Unlocking Learning Potential: Generative AI Chatbots as Study Partners in Online BS in Computer Science Degree Program

Yanzhen Qu
Colorado Techincal University
Colorado Springs , USA
yqu@coloradotech.edu

Daniel Letort
Colorado Techincal University
Colorado Springs , USA
dletort@coloradotech.edu

Howard Evans
Colorado Techincal University
Colorado Springs , USA
hevans@coloradotech.edu

Noura Abbas
Colorado Techincal University
Colorado Springs , USA
nabbas@coloradotech.edu

Richard Cai
Colorado Techincal University
Colorado Springs , USA
rcai@coloradotech.edu

Mazen Haj-Hussein
Colorado Techincal University
Colorado Springs , USA
mhaj-hussein@coloradotech.edu

Anastasia Biggs
Colorado Techincal University
Colorado Springs , USA
abiggs@coloradotech.edu

Janet Durgin
Colorado Technical University
Colorado Springs, USA
jdurgin@coloradotech.edu

***Abstract*— This research-to-practice full paper presents our innovative approach to integrating Generative Artificial Intelligence (GAI) chatbots into selected BSCS courses. These courses operate within an Online Learning Environment (OLE) based on asynchronous communication. We have detailed two distinct approaches to implement Retrieval-Augmented Generation (RAG) model-based GAI chatbots. The first approach utilizes Microsoft Copilot Studio, allowing us to create customized chatbots without the need for coding. The second approach involves Python coding to develop more advanced GAI chatbots. Both methods are straightforward and cost-effective, leveraging the latest advancements to enhance communication between students and faculty. By incorporating frequently asked questions and answers from various BSCS courses into these chatbots, we can enrich the learning experience and aid both students and faculty in achieving their goals. Our preliminary results suggest that with careful planning, design, and implementation, these chatbots can circumvent common issues such as misinformation, inaccuracy, ethical and safety concerns, and lack of contextual understanding, thereby enhancing the effectiveness of student learning in an asynchronous communication-based OLE.**

Keywords—GAI chatbot, RAG model, Online Learning Environment, Study Partner

I. INTRODUCTION

In the swiftly changing domain of online education, establishing effective communication between students and faculty is a significant hurdle. The asynchronous nature of interactions often leads to frustration, particularly when immediate assistance is not available [1,2,3]. Recognizing this challenge, we embarked on an initiative in 2023: the deployment of a Generative Artificial Intelligence (GAI) enabled chatbot as a study partner in a variety of online programming and software engineering courses.

Leveraging Artificial Intelligence (AI) enabled chatbots to support students in online education is a well-established concept. From a research perspective, AI chatbots have a rich history that dates to the 1960s [4,5]. Over the course of several decades, these chatbots have undergone significant evolution, paralleling advancements in computing technology. Designed to simulate human conversation through text or voice interaction, these chatbots deliver information in a conversational style. As of November 2022, the most recent advancement in AI chatbots is the incorporation of Generative AI. Henceforth, we will refer to these as GAI chatbots [6,7].

Most notable benefits of GAI chatbots used in online education can be classified into four categories: (a) GAI chatbots offer students real-time assistance in their online studies by answering questions, providing explanations, and offering additional resources. (b) GAI chatbots to tailor their responses to individual students, enhancing the learning process. They adapt to different learning paces and styles, providing a personalized learning experience. (c) GAI chatbots can help students to develop various competencies such as critical thinking, problem-solving, and communication skills. Finally, (d) GAI chatbots to act as virtual teaching assistants, handling routine queries, grading assignments, and providing real-time feedback. This reduces the workload for faculty [8,9].

However, so far, the GAI chatbots are still not used broadly in online education because of several challenges and risks associated with GAI chatbots: (a) Reliability and Accuracy: Ensuring that AI chatbots provide accurate and reliable information is crucial. Errors or misinformation can impact

student learning. (b) Ethical Considerations: As chatbots interact with students and collect data during conversations, clear guidelines and safeguards are necessary to address privacy and ethical concerns. (c) Academic Integrity and Plagiarism: The ease of accessing information through chatbots raises concerns about academic integrity. Students might rely on chatbots for answers without fully understanding the material, leading to potential plagiarism. (d) Data Security and Privacy: Storing student data within chatbot systems poses risks related to data breaches and privacy violations. And (e) Intellectual Property Rights: Chatbots might inadvertently provide copyrighted material or violate intellectual property rights when generating content [7,8,9,10].

Many of these potential risks are rooted from the underlying technology of GAI chatbots. Although many research studies have been conducted aiming to mitigate or manage these risks, we found that, through our own applications of GAI chatbots in several online programming and software engineering courses, within a well-designed online learning environment most of these risks can easily managed based on a set of "best practice rules or guidance for using GAI chatbots" within such online learning environment. In this paper, we will share our best practices, as well as the impact of these best practices on the class performance and students learning experience.

