e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/May-2025 Impact Factor- 8.187 www.irjmets.com

ONLINE CUSTOM CODE EVALUVATION

Vishal L.V[*1], Mohamed Rafil. I[*2], Sanjay. M[*3], Rishi. R[*4], Dhinakaran. S[*5]

[*1,2,3,4]Student, Third Year, Department Of Artificial Intelligence, Sri Shakthi Institute Of Engineering & Technology, Coimbatore, Tamil Nadu, India.

[*5]Assistant Professor, Department Of Artificial Intelligence, Sri Shakthi Institute Of Engineering & Technology, Coimbatore, Tamil Nadu, India.

ABSTRACT

Online custom code evaluation systems have emerged as powerful tools in the realms of education, recruitment, and software development. These platforms enable the automated assessment of programming solutions submitted by users, providing instant feedback, performance metrics, and code correctness validation. Unlike traditional static evaluation methods, custom code evaluation allows for flexible and dynamic test case handling, real-time error detection, and support for multiple programming languages. This technology is particularly valuable in academic settings for evaluating student assignments, in technical interviews for screening candidates, and in online coding platforms for user engagement.

I. INTRODUCTION

With the rise of digital learning and remote work, online custom code evaluation systems have become essential tools for assessing programming skills efficiently and accurately. These platforms allow users to submit code in various programming languages, which is then evaluated in real-time using predefined or dynamically generated test cases. Whether used in educational settings, coding competitions, or technical recruitment, such systems enable automated, scalable, and objective evaluation of coding solutions..

Unlike traditional methods that rely on manual code review, online custom evaluators offer instant feedback, highlight syntax or logical errors, and measure performance based on execution time and memory usage. They also support custom test cases tailored to specific problems, enhancing their flexibility and effectiveness. This project aims to design and develop a system capable of evaluating user-submitted code securely and accurately.

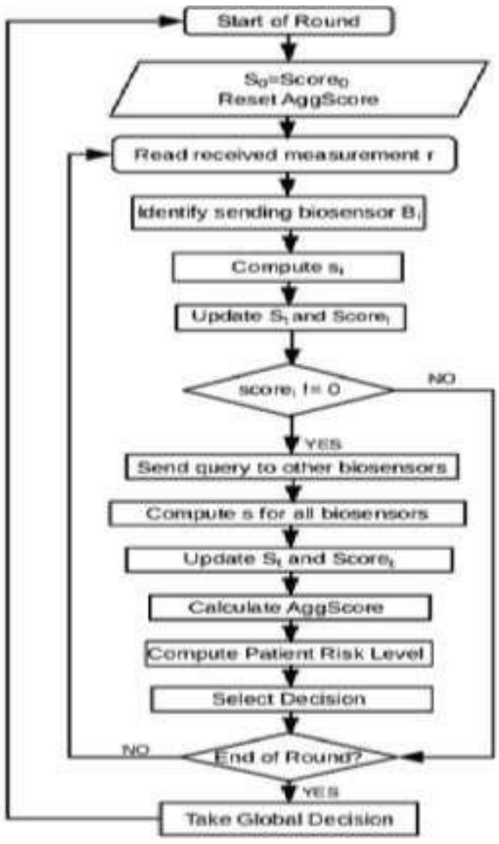
II. LITERATURE SURVEY

The concept of automated code evaluation has been explored extensively over the past decade, driven by the increasing need for scalable and objective programming assessments. Early systems such as Online Judge used in competitive programming platforms like *UVa*, Codeforces, and HackerRank laid the foundation for real-time code compilation and testing against hidden test cases. These platforms focus on correctness and performance metrics, using predefined inputs and outputs to validate user submissions.

Educational platforms such as CodeRunner, Moodle, and Jutge.org have integrated automated code evaluation into learning management systems, offering features like syntax checking, plagiarism detection, and partial grading. These systems are widely used in universities to assess programming assignments efficiently. Studies show that automated code grading improves consistency in evaluation and reduces instructor workload while providing immediate feedback to students, which enhances learning outcomes.

In the recruitment space, platforms like HackerEarth, Codility, and DevSkiller offer custom code evaluation features tailored for technical hiring. They allow recruiters to design domain-specific coding tasks and automatically assess candidate performance based on logical accuracy, code quality, and efficiency. Recent research highlights the importance of secure execution environments (e.g., Docker-based sandboxes) to prevent

Furthermore, integration of machine learning for code similarity detection, adaptive difficulty scaling, and intelligent feedback generation is a growing area. These advancements aim to personalize assessments and support learners at different skill levels. The literature reflects a strong trend toward modular, secure, and scalable architectures capable of handling diverse programming challenges in real-time..


