iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VII **Month of publication:** July 2026

DOI: https://doi.org/10.22214/ijraset.2026.84221

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

AI-Powered Real-Time Code Review and Bug Detection System Using MERN Stack and Python

Pedanayuni Amalakumari[1], R. Rajesh[2]
[1]*MCA Final Semester,* [2]*Assistant Professor, Master of Computer Applications, Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India*

Abstract: ***The AI-Powered Real-Time Code Review and Bug Detection System Using MERN Stack and Python is an intelligent web application designed to automate the software code review process. The system helps developers identify coding errors, security vulnerabilities, and code quality issues in real time. It is developed using the MERN Stack, where React.js provides the user interface, Node.js and Express.js handle backend operations, and MongoDB stores application data. Python is integrated as the AI engine to perform intelligent code analysis and bug detection. Users can upload source code or connect GitHub repositories for automated review. The AI engine analyzes the code using static analysis techniques and machine learning methods to detect syntax errors, logical bugs, security issues, and coding standard violations. The system also generates code quality scores and provides optimization suggestions to improve software performance and maintainability. JWT-based authentication ensures secure access, while Socket.IO enables real-time notifications and instant feedback. All review reports are securely stored in the database and displayed through an interactive dashboard. The proposed system reduces manual code review effort, improves software reliability, enhances developer productivity, and supports secure software development practices. Its scalable architecture makes it suitable for both academic projects and enterprise-level software development environments.***
Keywords :Artificial Intelligence (AI), Code Review, Bug Detection, MERN Stack, Python, Machine Learning, Static Code Analysis, MongoDB, React.js, Node.js.

I. INTRODUCTION

Software development has become increasingly complex with the rapid growth of modern technologies and large-scale applications. Ensuring high-quality, secure, and error-free code is one of the biggest challenges faced by software developers. Traditional manual code review is time-consuming, inconsistent, and requires significant human effort, making it difficult to identify all coding issues efficiently. To overcome these limitations, Artificial Intelligence (AI) is being integrated into software development processes to automate code analysis and improve development quality. The AI-Powered Real-Time Code Review and Bug Detection System Using MERN Stack and Python is designed to perform intelligent code reviews by automatically detecting bugs, security vulnerabilities, and coding standard violations. The system allows developers to upload source code or connect GitHub repositories for real-time analysis. The MERN Stack provides a responsive and scalable web platform, while Python-based AI models analyze the code and generate meaningful recommendations. The application also provides code quality scores, optimization suggestions, and detailed review reports through an interactive dashboard. Real-time communication ensures that developers receive instant feedback during the development process.

II. EXISTING SYSTEM

The existing code review process mainly depends on manual inspection by experienced software developers and conventional static code analysis tools. Developers review source code to identify syntax errors, logical mistakes, coding standard violations, and security vulnerabilities before deployment. Although manual review improves software quality, it is often slow, time-consuming, and prone to human error, especially in large-scale software projects. [12]Traditional static analysis tools can detect only predefined coding issues and usually fail to understand the overall context and logic of the program. These tools provide limited recommendations and cannot generate intelligent suggestions for improving code quality. Most existing systems also lack real-time feedback, forcing developers to wait until the review process is completed.[5]Integration with modern development workflows and version control platforms is often limited. Handling multiple programming projects simultaneously can also become difficult due to scalability issues. As software complexity increases, manual review becomes less efficient and more expensive. Therefore, existing systems are unable to provide fast, intelligent, and comprehensive code analysis required in modern software development environments.

