AI-POWERED CODE REVIWER AND QUALITY ASSISTANT

BY Sujana Guchhait

Code review using AI and static analysis

01

PROBLEM STATEMENT

Manual code reviews are time-consuming and inconsistent, especially in rapidly evolving codebases. Maintaining code quality and documentation standards becomes difficult as projects grow. This project proposes an AI-powered system to automatically review Python code and provide meaningful feedback to improve code quality and efficiency.

02

PROJECT OBJECTIVES

- Automatically analyze Python source code
- Detect code smells, missing docstrings, and quality issues
- Generate human-readable suggestions using AI
- Provide CLI and optional Streamlit web interface
- Integrate with Git pre-commit hooks and CI/CD pipelines

03

SYSTEM ARCHITECTURE

The system architecture consists of a source code parser that performs AST-based static analysis to extract code structure, an AI review engine that generates human-readable feedback, and a validation and reporting module that evaluates code quality metrics. The system provides user interaction through a command-line interface and an optional Streamlit web dashboard, while seamless integration with Git pre-commit hooks and CI/CD pipelines ensures automated and consistent code quality enforcement throughout the development workflow.

04

SYSTEM WORKFLOW

- Developer submits Python source code
- Code is parsed to extract functions, classes, and imports
- AI engine generates review feedback and suggestions
- Validation checks are applied to ensure quality
- Results are displayed via CLI or Streamlit UI
- Reports are generated and enforced in CI pipelines

05

MILESTONE 1 IMPLEMENTATION

Parsing & Baseline Generation

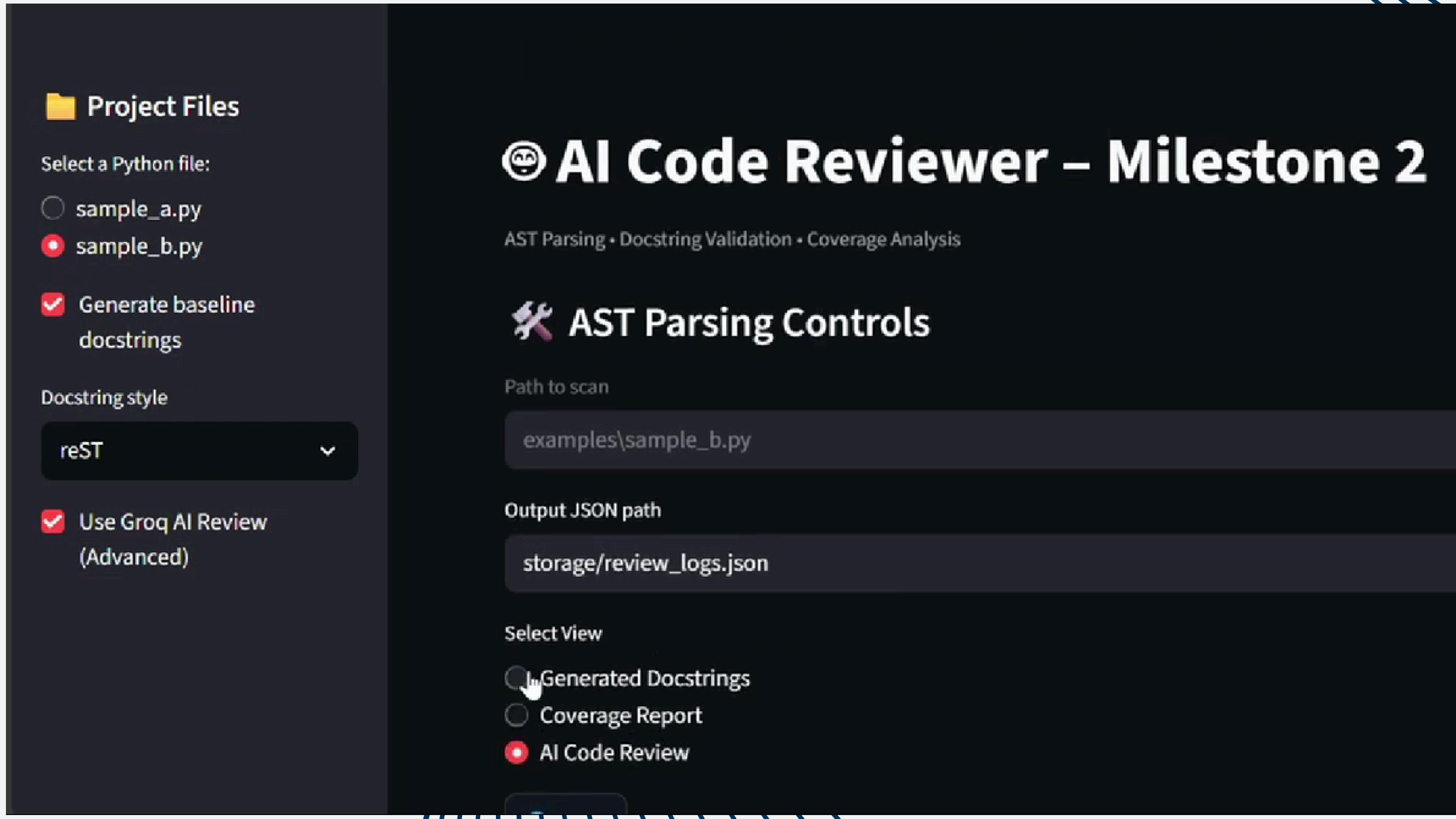
- AST-based extraction of code structure
- Automatic baseline docstring generation
- Docstring coverage report generation

