Journal of Computer and Electrical Sciences (JCES)- Volume-1 Issue-4, April 2026
ISSN: 3049-2602

AI-POWERED REAL-TIME CODE REVIEW AND SECURITY SCANNER

Y. Bhargav Kumar[1], Mrs. V. Rama Lakshmi[2]
1 - Student, Department of Computer Science and Engineering Andhra Loyola Institute of Engineering and Technology, Vijayawada, Andhra Pradesh, India
2 - Assistant Professor, Department of Computer Science and Engineering Andhra Loyola Institute of Engineering and Technology, Vijayawada, Andhra Pradesh, India
Email id: [22hp1a0539@gmail.com]

ABSTRACT:

The rapid growth of software development has led to increasingly complex applications involving large codebases and multiple developers. Ensuring code quality, maintainability, and security has become a major challenge. Manual code reviews are time-consuming and may fail to detect critical vulnerabilities.

This project presents an **AI-Powered Real-Time Code Review & Security Scanner**, which automates the process of analyzing source code repositories. The system uses a multi-agent architecture consisting of modules such as repository fetcher, parser agent, linter agent, metrics agent, decision agent, and reporter agent.

The proposed system performs static code analysis to detect coding standard violations, security vulnerabilities, and complexity issues. It integrates with GitHub repositories and generates structured reports using SARIF format. A user-friendly interface built with Streamlit allows developers to easily upload repositories and view results.

This system improves software quality, reduces manual effort, and helps detect issues early in the development lifecycle.

Keywords:

AI Code Review, Static Code Analysis, Security Scanner, Multi-Agent System, Software Quality, GitHub Integration, Streamlit, SARIF

1.INTRODUCTION

Software development has become highly complex due to large-scale applications and collaborative development environments. Ensuring code quality and security is a major concern for developers.

Manual code review processes are time-consuming and may overlook critical issues such as vulnerabilities and coding errors. Although static analysis tools exist, they often rely on rule-based systems and may produce inaccurate results.

https://doi.org/10.5281/zenodo.19606418

The AI-Powered Code Review & Security Scanner is designed to automate code analysis using intelligent agents. It scans source code repositories, detects issues, and provides structured reports. The system improves efficiency and accuracy while reducing human effort.

2. LITERATURE REVIEW

Literature Review

Existing systems for code review include:

- Manual code review processes
- Static analysis tools like Pylint, ESLint, and SonarQube
- Automated code review tools integrated with CI/CD pipelines

These systems help improve code quality but have limitations such as:

- High time consumption
- False positives
- Limited detection of complex issues

Base Paper Differentiation

The proposed system differs from existing approaches in the following ways:

- Uses **multi-agent architecture** instead of single-tool analysis
- Combines **code quality, security, and complexity analysis**
- **Provides** integrated reporting system
- **Supports** GitHub automation
- **Offers** user-friendly web interface

III. PROPOSED SYSTEM

Overview

The proposed system is an AI-powered automated tool that analyzes source code repositories and detects issues related to quality and security.

System Architecture

The system follows a **multi-agent architecture**, where each agent performs a specific task:

- Repository Fetcher
- Parser Agent
- Linter Agent
- Metrics Agent
- Decision Agent
- Reporter Agent

https://doi.org/10.5281/zenodo.19606418

Working Principle

1. User uploads repository
2. System fetches code files
3. Code is parsed and analyzed
4. Multiple agents detect issues
5. Results are combined
6. Report is generated

Advantages

- Reduces manual effort
- Improves accuracy
- Detects vulnerabilities early
- Provides structured reports

3.METHODOLOGY

➢ Step 1: Repository Collection

The system collects source code from GitHub or uploaded files.

➢ Step 2: Code Parsing

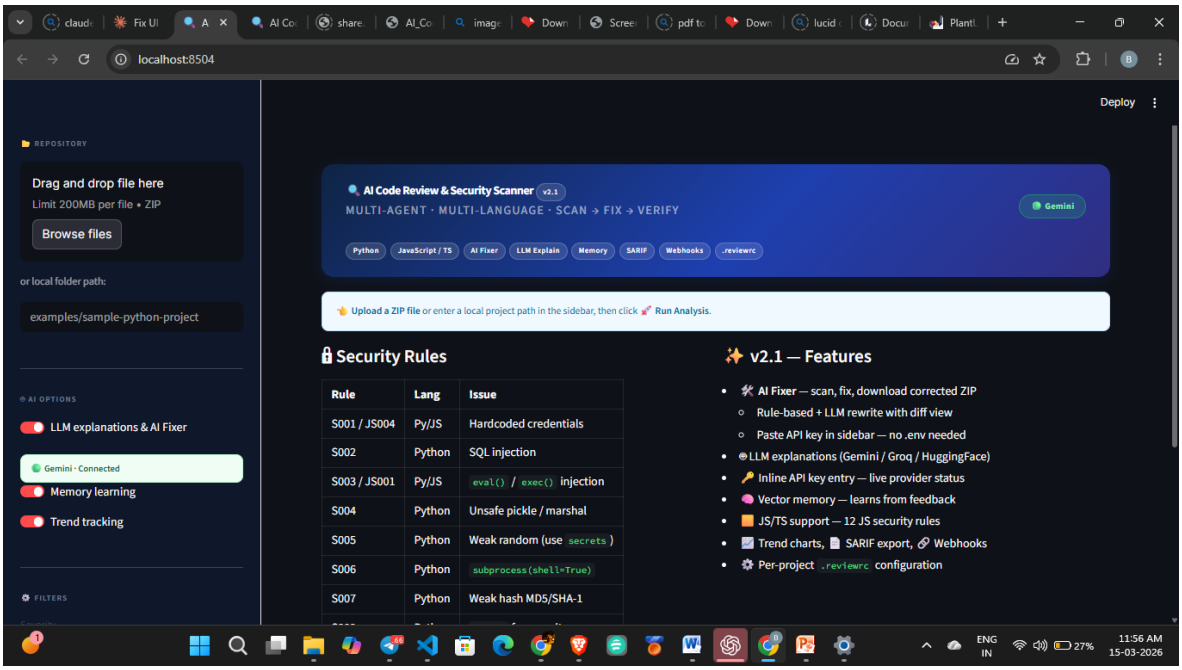
The parser agent analyzes code structure including functions, classes, and modules.