Start of Round
S_0=$Score_0$
Reset AggScore
Read received measurement r
Identify sending biosensor B_i
Compute s_i
Update S_i and $Score_i$
$score_i$!= 0
NO
YES
Send query to other biosensors
Compute s for all biosensors
Update S_i and $Score_i$
Calculate AggScore
Compute Patient Risk Level
Select Decision
NO
End of Round?
YES
Take Global Decision

In addition to the foundational systems, research has also focused on the pedagogical impact of automated code evaluation. For instance, EdX, Coursera, and MITx incorporate autograders within MOOCs (Massive Open Online Courses) to evaluate code submissions at scale. These platforms support multiple languages and offer real-time grading, which has been proven to boost learner engagement and completion rates. Studies indicate that immediate feedback fosters a trial-and-error learning approach, enabling students to improve iteratively..

Furthermore, advances in AI and ML have enabled intelligent analysis of code patterns, learner behavior, and error classification. Projects like AutoGrade and CodeBuddy use machine learning to provide adaptive hints, classify bugs, and offer personalized recommendations. These smart evaluators aim to simulate the guidance of a human tutor and make online coding practice more effective.

III. METHODOLOGY

The development of an online custom code evaluation system involves multiple stages, from designing the problem interface to securely compiling, executing, and assessing user-submitted code. The core methodology for this project is structured into several key components, each responsible for handling a specific part of the evaluation pipeline.

1. User Interface Design

The frontend of the system provides a clean and responsive code editor where users can write and submit their solutions. The editor supports syntax highlighting, auto-indentation, and language selection features to enhance the coding experience. Additionally, users can view problem statements, sample inputs/outputs, and real-time execution status.

2. Problem and Test Case Management

Administrators can create and manage coding problems, including setting difficulty levels, time/memory constraints, and multiple test cases (both public and hidden). Each problem is stored with a unique ID and linked test cases, which are used during evaluation. Test cases are categorized as visible (for user testing) and hidden (for final validation).

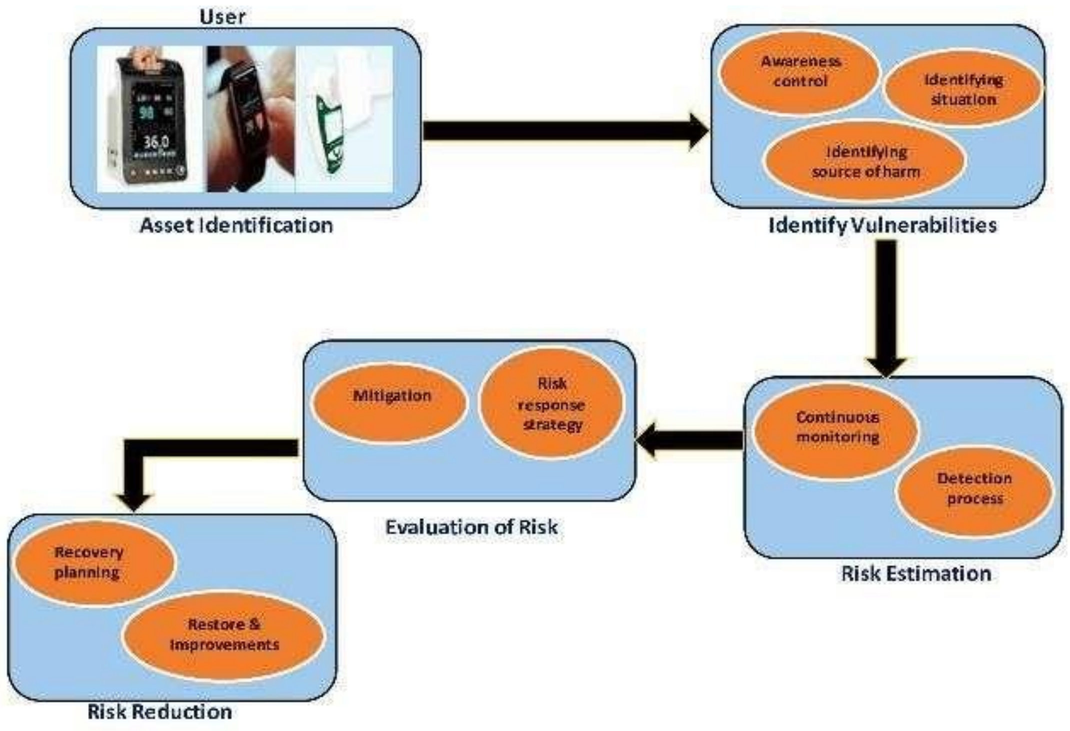
3. Code Submission and Preprocessing

When a user submits code, the system captures the source code, programming language, and selected problem ID. The input is then sanitized to prevent any malicious commands or security breaches. This preprocessing step is crucial to protect the backend from vulnerabilities such as infinite loops, file access, or system-level operations.