06

MILESTONE 2 IMPLEMENTATION

Synthesis & Validation

- Support for Google, NumPy, and reST docstring styles
- PEP 257 compliant docstring generation
- Validation using pydocstyle checks



Project Files

Select a Python file:

○ sample_a.py
● sample_b.py

☑ Generate baseline docstrings

Docstring style: reST

☑ Use Groq AI Review (Advanced)

AI Code Reviewer – Milestone 2

AST Parsing • Docstring Validation • Coverage Analysis

AST Parsing Controls

Path to scan: examples\sample_b.py

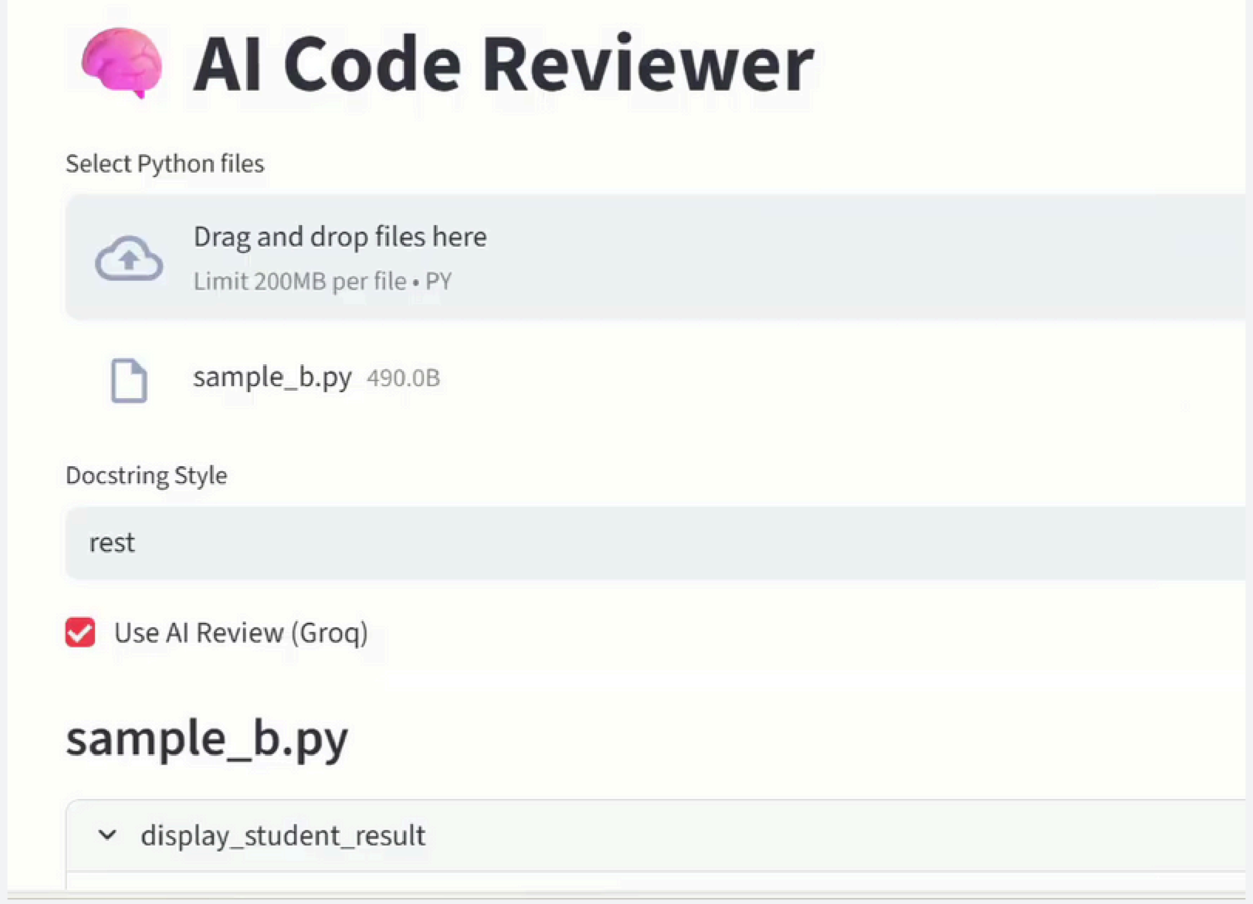
Output JSON path: storage/review_logs.json

