Journal of
Information
Systems
Education

Volume 37
Issue 1
Winter 2026

Teaching Tip

Teaching Undergraduate IS Students Hands-on Generative AI Development Skills

Yee Kit Chan, Erica Lai, and Yu Chen

Recommended Citation: Chan, Y. K., Lai, E., & Chen, Y. (2026). Teaching Tip: Teaching Undergraduate IS Students Hands-on Generative AI Development Skills. *Journal of Information Systems Education*, 37(1), 1-17. https://doi.org/10.62273/HPDA8033

Article Link: https://jise.org/Volume37/n1/JISE2026v37n1pp1-17.html

Received: December 9, 2024
First Decision: May 11, 2025
Accepted: September 3, 2025
Published: March 15, 2026

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Journal of Information Systems Education, 37(1), 1-17, Winter 2026
https://doi.org/10.62273/HPDA8033

Teaching Tip
Teaching Undergraduate IS Students Hands-on Generative AI Development Skills

Yee Kit Chan
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
yeekitchan@cmu.edu

Erica Lai
Donald Bren School of Information & Computer Sciences
University of California, Irvine
Irvine, CA 92697, USA
elai14@uci.edu

Yu Chen
Lucas College and Graduate School of Business
San Jose State University
San Jose, CA 95192, USA
yu.chen@sjsu.edu

ABSTRACT

Generative AI (GenAI) is increasingly essential in the professional workplace, making it crucial to equip students with the necessary GenAI skills. In information systems (IS) education, students often find themselves in a unique position: they may possess a solid knowledge of technology but tend to lack the deep programming expertise of computer science and engineering majors. To address this gap, we developed a series of labs designed to introduce IS undergraduate students to hands-on generative AI development. The labs aim to familiarize students with OpenAI API, teach students to create programs leveraging the API for tasks such text generation, image generation, and transcription, guide students in practicing prompt engineering techniques, and enable students to build web applications powered by OpenAI APIs. We implemented these labs in an upper-level undergraduate IS elective course in 2024 with 93 students. Students reported increased confidence in creating generative AI powered applications and a deeper understanding of the limitations of current models. This teaching tip explores the feasibility and adoption strategies for other instructors interested in incorporating the labs into their courses.

Keywords: Artificial intelligence, Generative AI, Generative AI in teaching, Hands-on project

1. INTRODUCTION

Generative AI (GenAI) has dramatically developed and been adopted both in industry and in our daily lives. According to a report from McKinsey Global Institute (2023), "although generative AI is still in the early stages, the potential applications for businesses are significant and wide-ranging" and there is a possibility of significant change in jobs and occupations. Therefore, it is imperative for students in information systems (IS) to have the proper GenAI education in their courses. Touretzky et al. (2022) proposed Five Big Ideas in AI, a framework that lays out the various dimensions of AI literacy and levels of AI skills required in K-12 education, which is also of high value for higher education. Despite the high demand for AI literacy, Chen (2022) found in a study on current and future AI curricula in business schools that AI curricula remain at a very early stage, with machine learning (ML) and data analytics as the main topics in the AI curriculum.

Unlike in computer science and engineering disciplines, where AI is traditionally taught, it is important to consider the appropriate level of difficulty and format for AI in business and IS education. Different from computer science and engineering (CSE) students, IS students are not expected to know the very technical details of generative AI models or possess the same level of programming proficiency as CSE students. Meanwhile, according to *AACSB Thought Leadership* (2026), as AI literacy becomes a foundational expectation for all business students, IS students require preparation that extends beyond prompt engineering – a skill that can be readily developed across majors – toward domain-specific capabilities in AI-enabled systems, data, governance, and organizational decision-making. IS education researchers have been exploring innovative methods for IS students to develop systems that match their technical background, such as through low-code or no-code approaches (Wang & Wang, 2021), to prepare IS students on how to use generative AI strategically in businesses and organizations in the future. What is the right balance for generative AI education?

Now, the rise of foundation models allows users to develop applications without having to know the technical details about the models while gaining the benefits of the models. In this teaching tip, we introduced a set of generative AI labs that support students to learn generative AI by building applications that directly use various generative AI models. In particular, the labs started by integrating OpenAI models. The labs include a variety of activities that engage students in managing Open AI accounts and understanding Application Programming Interface (APIs), writing programs using a variety of models, such as text generation, image generation, transcription, prompt engineering, the cost and economy of APIs, and creating web applications. The labs are designed to help students gain hands-on skills of generative AI. In this way, IS students can learn to create applications using generative AI models without needing extensive technical and programming skills. We implemented the labs in an upper-division IS elective course among 93 undergraduate students. The labs were well received by students and showed increased levels of self-efficacy in AI knowledge and learning after the modules.

These labs can be used in IS courses or related disciplines where students have basic programming knowledge. Example courses include business programming, data mining and analytics, database systems, information system development, business intelligence, cybersecurity, etc.

2. LITERATURE REVIEW

2.1 General Information

In the IS field, there has been an increasing amount of research on AI education. Before the rise of generative AI, some educators already started raising the importance of AI literacy in higher education, and especially among business students (Ma & Siau, 2019; Siau et al., 2018). There have been a few experiments on AI education among business students. For example, previous research by Chen and Hill (2020) presented an exploratory study on how to teach undergraduate business students to create chatbots using a no-code approach and found that the teaching modules would be most effective when it is suitable for students' technical backgrounds, relevant to students' life experiences, and connected to their career and professional development. Similarly, Chen et al. (2022) presented a gamified approach to teach business undergraduates to learn AI through the lens of gamified social entrepreneurship. The modules introduce students to the concepts of AI through demos, the applications of AI through case studies, and then invite students to identify a social problem in their communities and propose AI-powered solutions to address the problem they have identified. Students reported it as an engaging way to gain AI literacy and enhance their innovation skills. The no-code AI method is also used by Sundberg and Holmström (2024) to teach machine learning models among university students with different types of educational backgrounds, especially those without a technical background. Students found it an intuitive way to explore data and machine learning models in a visual way. Some educators also considered integrating ethical and responsible AI in IS education, as presented by Grøder et al. (2022) and Kong et al. (2023).

Most recently, with the rise of generative AI, there have been some efforts to integrate GenAI literacy into higher education. After conducting a study among organizational practitioners, Fleischmann et al. (2024) proposed a Diamond Model that advocates for refining educational approaches to equip students with the capabilities to apply GenAI, maintain accountability, ensure authenticity, and exercise agency in GenAI-driven workplaces.

