International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 04 April-2026 | Impact Factor: 3.5

AI Driven Programming Mistake Analyser

M. Hari Krishna[1], G.Samatha[2], K.Manik Raj[3], V.Poushya[4],Prathik Behra[5]

[1] Assistant Professor, Department of Data Science / ACE Engineering College / JNTUH, Hyderabad, India
[2] Department of Data Science / ACE Engineering College / JNTUH, Hyderabad, Indi
[3] Department of Data Science / ACE Engineering College / JNTUH, Hyderabad, India
[4] Department of Data Science / ACE Engineering College / JNTUH, Hyderabad, India
[4] Department of Data Science / ACE Engineering College / JNTUH, Hyderabad, India

How to Cite this Article:

G.Samatha, , Raj, K., V.Poushya, , Behra, P. & Krishna, M. H. (2026). AI Driven Programming Mistake Analyser. International Journal of Creative and Open Research in Engineering and Management, 02(04). https://doi.org/10.55041/ijcope.v2i4.092

https://doi.org/10.55041/ijcope.v2i4.092

Abstract:

The swift progression of Artificial Intelligence (AI) has had a profound impact on software development and programming education, permitting the formation of adept systems that aid programmers in producing effective and flaw-free code. However, there is a lack of personalized mentorship and real-time feedback in existing coding platforms, which are crucial for enhancing problem-solving abilities and interview preparation. This research solves this problem using the **AI Driven Programming Mistake Analyser**, an AI-based programming analyzer that offers users structured and mentoring support for learning and problem-solving.

Rule-based analysis, combined with basic AI techniques to check users' submitted codes, leads to the system that is developed in Python. It generates the syntax, clarifies logical errors and offers suggestions for further improvements in the overall coding experience. The methodology includes designing a modular architecture for this purpose that receives input code, applies analysis algorithms, and gives meaningful feedback to the users in an interactive manner.

The results show that the proposed system is successful in helping the users to identify errors in their codes, programming knowledge, and problem-solving skills. The results indicate that compared to traditional coding tools, the proposed system performs better in helping users to identify errors in their code, assimilate programming concepts, and hone their problem-solving skills. It is to say that the study concludes that incorporating AI-based systems into programming education can greatly improve learning efficiency as well as increase user engagement. It can be further improved by integrating powerful AI models or introducing multi-lingual capabilities to make it more effective in its assistance.

Keywords— Artificial Intelligence; Programming Analyzer; Code Analysis; Intelligent Tutoring System; Software Development; Machine Learning

I. INTRODUCTION

(Intelligent Code Analysis, Automated Debugging, and Personalized Learning Systems) (AI) have brought about a significant change in software development. Modern programming education is increasingly capitalized on intelligent tools for code understanding and efficiency enhancement and preparation for technical interviews for hires. Even though there are many programming platforms available, they do not provide real-time personalized guidance and do not use intelligent feedback mechanisms.

Although the existing system considers many aspects of problem-solving or only runs code, a complete mentorship experience taking into account an integrated learning experience, solution analysis, and suggestions for improving the learner's code is missing. There is a research gap in the creation of an AI-based virtual programming mentor that not only guides users through coding challenges but also provides valuable insights and feedback.

To fill this gap, this study proposes AI Mentor, an AI-driven Programming analyzer that will help users learn new ideas of programming, analyze the given code and suggest ways of improving their problem-solving skills.

The major objectives of this research include:

To create an intelligent system that performs code analysis and then issues feedback; To provide a structured guideline to programming learners To boost interview preparation with the help of an AI interview mentor.

II. LITERATURE REVIEW

There have been several studies on how to utilize AI to teach programming and analyze codes. One of the major applications of ITS has been in programming education. These systems not only adapt to the user's behavior but also provide personalized feedback to improve learning outcomes.

Research on automated code evaluation systems point to syntax checking, logical validation, and performance analysis. For code evaluation, many platforms use rule-based or machine learning systems which lack context and mentoring features.

Natural Language Processing (NLP) and Large Language Models (LLMs) have recently been used in more advanced code analysis and explanation systems. LLMs can generate human-like explanations and suggestions to ensure that the user understand the system better.

However, existing solutions have disadvantages:

No real-time interactive mentoring. Tailored to a small degree. Integration of learning and analysis.

