Tuijin Jishu /Journal of Propulsion Technology
ISSN:1001-4055
Vol. 44 No. 04 (2023)

CodexGuru: An Interactive Programming Assistance Platform for Instructors and Novice Programmers

D.I.De Silva[1], E. Weerasinghe[2], V.A.R.Maharanhindage[3], D.M.L.D.Jayakody[4], H.K.L.Shavinda[5], K.S.Madhubhashana[6]

Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka

Abstract:-In computer programming education, novice programmers and lab instructors grapple with persistent challenges, particularly in large lab environments, including repetitive problem-solving routines, limited student-instructor engagement, and an unsatisfactory learning experience. This research paper introduces *CodexGuru*, a comprehensive web-based solution to revolutionize the traditional coding lab paradigm. *CodexGuru* incorporates an AI-assisted code editor, a real-time discussion forum, an online examination environment, and robust user management and administrative functionalities, catering to the needs of lab instructors, students, and system administrators. Through the implementation of *CodexGuru*, this study showcases a transformative shift in coding lab dynamics. By streamlining coding processes and fostering dynamic interaction, the platform significantly enhances student-instructor engagement, ignites greater student interest in lab sessions, and boosts student motivation. This research paper explores how *CodexGuru* addresses longstanding challenges, fostering pedagogical excellence and ushering in a new era of computer programming education. Within this research paper, readers will encounter an in-depth analysis of *CodexGuru*'s functionalities, its influence on student learning experiences, and its capacity to transform traditional coding lab sessions for the advantage of all relevant stakeholders.

Keywords*: programming, novice programmers, online exams, coding assistance.*

1. Introduction

In the rapidly evolving landscape of education and technology, integrating innovative solutions has become paramount to addressing the challenges faced by traditional learning environments. This research paper presents a groundbreaking research initiative that seeks to transform the process of coding education and lab sessions in Sri Lankan universities. The project, *CodexGuru*, aims to revolutionize the conventional coding lab experience by harnessing the power of cutting-edge AI technology and other interactive online tools.

Like many educational institutions globally, Sri Lankan universities encounter many challenges within their programming lab environments. These challenges include limited communication between students and instructors, particularly in large class settings, difficulties in visibility due to projector screen constraints, the persistent struggles novice programming students face while resolving seemingly trivial code errors, and the repetitive nature of solutions provided by instructors. These obstacles collectively hinder novice programmers' and lab instructors' engagement, interaction, and overall learning experience.

The significance of this research project lies in its potential to impact the landscape of coding education profoundly. emerges as a comprehensive solution addressing the challenges above and as a beacon of innovation. This platform empowers novice programmers to navigate coding complexities more efficiently and intuitively by seamlessly integrating advanced AI technology. Moreover, *CodexGuru* is a bridge, fostering

effective communication and interaction between lab instructors and students, thus enriching the educational journey. The primary objectives of *CodexGuru* are threefold:

1) Enhanced Engagement and Interaction: *CodexGuru* strives to create a dynamic and interactive learning environment where students can engage with their peers and instructors, fostering a collaborative spirit that extends beyond the physical confines of the lab.
2) Supporting Novice Programmers: *CodexGuru* aims to alleviate the challenges faced by novice programmers by providing real-time assistance via a dedicated discussion forum and guidance through an AI-assisted intelligent code editor, thereby accelerating their coding journey.
3) Redefining Coding Lab Session: By introducing an online examination environment, the platform seeks to revolutionize the traditional process of conducting coding lab sessions and paper-related lab assessments, offering instructors a more efficient and effective means of evaluating student progress.

This research journey envisions *CodexGuru* as a practical solution to immediate challenges and a catalyst for the future of programming education. This research paves the way for a new paradigm in coding instruction that transcends the boundaries of traditional classroom settings, enriching the learning experiences of both instructors and novice programmers alike.

The subsequent sections of this paper delve into the comprehensive functionalities of *CodexGuru*, outlining how each component contributes to realizing the objectives of this research. Furthermore, the following sections explore the underpinning technologies that power this platform, shedding light on the integration of AI and interactive features that make *CodexGuru* a transformative force in programming education. Section II of this research paper details the literature review. The methodology is discussed in Section III, followed by the results of the research in Section IV. Sections V, VI, and VII represent the discussion of the research, the conclusion, and future work, respectively.