➢ Step 3: Code Analysis

Different agents perform:

- Linting (coding standards)
- Security checks
- Complexity analysis

https://doi.org/10.5281/zenodo.19606418



➢ Step 4: Decision Making

The decision agent combines results from all agents and evaluates overall code quality.

➢ Step 5: Report Generation

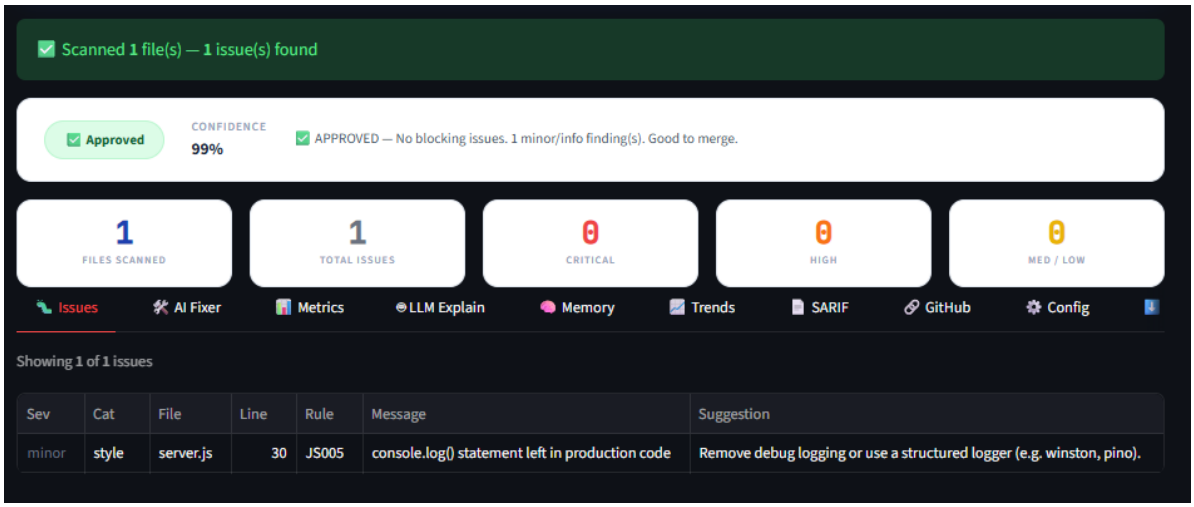
The reporter agent generates structured reports highlighting:

- Errors
- Warnings
- Vulnerabilities

➢ Step 6: Visualization

Results are displayed using a Streamlit web interface

https://doi.org/10.5281/zenodo.19606418



Reference

The following references were used during the development of the AI-Powered Code Review & Security Scanner project. These references include books, research papers, and online resources related to software engineering, static code analysis, and automated code review systems.

1. Ian Sommerville, *Software Engineering*, 10th Edition, Pearson Education, 2016.
2. Roger S. Pressman and Bruce R. Maxim, *Software Engineering: A Practitioner's Approach*, 8th Edition, McGraw-Hill Education, 2015.
3. Martin Fowler, *Refactoring: Improving the Design of Existing Code*, Addison-Wesley Professional, 2018.
4. Robert C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*, Prentice Hall, 2008.
5. M. Howard and D. LeBlanc, *Writing Secure Code*, Microsoft Press, 2003.
6. Google Developers, "Code Review Developer Guide," Available: https://google.github.io/eng-practices/review/
7. SonarSource Documentation, "Static Code Analysis Concepts," Available: https://www.sonarsource.com/
8. Python Software Foundation, "Python Documentation," Available: https://docs.python.org/3/
9. Streamlit Documentation, "Streamlit – The fastest way to build data apps in Python," Available: https://streamlit.io/
10. GitHub Documentation, "GitHub Webhooks and API Documentation," Available: https://docs.github.com/
11. OASIS Standard, "Static Analysis Results Interchange Format (SARIF)," Available: https://sarifweb.azurewebsites.net/

https://doi.org/10.5281/zenodo.19606418