Meanwhile, a growing amount of research has started to investigate the potential of using AI to support and augment learning in IS education by leveraging the opportunities while mitigating risks, such as misinformation and plagiarism (Kakhki et al., 2024; Van Slyke et al., 2023). While it has been a debatable topic of whether and how to use AI in education and schools (Kishore et al., 2023), AI tools, such as chatbots, could be used as a supplement resource to allow students to seek help in a more accessible and interactive way (Gupta & Chen, 2022). Previous research by Chen et al. (2023) also explored the feasibility of using chatbots to offer basic AI knowledge to undergraduate business students. The researchers created a chatbot that presented AI concepts and applications through dialog with students and created quizzes to test students' understanding. Given the technical difficulty of understanding regression for many business students when learning data analytics, Zhong and Kim (2024) studied how to involve students to use ChatGPT to facilitate their learning. They found that ChatGPT could help business students build solutions without being stuck in technical details and opened opportunities for students to understand how to work on data analytics tasks with the help of AI. In addition, research that investigates the impact of using generative AI to help CS students learn programming (Gottipati et al., 2023) is also increasing. However, AI education research among non-CS and non-technical students remains very limited.

3. LABS

The learning modules presented here are designed to help undergraduate IS students learn to prototype generative AI web applications from the backend to the frontend of a problem. The modules consist of six labs in a modular structure that requires students to use an OpenAI API. The labs are designed for students in technology-adjacent fields but without extensive generative AI and extensive coding experience. This section presents the six labs in detail so that the labs can be replicated by other instructors. Depending on the course goals and scope, the materials can be integrated in their entirety or in parts. The structure and sample screenshots of the labs are also included. Complete program code and exercise handout(s) are referred to in this section and included in the Appendix.

3.1 Lab 1: Setting Up for OpenAI Development

The first lab assignment guides students to set up their OpenAI environment by creating their OpenAI accounts and obtaining an API key. It familiarizes students with the OpenAI platform, including concepts such as API usage and the role of APIs in software development. The lab is designed to help students understand the role of APIs in software development, especially in the context of AI and the purpose of API keys in securing and managing access to APIs like OpenAI's platform.

This introductory lab requires students to create an OpenAI API key and learn to navigate the API usage panel. After being provided with the definition of an API key and how API keys are used in building applications, students create OpenAI API keys and learn about the importance of keeping the key secure, revoking, and regenerating API keys. Students are also introduced to the Usage page of their OpenAI dashboards where they can monitor their usage broken down by API key, as well as the remaining credits in their account (see Figure 1).

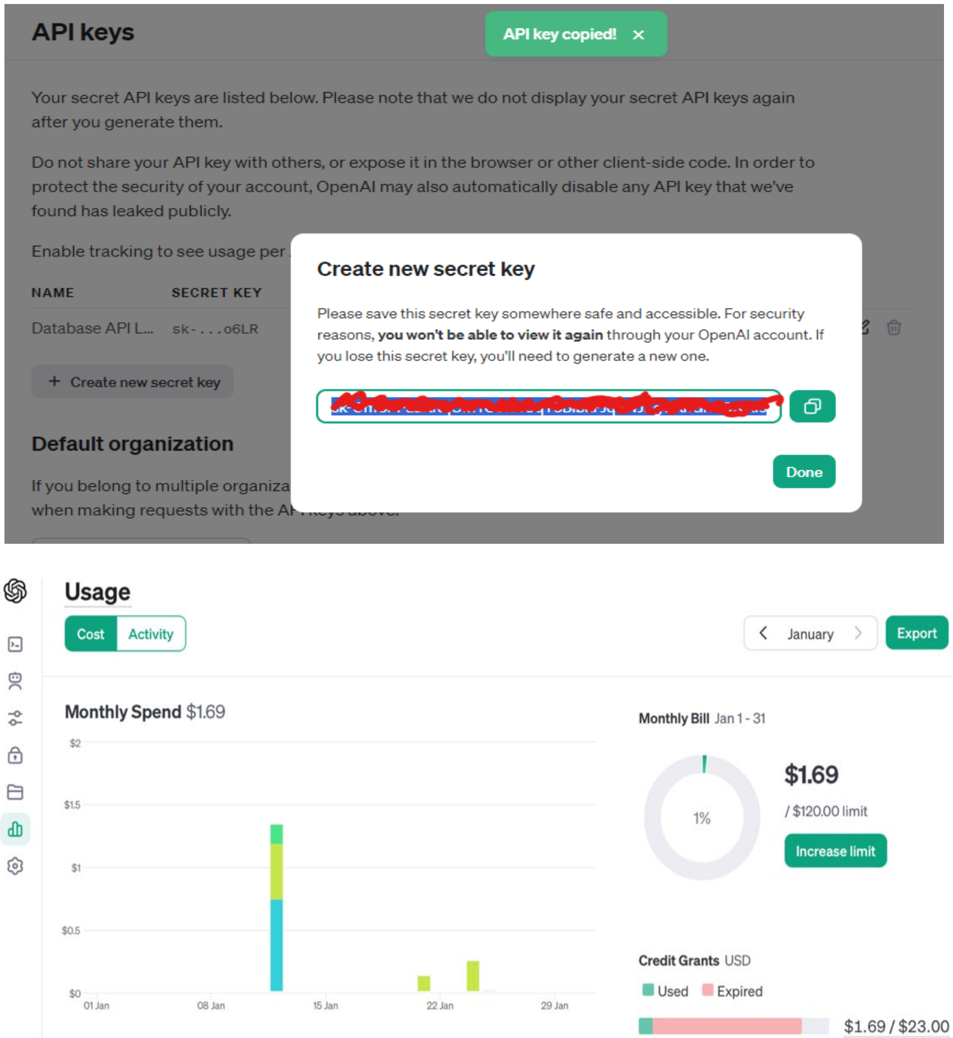

Figure 1. Generating an OpenAI API Key and Managing API Usage and Credits

3.2 Lab 2: Generating Text With OpenAI's GPT Models

Using OpenAI's text generation models, this lab is designed to guide students to create a simple application that can generate text. Lab 2 provides detailed instructions and sample codes to create a *Get_Completions* function which can produce responses from the GPT model that can be customized for specific purposes. This lab aims to help students: 1) understand the capabilities of OpenAI's Language Models: gain a foundational understanding of what OpenAI's language models like GPT-3.5-turbo are capable of, including their strengths and limitations in text generation, 2) develop skills in using OpenAI's API for text generation: learn to effectively utilize OpenAI's API to generate text, including setting up the necessary development environment, understanding API documentation, and executing API calls, and 3) explore and analyze different text generation techniques: experiment with various prompts and settings to understand how different inputs influence the AI's text output, thereby gaining insights into the mechanics of AI-driven text generation.