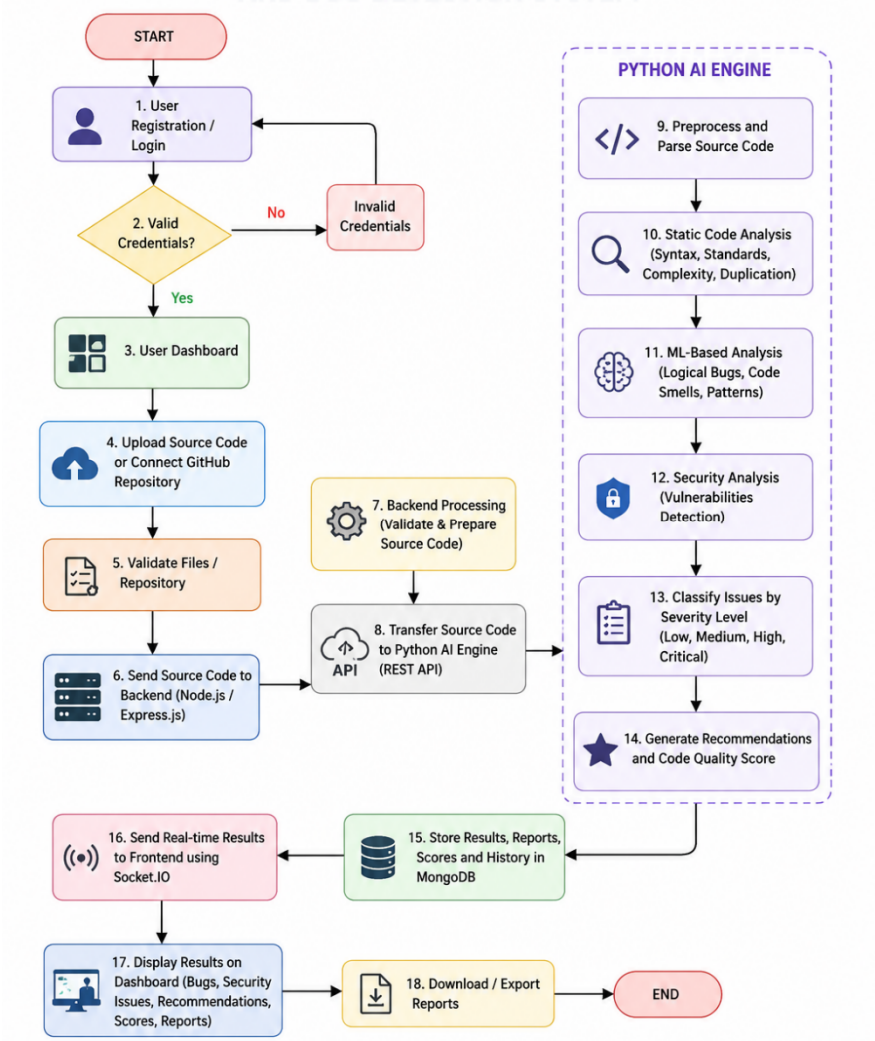


Figure1 : Flow Chart

CHALLENGES

Developing the AI-Powered Real-Time Code Review and Bug Detection System Using MERN Stack and Python involves several technical and operational challenges. [9]The AI engine must accurately analyze different coding styles and programming patterns without producing excessive false positives or false negatives. Supporting multiple programming languages and frameworks requires continuous model training and regular updates. Processing large source code files and GitHub repositories in real time demands high computational resources and optimized algorithms. Integrating the Python AI engine with the MERN Stack while maintaining fast communication is another significant challenge. Ensuring secure handling of uploaded source code and repository data is essential to protect user privacy and intellectual property.[2] Detecting complex logical errors and framework-specific issues is difficult because they often require contextual understanding. Maintaining high performance when multiple users access the system simultaneously is also challenging. The application must provide real-time feedback without affecting response time or system stability. Keeping AI models updated with evolving coding standards and newly discovered security vulnerabilities requires continuous maintenance. Achieving scalability, reliability, and accurate code analysis while ensuring seamless user experience remains one of the major challenges in developing an intelligent automated code review platform.

III. PROPOSED SYSTEM

The AI-Powered Real-Time Code Review and Bug Detection System Using MERN Stack and Python is an intelligent web-based application designed to automate the software code review process. [1]The system enables developers to upload source code or connect GitHub repositories for instant AI-based analysis.

It is developed using the MERN Stack, where React.js provides an interactive user interface, Node.js and Express.js manage backend operations, and MongoDB stores user and project information. Python is integrated as the AI engine to perform static code analysis and machine learning-based bug detection. The system automatically identifies syntax errors, logical bugs, security vulnerabilities, and coding standard violations in real time. It also generates code quality scores and provides intelligent recommendations to improve code readability, maintainability, and performance. JWT authentication ensures secure user access, while Socket.IO delivers instant notifications and live review updates. [8]All analysis reports are securely stored in the database and displayed through a user-friendly dashboard. The proposed system reduces manual review effort, improves software quality, enhances security, increases developer productivity, and provides a scalable solution suitable for modern software development environments.