Select View

○ Generated Docstrings
○ Coverage Report
● AI Code Review

07

MILESTONE 3 IMPLEMENTATION


AI Code Reviewer

Select Python files

Drag and drop files here
Limit 200MB per file • PY

sample_b.py 490.0B

Docstring Style

rest

☑ Use AI Review (Groq)

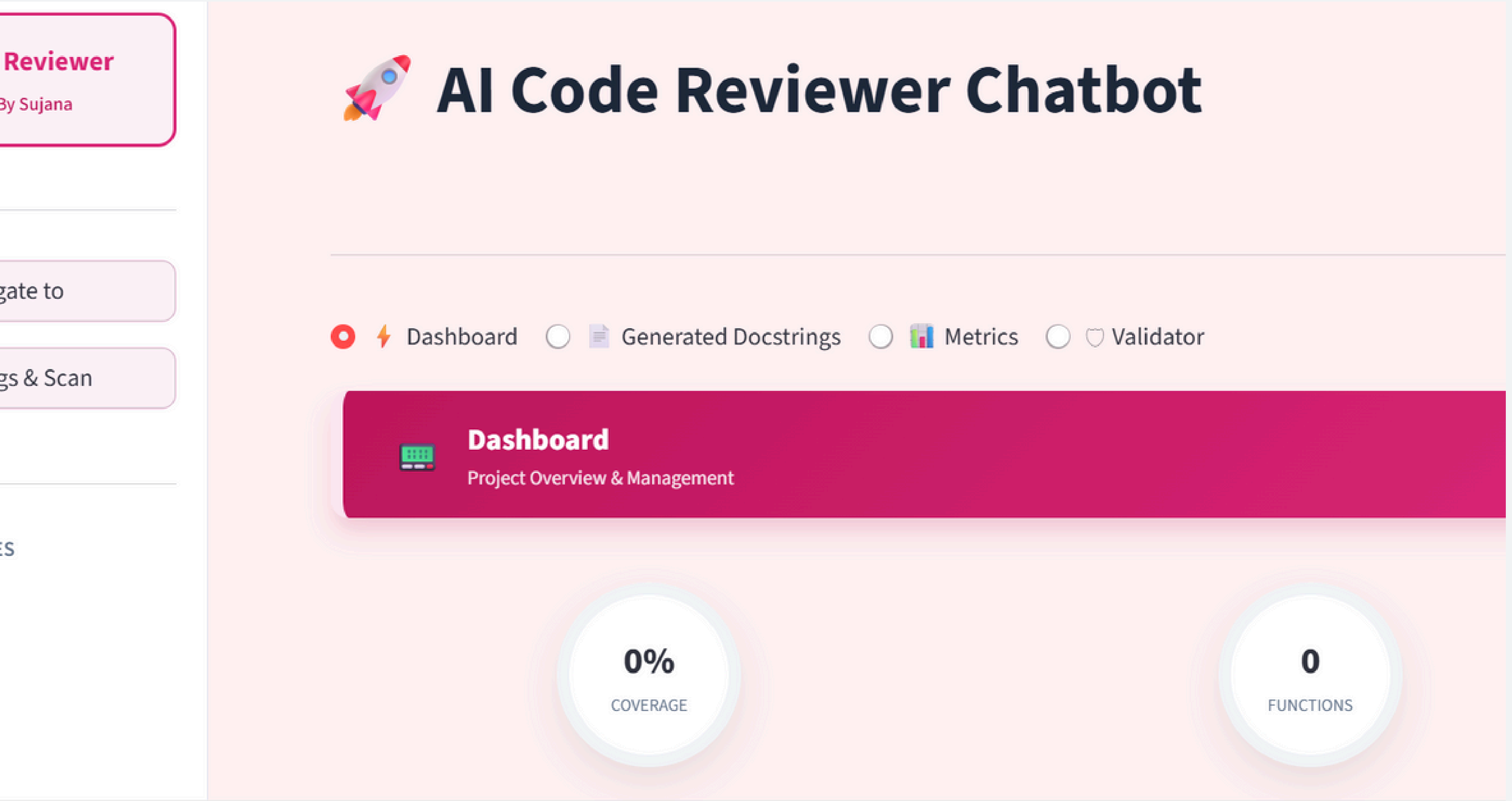
sample_b.py

display_student_result

Workflow & CI Integration

- Git pre-commit hook for automatic review
- CI pipeline with quality threshold enforcement
- Configuration through pyproject.toml
- Streamlit UI for previewing and managing suggestions

MILESTONE 4 IMPLEMENTATION



🚀 AI Code Reviewer Chatbot

Reviewer
By Sujana
gate to
gs & Scan
ES
● Dashboard ○ Generated Docstrings ○ Metrics ○ Validator
Dashboard
Project Overview & Management
0%
COVERAGE
0
FUNCTIONS

Packaging & Finalization

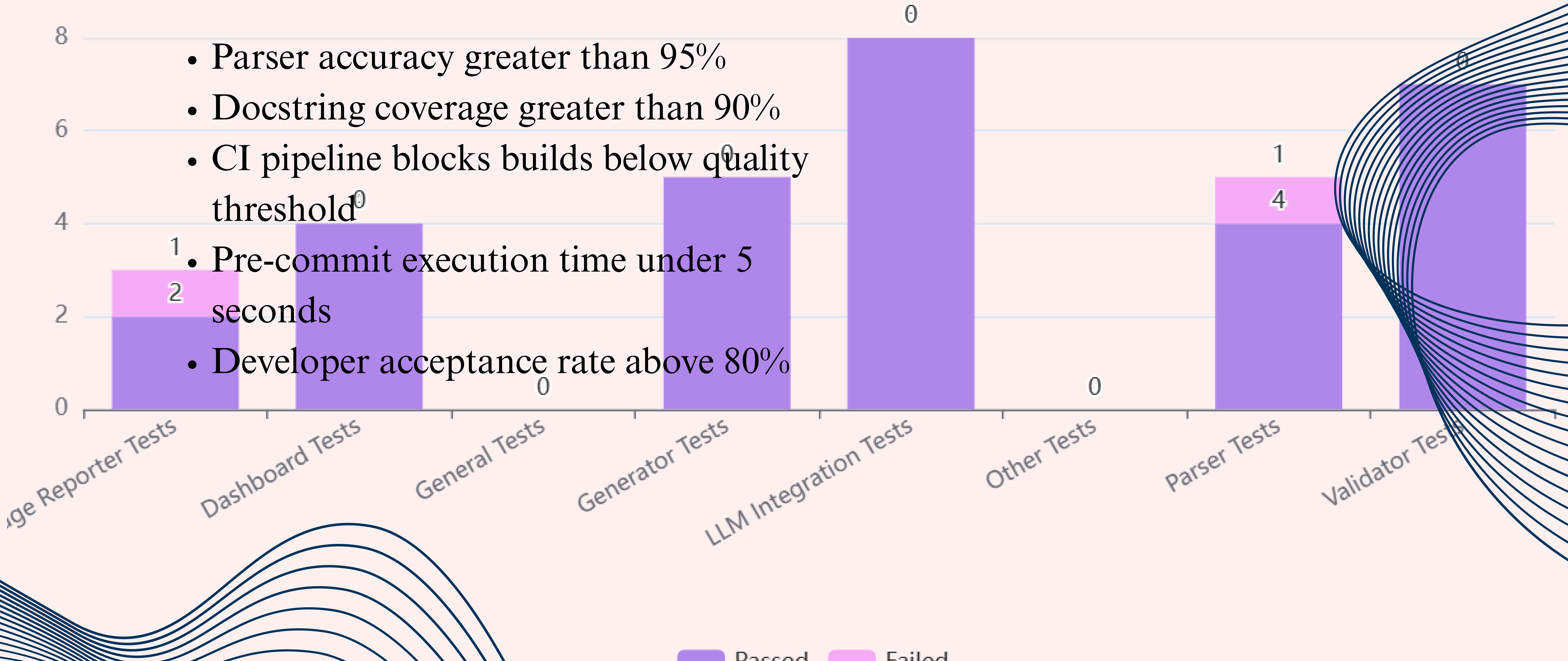
- Tool packaged as a pip-installable Python library
- Robust unit and integration testing implemented
- Enhanced Streamlit UI with filters, search, and tooltips
- Complete documentation and usage guides published

› Detailed Console Output

📊 Tests by Category

RESULTS & EVALUATION

- Parser accuracy greater than 95%
- Docstring coverage greater than 90%
- CI pipeline blocks builds below quality threshold
- Pre-commit execution time under 5 seconds
- Developer acceptance rate above 80%



CONCLUSION

This project successfully implements an AI-powered automated code review system that improves code quality, consistency, and developer productivity while reducing manual review effort. In the future, the system can be enhanced by supporting multiple programming languages, providing AI-based auto-fix suggestions, enabling a cloud-based dashboard, and integrating directly with GitHub pull request workflows.

THANK YOU

Sujana Guchhait