2. Literature Review

This section of the research paper explores the extant corpus of scholarly work pertaining to the objectives and challenges that are the focus of this pioneering project. The literature under consideration herein highlights the efforts of previous researchers in addressing analogous challenges across four principal domains:

A. Enhanced learning Environment with online code editors

Integrating online code editors within educational contexts has gained substantial attention due to its potential to enhance programming learning experiences. Robins et al. extensively reviewed learning and teaching programming, emphasizing the importance of interactive environments and adaptive tools. Their work establishes a foundation for integrating the *CodexGuru* platform's interactive code editor feature [1]. Kelleher et al. explored the use of storytelling and interactive tools to enhance programming education, particularly among middle school students. Their findings highlight the potential of interactive platforms in capturing and maintaining student engagement, aligning with the engagement-focused goal of *CodexGuru*[2]. ArTEMiS is another solution proposed by Krusche and Seitz. Its inclusion of an online code editor with interactive exercise instructions together with version control provides valuable insights into this project. Their work serves as a foundation for the *CodexGuru* platform's intelligent code editor feature integrated with artificial intelligence [3].

B. Collaborative learning platforms for student instructor interaction

Efforts to facilitate communication and collaboration between students and instructors have led to the developing of collaborative learning platforms. Lister et al. conducted a multi-national study on novice programmers' reading and tracing skills. Their work underscores the significance of effective communication and interaction between novice programmers and instructors, aligning with the communication-enhancing features of the *CodexGuru* platform [4].

C. Intelligent assistance for code error resolution

The challenges novice programmers face in debugging and resolving code errors have spurred research into intelligent assistance tools. The described research on a personalized e-learning system using Item Response Theory (IRT) can provide valuable insights for this project. It showcases the significance of personalized approaches in online learning, similar to this project's goal of assisting novice programmers with an interactive platform. The focus on considering learner ability and adjusting course material difficulty aligns with the need for tailored support and personalized guidance in this programming assistance platform. The study's findings on achieving effective and efficient learning through personalized approaches can inform the design and implementation of the *CodexGuru* platform's features [5]. The described research framework focusing on integrating computational thinking into elementary science education provides valuable insights for this research. It emphasizes the importance of effective teacher training, pedagogical strategies, and design-based research, which can inform the development and implementation of this interactive programming assistance platform for instructors and novice programmers. While this research project is centered on coding assistance, the alignment in educational objectives and emphasis on teacher support make the framework's insights relevant and applicable to this project's work [6].

D. Digital assessment environments for efficient evaluation

Exploring digital assessment environments reflects a broader trend in modernizing assessment methods. Gikas and Grant examined the use of mobile computing devices for learning in higher education. Their insights into leveraging technology align with the examination environment feature of *CodexGuru*, which aims to streamline the assessment process [7]. The programming assistance tools discussed in the referenced article could provide insights into the types of tools and features that have effectively aided novice programmers. It can inform the development of the *CodexGuru* platform and help design functionalities aligned with successful approaches [8].

The above-reviewed literature demonstrates a precise alignment with the objectives of *CodexGuru* in addressing the challenges faced by lab instructors and novice programmers. By integrating insights from previous research, *CodexGuru*endeavors to foster engagement, enhance student-instructor interaction, provide AI-powered assistance, and revolutionize the traditional coding lab session. The contributions of these existing works provide a valuable context for the development and potential impact of this interactive programming assistance platform.

3. Methodology

This section outlines the methodology adopted in the development and evaluation of *CodexGuru*. The methodology encompasses the project's design, implementation, and assessment phases, providing a comprehensive framework for the realization of the platform's objectives.

Initially, an in-depth analysis of existing research papers/articles, aligning with the project's specific objectives, was conducted. After thoroughly examining these sources of information, certain flaws and potentially lacking aspects of existing systems were identified. Afterward, group brainstorming sessions were carried out for new ideas that could serve as improved features in the project. These ideas were then presented to a panel of assessors for their feedback and approval.

Upon their approval, the project commenced, initially starting with the design phase. Figma was used for sketching the wireframes of *CodexGuru*. The Ant Design guidelines were followed during the design, and its components were majorly used during the implementation phase. Based on the identified requirements, the architectural design of *CodexGuru* was formulated. Fig. 1 depicts this high-level architecture.