This lab introduces students to text generation using OpenAI's GPT-3.5 Turbo model. Students learn how to initialize the OpenAI client, construct messages with appropriate roles (system and user), and generate responses based on user input. The lab emphasizes the students' understanding of the structure of API requests—including the use of model, message, role, and content fields—while allowing students to interact with the model via a simple Python script. A sample function demonstrates how a user-prompt can be used to generate a poem. The complete code and implementation details are available in the Lab 2 GitHub repository in the Appendix.

3.3 Lab 3: Image Generation With OpenAI's DALL-E Model

In this lab, students use code to prompt OpenAI's DALL-E model to generate images, download them in their local environment, and display them in Google *Colab*. The goal is to support students in understanding the capabilities of OpenAI's image generation models, develop skills in using OpenAI's API for image generation, and download the generated images.

The lab guides students in building a simple application using OpenAI's DALL-E 2 model. The lab requires students to complete three core functions: one to generate an image from a text prompt, another to save the generated image, and a third to display the image. The headers of these three functions are provided in the lab, and students complete the lab by following the step-by-step tutorials. They then use an input function to provide prompts and automatically generate and save the images. A sample function to generate images using OpenAI's API has been included in the accompanying Lab 3 GitHub repository in the Appendix.

3.4 Lab 4: Insights From Audio Transcription

Adapted from a meeting-minutes-generation tutorial published by OpenAI (2024a), this lab guides students to transcribe a brief audio recording of a meeting, generate insights, and store the insights in a text file. By the end of the lab, students have a program to obtain the summary, key points, action items, and overall sentiment of the meeting. The program is formatted modularly to help students cultivate coding best practices, with each function serving a well-documented purpose. While this lab builds upon previous knowledge, using learnings from Labs 1 and 2 to set up the development environment and gather text completion-driven insights, the lab aims to help students gain a foundational understanding of what OpenAI's speech transcription model *Whisper* is capable of, including its strengths and limitations, and deepen their understanding of text generation functions to gain insights from the audio transcription.

Students build a function that uses OpenAI's *Whisper* model to transcribe audio content into text. The function takes an audio file path as input and returns the transcribed text. After obtaining the transcript, they apply text generation (learned in Lab 2)—such as summarization, key point extraction, action item identification, and sentiment analysis—by crafting targeted prompts to generate a full analysis of the meeting content. The full implementation and code can be found in the Lab 4 GitHub repository in the Appendix.

3.5 Lab 5: Prompt Engineering and Tokens

In Lab 5, students get a first taste of the power of creative prompt construction by selecting and testing sample prompts from OpenAI's example library. To provide students with a framework for effective prompt engineering, a sample analysis of a successful prompt and its elements is presented, along with a mnemonic acronym (FACTORS – Format of Response, Action Words, Context, Tone and Formality, Output, Role, Specific Items to Avoid) and exercises for usage. Last, we discuss tokenization, highlighting its importance to generative large language models and optimizing their costs. As the least coding-intensive lab in this series, this lab can be easily adapted for students with various technical backgrounds.

To expose students to a variety of themes of prompts, they are guided to use OpenAI's prompts. They are guided to use their text generation program from Lab 2, explore OpenAI's library of example prompts, then try out two prompts using the program from Lab 2. This exposes them to a variety of scenarios of prompt engineering and gives them a better understanding of the potential of prompt engineering, as well as what a well-constructed prompt can look like. See Figure 2 for sample prompts.

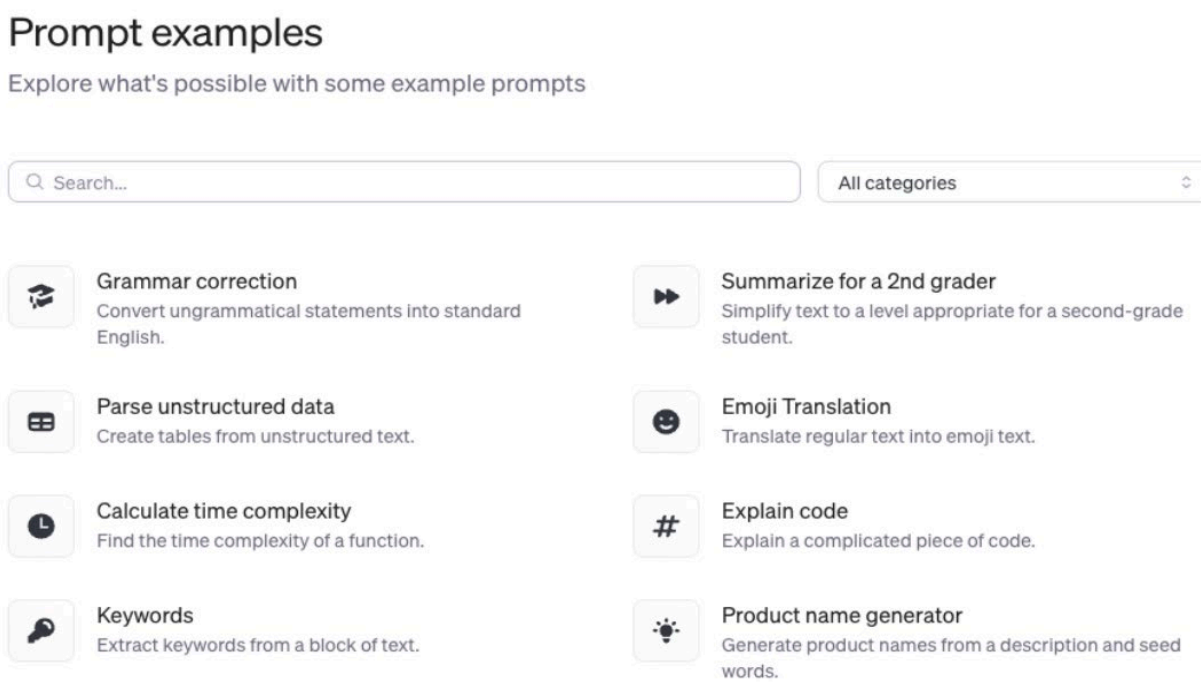

Figure 2. A Screenshot of the Prompt Examples Provided by OpenAI (2024b)

Students then engage in learning prompt engineering skills through a framework that we name as FACTORS (Format of Response, Action Words, Context, Tone and Formality, Output, Role, Specific Items to Avoid). Students are provided with good and bad prompt examples by analyzing them using FACTORS, highlighting where each technique is used effectively. With this new FACTORS familiarity, students then exercise these techniques by revising a "bad" prompt into a more effective one. The explanation of each factor is below.

