

Smart Code Mentor

Vaibhavi Palande, Khushi Kathane, Anushree Pate, Shloka Bhalerao, Denita Souza

Abstract

The Smart Code Mentor project focuses on creating a practical and user-friendly system that helps individuals improve their coding skills in a more efficient way. Many existing methods of learning and evaluating code depend heavily on manual checking or basic tools, which can be slow, inconsistent, and not very helpful for continuous improvement. This often makes it difficult for learners to understand their mistakes and progress effectively.

To overcome these issues, this project introduces an automated system that provides quick feedback and simplifies code management. The system allows users to write and analyze their code in one place, making the overall process smoother and more organized. It is designed with different modules such as login authentication, data handling, processing, and report generation, which work together to ensure accurate and reliable results. By reducing manual effort and automating key tasks, the system helps save time and 47°C to 55°C minimizes errors.

The developed solution improves both efficiency and user experience by making coding practice more interactive and accessible. It also helps users better understand their performance and identify areas for improvement. Overall, the Smart Code Mentor serves as a helpful tool for students and developers, supporting them in building stronger coding skills. In the future, the system can be further enhanced by adding advanced features like intelligent code suggestions using Artificial Intelligence.

1. Introduction

Background and importance of the problem

In today's digital world, programming has become an important skill for students and workers. But many people find it hard to understand coding ideas, fix mistakes, and get better at solving problems. Most learning websites offer

coding tools, but they often don't give real-time help, personal feedback, or smart support.

This project is about creating a smart and interactive coding mentor that connects theory with practice, making coding easier and more approachable for learners.

2. Motivation and Objectives

Motivation

What existing limitations or problems motivate the study?

Traditional learning resources and tools often fail to provide personalized guidance.

The C programming language is one of the most important foundational languages in computer science education. However, many beginners struggle with writing correct syntax, understanding logical errors, and debugging their programs effectively.

Why is IoT integration important here?

To solve this problem, a Smart Code Mentor System is developed using the Python programming language.

The system analyses C code written by students and identifies mistakes using pattern recognition techniques.

Instead of directly correcting the code, the system provides hints whenever an error is detected after code analysis.

These hints guide students step-by-step toward the correct solution, helping them understand their mistakes and improve their problem-solving skills.

Objectives

The main objectives of this project are:

- To help students in learning C programming effectively .
- Detect errors in code using pattern recognition .
- It provide hints instead of direct solutions .
- Improve debugging skills

3. Related Work

Several researchers in recent years have focused on developing intelligent coding assistance systems and AI-based learning platforms to enhance programming education. These systems mainly concentrate on automatic error detection, personalized learning experiences, code recommendations, and real-time feedback for students.

1. In 2024, researchers introduced AI-based programming tutors that assist students in understanding coding concepts through automated feedback systems.

These platforms utilized Machine Learning and Natural Language Processing techniques to detect syntax and logical errors in student code. The study revealed that students saw improvements in their debugging skills and coding confidence after using these systems.

2. Another research project centered on intelligent debugging assistants designed for beginner programmers. The system analyzed common programming mistakes in languages like C and Python and provided hints instead of direct answers. This approach encouraged students to think logically and solve problems independently, rather than relying entirely on ready-made solutions.

3. A cloud-based smart learning platform launched in 2025 offered personalized coding exercises tailored to each student's performance. The system monitored learning progress and suggested topics where students needed more practice. Researchers found that adaptive learning methods increased student engagement and reduced learning challenges.

4. Some studies explored chatbot-based coding mentors powered by Artificial Intelligence. These mentors interacted with students in real time and answered programming-related questions. While these systems improved the accessibility and availability of guidance, they sometimes struggled to accurately understand complex coding problems.

5. Research on automated code review systems demonstrated that AI tools can effectively identify syntax errors, inefficient code structures, and formatting issues.

These systems helped students enhance code quality and adhere to programming standards without constant teacher supervision.

6. Another important study looked at gamified coding platforms where students learned programming through quizzes, coding challenges, and hints. The results showed that interactive learning environments helped beginners feel more motivated and improved their problem-solving skills.

7. In recent educational research, intelligent tutoring systems were added to online learning platforms. These systems kept track of how students were doing and gave them personalized suggestions. However, most of these systems just gave direct answers instead of helping learners think for themselves.

8. A comparison between traditional teaching and AI-assisted coding education found that students using smart mentoring systems did better in debugging and understanding programming logic. The research stressed how important interactive feedback is in teaching programming.

9. Studies on pattern recognition in programming education showed that machine learning can spot repeated mistakes students make and offer specific suggestions. This helped cut down on mistakes and made coding more efficient.

10. Recent work on AI-driven software engineering tools combined automatic grading, code analysis, and educational support into one platform. These systems made learning more efficient, but they still didn't offer the same level of personalization or emotional understanding as human mentors.

Comparison with Proposed Work

Most existing systems focus mainly on automatic correction or giving direct answers. Some platforms offer instant feedback, but they don't explain mistakes clearly. Others don't provide personalized support or track repeated errors effectively. The proposed Smart Code Mentor is different from previous systems because it uses hints to guide students instead of giving direct answers. The system helps students think logically, improves their debugging skills, and tracks their mistakes to support better learning. It combines code analysis, error detection, and intelligent mentoring into one interactive platform, making coding education more practical and easier for students.

4. Research Gap

Although there are many coding help tools and online learning websites today, there are still some big problems with the technology that's currently available.

Most of these systems give direct answers or finished solutions, which can make it harder for learners to think through problems on their own. Students often end up relying too much on these automatic solutions instead of really understanding the programming ideas behind them.