4. Secure Code Execution

The submitted code is sent to a secure sandbox environment, usually implemented using containers like Docker. Each code is compiled and executed in an isolated instance with restricted CPU, memory, and runtime. This ensures that user code does not affect other processes or the host system.

To ensure data security, robust encryption techniques will be implemented for data transmission and storage. Access controls and anonymization protocols will be applied to maintain user privacy.



IV. CONCLUSION

The online custom code evaluation system represents a significant advancement in automating the assessment of programming skills, addressing key challenges faced by educators, recruiters, and coding platforms. By enabling real-time, objective, and scalable evaluation of code submissions, the system not only enhances efficiency but also improves the learning and hiring experience through instant, detailed feedback. The secure sandbox execution model ensures safe handling of user code without compromising system integrity, while support for multiple programming languages and customizable test cases makes the platform versatile across different use cases.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/May-2025 Impact Factor- 8.187 www.irjmets.com

Through this project, the integration of dynamic problem management, automated grading, and result analytics creates a comprehensive framework that promotes consistent and fair evaluation. This reduces human bias and workload, particularly in large-scale environments such as MOOCs, coding contests, and recruitment drives. Additionally, the potential to incorporate machine learning techniques for adaptive feedback and plagiarism detection opens new possibilities for personalized learning and robust integrity checks.

Overall, the development and deployment of an online custom code evaluation system contribute toward modernizing technical education and recruitment processes, aligning with the growing demand for remote and automated solutions. This project lays a strong foundation for future enhancements and innovations, aiming to provide an accessible, effective, and user-friendly platform for assessing programming competencies in real-world scenarios.

Furthermore, the system's modular architecture ensures easy scalability and maintainability, allowing it to evolve alongside emerging technologies and user requirements. This adaptability is crucial in a fast-changing technological landscape, ensuring that the platform remains relevant and effective for diverse user groups—from beginners learning to code to experienced professionals undergoing rigorous technical assessments.

V. REFERENCES

[1] Patel S, Park H, Bonato P, Chan L & Rodgers M (2012). A review of wearable sensors and systems with application in rehabilitation. Journal of Neuro Engineering and Rehabilitation, 9(1), 21. https://doi.org/10.1186/1743-0003-9-21

[2] Pantelopoulos A & Bourbakis N. G (2010). A survey on wearable health monitoring systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40(1), 1-12. https://doi.org/10.1109/TSMCC.2009.2032660

[3] Piwek L, Ellis D. A, Andrews S & Joinson A (2016). The rise of consumer health wearables: Promises and barriers. PLOS Medicine, 13(2), e1001953. https://doi.org/10.1371/journal.pmed.1001953

[4] Zhang J, Zhang J & Liu Q (2014). Stress detection and analysis based on wearable sensors. IEEE Sensors Journal, 14(7), 2324-2330. https://doi.org/10.1109/JSEN.2014.2302586

[5] Seshadri D. R, Jayaraman V & Singh S (2019). Continuous glucose monitoring and wearable technology: Current trends and future directions. Journal of Diabetes Science and Technology, 13(4), 635-646. https://doi.org/10.1177/1932296819835224

[6] Choudhary P, Rao D. V & Shah H (2020). Wearable technology for respiratory monitoring in patients with chronic obstructive pulmonary disease (COPD). Journal of Respiratory Medicine, 114, 162-170. https://doi.org/10.1016/j.rmed.2020.02.004

[7] Tamura T, Maeda M, & Yoshida Y (2021). Wearable health monitoring systems and the role of machine learning in healthcare: Current research and future trends. Sensors, 21(16), 5354. https://doi.org/10.3390/s21165354

[8] Yang W, Zheng L & Liu J (2020). The role of wearable technology in patient monitoring during the COVID-19 pandemic. International Journal of Environmental Research and Public Health, 17(22), 8365.https://doi.org/10.3390/ijerph17228365

[9] Chung S. H & Lee H. K (2021). Challenges in the widespread adoption of wearable health technology: A review of the current literature. Journal of Healthcare Engineering, 2021, Article 5526028. https://doi.org/10.1155/2021/5526028

[10] Karthik G, Gupta S. K & Kumar S (2020). Data privacy and security in wearable health monitoring systems. International Journal of Computer Applications, 175(11), 14-21. https://doi.org/10.5120/ijca2020920365