, and how in curriculum design and implementation. Computers and Education: Artificial Intelligence, 3, 100061. https://doi.org/10.1016/j.caeai.2022.100061

Zhao, H.-G., Li, X.-Z., & Kang, X. (2024). Development of an artificial intelligence curriculum design for children in Taiwan and its impact on learning outcomes. Humanities and Social Sciences Communications, 11(1), 1339. https://doi.org/10.1057/s41599-024-03839-z