Another big problem is that many platforms focus more on getting the right answer than on helping students improve their overall problem-solving skills. They don't usually explain why something went wrong or how to prevent similar mistakes in the future.

Current systems also have trouble keeping track of the same mistakes students make again and again. Because of this, teachers and students can't easily find out where someone is struggling or see how much they've improved over time. Personalized help is also limited, especially in big classes where individual attention is hard to give.

Plus, many systems don't offer detailed explanations for things like syntax errors, logic errors, or how to debug code. Beginners often get stuck trying to understand their mistakes, which can lead to frustration and a loss of confidence in programming.

Because of all these issues, there's a need for a smarter mentoring system that not only finds mistakes but also offers helpful hints, personalized guidance, tracks errors, and provides interactive learning support.

The Smart Code Mentor is designed to address these problems by creating a better, more helpful learning environment for coding students.

5. Proposed Approach

Even though there are many coding help tools and online learning websites available today, there are still some major issues with the technology that's currently being used.

Most of these systems give direct answers or completed solutions, which can make it harder for learners to think through problems on their own. Students often end up relying too much on these automatic solutions instead of really understanding the programming ideas behind them.

Another big problem is that many platforms focus more on getting the right answer than on helping students improve their overall problem-solving skills. They don't usually explain why something went wrong or how to prevent similar mistakes in the future.

Current systems also have trouble keeping track of the same mistakes students make again and again. Because of this, teachers and students can't easily find out where someone is struggling or see how much they've improved over time. Personalized help is also limited, especially in big classes where individual attention is hard to give.

Plus, many systems don't offer detailed explanations for things like syntax errors, logic errors, or how to debug code. Beginners often get stuck trying to understand their mistakes, which can lead to frustration and a loss of confidence in programming.

Because of all these issues, there's a need for a smarter mentoring system that not only finds mistakes but also offers helpful hints, personalized guidance, tracks errors, and provides interactive learning support. The Smart Code Mentor is designed to address these problems by creating a better, more helpful learning environment for coding students.

6. Advantages and Disadvantages

Advantages (Key Strengths of Smart Code Mentor)

1. Personalized Learning Experience:

The system adjusts to the learner's skill level, speed, and how they learn, offering tailored coding help.

2. Instant Feedback and Error Detection:

It gives real-time advice, helps fix mistakes, and explains things, which makes learning faster and smoother.

3. 24/7 Availability:

You can use it anytime, day or night, so you can practice and get help whenever you need.

4. Enhanced Productivity:

It handles repetitive tasks like correcting code or offering suggestions, helping you work more quickly.

5. Interactive Learning Environment:

It keeps you involved with quizzes, hints, step-by-step guides, and clear explanations.

6. Scalability in Education:

It can handle many users at once, which is great for classrooms and online learning platforms.

7. Bridging Skill Gaps:

It helps new learners grasp tough topics and supports experienced users in improving their code.

Disadvantages (Limitations / Challenges)

1. Dependence on Technology:

Users might rely too much on the system, which could weaken their ability to solve problems on their own.

2. Limited Context Understanding:

It might not fully understand complex or unclear problems, leading to wrong suggestions.

3. Quality of Suggestions Varies:

The suggestions depend on the data and algorithms used, which can sometimes result in poor or incorrect code.

4. Lack of Human Intuition:

It can't fully replace human mentors, especially when it comes to creative thinking, ethics, and deep understanding of concepts.

5. Data Privacy Concerns:

User code and input may be stored or processed, which could raise security and privacy issues.

6. Initial Development Cost:

Creating and keeping such a system running needs a lot of time, money, and skilled people.

7. Internet Dependency:

Most smart systems need a stable internet connection to work properly.

8. Bias in Training Data:

It might show biases from the data it was trained on, which could affect fairness and inclusivity.

7. Applications

1. E-Learning Platforms and Online Education:

Integrated into platforms like Coursera and Udemy to offer interactive coding help, instant feedback, and personalized learning paths.

2. Academic Institutions and Classrooms:

Used in schools and colleges to assist students during programming labs, assignments, and projects, reducing their reliance on instructors.

3. Software Development and Industry Training:

Helps developers in companies with debugging, code optimization, and learning new technologies, which improves productivity and code quality.

4. Coding Practice Platforms:

Enhances platforms like LeetCode and HackerRank by providing guided hints, explanations, and solution analysis.

5. Self-Learning and Skill Development:

Individuals preparing for interviews or learning programming on their own can use it as a personal coding tutor that's always available.

8. Conclusion

Summary of Findings

The study on Smart Code Mentor shows that AI-powered coding assistants can greatly improve the learning and development process in programming.

The system offers personalized guidance, instant feedback, and interactive support, making coding easier and more efficient for both new learners and experienced developers. It also helps increase productivity and reduces common mistakes during software development.

Contribution to the Field

This research contributes to the fields of Artificial Intelligence and Software Engineering Education by introducing an intelligent mentoring system that connects traditional teaching methods with modern technological advances.

Smart Code Mentor offers a scalable and adaptable approach to coding education, helping learners improve problem-solving skills and overall code quality.

Future Research Directions

Future work can focus on improving the system's ability to understand context and increase its accuracy using more advanced AI models.

Integrating the system with emerging technologies like adaptive learning platforms, voice-based assistants, and real-time collaboration tools could make it more user-friendly.

Additionally, research can explore better data privacy features, offline functionality, and support for more programming languages to make the system more robust and widely available.