AI CODE REVIEWER - Project Report

Introduction:

In today's fast-paced development environment, writing clean, efficient, and maintainable code is critical. Manual code reviews are time-consuming and often subjective. The **AI Code Reviewer** is a beginner-friendly, intelligent Streamlit application that automates code quality analysis using modern Python tools and artificial intelligence techniques. It provides real-time feedback, optimization tips, and intelligent insights, making it a powerful assistant for developers and students.

Abstract:

The AI Code Reviewer is an interactive web application built using Python and Streamlit. It uses static analysis tools like **flake8**, **black**, and **radon** to assess code style, formatting, and complexity. It also includes 14 advanced AI-powered features such as bug detection, missing logic detection, unit test generation, and impact forecasting. Users paste their code into the app and receive instant, actionable feedback with visual insights and simulated AI refactoring suggestions.

Tools Used:

- **Programming Language:** Python 3.10+
- **Framework:** Streamlit
- **Static Analysis Tools:** flake8, black, radon
- **Visualization Libraries:** Plotly, matplotlib
- **AI/NLP:** Custom logic using transformers-style reasoning and keyword-based heuristics

---

Steps Involved in Building the Project:

1. **Environment Setup**: Created a virtual environment and installed required packages listed in requirements.txt.
2. **Project Structure**:
    - app.py: Main Streamlit UI.
    - analyzer.py: Runs flake8, black, and complexity analysis.
    - modules/: Contains 14 feature modules like bug_scanner.py, code_mood.py, refactor_simulator.py, etc.

3. **Feature Integration**:
    - Each module performs a unique analysis (e.g., missing logic, test generation, code intent prediction).
    - Visuals are rendered via Plotly for interactivity.
    - AI-style messages simulate thought processes of a human reviewer.
4. **UI Design**:
    - Custom dark theme with orange highlight using Streamlit CSS injection.
    - Tabs and sections for organizing features.
5. **Testing**: Ran various test codes in the app to verify outputs and performance of each feature.

Conclusion:

The **AI Code Reviewer** project demonstrates how Python tools can be integrated into a user-friendly AI assistant to help developers write better code. It merges static analysis with intelligent, AI-like insights to make code reviews more interactive and instructive. The project is scalable, open to contributions, and provides a strong foundation for educational or enterprise-level code quality tools.

Submitted by:

AADI HARSHA VARDHAN

harshavardhanaadi464@gmail.com