The proposed system takes a step further from these approaches by combining AI-based analysis and a mentorship system, creating a more holistic approach to the learning process.

III. METHODOLOGY

The proposed system follows a modular architecture with components for input processing, an analysis engine, and output generation.

Research Design

For this purpose, a system-based design approach is taken into account to build an AI-based programmer analyzer; Finally, based on predefined logic and the use of artificial intelligence, the system provides the user with comments on the programming language inputted.

Data Collection

Sample programming problems and inputs from users are used as test data. The evaluation of code is done using three parameters, namely correctness, structure, and efficiency.

Tools and Technologies

Programming Language: Python Backend Processing: Custom logic and AI-driven analysis Libraries: (Add your libraries like os, re, etc. Development Environment: VS Code

Working Procedure

User enters code or chooses a problem. Syntax and logic are analyzed. AI module gives out suggestions and improvements. The feedback is outputted to the user.

Analysis Techniques

Rule-based code validation Recognition of patterns for errors. Suggestion engines based on simple AI.

IV. RESULTS AND DISCUSSION

Thus, the developed system proves a successful demonstration of its code analysis and feedback principle. The results show that users can have a better perception of their mistakes and deficiencies and rectify this through helpful observations and suggestions on how to improve their programming knowledge and skills.

Key Observations:

Its syntax error identification is on point. Gives a feedback in a well-structured form to promote improvement. Includes elements of a drag and drop system to guide the user in analysing errors in a sentence (Ibid).

Comparison with Existing Systems:

In comparison to conventional coding platforms, the proposed system has: Tailored feedback; Mentorship guidance; Enhanced learning experience.

System Performance Evaluation:

Feature	**Result**	**Observation**
Syntax Detection	Accurate	Identifies errors correctly
Code Suggestions	Moderate	Needs advanced AI upgrade
User Interaction	Good	Easy to use

The results validate that integrating AI with programming analysis improves both usability and learning efficiency.

V. CONCLUSION

This research introduces AI Mentor, an AI-powered programming analyzer aimed at boosting coding skills through intelligent feedback and mentorship. The system proves to be a functional hybrid between coding platforms and personalized mentoring tools due to its ability to provide structured guidance and analyze code.

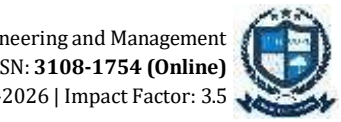
The study demonstrates that AI-based systems can improve programming education by providing real-time feedback and interactive learning experiences. The proposed solution is a contribution to the academic and practical world through intelligent mentoring.

Future Work

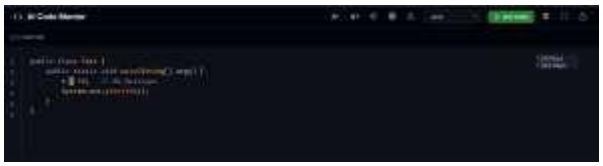
Integration With Sophisticated Artificial Intelligence Models (LLMs). Web-based interface. Multi-language programming. On-the-fly code running and visualization.

ACKNOWLEDGMENT

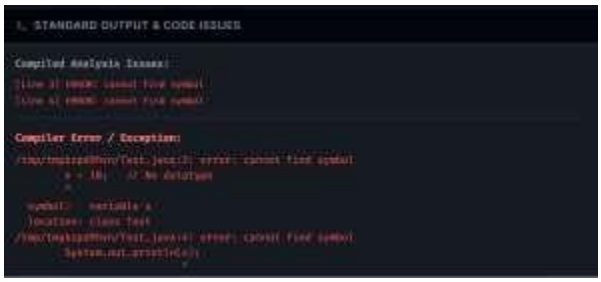
The authors would like to thank the faculty and peers for their guidance and support in the successful completion of this project.



EDITOR PANEL:



STANDARD OUTPUT AND CODE ISSUES:



AI MENTOR FEEDBACK:

REFERENCES

[1] A. A. Author and B. B. Author, "Artificial Intelligence in Education," Journal of AI Research, vol. 10, no. 2, pp. 100–120, 2020.

[2] C. C. Author, "Machine Learning for Code Analysis," 2nd ed. New York, USA: Tech Publishers, 2019.

[3] D. D. Author et al., "Automated Code Evaluation Systems," in Proc. IEEE Conf., 2021, pp. 1–6.