

CodeCoach: An Interactive Programming Assistance Tool

¹D.I. De Silva, ²S Vidhanaarachchi, ³S.B Kariyawasam, ⁴L.R.S Dasanayake, ⁵O.D Thawalampola, ⁶T.D.D.H Jayasuriya

Department of Computer Science and Software Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka.

Abstract:-The following research paper encapsulates a modern programming assistance tool developed to solve the challenges that students and instructors experience during fundamental programming lessons. Students frequently experience numerous kinds of coding issues during time-constrained lab environments, with limited opportunities for direct communication with instructors. CodeCoach makes use of the features of GPT-3.5 and provides straightforward explanations of errors identified, provides observations on student's programming approaches, and provides practical suggestions for code development. One of the unique features of CodeCoach is its extensive assistance for numerous programming languages, which ensures its flexibility across various learning environments. Additionally, the software's specially designed forum makes it easy for students to contribute knowledge. The posting of specific lab-related questions and errors by students allows collaborative problem-solving. Instructors and students can take part in offering comprehensive responses and solutions. The implementation of CodeCoach demonstrated the effectiveness it is in optimizing the learning process. Its adaptability and integration of advanced language models make it an important achievement in the context of programming educational support systems.

Keywords: *programming assistance tool, gpt 3.5, programming error handling, online learning, programming approaches.*

1. Introduction

Computer programming is a fundamental skill in the field of Information Technology, requiring a solid grasp of programming languages, concepts, and problem-solving abilities, especially for beginners. The transition from secondary to tertiary education, combined with the introduction of complex programming concepts and languages, can leave students feeling overwhelmed [1], as pointed out by [1], particularly when delving into introductory programming courses. While educators aim to provide students with the best learning experience, they all share a common problem: how to effectively help students in a constantly changing and complex programming world. This challenge becomes even more significant when students have to deal with abstract programming ideas and the logical thinking required for programming according to [2]. These challenges persist, as shown in [3]. Their research found that most students learning programming struggle with two main issues: figuring out how to create programs for specific tasks and finding mistakes in their own code [3].

As Carlsen, Nygaard, and Stage (2006) have shown, practical experience in programming is essential for producing high-quality designs. Even high-performing students, as their study found, do not always produce the best designs without practical experience [4]. This underscores the importance of practical problem-solving skills over academic achievement in programming education. The popular method that many universities follow, which involves providing greater practice of problem-solving skills to students, is to conduct practical programming labs. However, when considering large labs with more students, communication can be very limited between students and instructors. This decreases the amount of time that an instructor can devote to each student.

While several tools and systems have emerged to address these challenges, most have primarily focused on offering feedback to assess students' abilities and performance during lab sessions or enhance their programming skills through various tutoring mechanisms. However, a critical gap exists in the realm of tools that help students correct the errors they encounter while writing programs during lab sessions. The limited available tools are often tailored to a specific programming language, hindering students' flexibility in their language of choice.

This is where this innovative tool, CodeCoach, steps in to bridge the gap. CodeCoach addresses the crucial need for real-time, error-specific programming assistance for students. Unlike existing tools, this solution empowers students to understand and rectify their coding mistakes independently, reducing the burden on lab instructors. The backbone of CodeCoach is the power of Artificial Intelligence (AI), driven by a state-of-the-art GPT(Generative Pre-trained Transformer) 3.5 model.

2. Literature Review

To assist beginners with their programming challenges, numerous colleges have developed a wide variety of tools. The "SmartLab" program was created by Alammery, Carbonne, and Sheard (2012) at Monash University, offering professors quick and comprehensive feedback on their student's performance during lab sessions [5]. However, this technology does not help the students in rectifying their mistakes.

At Middle Tennessee State University, Yoo et al. have developed a web-based tutoring system that includes a question tutor, program tutor, and course management system [6]. Lab supplies can be managed by teachers who have access to the course administration system. The question tutor and program tutor cover learning concepts and programming techniques. Even so, they do not prioritize providing solutions for students' mistakes as CodeCoach does.

Another area of research has focused on providing automated assistance to students. One example is "WebToTeach," a web-based interactive programming exercise system created by Arnow and Barshay in 2002 [7], allowing for automated assignment checking.

The Java compiler messages are refined into more comprehensive and understandable error messages using a tool called "Espresso," designed for novice Java programmers [8]. However, there is no mechanism currently in place that can suggest solutions for errors not only in Java but in any programming environment. In contrast to the "iPAT" tool [9], which only supported the C# language, the web-based tool CodeCoach, with the aid of modern AI, can support any programming language.