IV. ALGORITHM

Algorithm: AI-Powered Real-Time Code Review and Bug Detection
Step-by-Step Algorithm
Step 1: Start the application and initialize the frontend, backend, AI engine, and database services.
Step 2: Register a new user or authenticate an existing user using JWT-based login credentials.
Step 3: Display the user dashboard after successful authentication.
Step 4: Allow the user to upload source code files or connect a GitHub repository.
Step 5: Validate the uploaded files and verify repository access permissions.
Step 6: Transfer the source code from the Node.js backend to the Python AI Engine through secure REST APIs.
Step 7: Preprocess and parse the source code for static code analysis.
Step 8: Extract important code features such as syntax, complexity, coding standards, variable usage, and function structures.
Step 9: Apply AI and Machine Learning models to detect syntax errors, logical bugs, code smells, and security vulnerabilities.
Step 10: Classify detected issues based on severity levels such as Low, Medium, High, and Critical.
Step 11: Generate code quality scores, optimization suggestions, and best practice recommendations.
Step 12: Store the complete analysis results, reports, and review history in the MongoDB database.
Step 13: Send the generated results to the React.js frontend using Socket.IO for real-time updates.
Step 14: Display the bug report, security analysis, AI recommendations, and downloadable report on the user dashboard.
Step 15: End the process or wait for the next code review request.

A. Input

The system accepts source code files or GitHub repository links from authenticated users. The uploaded code is validated and securely forwarded to the Python AI engine for intelligent analysis. [7]User authentication details and project information are also collected to manage review history and reports efficiently.

B. Processing

The backend preprocesses the uploaded source code and transfers it to the AI engine. Static code analysis and Machine Learning techniques examine the code for syntax errors, logical bugs, coding standard violations, performance issues, and security vulnerabilities. The AI engine evaluates code quality, assigns severity levels, generates optimization suggestions, and stores the complete review results in MongoDB for future reference.

C. Output

The system produces a comprehensive code review report containing detected bugs, security vulnerabilities, code quality scores, severity levels, optimization recommendations, and best coding practices. [3]The results are displayed instantly through an interactive dashboard, allowing developers to download reports and improve their source code before deployment.

D. Advantages of the Algorithm

The proposed algorithm automates the entire code review process, significantly reducing manual effort and review time. It provides real-time bug detection, intelligent AI-based recommendations, and security analysis, enabling developers to write secure, efficient, and maintainable software. Its scalable architecture supports continuous software quality improvement and enhances overall developer productivity.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

V. TECHNIQUES

The AI-Powered Real-Time Code Review and Bug Detection System Using MERN Stack and Python utilizes several advanced techniques to automate the software code review process. These techniques enable the system to detect bugs, identify security vulnerabilities, evaluate code quality, and provide intelligent recommendations.[15] By combining Artificial Intelligence, Machine Learning, Static Code Analysis, and Security Analysis, the system improves software reliability, reduces manual effort, and enhances developer productivity.

A. Artificial Intelligence (AI)

Artificial Intelligence serves as the core technology of the proposed system. The AI engine analyzes the uploaded source code to understand its structure, detect coding issues, and generate intelligent suggestions.[13] It evaluates programming logic, coding standards, readability, and maintainability to provide developers with meaningful recommendations. AI automates repetitive review tasks, reduces human intervention, and improves the overall software development process.

B. Machine Learning (ML)

Machine Learning algorithms are used to identify bug patterns and improve the accuracy of code analysis. The ML models examine code features such as syntax, variables, functions, loops, and complexity to classify programming errors and security issues. Based on trained models, the system predicts the severity of detected bugs and continuously improves its performance by learning from previous code review results.

C. Static Code Analysis

Static Code Analysis examines the source code without executing the program. It detects syntax errors, duplicate code, coding standard violations, memory-related issues, and poor programming practices during the early stages of development. This technique helps developers identify defects before deployment, reducing debugging time and improving code quality, maintainability, and overall software performance.

D. Security Vulnerability Detection

Security Vulnerability Detection is used to identify common software security risks within the source code. The system scans for vulnerabilities such as SQL Injection, NoSQL Injection, Cross-Site Scripting (XSS), hardcoded passwords, insecure API usage, weak authentication methods, and improper input validation. The AI engine assigns severity levels to each vulnerability and provides appropriate security recommendations, helping developers build secure, reliable, and high-quality software applications.

