

Overview

The Code Review SaaS service is designed to automate code reviews using a Large Language Model (LLM)-based AI agent. The service takes a GitHub pull request (PR) URL as input and generates review comments, which are then posted back to GitHub.

System Components

1. **Frontend**
 - Web-based interface for submitting PRs and viewing results.
 - Dashboard for managing reviews and configurations.
2. **Backend API**
 - Handles PR submission and processing.
 - Interfaces with the LLM model for code analysis.
 - Communicates with GitHub API to fetch PR details and post comments.
3. **LLM Review Engine**
 - Uses an LLM model to analyze code changes.
 - Evaluates code quality, best practices, and potential bugs.
 - Generates structured review comments.
4. **Database**
 - Stores metadata about reviews, users, and historical data.
 - Tracks review feedback for improving recommendations.
5. **Job Queue & Worker**
 - Manages asynchronous processing of PRs.
 - Ensures scalability for handling multiple requests concurrently.
6. **GitHub API Integration**
 - Retrieves PR information (changed files, commit history, author details).
 - Posts generated comments back to the PR.

Architecture Diagram

A high-level diagram illustrating the architecture:

User -> Frontend -> Backend API -> Job Queue -> Worker -> LLM Review Engine -> GitHub API -> PR Comments

Workflow

1. User submits a PR URL via the frontend.
2. Backend fetches PR details from GitHub.
3. The PR is added to the job queue for processing.

4. Worker fetches the PR from the queue and sends relevant code snippets to the LLM.
5. LLM analyzes the code and generates review comments.
6. Worker posts the comments back to the PR via GitHub API.
7. User can view comments in the GitHub PR interface.

Technology Stack

- **Frontend:** React.js, Next.js, or similar.
- **Backend:** FastAPI.
- **LLM:** OpenAI API, Hugging Face models, or a custom-trained model.
- **Database:** PostgreSQL or MongoDB.
- **Queue System:** Celery with Redis.
- **Infrastructure:** Docker, Kubernetes for scalability.