1. **Format of Response** – specify the desired level of detail, brevity, and overall format of the response you wish to receive.
2. **Action Words** – use action words throughout your prompt to guide the model.
3. **Context** – offer background information to frame your prompt within a specific scenario/goal.
4. **Tone and Formality** – define the style and seriousness appropriate for the intended audience.
5. **Output** – provide an example output to illustrate your expectations.
6. **Role** – clarify the model's assumed identity, profession, expertise, etc.
7. **Specific Items to Avoid** – Identify any content to exclude from the model's response.

Finally, to help students understand the cost of API calls, we introduce the concept of tokenization and tokens. Since tokens are the currency of AI interactions, managing their usage is key to balancing cost and output quality. Students are instructed to provide one of the prompts they have written before and use the OpenAI (2024c) tokenizer to calculate the number of tokens it corresponds to. Students then revise the prompt to reduce the number of tokens, using strategies such as including only essential information only, rephrasing for conciseness, and replacing longer words with shorter synonyms. Figure 3 shows an example of tokenization calculation.

Journal of Information Systems Education, 37(1), 1-17, Winter 2026
https://doi.org/10.62273/HPDA8033

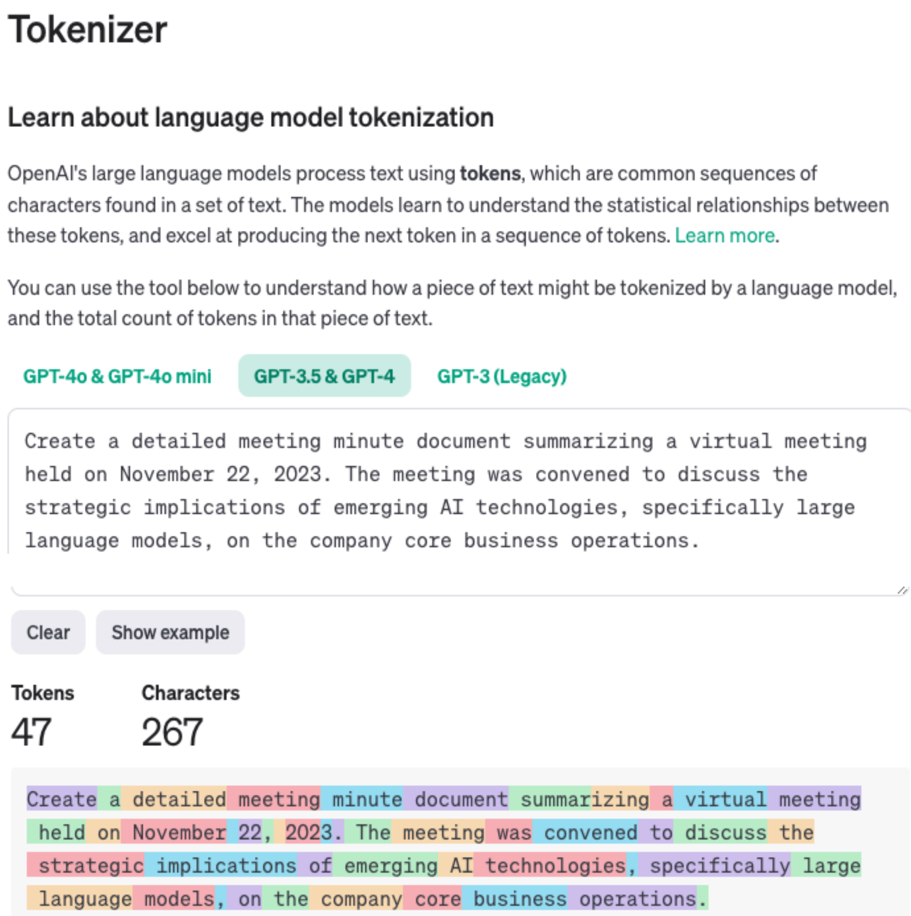

Figure 3. An Example of the OpenAI Tokenizer Calculating Token Usages

3.6 Lab 6: Creating User Interface for Generative AI Programs

So far, the labs focus on content generation in the IDE terminal. While the terminal is an effective way to display results during development and learning, it is not user friendly for a lay user. We pivot to *Visual Studio* (VS) Code in order to introduce *Streamlit*, an open-source framework for building web interfaces with only a few lines of code. In this lab, students are introduced to *Streamlit* as a tool for rapidly prototyping basic web applications, equipping them with a user interface to package and present their future work. In the process, students also familiarize themselves with Visual Studio Code, an industry-standard IDE, heralding a foray into local development with OpenAI and beyond.

Students will first set up the environment by installing Python and VS Code. After installing OpenAI and *Streamlit* libraries, students are directed to reuse their text generation program from Lab 2 as a sanity check for a successful development environment setup. They then move on to creating user interfaces with *Streamlit*, an open-source framework for building web interfaces, suitable for students with little to no technical experience in this area. Students are provided with sample code (available on Lab 6 GitHub repository in the Appendix) to create a basic *Streamlit* web interface to display the result of their text generation program from Lab 2. After students complete the lab, they should be able to view the web application with a graphical user interface (Figure 4).

The six labs serve as the foundation for students to gain some experience in building applications using OpenAI API. We list the APIs and platform used, estimated time and cost in Table 1.