II. RELATED WORKS

A. Online Learning Environemnet

The Online Learning Environment (OLE) is a web-based application designed to deliver courses through defined student learning objectives. The OLE's backend is powered by a Learning Management System (LMS), while the frontend, known as the Virtual Campus (VC), provides an interactive interface and learning tools for students and faculty to conduct online classes. According to [11], cognitive, social, and teaching presence significantly contribute to learners achieving high-level outcomes in OLEs. Further studies emphasized the importance of treating online learners with the same value as face-to-face learners, particularly in terms of communication frequency. Instructor responsiveness was also identified as a crucial factor for success [12]. Interactions between learners and content, as well as between learners and instructors, were highlighted as primary factors in [13]. In recent years, OLEs have proven effective in teaching programming to students. Roshni & Choon [14] noted that peer reviews in an online programming environment enhance student learning. The perceived value of online learning has surged as the education system grappled with the constraints imposed by the COVID-19 pandemic. These challenges encompassed social, emotional, and cognitive hurdles, necessitating the instructor's pivotal role in facilitating interactions between students and the learning system. Tools like Zoom, which enable effective and frequent communication, have been instrumental in motivating students and achieving desired learning outcomes [15]. Lastly, [16] found that written feedback, when provided suggestively, also proved effective in an OLE. This highlights the importance of constructive feedback in online learning environments..

B. Impact of GAI Chatbots on Students Learning in Higher Education

In the last two years, GAI chatbots have gained popularity in higher education, revolutionizing the way students learn and interact. These chatbots offer a myriad of advantages, one of which is enhancing student performance. By providing personalized assistance, instant feedback, and adaptive learning paths, chatbots empower students to grasp complex concepts more effectively. For instance, a language chatbot can engage students in interactive dialogues, improving their conversational skills. Moreover, the 24/7 availability of chatbots ensures that students can seek help at any time, whether they're preparing for an exam or working on assignments. However, responsible implementation is crucial. Institutions must establish clear policies to address potential challenges, such as overreliance on chatbots or privacy concerns. When implemented thoughtfully, these AI-driven tools can maximize the positive impact on learning in online environments, fostering a more efficient and engaging educational experience for students [17,18,19,20, 21,22,23].

C. The RAG Model and its Main Functionality

The Retrieval Augmented Generation (RAG) model is an innovative technique that combines the power of pre-trained large language models (LLMs) with external data sources. The key functionalities of RAG model can be classified into the following aspects: (a) The RAG model can integrate LLMs with specialized data search mechanisms. This combination allows for nuanced responses that bridge the gap between generic LLM answers and specific information needs. (b) The RAG model can combine retrieval models (which extract data from vast knowledge repositories) with generative models (which formulate pertinent responses). By doing so, The RAG model produces informative and accurate responses rooted in information. (c) The RAG model can enhance traditional language model responses by incorporating real-time data retrieval from external sources. User input is leveraged to fetch relevant information, enriching the context and content of the language model's response. (d) The RAG can explore different paradigms, including Naive RAG, Advanced RAG, and Modular RAG. It meticulously scrutinizes the tripartite foundation of the RAG model's frameworks: retrieval, generation, and augmentation techniques. (e) The RAG model can also optimize LLM outputs by referencing authoritative knowledge bases outside their training data. Context-aware responses are ensured by incorporating real-time external data retrieval [24,25,26,27,28].

III. PROPLEM STATEMENT AND RESEARCH QUESTIONS

A. Problem Statement

One of the biggest challenges for students studying in an OLE based on asynchronous communication is to receive feedback from either faculty or classmates in a timely manner. Students mostly depend on listening to archived lecture recordings to learn the nuances of course material, and the delayed faculty or classmates' comments.

B. Hypothesis Statement

If an effective pedagogical framework can be defined to ensure that not only GAI chatbots can offer the functionality to enrich learning and facilitating critical thinking, but also avoid known issues such as misinformation, inaccuracy, data security, etc., to provide both students and faculty timely and personalized assistance via both natural language interface and course learning objectives oriented LLM (Large Language Model), then students learning experience, faculty teaching experience and class performance under an OLE will be improved greatly.

C. Research Questions

Based on our problem and hypothesis statements we have come up with the following research questions:

Research Question 1: What is an effective pedagogical framework to ensure the successful integration of GAI chatbots into OLE?

Research Question 2: What model can be applied as a foundation for building a unified solution to resolve multiple known issues of LLMs?