VI. METHODOLOGY

The methodology of the AI-Powered Real-Time Code Review and Bug Detection System Using MERN Stack and Python follows a systematic approach to automate the software code review process using Artificial Intelligence and Machine Learning techniques. The process begins with user registration and secure authentication using JWT, allowing only authorized users to access the application. [2]After successful login, the user can upload source code files or connect a GitHub repository through the React.js frontend. The Node.js and Express.js backend receives the request, validates the uploaded files, and securely transfers the source code to the Python AI Engine through REST APIs. The AI engine preprocesses the source code by parsing its structure and extracting important programming features such as variables, functions, classes, loops, and coding patterns. Static code analysis techniques are then applied to identify syntax errors, duplicate code, coding standard violations, and code complexity. Machine Learning algorithms further analyze the source code to detect logical bugs, runtime issues, and potential security vulnerabilities. The system also performs security analysis to identify risks such as SQL Injection, Cross-Site Scripting (XSS), hardcoded credentials, insecure API usage, and weak authentication mechanisms.[8] Based on the analysis, the AI engine generates a code quality score, optimization suggestions, and best practice recommendations. All review results, detected bugs, and generated reports are stored securely in the MongoDB database. Using Socket.IO, the backend sends real-time notifications and analysis results to the React.js dashboard, where users can view bug reports, security findings, AI recommendations, and downloadable reports.[11] This methodology ensures accurate code analysis, reduces manual review effort, improves software quality, enhances security, and increases developer productivity while providing a scalable and efficient platform for modern software development.