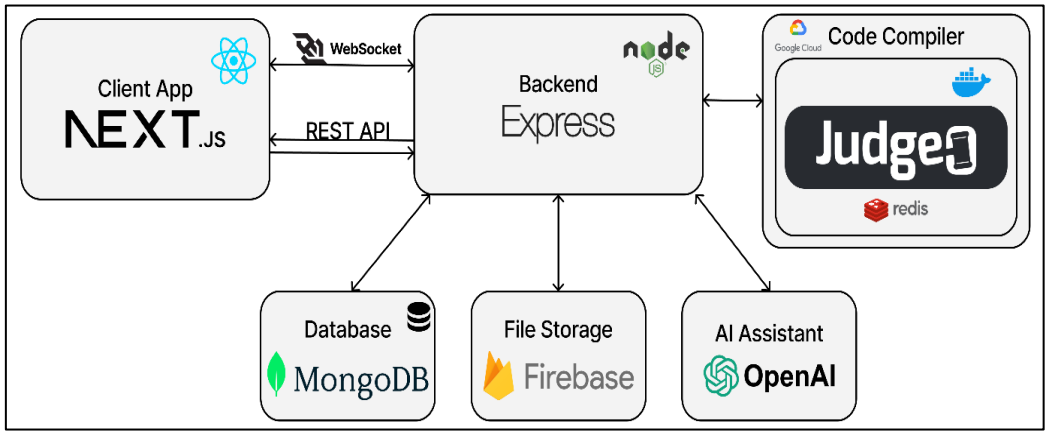


Fig. 1. CodexGuru's high-level architecture diagram

The implementation phase involved selecting the appropriate technologies to bring *CodexGuru* to life. *CodexGuru* is a web platform developed using Next.js (a React framework) for the front end, Node.js and Express.js for the back end, and MongoDB as the database. Moreover, Firebase was also used for specific file handling/storage purposes. The project used Representational State Transfer (REST) APIs for service inter-communication. The interface designs and styles were crafted with the Ant Design components and Tailwind CSS framework. Additionally, Judge0, an open-source code execution system, was hosted within a Docker environment, and OpenAI's API was integrated. Web sockets technology was also used slightly as well. This section provides a comprehensive account of the step-by-step development process for each component of *CodexGuru*:

A. Code Editor and code execution environment

This is the most significant feature of *CodexGuru*. It is an online code editor that allows novice programmers to execute their code without needing a separate Integrated Development Environment (IDE). It supports an extensive array of programming languages, totaling 46 in specificity. Furthermore, the code editor highlights syntax for all supported languages, aiding in code comprehension and error detection. Additionally, the implementation of intelligent code completion, often referred to as IntelliSense, is offered for selected languages, enhancing coding efficiency and accuracy.

The code editor leverages the Judge0 open-source code execution system hosted within a Docker environment. Once the user chooses the "Compile" option in the code editor to compile the code, the code and user inputs (keyboard inputs) are sent as a JavaScript Object Notation (JSON) object to the backend via a REST API. Afterward, the backend makes another REST API request to Judge0 to compile the code. Judge0 responds with appropriate JSON data, whether the code has been successfully compiled or not. Finally, this response is formatted and displayed to the user as the code execution output. This choice of infrastructure ensures reliable and secure execution of code snippets, contributing to a seamless coding experience.

This code editor provides light and dark mode themes to cater to individual user preferences and promote user-friendly interaction. This versatile visual customization feature allows users to tailor the interface to their comfort, reducing eye strain during extended coding sessions.

B. AI-Assisted chat

This Artificial Intelligence (AI)-driven chat support system is designed to aid novice programmers with coding-related queries and clarifications. Students can seek help and guidance on a broad spectrum of coding-related topics, making it an indispensable component of *CodexGuru*.

The AI assisted chat support is powered by integrating OpenAI's API to access the Generative Pre-Trained Transformers (GPT)-3.5 Turbo Model. The student inputs a prompt describing the particular issue, asking for AI

assistance. This prompt is once again sent as a JSON object to the backend. Afterward, the backend makes an API request to the GPT 3.5 Turbo model. The model responds with the appropriate data, which are then formatted and displayed as the final response to the student. This entire conversation is maintained as a continuing chat, where the AI maintains prompt chaining – "uses the output of one prompt as the input to the next prompt or part of your conversation."

C. Discussion forum

This component represents a real-time discussion forum, which utilizes web sockets for instant communication. "Web sockets technology enables efficient, real-time bidirectional communication between clients and server." Apart from AI-assisted chat in the code editor, this discussion forum allows students to engage with the instructor(s) and peers for additional assistance.

Per each lab session, a new discussion forum is initialized. This forum allows any student to post questions and provide answers. The lab instructors could also take part in this discussion by addressing student questions or marking a student's response as a correct answer. An integrated markdown editor in the forum with proper highlighting allows participants to share readable and formatted code segments directly. Additionally, this forum comprises a point-based system and a leaderboard. Students earn points when they receive upvotes or recognition from instructors for providing accurate answers. At the end of a lab session, the leaderboard displays the top performances of the students.