Research Question 3: How has the usage of GAI chatbots as a learning partner changed the students learning experience and faculty teaching experience in their BSCS courses in an OLE?

Research Question 4: How has the usage of GAI chatbots as a learning partner affected the performance of various BSCS courses in an OLE?

IV. Research Design

In this section, we will first introduce our pedagogical framework, which illustrates the metrics guiding all stakeholders to collaborate effectively in ensuring the success of adopting GAI chatbots project. Subsequently, we will present a critical unified solution that effectively addresses the top three known issues of LLMs (Large Language Models), issues that directly impact student learning. Specifically, we'll delve into the Retrieval-Augmented Generation (RAG) model and discuss how it effectively resolves these three challenges. Lastly, we'll outline our chosen platform for building RAG model-based GAI chatbots, and two approaches to implement the platform.

A. Pedagogical Framework

To ensure the project of integration of GAI chatbots into the OLE to be a successful one, we have made the effort to design a pedagogical framework which consists of a set of metrics, each corresponding to one specific type of stakeholder, who are either the end users, or service provider, or computer system administrator, or IT supporting team, or academic administrator. The details of these metrics are shown in TABEL I. The main takeaways from reading TABLE I are (a) using GAI chatbots is a system engineering project which requires cooperation from all stakeholders, (b) it is critical to pre-define the goals, policies and guidelines on how the GAI chatbots should be used ethically, (c) must have effective solutions to resolve known issues that have direct impact to students learning ahead of time; and (d) provide enough trainings to all stack holders via various communication channels, including live demonstrations, workshops, competitions, tutorials, emails, and videos, etc.

B. Resolve Known Issues of LLMs

All GAI chatbots are using LLMs which have some issues or risks triggered by the underlying technology. Among all the known issues, the top three that directly impact whether a GAI chatbot can be used for education are "misinformation and inaccuracy of information", "ethical and safety concerns", and "lack of contextual understanding". For each of these issues, many solutions or principles have been proposed and applied in recent years. For example, (a) fact-checking: apply a fact-checking mechanism to verify the accuracy of the generated content; (b) user feedback: allow users to report inaccuracies and use this feedback to improve the model; and (c) regular updates: update the model's knowledge base regularly to keep it current.

To effectively resolve all three issues mentioned above, we have applied the RAG model as the technical foundation to provide an unified solution to resolve all three issues. A RAG model is a language model that merges pre-trained models and information retrieval. It retrieves pertinent documents from a large text corpus based on input, then generates a response synthesizing this information. This process enables RAG models to deliver precise, detailed, and contextually apt responses. They excel in tasks needing long conversation context understanding and providing current, accurate information. How the RAG model can used to resolve the three issues have been presented in TABEL II

TABLE I Stakeholder Metrics

Stakeholder	Metrics
Chatbot Function Designer	Define Acceptable Output Rules for chatbots. Define Acceptable Prompt Template. Define Risk Alerts or Reminder Messages. Define training courses or guideline.
Chatbot User: Students	Understand pros and cons of LLM. Define the purpose. Use the recommended prompts. Focus on the learning process via leading questions. Practicing critical thinking to any outputs from the chatbots. Leverage chatbots functionality to enrich learning (e.g. quiz, more examples, hands-on) Avoid submitting AI-generated content without proper attribution.
Chatbot User: Faculty	Teach students how to use AI resources responsibly. Emphasize original thought over copied content. Define the purpose of using ChatGPT-4 in the classroom. Encourage peer feedback facilitated by ChatGPT-4. Foster critical thinking by asking open-ended questions.

TABLE II LLM ISSUES CAN BE SOLVED VIA THE RAG MODEL

LLM Issue	How can the RAG model help
Misinformation and Inaccuracy	The RAG model can help address this issue by retrieving and synthesizing information from a large corpus of documents in real-time to provide more accurate and up-to-date responses.
Ethical and Safety Concerns	The RAG model can be combined with safety mitigations and content filters to reduce the risk of generating harmful content, and the model can be trained to recognize and avoid generating responses that could be potentially harmful or offensive.
Lack of Contextual Understanding	The RAG model can help address this by using a document retrieval mechanism that considers the entire conversation context, so that the model can maintain a better understanding of the conversation and provide more relevant responses.

C. Implementation Method to Customize GAI Chatbots by Applying the RAG Model

There are two types of implementation methods to customize an existing GAI chatbot that uses a public LLM by applying the RAG model. We introduce both of these methods in this section.

A. Low-code / No-code Implementation Method

As an example of the low-code/no-code approach, in this section, we provide the steps by using Microsoft Copilot Studio as the tool to customize a public LLM via the RAG model to create GAI chatbot based on specific course materials.
Figure1 shows the architecture of Copilot Studio.