A. INPUT

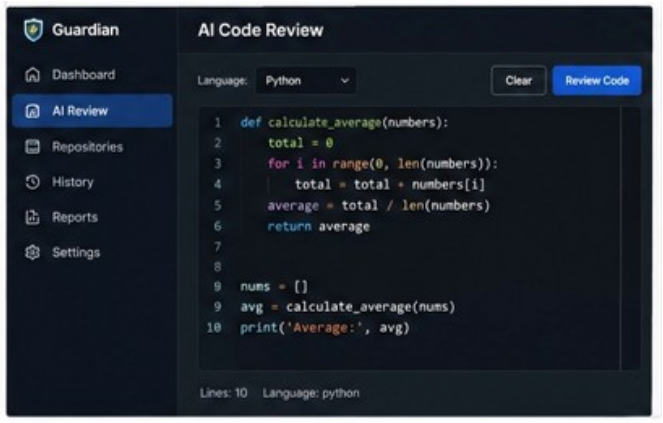


Figure 2: Code With Issue

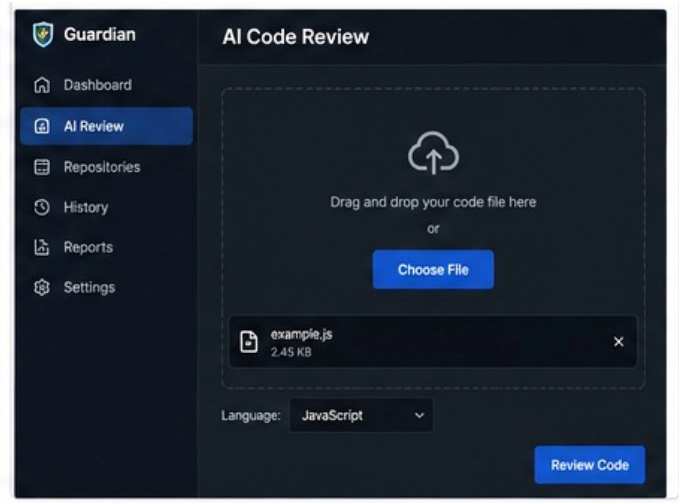


Figure 3:Upload File

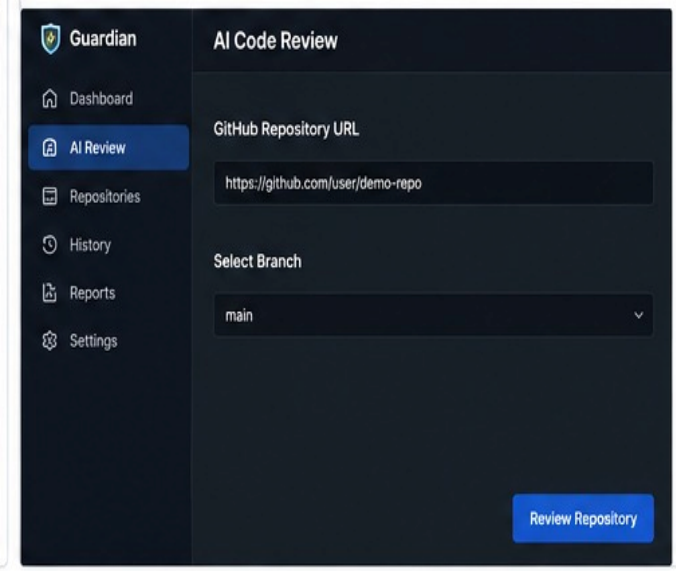


Figure 4: GitHub Repository

B. OUTPUT:

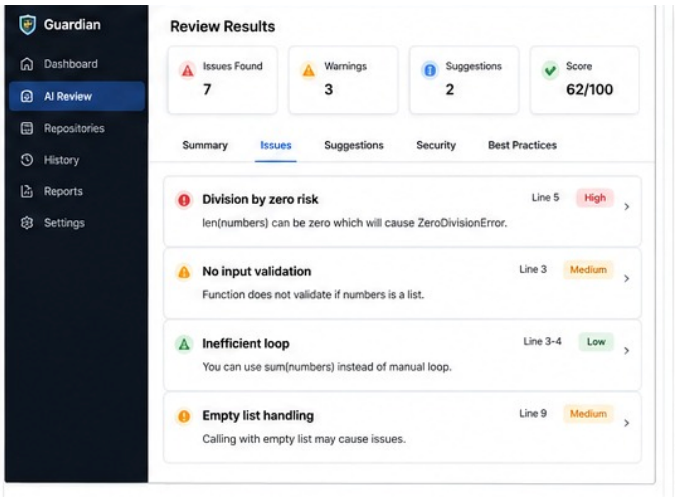


Figure 5: AI Review Results (Code)

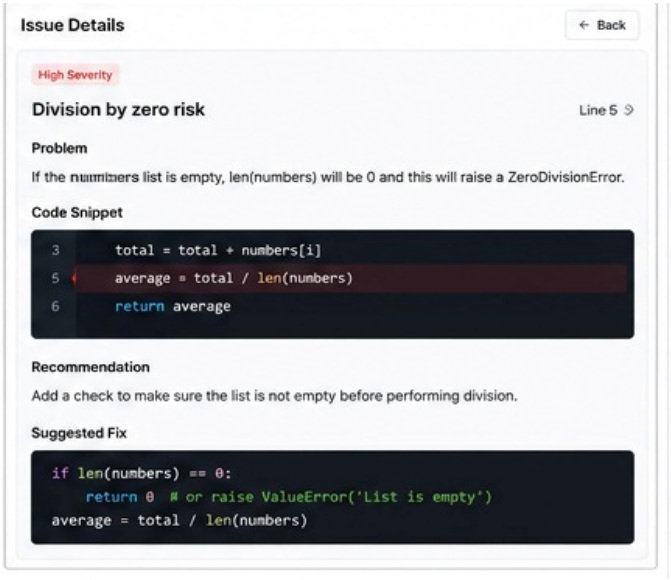


Figure 6: Detailed Issue View

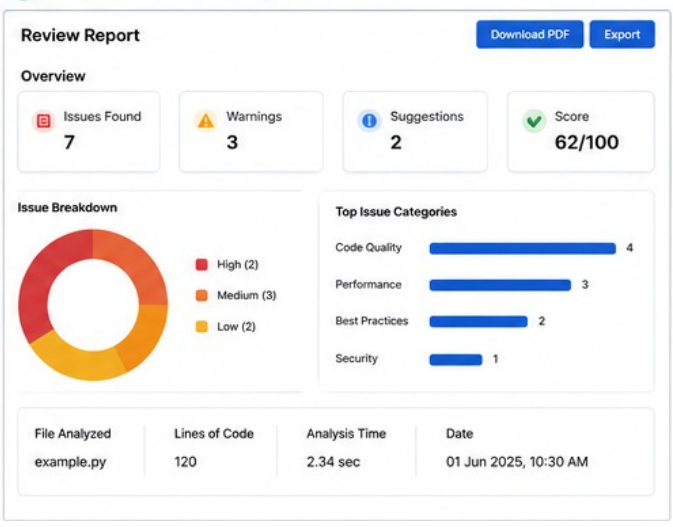


Figure 7: Review Summary Report

VII. CONCLUSION

The AI-Powered Real-Time Code Review and Bug Detection System Using MERN Stack and Python successfully demonstrates how Artificial Intelligence can be integrated with modern web technologies to automate the software code review process. The proposed system provides an efficient, reliable, and intelligent platform for analyzing source code, detecting programming bugs, identifying security vulnerabilities, and generating meaningful recommendations for improving software quality. By combining the MERN Stack with a Python-based AI engine, the application delivers real-time code analysis and instant feedback, helping developers identify and resolve issues during the early stages of software development.