Any posted question in the forum is saved as a record in the database. The answers to a particular question are also saved in the database. For any change in the discussion forum, web sockets notify this event to all the connected users in the forum. Due to the usage of web sockets, these changes are reflected in real-time to the participants. Similarly, the upvotes and point calculations are also processed in real time. Finally, the leaderboard is visualized by extracting all relevant data from the database.

D. Online examination environment

This component represents an online examination environment featuring automated grading capabilities. Primarily designed for Multiple Choice Question (MCQ) based assessments, it also offers the optional inclusion of images in questions. Incorporating security measures, it empowers lab instructors with various administrative functionalities and students to take online exams during lab sessions. Subsequent sections will expound upon these two user functionalities in further detail:

1) Lab instructors: Lab Instructors possess the ability to create, modify, and delete exams, as well as manipulate associated questions with ease. Furthermore, they have the discretion to alter the status of an exam, toggling it between active and inactive states as per the current phase of the examination process, whether it is ongoing or concluded. Moreover, a comprehensive report generation feature per exam affords instructors with detailed insights for an exam. This report includes vital statistics, such as the number of students who undertook the examination, the percentage of pass and fail outcomes, and related performance metrics. All these operations are performed via REST APIs between the front and back end. The exams and related questions are stored as separate records in the database. The report includes data that are extracted from the database.

2) Students: Students authenticate themselves with a valid password before attempting an exam. This precaution ensures the integrity of the examination process. It ensures that no external student, who is not in the particular lab session, can attempt the exam remotely. Moreover, the platform has been implemented to prevent a student from attempting the same exam multiple times. Additionally, this component incorporates a count-down timer and an auto-submit feature, streamlining the examination experience for students while maintaining the integrity of the assessment. The automated submission guarantees students that even if the allocated time runs out, their currently attempted questions are submitted and graded without any issue. At the end of an exam attempt, all the relevant data is stored as a new record in a separate table in the database.

E. User management and administrative functionalities

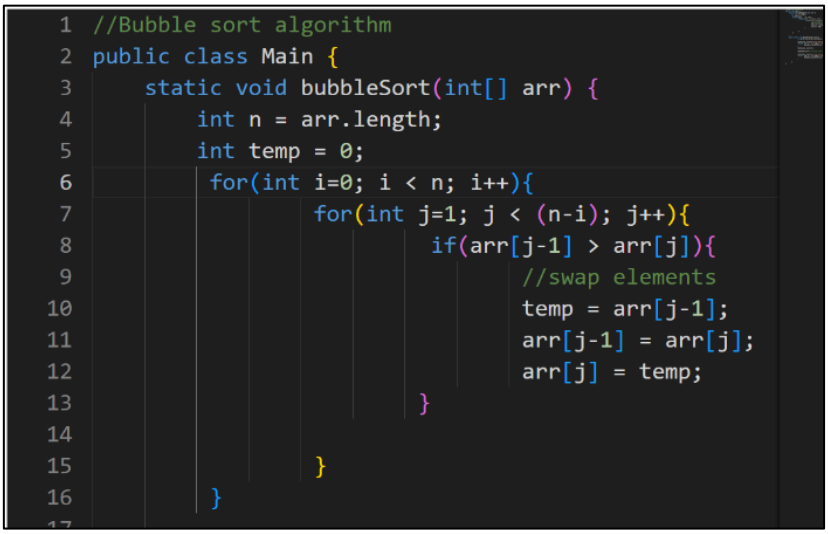
It assumes a critical role in *CodexGuru*'s administration, primarily catering to managing two key user categories: students and lab instructors. Within this context, administrators wield complete control and authority over all administrative functions associated with these user groups. It encompasses user registration, account maintenance, access provisioning, and account deactivation. Moreover, a robust report generation feature empowers administrators to compile and retrieve essential user-related data and statistics of both students and instructors.

Furthermore, administrators can grant the necessary permissions for creating and managing lab sessions for the instructors. Only lab instructors are authorized to create and manage lab sessions as necessary. It involves uploading necessary lab materials, setting up passwords, and allocating an estimated time duration. The uploading of lab materials is implemented via file handling to Firebase and then storing a reference in the database. All the user management tasks, and authorization are implemented via REST APIs and JSON Web Tokens (JWT), respectively.

4. Results

The proposed system was effectively constructed, incorporating distinctive features and functionalities tailored to cater to the specific requirements of its users.