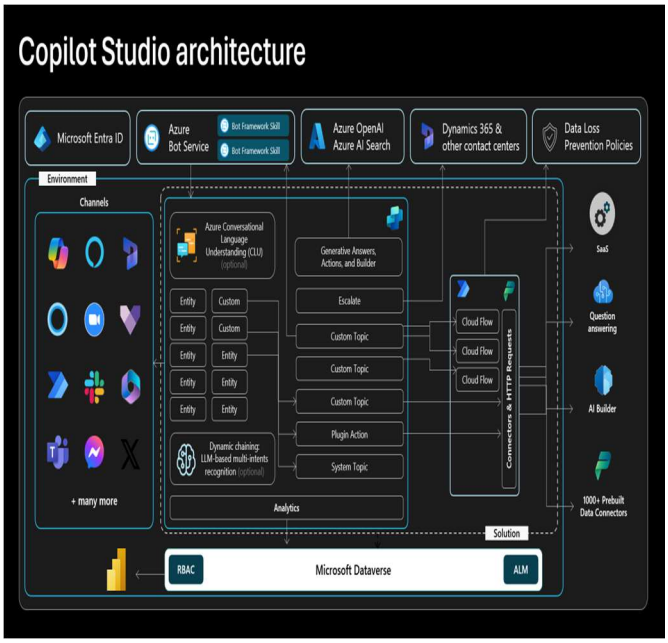

Figure 1. Copilot Studio Architecture

1. Environment Setup: Ensure access to Microsoft Copilot Studio within Microsoft Account/Tenet.
2. Data Collection and Preprocessing:
   - Gather relevant datasets for training the AI assistant, including prompts, responses, and any additional knowledge sources.
     - Private Data to be used (Structured or Unstructured Data)
     - Ground Truth Testing file with expected questions, authoritative answer, and link to private dataset/documentation.
     - Synonyms and other language-specific terms for the assistant.
   - Preprocess the data to clean, format, and structure it as needed for training and retrieval.
     - This may include using The Power Platform tools, depending on preprocessing needs.
3. Model Selection and Fine-Tuning:
   - Choose a pre-trained language model suitable for the task, such as ChatGPT, Gemini, LLAMA, Claude, etc. GPT-4 is commonly used with Copilot Studio.
   - Fine-tune the selected model on the collected data within Microsoft Copilot Studio to adapt it to the specific task or domain.
4. Implement Retrieval Augmented Generation (RAG):
   - Utilize Copilot Studio's capabilities for combining retrieval and generation approaches.
   - Implement logic for retrieving relevant information from external knowledge sources based on user queries.
   - Combine the retrieved knowledge with the model's generation capabilities to produce informative responses.
5. Integration with Microsoft Copilot Studio:
   - Integrate the fine-tuned model and retrieval mechanisms into Microsoft Copilot Studio.
   - Utilize Copilot Studio's interface and tools for managing prompts, responses, and interactions with the AI assistant.
6. Testing and Evaluation:
   - Test the AI assistant within Microsoft Copilot Studio to ensure its functionality and performance.
   - Evaluate the assistant's responses for accuracy, coherence, and relevance to user queries.
   - Collect feedback from users to identify areas for improvement and refine the assistant's capabilities.
7. Deployment and Maintenance:
   - Deploy the AI assistant built in Copilot Studio to production environments or integrate it into other platforms as needed.
   - Monitor the assistant's performance and user interactions in real-world scenarios.
   - Maintain the assistant by updating prompts, responses, and underlying models as necessary to improve its effectiveness and relevance over time.

B. Code-based Implementation Method

As an example of the cod-based implementation method, in this section, we provide steps by using Langchain, which is a language model integration framework, and Python code to interact with the "private" data and combine it with the public LLM model, to create a customized GAI chatbot. Figure 2 shows the architecture of the implemented via the code-based approach.