The system integrates several important modules, including User Authentication, Code Upload, AI Analysis Engine, Bug Detection, Security Vulnerability Detection, Code Review Engine, Report Generation, and Dashboard Analytics. These modules work together to provide a complete software quality assurance solution. The use of Machine Learning and Static Code Analysis enables the system to evaluate coding standards, detect syntax and logical errors, measure code complexity, and generate optimization suggestions that improve software maintainability and performance.

One of the major strengths of the proposed system is its ability to reduce the time and effort required for manual code reviews. Instead of relying completely on experienced developers, the AI engine performs automated analysis with high speed and accuracy. The integration of GitHub repositories, REST APIs, JWT authentication, MongoDB, and Socket.IO ensures secure communication, efficient data management, and real-time notifications. The interactive dashboard provides developers with detailed bug reports, security analysis, code quality scores, and AI-powered recommendations, making the review process more user-friendly and effective.

The proposed system also promotes secure software development by identifying common vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), insecure authentication mechanisms, hardcoded credentials, and improper input validation. Early detection of these issues helps developers build reliable applications while minimizing security risks and reducing maintenance costs. The scalable architecture of the system allows it to support multiple users, large software projects, and future enhancements without significant performance degradation.

Overall, the AI-Powered Real-Time Code Review and Bug Detection System Using MERN Stack and Python provides a modern, scalable, and intelligent solution for improving software quality and developer productivity. With future enhancements such as support for additional programming languages, advanced Deep Learning models, IDE integration, CI/CD pipeline automation, cloud deployment, predictive bug analysis, and continuous AI learning, the system can evolve into a comprehensive software quality assurance platform that meets the growing demands of the software industry. The successful implementation of this project proves that Artificial Intelligence has the potential to transform traditional code review practices into a faster, smarter, and more efficient development process.