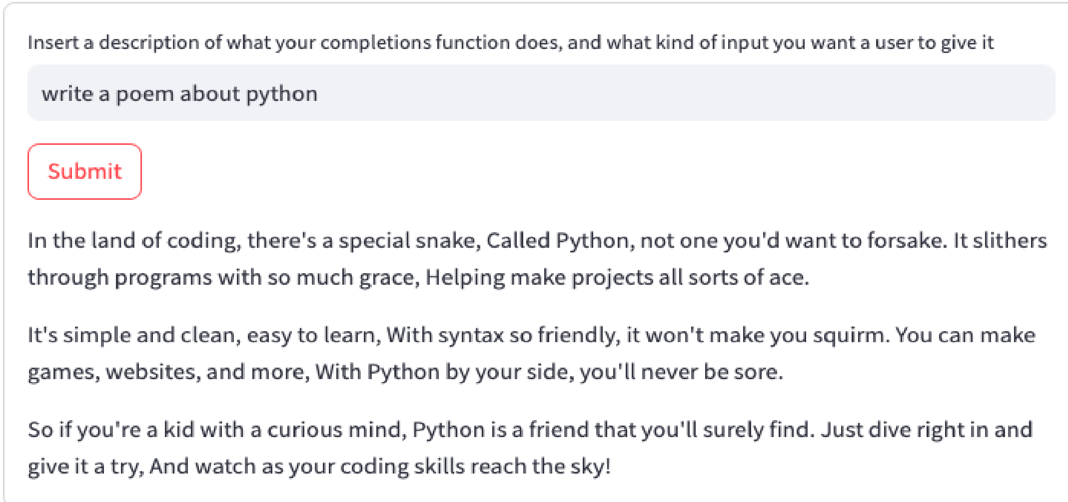

Figure 4. Sample Display of a *Streamlit* Webpage

ID	Goal	Learning Outcomes	API	UI	IDE	Est. Time	Cost
1	Setting up the environment for using OpenAI API	1) Understand APIs in AI applications; 2) Create, use and secure API keys; 3) Navigate the OpenAI dashboard	N/A	N/A	N/A	15-30 min	N/A
2	Develop a text generation application	1) Understand GPT-3.5-turbo; 2) Use OpenAI API for text generation; 3) Try prompts to analyze AI output	GPT 3.5	Text	Colab	30-60 min	$0.02
3	Develop an image generation application	1) Understand DALL-E 2 capabilities; 2) Use OpenAI API to generate images; 3) Download images from URLs with Python	Dall-E 2	Text	Colab	30-60 min	$0.05
4	Develop a transcribing app and analyze the content	1) Understand Whisper's transcription capabilities; 2) Use OpenAI API for analyze text	Whisper	Text	Colab	30-60 min	$0.05
5	Prompt engineering and development using applied scenarios	1) Use the FACTORS framework for prompt design; 2) Understand tokenization and its cost impact; 3) Refine prompts for efficiency and quality	All the above	Text	Colab	60-90 min	$0.02
6	Develop a web application	1) Use VS Code for Python coding; 2) Configure OpenAI locally; 3) Build Streamlit apps with OpenAI integration	GPT 3.5 + **Streamlit**	GUI	VS Code	60-90 min	$0.02

Table 1. Overview of Associated Learning Outcomes, Tools, and Costs for Each Lab

3.7 Innovation Project

After completion of the first four labs, students have gained familiarity with various functionalities of the OpenAI API, including text generation, image generation, speech generation, and transcription. To synthesize their learnings, students are instructed to work as a team to create a web application powered by GenAI that addresses a real-world problem of their choice.

They are instructed to identify a social problem in their community, aligning with the United Nations Sustainable Development Goals (UN SDGs), brainstorm solutions, and then create a web application leveraging the Generative AI technical skills they have gained so far. Students are provided with four weeks to work on a project with weekly iterations and feedback from the instructors.

4. IMPLEMENTATION

Collectively, the six labs of the GenAI modules aim to provide students with a progressively scaffolded learning experience in generative AI, from initial API configuration to the design of AI-driven web applications. To benefit from all six labs, students need to have basic Python coding experience. They do not necessarily need to have specific IDE experience like Colab or VS Code because the labs provide instructions. However, students without a coding background could also benefit from engineering knowledge on prompts presented in Lab 5. The sequence equips students with both practical skills and concrete projects, preparing them to undertake the culminating innovation projects with creativity and technical fluency. After finalizing the modules, we implemented them in an upper-division undergraduate IS elective course titled *Digital Innovation*. The pre-requisites of this course include two other IS courses that focus on introduction to business programming (such as Python). The 6-week labs were introduced to set a solid foundation for students to fast-prototype generative AI systems, which will be reported in a separate paper.

The labs were piloted in an upper-division undergraduate IS course in Spring 2024 with 93 students. Most students in the course were majoring in IS, except for 8 students majoring in general business, chemical engineering, computer engineering, human resources, biomedical engineering, marketing, finance, and kinesiology. The six labs were deployed as weekly take-home lab assignments. As a reference for course planning purposes, Lab 1 to Lab 5 were deployed between Week 2 and Week 6, and after a midterm exam, Lab 6 was deployed in Week 8. The innovation project ran from Week 9 to Week 13. Throughout the course, the instructor and teaching assistant (TA) provided additional Zoom-based office hours for students seeking help outside of class. Overall, all students completed Lab 1 to Lab 5 without major issues. When encountered, all technical issues were resolved with the help of the instructor or the TA.

The project's effectiveness was measured with a survey administered before and after the learning modules. The survey included five-point Likert scale items adapted from existing items in literature. These survey questions were mainly designed to measure student perceived self-efficacy of AI based on the work of Alvarez et al. (2022), Katuka et al. (2023), and Vandenberg and Mott (2023), focusing on their individual beliefs in their capability to use and interact with AI systems. Among 93 students enrolled in the course, 46 students completed both surveys. We analyzed the effectiveness of learning modules by conducting a paired-sample t-test using SPSS.

In addition to pre- and post-surveys used to capture overall learning outcomes, the idea to include brief written reflections arose after Lab 2 as a way to gather immediate, qualitative feedback. As a result, only Lab 3 and Lab 4 incorporated structured reflection prompts focused on students' experiences with AI models. The reflections were limited to being 50 to 100 words in length for the image generation lab (Lab 3) and audio transcription and summary lab (Lab 4). These two labs involve the use of three models: DALL-E 2 for image generation, GPT 3.5 for text generation (Lab 4) and *Whisper* for transcription (Lab 4). We analyzed students' reflections to gain further insights about their experiences of these labs.

5. STUDENT EXPERIENCE

Table 2 shows the detailed results of the analysis. Overall, students reported positive changes in all the survey questions. In particular, students indicated a significant increase in their knowledge about "building artificial intelligence applications" (Mean_before=2.63, Mean_after=3.63, p<0.001) and their confidence in "understanding the ideas behind how AI works" (Mean_before=3.80, Mean_after=4.20, p<0.001). In addition, students also reported significantly enhanced perception about "I can do well in AI" (M_before=3.76, Mean_after=4.02, p=0.022), "I can be successful in learning AI" (M_before=4.13, M_after=4.37, p=0.027), and "I can explain what AI is to my friends" (M_before=3.83, M_after=4.07, p=0.031). In addition, there is a slight increase on their perception about "I can use AI to help solve problems around me" (M_before=4.2, M_after=4.39, p=0.076), though the increase is not significant. It looks like after the learning modules, students were able to gain confidence on the concepts, applications, ideas behind AI, as well as their confidence in learning AI in the future. Further, their perceived increase in the ability to explain AI concepts to their friends indicated a deeper level of understanding and mastery of the AI knowledge.

Overall, students reported positive experiences from their interaction with Lab 3 and Lab 4. Many students reported that it was their first time building simple applications using Python code to interact with the models. Some students were pleasantly surprised to discover that they could write generative AI-powered programs with only a few lines of code. As one student reflected: "*This lab was interesting, and it was fun to be able to build an image generator from pretty simple code. Moreover, the code being able to call, give a prompt, and show the image of whatever was typed is super interesting. It shows how easy and simple it is to create code and AI together.*" Not all students found it a very smooth experience. For some students, "*Although I struggled a bit more than I would have expected, this assignment was enjoyable.*"

Item	Mean (SD)			P value (one-sided)
	Pre	Post	Diff.	
How much do you know about how to build artificial intelligence applications?	2.63 (0.97)	3.63 (0.83)	1 (1.16)	<.001***
I can do well in AI.	3.76 (0.71)	4.02 (0.86)	0.261 (0.86)	0.022*
I can figure out how to solve hard AI problems if I try.	3.70 (1.01)	3.85 (0.97)	0.152 (0.99)	0.151
I can use AI to help solve problems around me.	4.20 (0.78)	4.39 (0.71)	0.196 (0.91)	0.076
I can be successful in learning AI.	4.13 (0.81)	4.37 (0.68)	0.239 (0.82)	0.027*
I am confident I can understand the ideas behind how AI works.	3.80 (0.72)	4.20 (0.81)	0.391 (0.77)	<.001***
I can explain what AI is to my friends.	3.83 (0.77)	4.07 (0.71)	0.239 (0.85)	0.031*

N=46; *0.05<p<0.1; **p<0.05; ***p<0.01

Table 2. Results of Paired Sample T-Test Comparing Pre- and Post-Intervention Scores

For Lab 3, the image generation model, most students found it interesting and provided innovative application scenarios. One student shared, "*I think image generation has numerous opportunities for creative individuals. If used correctly, I believe artists and even ordinary people can leverage the technology to explore creative avenues, especially if they learn how to tailor their prompts according to their needs.*" Meanwhile, students also pointed out that the image generation capabilities still have room for improvement in terms of image quality and relevance. Some students have noticed that the quality of the generated images is also influenced by the quality of the prompts. As one student wrote, "*It is quite clear that when it comes to generative AI producing images, the quality is dependent on the specifications put into it. The more details and description, the better.*" Some also expressed concerns about image generation: "*With its development, an issue that could stem from this though, could be the production of fake photos and how the content is used. It's going to be very interesting where the ethical usage of this service goes as time progresses.*"

For Lab 4, the audio transcription model and text generation model, most students found it accurate in their lab. As one student put it, "*The transcription for my audio file was impressive and was very accurate in terms of how accurate the key points and action items were. I was surprised that the sentiment analysis was also pretty spot on (…) I think that audio transcription could be used to solve issues like inequality, specifically for those who are hard of hearing. It would be able to help them capture the main points presented in an audio file or even transcribe the entire audio, so they have full access to the same information as someone who is not hard of hearing.*" Students also proposed various scenarios where they can apply the models, such as in criminology and police interrogations. Similar to image generation models, students also pointed out concerns and limitations of the audio transcription and text generation models. As one student wrote: "*(...) students could use this as a short-cut and not put in the effort to really learn the class material through doing an assignment. Also, if not trained perfectly, I believe audio transcription can also inherit human bias, and potentially transcribe content that may be deemed as inappropriate, offensive, or socially unacceptable. Also, similarly to other forms of AI, privacy of data is always a concern. While these tools may be extremely useful, they must be used carefully, and it is difficult to discern who is responsible for keeping user data secure.*"

Overall, the data suggests that the labs serve as an appropriate tool for business students with entry-level programming skills to get a taste of building simple generative AI applications by using the models, playing with the models, and evaluating the opportunities and limitations of the models critically.

6. DISCUSSION

Notably, students reported being surprised at how easy it is to develop an application powered by generative AI, especially with limited programming experience. They also expressed their excitement and motivation in learning generative AI hands-on skills with the labs. In this section, we discuss areas where the labs can be extended and issues we have encountered.

6.1 Implementation Suggestions

The labs are most suitable for courses that are open to students with basic coding skills as prior knowledge and have a need for or interest in learning generative AI. Instructors can then tailor the lab series to align with students' prior programming knowledge, specific course objectives, learning outcomes, and time constraints, choosing to implement some or all of the six labs listed below. By carefully selecting and adapting these labs, instructors can empower students to harness the potential of generative AI and develop innovative applications. Table 3 presents the overview of the lab grouped by theme.

If the goal is to help students get a taste of the labs, Lab 1 and Lab 2 are the most relevant and could be easily integrated as a lab. If there are time constraints, the instructor could also consider directly giving a temporary API key to students for Lab 1 and deactivating the key afterwards. For students with limited or no coding background, Lab 1 and Lab 2 can still provide exposure to real-world tools and help develop computational thinking – such as inputs, outputs, prompts, responses, and how AI models interpret instructions – without requiring deep coding knowledge. If the goal is to help students explore various models, then Lab 2, Lab 3, and Lab 4 are most relevant for students to gain hands-on experience about text generation, image generation, and transcription features. If students have no coding background and the primary learning goal is prompt engineering, Lab 5 would be the most relevant, and can be further divided into sections such as prompt engineering playground to explore various prompts, prompt engineering principles, and tokens. Finally, Lab 6 is helpful if the instructor would like to help students build applications with graphical user interfaces.

6.1.1 Pedagogy. To further solidify students' understanding and encourage innovation, instructors may also consider adding a final project after the core labs. The labs themselves are designed to provide a solid foundation for students to learn generative AI API usage, prompt engineering, and application development. The above knowledge could potentially empower students to address a range of problems and create AI powered solutions. The culminating project, with specific parameters set by the instructors, could allow students to learn AI innovation by applying their learnings to topics of their choice. We have already experimented with two types of projects after introducing the labs. The process and outcome of students' innovation projects will be reported in a separate paper.

Purposes	Lab	Usage
Introductory Labs	Labs 1-2	Provide an introduction to generative AI, covering API usage and basic prompt engineering. These labs can be integrated into a course as standalone labs or by providing temporary API keys for a hands-on experience.
Model Exploration	Labs 2-4	Delve into generative AI models, offering practical experience with text generation, image generation, and transcription.
Prompt Engineering Deep Dive	Lab 5	Focuses on advanced prompt engineering techniques, including prompt engineering principles and token economy. It can be divided into smaller sections to accommodate different expertise.
Application Development	Lab 6	Guides students to build user-friendly apps with graphical user interfaces, leveraging the knowledge gained from previous labs.

Table 3. Overview of the Labs Grouped by Theme

6.1.2 Instructional Formats. While we implemented these labs as homework assignments with supplemental office hours, they can be adapted to various instructional formats. First, the labs can be conducted in-class, allowing for real-time guidance and collaboration among students. Alternatively, instructors could consider a hybrid approach, combining in-class exercises and out-of-class assignments to balance structured learning and independent study. Also, depending on the course learning outcomes, instructors may selectively incorporate specific lab components or exercises to introduce key concepts or reinforce learning objectives. In this way, there is room for instructors to tailor the delivery methods to the needs of the course and students to optimize the learning experience and maximize the benefits of the labs. Although the modules in this teaching tip were implemented in an in-person class, we do not anticipate major challenges for online delivery. In fact, the labs are assigned as homework, with student teaching assistants offering support via Zoom. Therefore, we expect the modules to work equally well in online or hybrid formats.

6.1.3 Resources. As outlined in Table 2, the primary cost associated with these labs are API calls. Among the various models used in the labs, the text generation models such as GPT-3.5, are relatively cost-effective, which are suitable for most students. While image generation and transcription API calls incur slightly higher costs, overall expenses remain manageable. In fact, most students reported spending less than $0.50 for the entire six-lab series. If budget and resources permit, instructors may also consider generating a limited number of API keys and distributing them to students for a specific timeframe so that students do not need to pay themselves. In addition to the API cost, instructors and students also need access to computers. For Labs 1-5, a web browser is sufficient; for Lab 6, students might need to download and install IDE, such as Visual Studio Code or PyCharm. It is also possible that there might be a cloud-based IDE that would not require installing software locally.

Given the rapid development of generative AI technologies, specific code and AI models used may become outdated. To ensure sustainability, all labs have been uploaded to a GitHub repository (see Appendix), allowing ongoing updates. Finally, while the modules were designed in the United States, the underlying principles are broadly applicable. Instructors in regions where OpenAI is not accessible can adapt the framework and substitute OpenAI's APIs with other generative models.

6.1.4 Instructor Readiness. While some prior coding experience is necessary to complete all six labs, we acknowledge that IS instructors possess varying levels of technical expertise. To accommodate this diversity, the labs are designed with flexibility in mind. For example, Lab 5 includes components focused on prompt engineering that do not require any programming, which makes it accessible to instructors with limited coding backgrounds. Instructors with experience using Google *Colab* may choose to integrate any combination of Labs 1 through 5, depending on their course objectives. Lab 6, which guides students in developing web-based applications, involves more advanced tasks such as interacting with APIs and working in IDEs. This lab may be more suitable for instructors already comfortable with such tools or seeking to extend their course in more technical applications.

6.1.5 Scalability Across Courses. The modular structure of the labs enables instructors to select content based on their own expertise, student preparedness, and specific course learning outcomes. The focus on prompt engineering in Lab 5 is particularly adaptable and can be integrated into a variety of courses both within and beyond the IS curriculum, which shows its flexibility across disciplines. Labs 1-4 and Lab 6 are particularly suited for courses that explore how generative AI can support business innovation and organizational effectiveness. In particular, Labs 2, 3, and 4 each address a distinct data modality – text, image, and audio – which allows instructors to tailor content to pedagogical needs and subject focus. These labs may be applied in a range of courses such as business analytics, database systems, data analysis, big data, systems design, and introductory business programming in Python.

6.2 Problems Occurred and Suggested Solutions

First, we noticed that providing a large code chunk to students often led to copy-pasting behavior. Some students tended to skip explanations and teaching components, focusing solely on the code. This could have resulted in missed opportunities to learn the underlying concepts and techniques. To address this, we are planning to integrate quiz questions into the next version of the labs to motivate students to delve deeper into the material and understand the design rationale. Second, we encountered challenges in Lab 6 when students ran their labs on VS Code using *Streamlit*, particularly due to varying operating systems, Python versions, and other environmental factors. Troubleshooting these issues consumed more time than anticipated. Based on our experiences with 93 students, we expanded the lab instructions to include a list of common issues and their corresponding solutions. As a result, we observed a significant decrease in Lab 6 issues in the following semester. However, we recognize that technology and systems are constantly evolving, so we will continue to update the list of potential issues and solutions accordingly. Ideally, a cloud-based IDE like Google *Colab* that supports *Streamlit* could mitigate issues arising from a variety of student computing environments.

7. CONCLUSIONS

This teaching tip introduced a set of six labs designed to teach hands-on generative AI skills. The labs were designed to cater to IS students with programming experience, but not at an advanced level. The labs were successfully piloted among 93 undergraduate IS students. The labs were well-received by students. Most students felt the labs were an engaging way for them to learn about hands-on generative AI development skills and their confidence in developing generative AI applications increased. Meanwhile, they noticed the limitations of the models and noted the caution of generative AI use responsibly. We hope the labs can inspire fellow IS instructors to adopt them in a variety of classes with varied versions.

8. ACKNOWLEDGEMENTS

This research was sponsored by the National Science Foundation (NSF) Grant No. 2142783. We thank the anonymous reviewers who offered valuable feedback.

9. REFERENCES

AACSB Thought Leadership. (2026, January 13). A Framework for Artificial Intelligence in Business Education: Exemplars and Critical Themes for Successful Integration. AACSB. https://www.aacsb.edu/insights/reports/2026/a-framework-for-artificial-intelligence-in-business-education?utm_source=chatgpt.com

Alvarez, L., Gransbury, I., Cateté, V., Barnes, T., Ledéczi, Á., & Grover, S. (2022, June). A Socially Relevant Focused AI Curriculum Designed for Female High School Students. *In Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 36, No. 11, pp. 12698-12705). https://doi.org/10.1609/aaai.v36i11.21546

Chen, L. (2022). Current and Future Artificial Intelligence (AI) Curriculum in Business School: A Text Mining Analysis. *Journal of Information Systems Education*, 33(4), 416-426.

Chen, Y., Albert, L. J., & Jensen, S. (2022). Innovation Farm: Teaching Artificial Intelligence Through Gamified Social Entrepreneurship in an Introductory MIS Course. *Decision Sciences Journal of Innovative Education*, 20(1), 43-56. https://doi.org/10.1111/dsji.12253

Chen, Y., & Hill, T. (2020). Teaching Business Students Chatbots: First Forays and Lessons Learned. *Proceedings of the 25th Americas Conference on Information Systems*.

Chen, Y., Jensen, S., Albert, L. J., Gupta, S., & Lee, T. (2023). Artificial Intelligence (AI) Student Assistants in the Classroom: Designing Chatbots to Support Student Success. *Information Systems Frontiers*, 25(1), 161-182. https://doi.org/10.1007/s10796-022-10291-4

Fleischmann, C., Logemann, M. K., Heidewald, J., Cardon, P., Aritz, J., & Swartz, S. (2024). Fostering GenAI Literacy in Higher Education for Future Workplace Preparedness: A Mixed-Methods Study. *Proceedings of the 32nd European Conference on Information Systems*, 5.

Gottipati, S., Shim, K. J., & Shankararaman, V. (2023). AI for Connectivism Learning - Undergraduate Students' Experiences of ChatGPT in Advanced Programming Courses. *Proceedings of the 29th Americas Conference on Information Systems*, 16.

Grøder, C. H., Schmager, S., Parmiggiani, E., Vasilakopoulou, P., Pappas, I., & Papavlasopoulou, S. (2022). Educating About Responsible AI in IS: Designing a Course Based on Experiential Learning, *Proceedings of the 43rd International Conference on Information Systems*, 10.

Gupta, S., & Chen, Y. (2022). Supporting Inclusive Learning Using Chatbots? A Chatbot-Led Interview Study. *Journal of Information Systems Education*, 33(1), 98-108.

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

Katuka, G. A., Auguste, Y., Song, Y., Tian, X., Kumar, A., Celepkolu, M., Boyer, K. E., Barrett, J., Israel, M., & McKlin, T. (2023, March). A Summer Camp Experience to Engage Middle School Learners in AI Through Conversational App Development. *In Proceedings of the 54th ACM Technical Symposium on Computer Science Education* (Vol. 1, pp. 813-819). https://doi.org/10.1145/3545945.3569864

Kishore, S., Hong, Y., Nguyen, A., & Qutab, S. (2023). Should ChatGPT Be Banned at Schools? Organizing Visions for Generative Artificial Intelligence (AI) in Education. *Proceedings of the 44th International Conference on Information Systems*.

Kong, S. C., Cheung, W. M. Y., & Zhang, G. (2023). Evaluating an Artificial Intelligence Literacy Programme for Developing University Students' Conceptual Understanding, Literacy, Empowerment and Ethical Awareness. *Educational Technology & Society*, 26(1), 16-30.

Ma, Y., & Siau, K. (2019). Higher Education in the AI Age. *Proceedings of the 25th Americas Conference on Information Systems*.

McKinsey Global Institute. (2023, July 26). *Generative AI and the Future of Work in America*. https://www.mckinsey.com/mgi/our-research/generative-ai-and-the-future-of-work-in-america

OpenAI. (2024a). *Meeting Minutes*. https://platform.openai.com/docs/tutorials/meeting-minutes

OpenAI. (2024b). *Prompt Examples*. https://platform.openai.com/docs/examples

OpenAI. (2024c). *Tokenizer*. https://platform.openai.com/tokenizer

Siau, K., Lacity, M., & Sauter, V. (2018). Panel: Artificial Intelligence and The Future of Work. *Thirteenth Midwest Association for Information Systems Conference* (pp. 5-8).

Van Slyke, C., Johnson, R. D., & Sarabadani, J. (2023). Generative Artificial Intelligence in Information Systems Education: Challenges, Consequences, and Responses. *Communications of the Association for Information Systems*, 53, 1-21. https://doi.org/10.17705/1CAIS.05301

Vandenberg, J., & Mott, B. (2023, June). "AI Teaches Itself": Exploring Young Learners' Perspectives on Artificial Intelligence for Instrument Development. *In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education* (Vol. 1, pp. 485-490). https://doi.org/10.1145/3587102.3588778

Sundberg, L., & Holmström, J. (2024). Teaching Tip: Using No-Code AI to Teach Machine Learning in Higher Education. *Journal of Information Systems Education*, 35(1), 56-66. https://doi.org/10.62273/CYPL2902

Touretzky, D., Gardner-McCune, C., & Seehorn, D. (2022). Machine Learning and the Five Big Ideas in AI. *International Journal of Artificial Intelligence in Education*, 33, 233-266. https://doi.org/10.1007/s40593-022-00314-1

Wang, S., & Wang, H. (2021). Teaching Tip: A Teaching Module of No-Code Business App Development. *Journal of Information Systems Education*, 32(1), 1-8.

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

AUTHOR BIOGRAPHIES

Yee Kit Chan is a senior at Carnegie Mellon University majoring in Computer Science and Human-Computer Interaction, with a minor in Chinese Studies. She is broadly interested in novel software development, human-AI interaction, and social computing, with an emphasis on innovating solutions which improve the quality of life of diverse populations. Outside of her schoolwork and the AI for Social Good initiative, she is also involved in research with the Computational Physicality Lab at National Taiwan University, as well as in the Human-Computer Interaction Institute at Carnegie Mellon University.

Erica Lai is a sophomore majoring in Computer Science and Engineering at the University of California, Irvine. She is passionate about cybersecurity, artificial intelligence, and computer hardware architecture. Her academic interests span machine learning applications, digital systems design, and the integration of AI into education.

Yu Chen is an Associate Professor of the School of Information Systems and Technology in the Lucas College and Graduate School of Business at San José State University. Her research interests include Artificial Intelligence (AI) for Social Good and AI for education. She has extensive experience in Computer Science and AI curriculum development and qualitative and quantitative research methods. She was a postdoctoral researcher at University of California, Irvine. Dr. Chen received her Ph.D. in Communication and Computer Sciences from Swiss Federal Institute of Technology at Lausanne (EPFL) in Switzerland and Master's degrees in Security and Mobile Computing from Aalto University in Finland and Norwegian University of Science and Technology in Norway.

APPENDIX

GitHub Pages of the Lab Tutorials

Lab 1: Setting Up for OpenAI Development
https://github.com/TEAMAI4SG/Lab1-openai-setup

Lab 2: Generating Text with OpenAI's Models
https://github.com/TEAMAI4SG/Lab2-Generating-Text

Lab 3: Generating Images with OpenAI's Models
https://github.com/TEAMAI4SG/Lab3-Generating-Images

Lab 4: Insights from Audio Transcription
https://github.com/TEAMAI4SG/Lab4-Audio-Transcription

Lab 5: Prompt Engineering for Social Good
https://github.com/TEAMAI4SG/Lab5-Prompt-Engineering

Lab 6: Visual Studio Code and Streamlit
https://github.com/TEAMAI4SG/Lab6-VisualStudio-Streamlit

Information Systems & Computing Academic Professionals

iscap

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the ***Journal of Information Systems Education*** have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN: 2574-3872 (Online) 1055-3096 (Print)