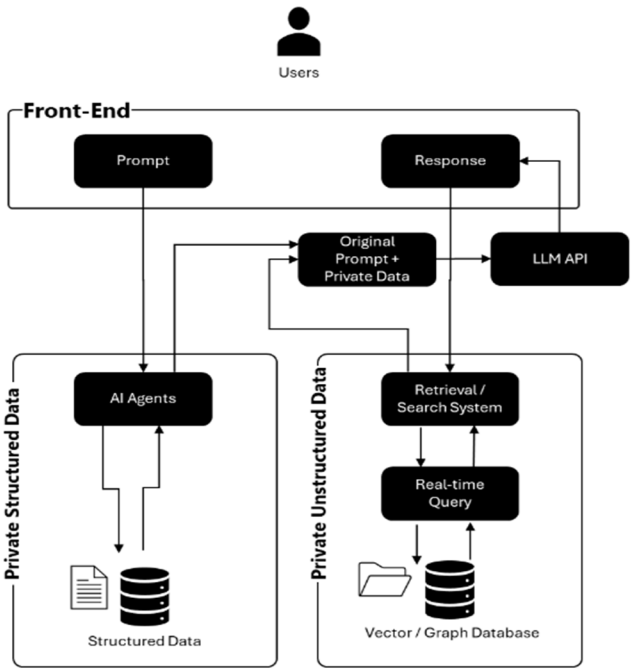

Figure 2. Architecture of the RAG Model-based Platform Implemented by the Code-based Approach.

1. Environment Setup: Will use JavaScript / Web Front-end for prompt/response management.
   - Install Node.js and npm (Node Package Manager) to manage dependencies.
   - Set up Angular CLI (Command Line Interface) globally to scaffold Angular projects.
2. Angular Project Creation:
   - Create a new Angular project using Angular CLI.
   - Create scaffold components, services, and modules required for managing prompts and responses.
3. Data Collection and Preprocessing:
   - Gather relevant datasets for training the AI assistant, including prompts, responses, and any additional knowledge sources.
     - Private Data to be used (Structured or Unstructured Data)
     - Ground Truth Testing file with expected questions, authoritative answer, and link to private dataset/documentation.
     - Synonyms and other language-specific terms for the particular assistant.
   - Preprocess the data to clean, format, and structure it as needed for training and retrieval.
4. Model Integration and Fine-Tuning:
   - Choose a pre-trained LLM suitable for the task, such as GPT, BERT, LLAMA, Claude, etc.
   - Integrate the selected LLM model into the Angular project using libraries like TensorFlow.js or Hugging Face Transformers.
   - Fine-tune the LLM on the collected data within the Angular environment to adapt it to the specific task or domain using evaluation metrics.
5. Implement Retrieval Augmented Generation (RAG):
   - Develop components and services for retrieval and RAG model-based Platform generation mechanisms within the Angular project.
   - Implement logic for retrieving relevant information from external knowledge sources based on user queries.
   - Combine the retrieved knowledge with the LLM's generation capabilities to produce informative responses.
6. Angular-Langchain Integration:
   - Explore and utilize Angular libraries or APIs for interacting with Langchain, if available.
   - Integrate Langchain functionalities such as data management, model orchestration, and computation coordination into the Angular project.
   - Implement communication protocols to interact with Langchain nodes for decentralized AI workflows.
7. User Interface Development:
   - Design and develop user interfaces using Angular components, templates, and stylesheets.
   - Create user-friendly interfaces for managing prompts, viewing responses, and interacting with the AI assistant.
   - Implement features such as input forms, buttons, and text areas for user interactions.
8. Testing and Evaluation:
   - Conduct thorough testing of the Angular application to ensure functionality and responsiveness across different browsers and devices.
   - Evaluate the AI assistant's performance by testing various prompts and analyzing generated responses.
   - Collect feedback from users to identify areas for improvement and iterate on the design and functionality.
9. Deployment and Maintenance:
   - Deploy the Angular-based AI assistant to web servers or cloud platforms for public access.
   - Monitor the application's performance and user interactions, addressing any issues or bugs that arise.
   - Maintain the AI assistant by updating prompts, responses, and underlying models as needed to keep pace with evolving requirements and user feedback.

V. Use Cases and Results Anlysis

In this section, we illustrate several use cases of GAI chatbots in various BSCS classes operating on an OLE based on asynchronous communication. Each use case demonstrates how these chatbots, acting as study partners, can enrich either the learning experience students or teaching experience for faculty in an asynchronous communication-based OLE. They contribute to enhancing the effectiveness of both student learning and faculty teaching, thereby positively impacting the overall performance of these classes.

A. Use Case 1: Applying Chatbot in Discussion Board

During Java Programming classes, the integration of a GAI chatbot (in this case, ChatGPT-4) into discussion boards was used to enhance student engagement. Despite initial apprehension from students about using GAI chatbots for academic purposes, those with prior experience found it beneficial. Faculty incorporated AI into the final reflective discussion board to alleviate student concerns and to promote GAI chatbot's use. This course module required students to use GAI chatbots to articulate and evaluate the benefits of the zyBooks learning tool (a courseware with the embedded SDK for a few popular programming languages), fostering their comfort with GAI chatbots in academic settings.