The systematic literature review at the University of Tabuk based on Kitchenham and Charters' guidelines [10] delves into the challenges confronting students and the burgeoning interest among scholars in unraveling these complexities. While a plethora of factors, both positive and negative, have been identified in the process of learning programming languages, the adoption of online tools emerges as a favorable solution.

The study [11], investigates the impact of AI code generators, particularly OpenAI Codex, on novice programmers in introductory programming education. This study addresses concerns about overreliance on AI-generated code by conducting a controlled experiment with 69 novice learners aged 10 to 17. Key findings reveal that Codex significantly improves code-authoring performance without impairing manual coding skills, as evidenced by a 1.15x increase in completion rates and 1.8x higher scores. Learners with access to Codex during training performed slightly better on subsequent evaluations, though not statistically significant. Notably, students with stronger prior programming knowledge benefited more from Codex, suggesting its potential to support learners while emphasizing the importance of considering their initial skill levels. This research highlights the promise of AI code generators in enhancing introductory programming education and the need for tailored implementation strategies.

Taking all the research conducted into consideration, CodeCoach stands out as the sole tool that possesses capabilities such as compiling and AI-assisted error handling, lab management features, community forums for each lab, and code challenges. This empowers lab instructors to address every student error from a central

location within the classroom, thereby enhancing interaction between students and instructors during programming lab sessions. The goal is to introduce a novel approach to teaching programming, employing a more streamlined and effective method through an enhanced web application for lab assistance.

3. Methodology

This research was carried out with the intention of building an Interactive Programming Assistance Tool which is designed to teach and learn programming in an effective way. The Tool does not require any special hardware support and it can be easily accessed using an internet connection since the tool is a web application. This research is mainly focused on improving a previous IPAT Tool [9] and to minimize the difficulties of the novice programmers when learning a new language and difficulties faced by the instructors when instructing a large number of students. To achieve this the system has the ability to compile many number of programming languages giving students a new experience with an improved system which consist of AI based programming assistance tool & capability to compile multiple programming languages, A forum to raise questions about lab tasks, A feature for instructors to create their own code challenges and test cases for complete automated evaluation process.

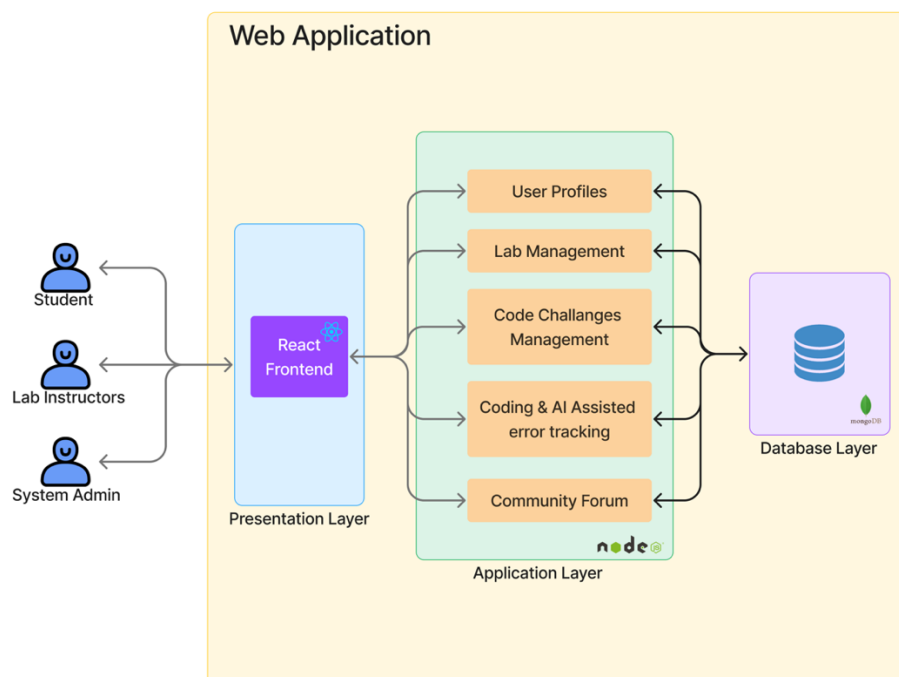


Fig. 1 System Architecture

A. Lab Management Feature

For easing the lab conducting for the instructors the lab Management feature will give the ability to create lab groups and enable or disable the specific forums for the given lab. The aim of this feature is to help organize the labs conducted by the instructors into lab groups so that it's easier to conduct these labs. The instructors can add specific code challenges for the labs. Getting data on how the students have performed during the labs and how much progress that students have made is very important when evaluating them. To do that this feature will provide analytical data on entire lab groups.

