

Code Review Assistant

Santhosh Kumar S P

Roll No: 22BCE3084

1. Abstract

The **Code Review Assistant** is an AI-driven web application that automates the process of reviewing programming code.

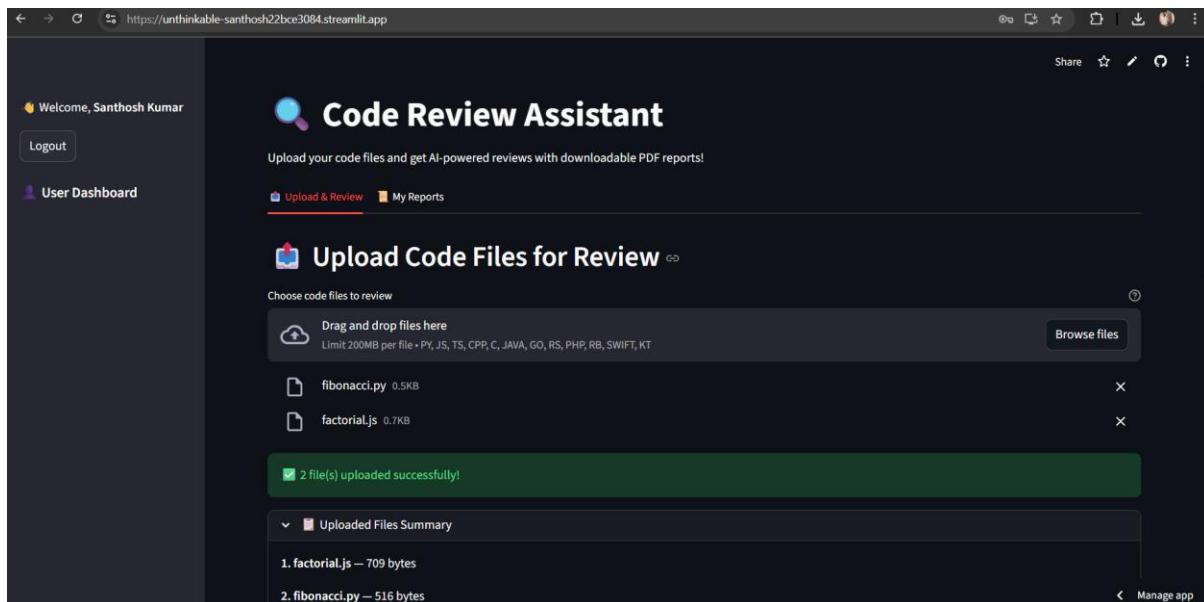
Built using **Streamlit**, **Python**, and **LLMs (Large Language Models)** via **OpenRouter**, this system enables users to upload code files, receive AI-generated feedback, visualize algorithmic complexity, and download structured PDF reports.

The project also implements **multi-user authentication**, where individual users can maintain private code reviews while administrators can oversee all user submissions.

Reports are persistently stored in an SQLite database with metadata tracking for audit and performance insights.

Live Application:

👉 <https://unthinkable-santhosh22bce3084.streamlit.app/>



2. Objectives

- Automate the **code review process** using AI/LLM models.
- Provide **detailed feedback** and improvement suggestions for uploaded source code.
- Allow **PDF report generation** with algorithmic complexity graphs.
- Enable **role-based access control** (Admin & User).
- Ensure **data persistence** for reports using SQLite.

- Deploy securely on the **Streamlit Cloud** platform for global access.

3. System Architecture

Core Components

Component	Description
Frontend	Streamlit-based web interface for code upload, review display, and PDF download.
Backend Logic	Python modules for LLM interaction, PDF generation, and database handling.
Database	SQLite for storing reports, metadata, and user associations.
AI Integration	OpenRouter API connected to qwen/qwen-2.5-coder-32b-instruct model.
Hosting	Streamlit Cloud for deployment and secure environment variable handling.

4. Technologies Used

Category	Tools / Libraries
Frontend	Streamlit
Backend	Python 3.12
AI / NLP	OpenRouter (Qwen Coder LLM)
Visualization	Matplotlib
PDF Generation	ReportLab
Database	SQLite
Environment Management	python-dotenv, Streamlit Secrets
Version Control	Git + GitHub
Hosting	Streamlit Cloud

5. Features

AI-Powered Code Review

- The LLM analyzes uploaded source code and generates feedback, suggestions, and optimization advice.
- Supports multiple languages — Python, C++, Java, JavaScript, etc.

Authentication & Roles

- Simple credential-based login system.
- **Admin:** View all reports from all users.
- **User:** Upload code, generate reports, and view their own review history.

PDF Generation

- Automatically compiles AI feedback and file data into a professional PDF report.
- Includes **time and space complexity graphs** for visual context.

Complexity Graphs

- Uses matplotlib to dynamically plot sample computational complexity growth.

Database Management

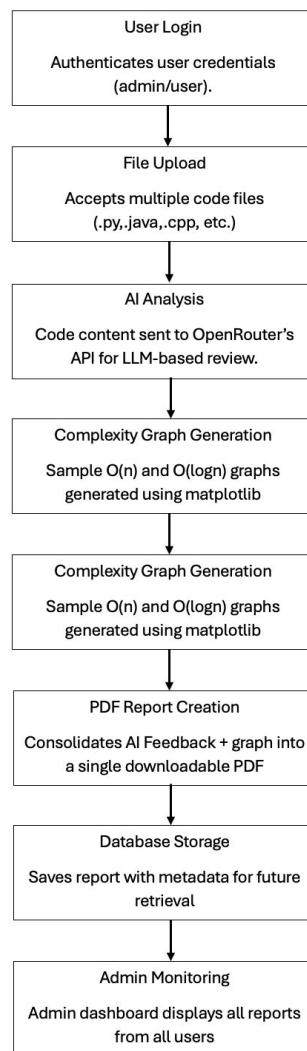
- SQLite-backed persistence layer stores:
 - Uploaded file metadata
 - Review text
 - PDF paths
 - Timestamped entries
 - User ownership

Cloud Deployment

- Hosted on Streamlit Cloud with API keys managed securely via **TOML-based Secrets Manager**.

6. System Flow (Summary)

1. **User Login** – Authenticates users (admin/user) and grants role-based access.
2. **File Upload** – Accepts multiple code files (.py, .java, .cpp, etc.) for review.
3. **AI Analysis** – Sends code to OpenRouter's API (Qwen 2.5 Coder) for automated LLM-based review.
4. **Complexity Graph Generation** – Creates visual time and space complexity graphs using matplotlib.
5. **PDF Report Creation** – Combines AI feedback and graphs into a structured downloadable PDF.
6. **Database Storage** – Saves reports with metadata and timestamps in SQLite for future access.
7. **Admin Monitoring** – Admin dashboard displays and manages reports from all users.



7. Deployment

Platform:

Streamlit Cloud — Deployed via GitHub repository

👉 https://github.com/isanthosh2004/unthinkable_submission

Live Application:

👉 <https://unthinkable-santhosh22bce3084.streamlit.app/>

Demo video

👉 https://drive.google.com/file/d/16rE-leC6FqorXCXd6Yesl_tUIO-rmYbj/view