The benefits of this GAI chatbot integration have been noteworthy, offering a fresh approach to classroom engagement and reducing the stigma associated with GAI chatbot's use. This has not only enhanced inclusivity and accessibility in online learning environments but also empowered students to express their ideas more effectively. Many students have voiced their appreciation for GAI chatbot, particularly for its role in reducing apprehension associated with GAI chatbot use in academic contexts. Ultimately, the integration of AI in student discussion boards is cultivating a culture of inclusivity, accessibility, and academic excellence across higher education institutions.

B. Use Case 2: Applying Chatbot in Individual Project

In the Fundamentals of Software Requirements course, a unique application of GAI chatbot was explored through an individual project where students were tasked with writing a research paper. The assignment required students to employ prompt engineering to formulate a detailed written description of a new system under development. This description included a comprehensive account of the system's functional capabilities and roles, enabling the generation of ten specific system requirements.

Following the generation of these requirements, students were instructed to construct a Requirements Traceability Matrix (RTM), which involved mapping and enhancing the generated requirements based on their initial prompts. This activity utilized prominent GAI chatbot platforms, including OpenAI ChatGPT-4, Google Gemini, and Microsoft Copilot, all of which produced comparable outcomes.

Feedback from the students revealed that the effectiveness of the GAI-generated responses was significantly influenced by the quality of their prompts. The ability of these tools to rapidly synthesize and expand upon other course materials was particularly valued, as it enriched the context and depth of their project work. Moreover, students recognized these tools as potent assets within their academic toolkit, enhancing their research efficiency, validating their knowledge, and highlighting further research opportunities. This positive reception underscores the potential of GAI chatbots to significantly benefit educational outcomes throughout the Computer Science program.

C. Use Case 3: Applying Chatbot in Trouble Shooting

In a data structures and algorithms course, students engage in practical coding exercises aimed at implementing various data structures and algorithms in Python. Throughout the course, students encounter a myriad of challenges in software coding and debugging. These challenges often include understanding complex data structures such as linked lists, trees, and graphs, as well as implementing algorithms for tasks like sorting, searching, and linked lists traversal. Additionally, students face difficulties in debugging their code, which may involve identifying logic errors, resolving syntax issues, and optimizing the efficiency of their implementations. The complexity of the subject matter combined with the hands-on nature of the course demands a robust support system to assist students in overcoming these challenges effectively.

To address the challenges encountered by students in software coding and debugging, both faculty and students leverage GAI chatbots as a valuable tool for troubleshooting Python code during labs and assignments. These chatbots utilize advanced natural language processing (NLP) algorithms to interpret students' coding queries and provide real-time assistance and feedback. Students can interact with the chatbots to articulate their coding problems, receive immediate suggestions for debugging, and access relevant resources for further assistance. For instance, students may submit their code to the chatbot and inquire, "What is wrong with my code?" They can also provide an error message they are encountering and request the chatbot to explain it, along with providing examples of how to rectify it. Another scenario where chatbots prove beneficial is in addressing logical errors, where the code executes but doesn't yield the expected result. Faculty members, in turn, can utilize the chatbots to monitor student progress, identify common coding pitfalls, and provide targeted guidance and support tailored to individual student needs. By integrating GAI chatbots into the learning environment, both faculty and students can collaborate more effectively, leading to improved learning outcomes and enhanced proficiency in software coding and debugging.

D. Use Case 4: Applying Chatbot in Live Chat

In Java Programming classes, students have repeated the course with challenges involving comprehension of the Java programming concepts, and application of the conceptual knowledge toward completing the labs. Students have trouble applying the concepts to their labs. Many students repeat the class and experience additional challenges in other programming classes utilizing the classroom tools such as

zyBooks. The faculty attempt to demonstrate Java programming in campus class using zyBooks labs. Many students continue to experience challenges as they are still acquiring basic programming concepts.

Several repeat students in the Java programming class began using Google Bard or Microsoft Copilot to understand the concepts to facilitate the completion of their Java labs. They were instructed in lectures regarding the use of Microsoft Copilot in searching for concepts and completing programming labs. After a few attempts, they became proficient in the application of what they learned from Microsoft Copilot to successful programming labs. Microsoft Copilot was demonstrated in our lecture series to the students. By the end of the five-week session, they had attained the necessary knowledge to complete the course.

E. Use Case 5: Applying Chatbot in Providing Advice

One of the BSCS courses is Advanced C++ Programming. In this course, students provided copies of their programs that had syntax and/or logical errors. Combining our own feedback with GAI chatbots responses, we were able to provide constructive information and feedback that aided in advising students as to how they could correct those higher-level issues. The students were able to follow the instructions and get their programs to work correctly.

Discussion board forums was an additional area where we employed the use of GAI chatbots to provide student feedback on their forum response posts. By providing discussion questions and student responses to multiple GAI chatbots tools, we were able to create constructive feedback that helped students remain engaged. After GAI enabled generated responses, we ran a careful review of the information to ensure it was valid and directly aligned to the topic and student assignment.

F. Use Case 6: Applying Chatbot in Preparing Lecture

GAI chatbots were also used in the Fundamental of Networking course which teaches students concepts such as network topologies, hardware and media, IP addressing, and subnetting. However, students with no prior networking experience may encounter challenges in comprehending these concepts, particularly subnetting. To address this, the faculty has incorporated the GAI chatbots into the lecture preparation. These chatbots have proven instrumental in elucidating complex concepts.

For example, they have suggested visual aids and analogies, such as likening subnetting to highway lanes and apartments in a building. These analogies have been effective in enhancing students' understanding of the subnetting concept and process. Furthermore, these chatbots have the capability to identify areas of weakness based on the questions posed and the students' responses. They then utilize this information to provide recommendations on topics that should be emphasized in lectures to bolster student learning. As a result of integrating chatbots into the teaching process, the faculty member has reported a significant increase in the participation rate for the optional lectures.

VI. Conclusion and Future Works

In this paper, we have presented our approach to integrating GAI chatbots into several selected BSCS classes. These classes operate on an OLE based asynchronous communication. The chatbots serve as study partners, enhancing student learning through their unique capabilities. Our preliminary results suggest that with careful planning, design, and implementation, RAG model-based GAI chatbots can circumvent common issues such as misinformation, inaccuracy, ethical and safety concerns, and lack of contextual understanding. Furthermore, these chatbots can generate meaningful and accurate content. This capability enriches the learning experience and aids both students and faculty, thereby enhancing the effectiveness of student learning and faculty teaching in an asynchronous communication-based OLE. Through the piloting use of the GAI chatbots in quite a few BSCS classes, we are now feeling more confident in applying this new tool to a broader range of classes of BSCS and BSIT degree programs. Based on all the above, we can claim that we have provided the positive answer to all our four research questions.

We have outlined the comprehensive steps for two distinct methods to implement chatbots enabled by the RAG model-based GAI. The first method utilizes Microsoft Copilot Studio, allowing us to create customized GAI chatbots for all our use cases without the need for coding. The second method, on the other hand, involves some Python coding and is used to develop more sophisticated GAI chatbots. Both methods are straightforward to implement. This simplicity presents an advantage as it allows us to leverage the latest advancements to cost-effectively improve communication between students and faculty in an Online Learning Environment (OLE) based on asynchronous communication. This is made possible because we can seamlessly integrate frequently asked questions and answers from various BSCS courses into the customized GAI chatbots by applying the RAG model.

The future works include collecting class performance data and student artifacts from the relevant BSCE courses, running on the OLE, that have already used GAI chatbots to conduct assessment based on the predefined assessment rubrics, which reflects both the courses' learning objectives and/or program outcomes and analyzing the assessment results so that we can find the room for further improvement.

Acknowledgment

We sincerely thank the reviewers who have not only confirmed the importance of our research work, but also provided us some recommendations to further improve the quality of this paper.

REFERENCES

[1] Vlachopoulos D, Makri A. Online communication and interaction in distance higher education: A framework study of good practice. International Review of Education. 2019 Aug 15;65(4):605-32.

[2] Jensen J, Smith CM, Bowers R, Kaloi M, Ogden TH, Parry KA, Payne JS, Fife P, Holt E. Asynchronous online instruction leads to learning gaps when compared to a flipped classroom. Journal of Science Education and Technology. 2022 Dec;31(6):718-29.

[3] Salarvand S, Mousavi MS, Rahimi M. Communication and cooperation challenges in the online classroom in the COVID-19 era: a qualitative study. BMC Medical Education. 2023 Mar 30;23(1):201.

[4] Baidoo-Anu, David, and Leticia Owusu Ansah. "Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning." *Journal of AI* 7, no. 1 (2023): 52-62.

[5] Akram, Bita, Juho Leinonen, Narges Norouzi, James Prather, and Lisa Zhang. "AI in Computing Education from Research to Practice." In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*, pp. 1521-1522. 2024.

[6] Cooper, Grant. "Examining science education in ChatGPT: An exploratory study of generative artificial intelligence." *Journal of Science Education and Technology* 32, no. 3 (2023): 444-452.

[7] Bahroun, Zied, Chiraz Anane, Vian Ahmed, and Andrew Zacca. "Transforming education: A comprehensive review of generative artificial intelligence in educational settings through bibliometric and content analysis." *Sustainability* 15, no. 17 (2023): 12983.

[8] Krause, Stefanie, Bhumi Hitesh Panchal, and Nikhil Ubhe. "The Evolution of Learning: Assessing the Transformative Impact of Generative AI on Higher Education." *arXiv preprint arXiv:2404.10551* (2024).

[9] Han, Songhee, et al. "Knowledge-based chatbots: a scale measuring students' learning experiences in massive open online courses." *Educational technology research and development* 71.6 (2023): 2431-2456.

[10] Ilieva, Galina, et al. "Effects of generative chatbots in higher education." *Information* 14.9 (2023): 492

[11] Taghizade, A., Hatami, J., Noroozi, O., Farrokhnia, M., & Hassanzadeh, A. (2020). Fostering learners' perceived presence and high-level learning outcomes in online learning environments. *Education Research International, 2020*http://dx.doi.org/10.1155/2020/6026231.

[12] O' Shea, S., Stone, C., & Delahunty, J. (2015). "I 'feel' like I am at university even though I am online." Exploring how students narrate their engagement with higher education institutions in an online learning environment. Distance Education, 36(1), 41–58. https://doi-org.coloradotech.idm.oclc.org/10.1080/01587919.2015.1019970.

[13] Alqurashi, E. (2019). Predicting student satisfaction and perceived learning within online learning environments. Distance Education, 40(1), 133–148.https://doi-org.coloradotech.idm.oclc.org/10.1080/01587919.2018.1553562.

[14] Roshni, S., & Quek Choon, L. G. (2020). A case study investigating programming students' peer review of codes and their perceptions of the online learning environment. Education and Information Technologies, 25(5), 3553-3575. http://dx.doi.org/10.1007/s10639-020-10111-9.

[15] Lamanauskas, V., Makarskaite-Petkeviciene, R., Gorghiu, G., Santi, E.-A., & Pribeanu, C. (2021). Exploring the Students' Perceived Academic Value in an Exclusive Online Learning Environment. Romanian Journal for Multidimensional Education / Revista Romaneasca Pentru Educatie Multidimensionala, 13(4), 1–16. https://doi-org.coloradotech.idm.oclc.org/10.18662/rrem/13.4/467.

[16] Guasch, T., Espasa, A., & Martinez-Melo, M. (2019). The art of questioning in online learning environments: the potentialities of feedback in writing. Assessment & Evaluation in Higher Education, 44(1), 111–123. https://doi-org.coloradotech.idm.oclc.org/10.1080/02602938.2018.1479373.

[17] Lawan, Abdulmalik Ahmad, Basheer Ridwan Muhammad, Ahmad Muhammad Tahir, Kamaluddeen Ibrahim Yarima, Abubakar Zakari, Aliyu Hassan Abdullahi II, Adamu Hussaini et al. "Modified flipped learning as an approach to mitigate the adverse effects of generative artificial intelligence on education." *Education Journal* 12, no. 4 (2023): 136-43.

[18] Bozkurt, Aras, Xiao Junhong, Sarah Lambert, Angelica Pazurek, Helen Crompton, Suzan Koseoglu, Robert Farrow et al. "Speculative futures on ChatGPT and generative artificial intelligence (AI): A collective reflection from the educational landscape." *Asian Journal of Distance Education* 18, no. 1 (2023): 53-130.

[19] Chang, Chew-Hung, and Gillian Kidman. "The rise of generative artificial intelligence (AI) language models-challenges and opportunities for geographical and environmental education." *International Research in Geographical and Environmental Education* 32, no. 2 (2023): 85-89.

[20] Liu, Rongxin, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J. Malan. "Teaching CS50 with AI: leveraging generative artificial intelligence in computer science education." In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, pp. 750-756. 2024.

[21] Prather, James, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning et al. "The robots are here: Navigating the generative ai revolution in computing education." In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*, pp. 108-159. 2023.

[22] Prasad, Prajish, and Aamod Sane. "A Self-Regulated Learning Framework using Generative AI and its Application in CS Educational Intervention Design." In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, pp. 1070-1076. 2024.

[23] Glynn, Colin, Emily Hed, Abbigail Pexa, Tyler Pohlmann, Imad Rahal, and Robert Hesse. "CAET: Code Analysis and Education Tutor." In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*, pp. 1656-1657. 2024.

[24] Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." *Advances in Neural Information Processing Systems* 33 (2020): 9459-9474.

[25] Siriwardhana, Shamane, et al. "Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering." *Transactions of the Association for Computational Linguistics* 11 (2023): 1-17.

[26] Gao, Yunfan, et al. "Retrieval-augmented generation for large language models: A survey." *arXiv preprint arXiv:2312.10997* (2023)

[27] Shuster, Kurt, et al. "Retrieval augmentation reduces hallucination in conversation." *arXiv preprint arXiv:2104.07567* (2021).