B. *Compiling and AI Assisted Error Handling*

This module is designed with two parts, the Compiler has the ability to compile multiple languages which will increase the flexibility and the usefulness of the application exponentially. Secondly the Error handling will be done with the assistance of the GPT 3.5 Turbo AI model which will take the code implemented by the students and the error generated by the compiler or in case of a test case failure the failed test case will be taken. The AI will process this data and suggest hints to the student to complete the coding challenge in a way that will improve their skills in the chosen programming language. Its important for an instructor know how much his/her students are dependent on hints generated by the AI assistance tool, to achieve these goals this feature has the ability to generate a report on average amount of hints used by a specific student.

C. *Managing Code Challenges*

There are two types of code challenges.

- 1) Lab specific code challenges: These code challenges are only accessible by the students and instructors of a specific lab group.
- 2) Common code challenges: These code challenges can be accessed by all the instructors and students that use the system.

To fully automate the process of assigning and completing lab sheets by students the lab tasks are divided into coding challenges which can freely add to the specific lab group from the common Coding challenges pool or create coding challenges specific for a one lab group by the instructor. This functionality will enable the creation of code challenges, editing and deleting these code challenges. Furthermore, to evaluate whether the student correctly completed the task the instructor can create test cases which is comprised of sample output values for a given code challenge. The lab instructors can add these lab specific code challenges to the common code challenges pool and System administrators can add code challenges to the common code challenges pool directly. To evaluate the progress of the students this feature will generate a report on how many tries a student needed to pass the test cases of a specific code challenge.

D. *Community Forums*

There are two types of community forums.

- 1) Community forums which are only accessible by the students and instructors for a given code challenge.
- 2) The common community forum for all the students and lab instructors in a lab group.

Community forums are a great way to communicate and collaborate on coding problems that arise during coding. To give a familiar experience to stack overflow and other community forums for debugging errors, this feature will have a community forum where students can communicate with their peers in the lab and the instructors of the lab by using the community forum that is specific for a code challenge and on a common community forum for all the students for a lab group using this system. The access to common community forum and the labs community forum can be disabled for the students by the instructor as necessary.

E. *User Roles*

According to the three major user categories that interact with this tool, four core components are created for the system: The lab instructor, lab student, System administrator are the three user categories.

Lab Instructor: The instructor or the lecturer who is using the system to conduct the lab session. He/ She can access the common code challenges pool and add them to a lab group as lab tasks or they can create new code challenges and test cases for them which are only accessible to the specific lab group they are created for. Instructors can save these code challenges if needed for future use by adding it to the common code challenges pool. They can answer the questions raised by students in the community forums and post instructions for a lab for all the students using these community forums. To evaluate the progress of the student's instructor has the ability to check which code challenge a student is currently working on and has the ability to check the

submitted code by the students for the previous code challenges and have access to various reports generated for specific lab groups.

Lab Student: The student who is using the system to complete the lab tasks. When a student encounters an error or when he fails a test case for a code challenge, he/she can request hints using the AI assisted error tracking feature. Furthermore, they can use both types of community forums to upload a code snippet to get help from their peers or an instructor. Where others can post solutions to the questions raised by a student via the community forum.

System Administrator: The person who is responsible for creating accounts for instructors and generally administrating the system. He or she is the only one that has access to add or delete code challenges directly from the common code challenges pool. They are responsible for creating user accounts for lab instructors, monitoring, and managing the common community forums.

F. Technologies Involved

There are key technologies that have been used to develop the compiler of the system. The core of the compiler is comprised of a containerize version of Judge0 open source code execution environment using docker [12] The AI assisted error handling feature is built around a custom trained GPT 3.5 model [13]. Node.js [14] and React.js [15] was used to develop the backend and the frontend of the system respectively, while MongoDB which is a non-relational document database was used as the primary database of the CodeCoach system.

G. System Evaluation

There are several decisive measures that have been taken to evaluate the system's effectiveness and accuracy. The evaluation of CodeCoach was carried out in a virtual lab environment which was attended by 15 students and 2 lab instructors. With these participants a lab session was conducted which comprised of 20 coding tasks for the students which were created by their lab instructors. At the end of the session participants were interviewed and asked to rate the usability, effectiveness, and usefulness of the functionalities of the CodeCoach system on a five-point scale.

The lab instructors' average rate was 4.2 and they further commented that giving hints and actively guiding the student to get the answer by themselves using GPT 3.5 model will be a very helpful feature for the students to learn and develop thinking patterns to successfully tackle problems arise while coding. They further commented that the progress tracking feature of the lab session is very useful to track the progress of students individually.

The students gave an average rating of 4. They positively commented on the error tracking feature where a trained GPT model gives personalized hints based on the students submitted code stating that it helped them to solve their errors on their own while giving useful hints to guide them. One student commented that this feature helped him to learn how to think when facing a coding problem or when debugging an error.

4. Result Of The Research

A. Compiler

The compiler feature of the CodeCoach system is a noteworthy accomplishment. It has the ability to compile a variety of programming languages, supporting a total of 38 languages. The ability to accommodate both student preferences and the unique demands of programming projects depends on this diversity. With this capability, instructors can assign tasks using different programming paradigms while giving students the freedom to choose their favorite language.

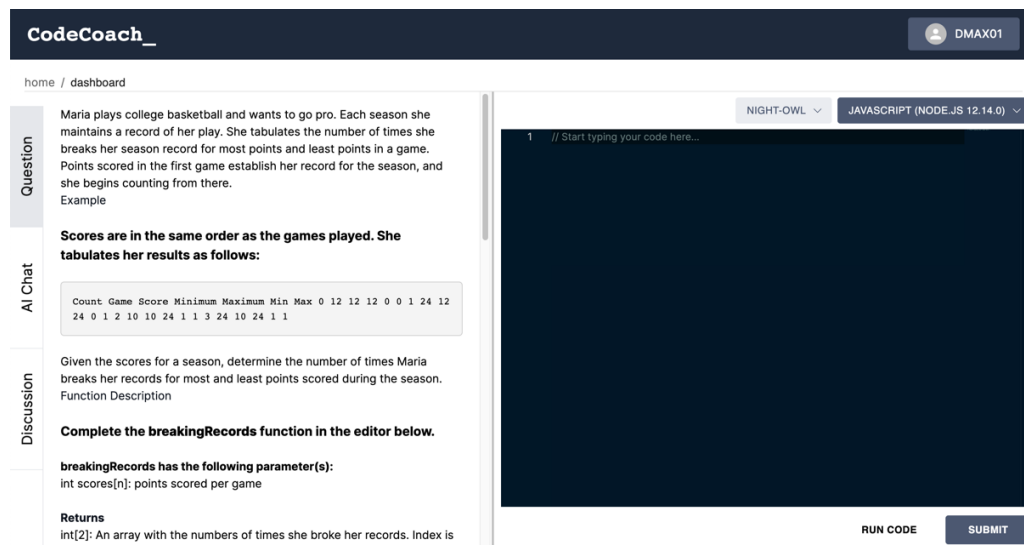


Fig. 2 Snapshot of the Compiler

B. AI-Powered Error Handling

The AI-assisted error handling of the CodeCoach system is one of its most notable features. This function makes use of the GPT 3.5 Turbo AI model, which is essential in assisting students in overcoming coding difficulties. The system records pertinent information when a student experiences an error or fails a test case, including the student's code, the error message, and test case specifics. After processing this data, the AI creates tailored tips and ideas to aid the student.