The online code editor successfully compiles the source code and displays the relevant output to the user. It also allows users to input values, in the case where the program needs to accept keyboard input. The AI-assisted chat works as expected to deliver code improvements, identify potential flaws in the code, and recommendations to help students improve their coding skills. Fig. 2 and Fig. 3 depict this scenario.



```
1  //Bubble sort algorithm
2  public class Main {
3      static void bubbleSort(int[] arr) {
4          int n = arr.length;
5          int temp = 0;
6           for(int i=0; i < n; i++){
7                   for(int j=1; j < (n-i); j++){
8                           if(arr[j-1] > arr[j]){
9                                   //swap elements
10                                  temp = arr[j-1];
11                                  arr[j-1] = arr[j];
12                                  arr[j] = temp;
13                          }
14
15                  }
16          }
```

Fig. 2. Snapshot of the code editor

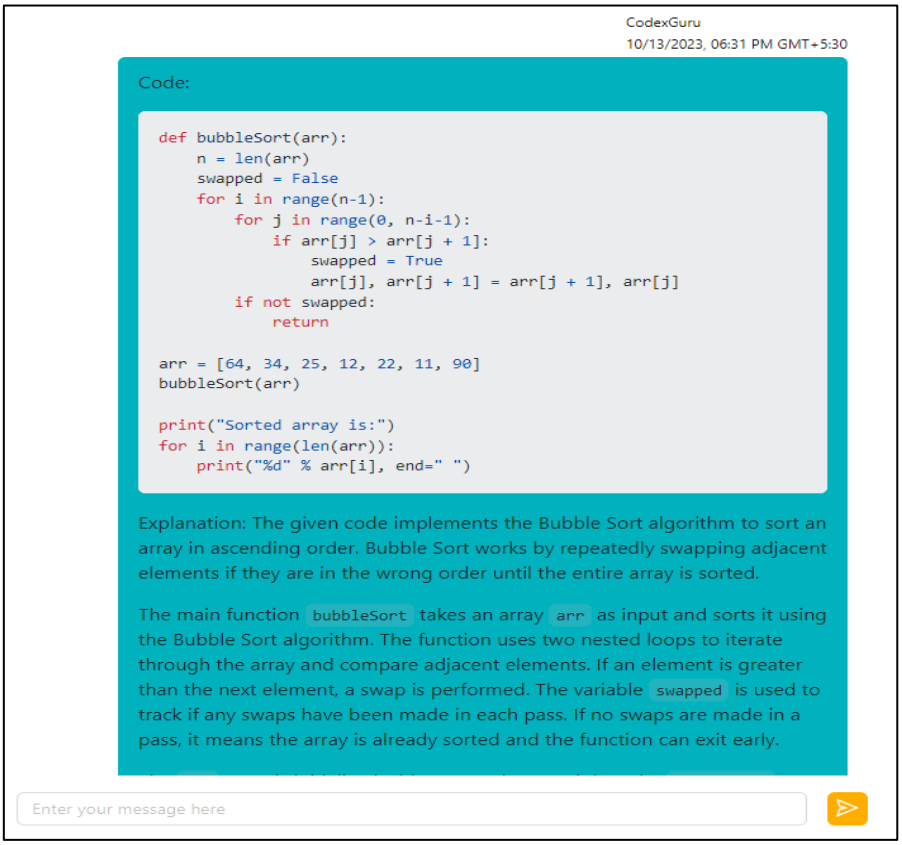


Fig. 3. Snapshot of the AI coding assistance

Fig 4. shows that the online discussion forum offers a real-time discussion experience for the students among their peers and lab instructors. The point-based system and voting techniques have improved student interaction and skill improvement, where they learn to appreciate the suggestions of their peers as well.

Students experience a seamless experience when taking online exams via *CodexGuru*. The auto-submit feature has been an important aspect, and students see feedback indicating that their responses have been successfully submitted by the platform as shown in Fig. 5.

Lab instructors can easily manage exams and questions via *CodexGuru*'s online exam environment. Fig. 6 illustrates that it offers instructors a smooth experience when managing exam questions.

The system administrators have complete control over the management of students and lab instructors. The administrators can efficiently manage the details of students and lab instructors. The report generation feature works as expected to obtain facts and figures related to the users of *CodexGuru*. Fig. 7, Fig. 8, and Fig. 9 illustrate this process.