REFERENCES

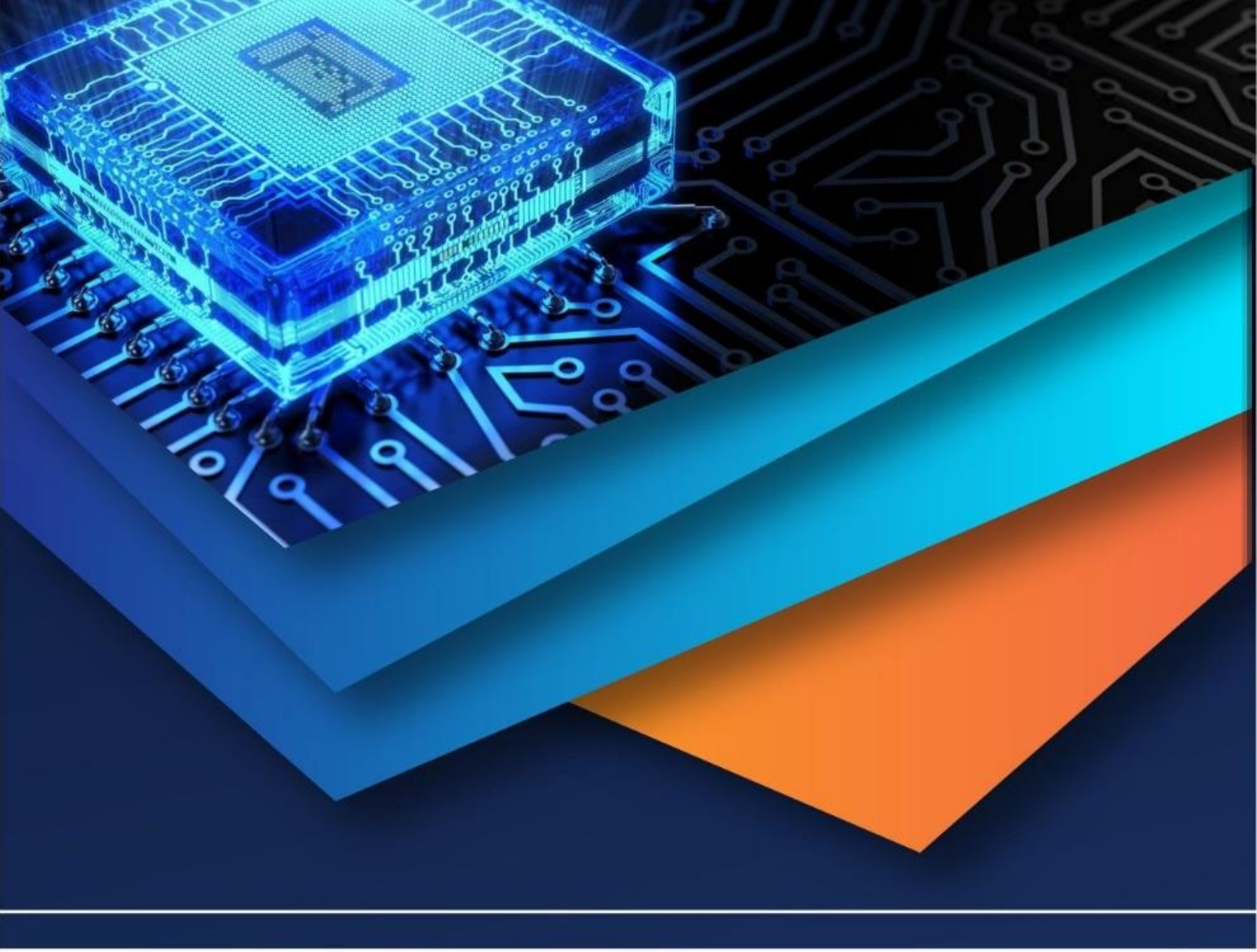
[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. https://www.deeplearningbook.org
[2] Russell, S., & Norvig, P. (2021). Artificial Intelligence: A Modern Approach (4th ed.). Pearson. https://aima.cs.berkeley.edu
[3] Ayewah, N., Pugh, W., Morgenthaler, J. D., Penix, J., & Zhou, Y. (2008). Using Static Analysis to Find Bugs. IEEE Software. https://ieeexplore.ieee.org
[4] OWASP Foundation. (2024). OWASP Top 10: Web Application Security Risks. https://owasp.org/www-project-top-ten
[5] Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall. https://www.pearson.com
[6] React Team. (2024). React Documentation. https://react.dev
[7] Node.js Foundation. (2024). Node.js Documentation. https://nodejs.org/docs
[8] Express.js Contributors. (2024). Express.js Documentation. https://expressjs.com
[9] MongoDB Inc. (2024). MongoDB Documentation. https://www.mongodb.com/docs
[10] Python Software Foundation. (2024). Python Documentation. https://docs.python.org/3
[11] FastAPI Contributors. (2024). FastAPI Documentation. https://fastapi.tiangolo.com
[12] Socket.IO Team. (2024). Socket.IO Documentation. https://socket.io/docs
[13] GitHub Inc. (2024). GitHub REST API Documentation. https://docs.github.com/en/rest
[14] JWT.IO. (2024). JSON Web Token (JWT) Documentation. https://jwt.io
[15] Docker Inc. (2024). Docker Documentation. https://docs.docker.com
[16] Nginx Inc. (2024). NGINX Documentation. https://nginx.org/en/docs

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

BIBLOGRAPHY

Rongala Rajesh is serving as an Assistant Professor at Sanketika Vidya Parishad Engineering College. He has one year of teaching experience in the field of Computer Science and Engineering. He is committed to academic excellence and is actively involved in guiding and mentoring students in their academic and project work. His areas of interest include Machine Learning and emerging technologies, and he continuously strives to enhance students' technical skills and research capabilities through effective teaching and practical exposure. Undertaken

Pedanayuni AmalakumariIs Currently Pursuing Her Final Semester OfMaster OfComputer Applications (MCA) At Sanketika Vidya Parishad Engineering College, Which Is Accredited With An 'A' Grade By NAAC, Affiliated ToAndhra University, And Approved By AICTE. With A Keen Interest InMachine Learning AndArtificial Intelligence, She Has Undertaken Her Postgraduate Project Titled "**AI-Powered Real-Time Code Review and Bug Detection System Using MERN Stack and Python**" The Project Has Been Successfully Carried Out Under The Guidance OfMr. Rongala Rajesh, Assistant Professor, SVPEC.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY