CAPSTONE PROJECT PHASE-1 (UQ24CA741A)

AUTOMATED PYTHON CODE SCORING WITH KNOWLEDGE SUPPORT

By Siddharth Puhan - PES1PG24CA171

UNDER THE GUIDANCE OF
Dr. S. Thenmozhi
Professor, Department of Computer Applications

ABSTRACT

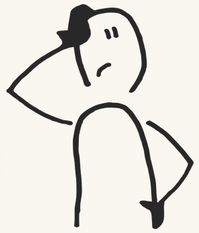
- This project focuses on building a system that evaluates Python programs automatically.

- Users can paste or upload their code and select the criteria on which they want it scored, such as **time complexity, space complexity, number of lines of code, or readability.**

- To enhance feedback, the system includes a **knowledge base of optimal implementations** for common problems, enabling direct comparison between user submissions and best-practice solutions.

- This project aims to support both **automated evaluation and learning through feedback**, providing detailed explanations alongside scores, ensuring transparency and learning.

PROBLEM SCENARIO



- **Code Understanding:** Beginners and students often write Python code without a clear understanding of efficiency, readability, or adherence to coding best practices.

- **Time-consuming evaluation:** Manual code evaluation is time-consuming, subjective, and prone to error, and often focuses more on correcting the given code rather than focusing on more efficient approaches to the same problem.

- **Learning through Feedback:** A system that can automatically analyze Python code, assign scores, and provide meaningful feedback based on both computational metrics and a knowledge base of optimal solutions will help learners improve their coding skills more effectively.

DOMAIN

PES UNIVERSITY
CELEBRATING 50 YEARS

Artificial Intelligence & Software Engineering

focus on creating tools that can **automatically evaluate and improve code quality,** reducing manual effort while maintaining accuracy

Static Code Analysis

analyzing source code without executing it. By parsing code, we can detect metrics like complexity, efficiency, and adherence to coding standards

Educational Technology

automated scoring and knowledge-base feedback act as an **intelligent tutor**, guiding students towards better coding practices

Knowledge-Based Systems

by comparing user submissions with stored optimal solutions, we can explain **why a solution is better**, not just assign a score

FUNCTIONALITIES

Input

The user can paste their code or upload the .py file

Code Evaluation

Based on the evaluation criteria, the code will be evaluated using different python libraries

Error Handling

Syntactically Invalid Code: Return clear error message with feedback.
Code Not in Knowledge Base: Fallback to metric-based scoring only.

Criteria Selection

Select the criteria on which the code should be evaluated like time or space complexity, lines of code, readability and style, cyclomatic complexity, etc.

Knowledge Base

Stores optimal implementations of common problems along with expected complexity, space usage, and sample solutions.
Used for comparing user code with best practices and generating tailored feedback.

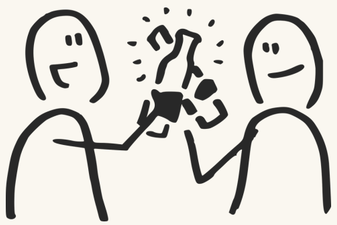
Output

Numeric scores for each chosen criterion along with feedback and suggestions

TOOLS AND TECHNOLOGIES

- **Python –** Backend language
- **AST (Abstract Syntax Tree) –** For parsing code structure
- **Radon / McCabe –** Measures code complexity and maintainability, and Cyclomatic complexity analysis
- **Pylint / Flake8 –** Style and readability checks. Checks code against PEP8 guidelines.
- **Streamlit –** Simple interactive frontend for taking input & displaying results
- **SQLite / JSON –** Knowledge base storage using JSON and SQLite for scalability

APPLICATIONS IN REAL WORLD



- **Educational Platforms:** Automated feedback for programming assignments or while learning coding in class.
- **Assignment Scoring:** Rather than giving absolute scores, programming assignments can be scored relatively
- **Coding Bootcamps:** Provide students with instant feedback & comparisons.
- **Competitive Programming:** Quick evaluation of efficiency before submission.
- **Software Engineering Training:** Reinforce best practices and optimal coding standards.

THANK YOU