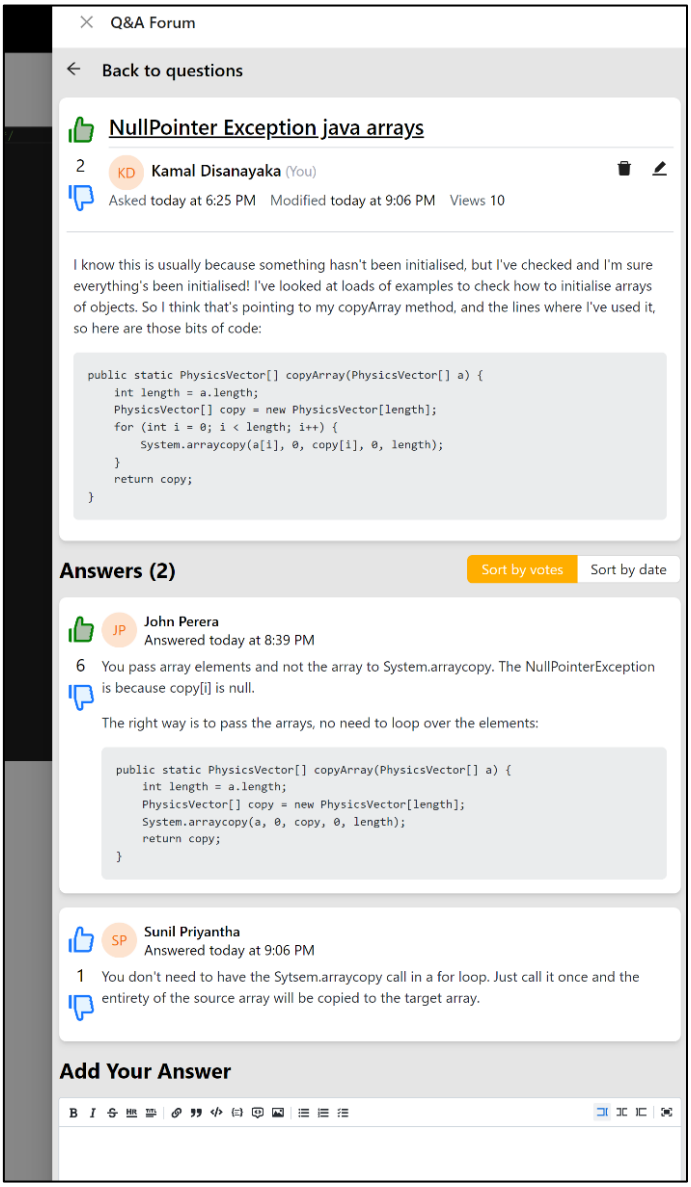


Fig. 4. Snapshot of the discussion forum

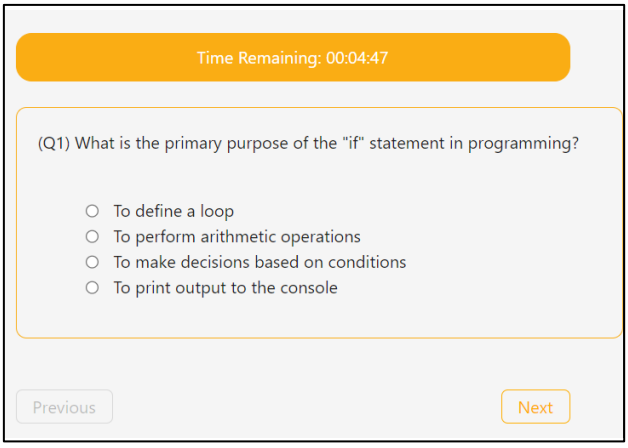


Fig. 5. Snapshot of the exam environment

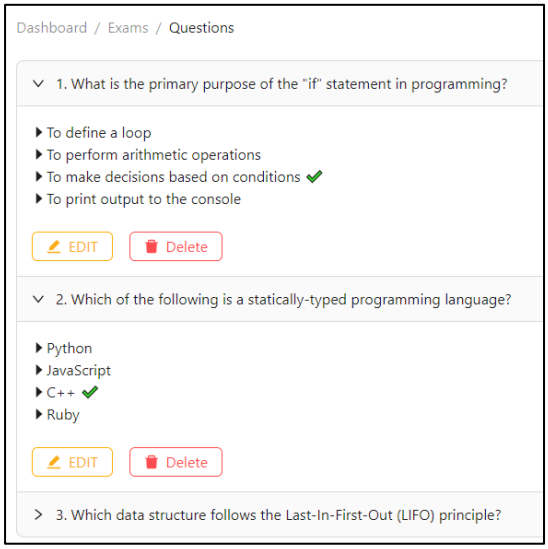


Fig. 6. Snapshot of instructor managing exam questions

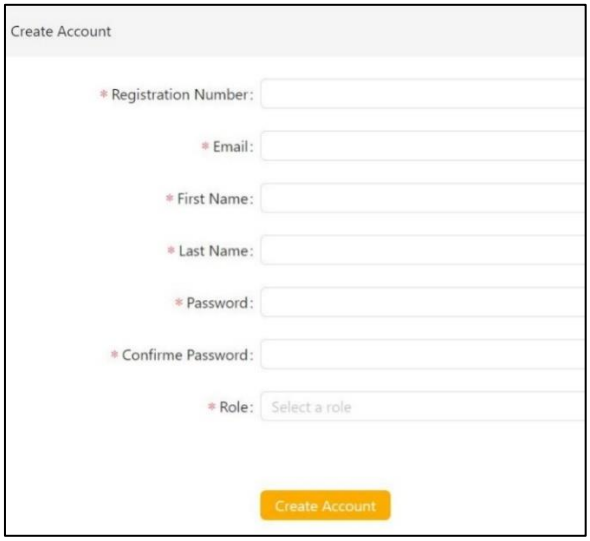


Fig. 7. Snapshot of administrator creating account

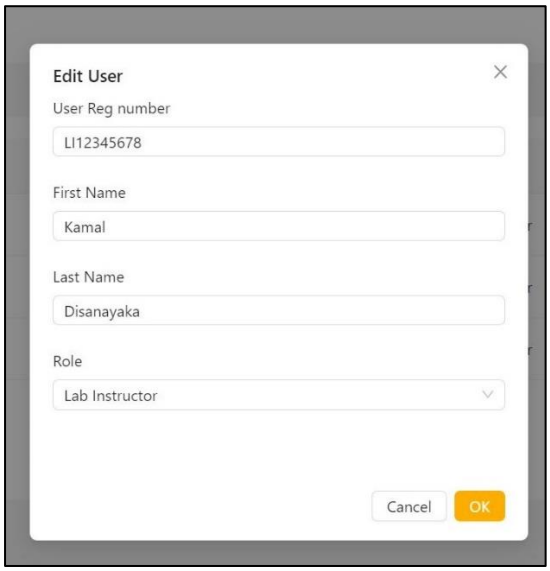


Fig. 8. Snapshot of administrator updating an account

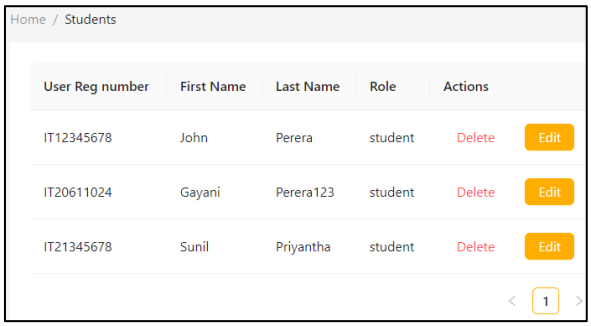


Fig. 9. Snapshot of student list

5. Discussion

This research paper discusses the *CodexGuru* platform and its features, including the AI-assisted code editor, real-time discussion forum, online exam environment, and user management and administrative functionalities. The overarching objective of this research endeavor was to catalyze a transformative shift in traditional coding lab sessions, with specific emphasis on enhancing engagement, arousing heightened student interest, and improving the overall coding experience, particularly for novice programmers and instructors.

Throughout this investigation, the *CodexGuru* platform exhibited promising results, showcasing its potential to be more effective than an existing solution, theiPaT platform [9]. Similarly, *CodexGuru*, with a code editor supporting multiple languages, surpasses the capacity of another cloud-based online coding platform, which only supports the C programming language [10]. Incorporating advanced AI assistance not only streamlined

coding processes but also promoted active learning and engagement among students. The real-time discussion forum facilitated dynamic interaction between students and instructors, promoting collaborative problem-solving and knowledge sharing. Furthermore, the online examination environment effectively automated the assessments.

Nevertheless, it is essential to acknowledge the project's limitations. *CodexGuru* currently relies on the trial version of OpenAI's API, posing potential future scalability and resource availability challenges. Additionally, the platform remains within a developmental environment and has not yet been hosted for broader accessibility. Although the solution has not undergone rigorous testing in an actual coding environment, it is noteworthy that the project received guidance from university assessors, and overall feedback was positive.

Overall, this research approach underscores the promising potential of *CodexGuru* in revolutionizing coding lab sessions and fostering a more engaging and productive learning environment for novice programmers and instructors. Despite the noted limitations, the findings suggest that *CodexGuru* represents a significant step toward achieving the stated objectives and merits further exploration.

