İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

SOFTWARE DEVELOPMENT DESIGN AND PRACTICE PROJECT PROPOSAL

BY :

ID	Name	Surname
1306220015	Yusuf Şamil	Görmüş
1306210007	Baran	Uygun

PROJECT PROPOSAL: AUTOMATED CODE REVIEW SYSTEM

1) Project Overview: This project provides an automated code review system to improve code quality and accelerate error detection for developers. The system analyzes uploaded code files to identify errors, security vulnerabilities, style violations, and performance issues. The results are presented to the user in a detailed report.

2) Detailed Description of the Project:

2.1) Purpose

The purpose of the Automated Code Review System is to:

- Automatically review Java code to detect potential issues.
- Provide feedback on code quality, best practices, and security vulnerabilities.
- Allow developers to run the code review process locally, without relying on external API services, by using a pre-trained LLM model.
- Help developers improve their coding practices by providing consistent, automated feedback.

2.2) Project Scope

- Frontend: A user interface where developers can input their Java code for analysis and view the resulting feedback.
- Backend: A Java-based application (using Spring Boot) that handles the logic of receiving code from the frontend and forwarding it to the Python-based model server for analysis.
- Model Hosting: We decided to use a locally hosted LLM (DeepSeek or LLaMA) to analyze the code and provide suggestions. (Still haven’t decided yet, will provide further updates on the prefered LLM)
- Local Execution: Users will follow installation instructions to set up the LLM model on their machines, enabling them to run the system locally without API costs.

- Code Suggestions: The system will provide suggestions on code optimization, error checking, style improvements, and security fixes based on Java best practices.

3) Software Requirements Specifications (SRS):

3.1) User Input:

The user must be able to input Java code into the system via the frontend interface.

3.2) Code Review Process:

The backend will receive the code from the frontend, process it, and send it to the local model server for analysis. The system will return suggestions on code optimization, style, and security improvements.

3.3) Code Analysis and Feedback:

The LLM will analyze the provided Java code and provide feedback on:

- Code quality (e.g., variable naming, readability).
- Performance improvements (e.g., algorithmic optimizations).
- Security issues (e.g., SQL injection risks, improper exception handling).

3.4) Local Model Hosting:

The system will allow users to run the LLM model locally without requiring external APIs. Installation instructions for setting up the LLM model will be provided.

3.5) User Interface:

The frontend will allow the user to view the feedback in a clear, organized format. It will also display suggestions for improving the code.

3.6) Security and Privacy:

The system will ensure that all code is processed locally, meaning no external servers will store or analyze the user's code, preserving privacy.

4) Tech-Stacks:

4.1) Backend

- Programming Language: Java
  - Framework: Spring Boot – for creating RESTful APIs and managing backend logic.

- Libraries:
    - Spring Web – for handling HTTP requests.
    - Spring Boot Data

4.2) Model Hosting

- Programming Language: Python
  - Model: DeepSeek or LLaMA
  - Framework: Flask or FastAPI – for exposing the model as a local HTTP endpoint.
  - Library: Hugging Face Transformers – for loading the model and performing inference.

4.3) Frontend

- Framework: React.js – for building the web interface that allows users to input Java code and view the results.
  - Libraries:
    - Axios – for making HTTP requests to the backend.
    - React Hooks – for managing state and rendering dynamic content.

4.4) Database

- Database: PostgreSQL or MongoDB (this is needed in case we need data persistency, we can come up with an early version of the project that doesn't handle data persistency, and later update the project)

4.5) Deployment and Containerization

- Containerization: Docker – to package the backend, frontend, and Python server into a containerized environment, making it easier to deploy and run on different systems.
- Deployment: AWS

5) URL to GitHub repo

https://github.com/YusufSamilGormus/Automated-Code-Review-System