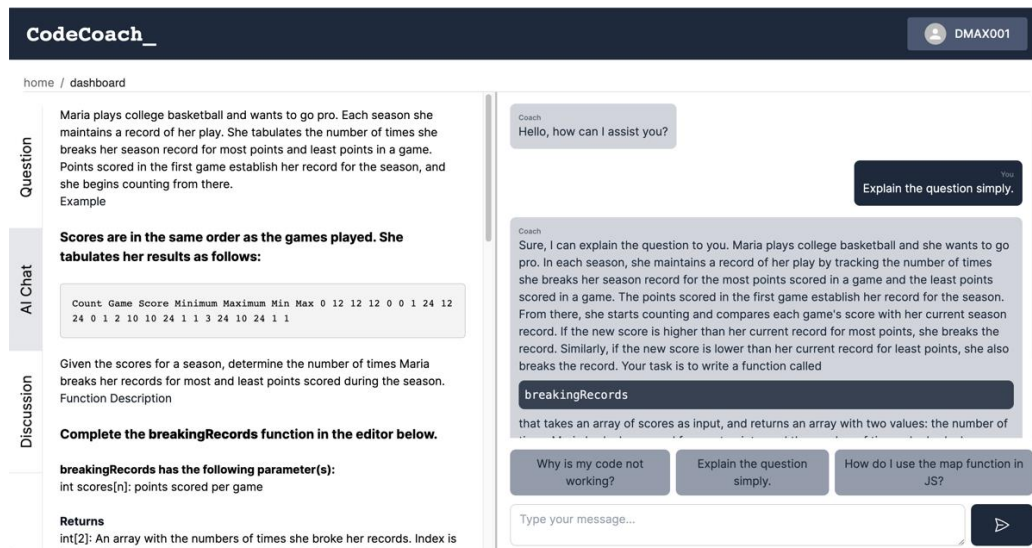


Fig. 3 Snapshot of the AI assistant tool

C. Reports on AI Hint Usage

The system also creates reports on the typical number of tips utilized by certain students, which may be used to evaluate the efficacy of the AI-assisted error management and determine how students are benefiting from this feature. This information is helpful to teachers since it sheds light on how much each student relies on AI-generated hints. These findings can be used by teachers to modify their pedagogical strategies and offer greater assistance to struggling students.

D. Community Forums

The CodeCoach system's community forums have established themselves as useful channels for collaboration and communication. They are set up to encourage dialogue and information exchange around certain coding challenges. These forums are designed to address individual coding challenges. They establish a setting where students can ask for assistance and teachers can offer tips for resolving specific code challenges. This narrow focus guarantees that conversations stay pertinent to the current work, encouraging beneficial interactions. Additionally, this forum's collaborative nature is mirrored by other community-driven coding forums and websites like Stack Overflow.

The ability to enable or disable access to these forums by instructors allows for customization based on the specific requirements of a lab group or the nature of the coding challenge.

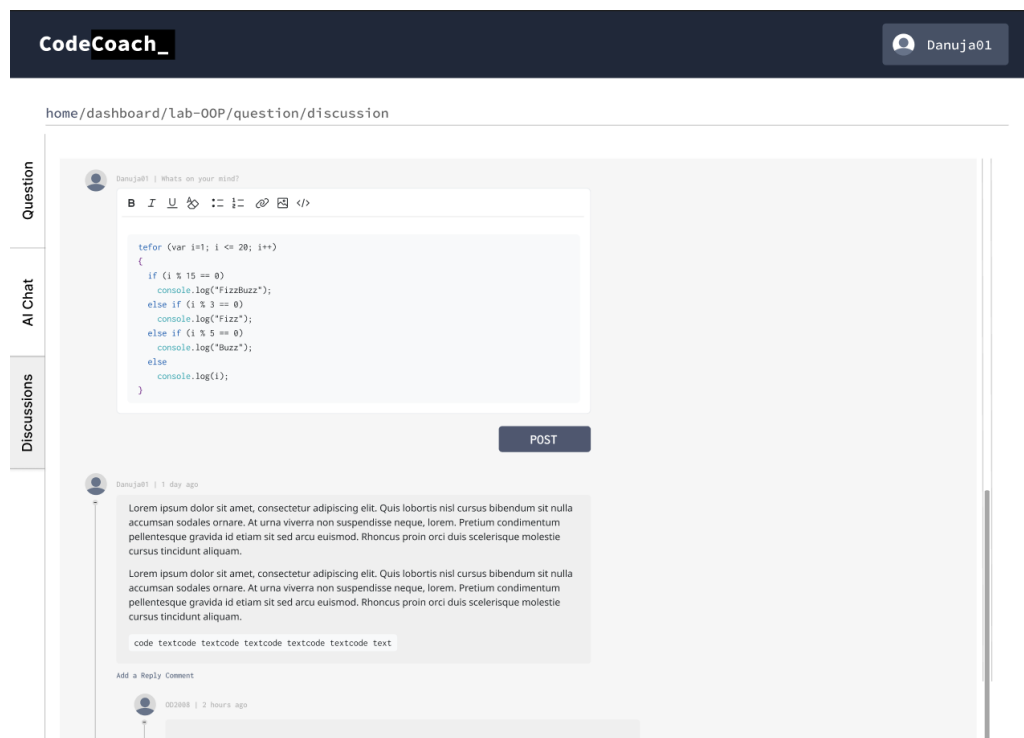


Fig. 4 Snapshot of the Forum

5. Conclusion

According to the "An Interactive Programming Assistance Tool (iPAT) for Instructors and Novice Programmers" research [9], a comprehensive software solution has been crafted specifically for the C# programming language. This tool encompasses features such as robust error handling, screen sharing capabilities, a proficient lab resource management system, and a comprehensive solution archive dedicated to addressing common errors encountered in C# programming but CodeCoach, was developed upon an extensive investigation of the existing instructional tools and techniques. CodeCoach is extremely flexible, supporting numerous types of programming languages. This inclusivity makes sure that students, irrespective of the study language they have selected, receive individual assistance. Furthermore, CodeCoach constitutes a smart tool for programming which takes advantage of GPT-3's impressive abilities to clarify and explain code in an approachable way. This tool's capacity to extend farther than traditional error identification by offering explanations on error causes is impressive. This approach to instruction greatly enhances comprehension of programming concepts and CodeCoach builds an environment of autonomous learning and collaborative problem-solving through the inclusion of an individual forum for lab groups, where both students and instructors can ask questions, engage in knowledge-sharing, and enhance their programming skills. The forum interface gives users the flexibility to edit or remove posts. Concluding the whole idea behind this research project, CodeCoach is a highly valuable assistance. It is a valuable tool for both students and instructors due to

its ability to clearly explain complex ideas, language adaptability, and emphasis on independent learning. With the introduction of CodeCoach, the practice of programming education has been established by increased accessibility, contribution, and effectiveness, establishing an improved learning environment for every learner.

References

- [1] Butler, J., & Morgan, J. (2007). The transition from secondary to tertiary education in computer science: A student perspective. *ACM Transactions on Computing Education*, 7(1), 1-22
- [2] Bruce, K. (2005). The challenge of teaching programming. *ACM SIGCSE Bulletin*, 37(1), 15-20.
- [3] Lahtinen, L., Jarvela, S., & Hakkarainen, K. (2005). The role of prior knowledge and problem solving skills in learning to program. *Computer Science Education*, 15(2), 129-150.
- [4] Carlsen, S., Nygaard, K., & Stage, J. (2006). Can graduating students design software systems? *Computers & Education*, 46(4), 379-393.
- [5] A. Alamy, A. Carbone and J. Sheard, "Implementation of a Smart Lab for Teachers of Novice Programmers", *Proc. Fourteenth Australasian Computing Education Conference (ACE2012)*, Melbourne, Australia.
- [6] J. Yoo, C. Pettey, S. Yoo, J. Hankins, C. Li, and S. Seo, 2006, "Intelligent tutoring system for CS-I and II laboratory", *Proc. 44th annual south east regional conference (ACM-SE 44)*. ACM, New York, NY, USA, 146-151.
- [7] D. Arnaw and O. Barshey, "WebToTeach: An interactive focused programming exercise system", *Proc. 29th ASEE/IEEE Frontiers in Education Conference (FIE 1999)*, San Juan, Puerto Rico.
- [8] M. Hristova, A. Misra, M. Rutter, and R. Mercuri, "Identifying and correcting Java programming errors for introductory computer science students", *Proc. SIGCSE technical symposium on Computer Science Education (SIGCSE '03)*. ACM, New York, NY, USA, 153-156.
- [9] M. Amaratunga, G. Wickramasinghe, M. Deepal, O. Perera, D. De Silva and S. Rajapakse, "An Interactive Programming Assistance tool (iPAT) for instructors and novice programmers," 2013 8th International Conference on Computer Science & Education, Colombo, Sri Lanka, 2013, pp. 680-684.
- [10] A. Alaqsam, F. Ghabban, O. Ameerbakhsh, I. Alfadli, and A. Fayez, "Current trends in Online Programming Languages Learning Tools: A systematic literature review," *Journal of Software Engineering and Applications*, vol. 14, no. 07, pp. 277-297, 2021. doi:10.4236/jsea.2021.147017
- [11] M. Kazemitabaar et al., "Studying the effect of AI code generators on supporting novice learners in introductory programming," *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023. doi:10.1145/3544548.3580919
- [12] Judge0 CE - API Docs, <https://ce.judge0.com/> (accessed: 17 September 2023).
- [13] GPT-3.5 Turbo fine-tuning and API updates, <https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates> (accessed Sep. 17, 2023).
- [14] "Documentation Node.js", <https://nodejs.org/en/docs> (accessed Sep. 17, 2023).
- [15] "Introducing react.dev," React, <https://react.dev/blog/2023/03/16/introducing-react-dev> (accessed Sep. 17, 2023).