6. Conclusion

In conclusion, this research underscores the persistent challenges encountered by novice programmers and lab instructors within the context of traditional coding lab sessions. The observed need for repetitive solution dissemination by instructors and restricted student-instructor engagement contributes to a conventional lab experience that often falls short of satisfaction. Furthermore, students' substantial time investments in rectifying minor code issues limit skill enhancement opportunities and overall progress within these educational settings. Addressing these challenges is imperative for enhancing the effectiveness and efficiency of coding lab instruction and fostering a more engaging and productive learning environment for novice programmers.

CodexGuru, the proposed solution, is a web platform offering a comprehensive remedy to the above-mentioned challenges. Its multifaceted approach integrates innovative features, including an AI-assisted code editor, a real-time discussion forum, an online examination environment, and robust user management functionalities. These features collectively cater to the diverse needs of the primary stakeholders involved in coding lab sessions, namely lab instructors, students, and system administrators.

By harnessing the capabilities of *CodexGuru*, the traditional coding lab experience is revolutionized. This transformative tool streamlines and automates routine coding tasks and facilitates dynamic real-time interaction among students and instructors, transcending the limitations of conventional teaching methods. As a result, it fosters a heightened level of student-instructor engagement, augments student interest in lab sessions, and catalyzes a substantial increase in student motivation. The implementation of *CodexGuru* thus holds the potential to significantly enhance the overall quality and effectiveness of coding labs, ushering in a new era of pedagogical excellence in computer programming education.

7. Future work

CodexGuru has embarked on a forward-looking journey, exploring areas that warrant additional exploration and offering insights into how it can serve as a foundation for future endeavors. The *CodexGuru* platform could be improved in the following ways:

- Develop and train a dedicated model specializing in coding and providing coding support. It could serve instead of the OpenAI’s API.
- Offer an in-depth examination of the typical programming errors made by each student and furnish automated suggestions and tips for enhancing the specific skills they need to develop.
- The code editor could be improved to become an IDE to aid in building small to medium-scale projects online, maintaining multiple coding files/ file structures.
- The online exams are MCQ-based, with just one correct answer per question. It could be improved to support multiple answers.

References

[1] A. Robins, J. Rountree and N. Rountree, "Learning and Teaching Programming: A Review and Discussion," *Computer Science Education,* vol. 13, no. 2, pp. 137-172, June 2003.

[2] C. Kelleher, R. Pausch and S. Kiesler, "Storytelling Alice motivates middle school girls to learn computer programming," in *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* San Jose, California, USA, 2007.

[3] S. Krusche and A. Seitz, "ArTEMiS: An Automatic Assessment Management System for Interactive Learning," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education,* Baltimore, Maryland, USA, 2018.

[4] R. Lister, E. Adams, S. Fitzgerald, W. Fone, J. Hamer, M. Lindholm, R. Mccartney, J. E. Moström, K. Sanders, O. Seppälä, B. Simon and L. Thomas, "A Multi-National Study of Reading and Tracing Skills in Novice Programmers," *ACM SIGCSE Bulletin,* vol. 36, no. 4, pp. 119-150, January 2005.

[5] C.-M. Chen, H.-M. Lee and Y.-H. Chen, "Personalized e-learning system using Item Response Theory," *Computers & Education,* vol. 44, no. 3, p. 237–255, March 2005.

[6] L. Cabrera, D. J. Ketelhut, K. Mills, H. Killen, M. Coenraad, V. Byrne and J. D. Plane, "Designing a framework for teachers' integration of computational thinking into elementary science," in *Journal of Research in Science Teaching*, 2023.

[7] J. Gikas and M. Grant, "Mobile Computing Devices in Higher Education: Student Perspectives on Learning with Cellphones, Smartphones & Social Media," *The Internet and Higher Education,* vol. 19, pp. 18-26, October 2013.

[8] M. Koorsse, C. Cilliers and A. Calitz, "Programming assistance tools to support the learning of IT programming in South African secondary schools," *Computers & Education,* vol. 82, pp. 162-178, March 2015.

[9] M. Amaratunga, G. Wickramasinghe, M. Deepal, O. Perera, D. De silva and R. Samantha, "An Interactive Programming Assistance tool (iPAT) for instructors and novice programmers," *2013 8th International Conference on Computer Science & Education,* pp. 680-684, April 2013.

[10] J. Liao, S. Chen and H. Xiong, "A cloud-based online coding platform for learning coding-related courses of computer science," *ICIC Express Letters, Part B: Applications,* vol. 8, no. 1, p. 109–115, January 2017.