

PROJECT REPORT

Title: Sustainable Smart City Assistant using Granite IBM Cloud

1. INTRODUCTION

1.1 Project Overview

SmartCityAI is an intelligent civic assistant powered by **Granite IBM Cloud** and **Generative AI**. It aims to improve urban living by providing citizens with quick, accurate, and personalized access to smart city services. The system enhances communication between citizens and government bodies through voice-enabled complaint registration, real-time environmental updates, and an AI-powered chatbot.

The purpose of SmartCityAI is to serve as a virtual civic companion, enabling citizens to report issues, access real-time environmental data, and receive AI-driven guidance on public services through a user-friendly interface.

2. IDEATION PHASE

2.1 Problem Statement

2.2 In today's world, citizens often face delays and confusion when accessing civic services or reporting issues. SmartCityAI addresses this by providing trusted, AI-powered urban assistance and real-time service interaction.Empathy Map Canvas

The empathy map helps understand the target users' mindset and expectations when interacting with a virtual civic assistant like **SmartCityAI**.

Category	Description
Says	"I need to report a civic issue quickly." "Who will respond to my complaint?"
Thinks	"Will the authorities take action?" "Is this platform reliable?"
Does	Tries to call helplines, visits government websites, posts complaints on social media

Feels	Frustrated by delays, unsure about where to report, hopeful for quick resolution
Pains	Slow or no responses, complex complaint systems, lack of transparency
Gains	Fast complaint registration, real-time updates, AI-based suggestions for better urban living

2.3 Brainstorming

The team explored various ideas such as AI-based waste management, smart traffic monitoring systems, and IoT-enabled pollution trackers before settling on an all-in-one intelligent civic assistant with voice interaction, complaint registration, and real-time environmental analytics powered by Granite IBM Cloud..

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The customer journey in the **SmartCityAI** application follows a seamless, accessible, and intelligent interaction flow:

1.Start Application – The user opens the SmartCityAI web interface (built using HTML, CSS, javascript,and IBM Granite backend).

2.Select Service Mode – The user chooses between voice input, chatbot, or map-based interaction for civic assistance.

3.Report or Query – The user submits a civic issue (e.g., garbage, pothole) or queries about services like water/electricity using AI-powered chat or speech.

4.Receive AI Response – The Granite AI model processes the request and provides a structured, context-aware response or service routing.

5.Access Environment Info – The user can view real-time air quality, temperature, and other environmental data through integrated APIs.

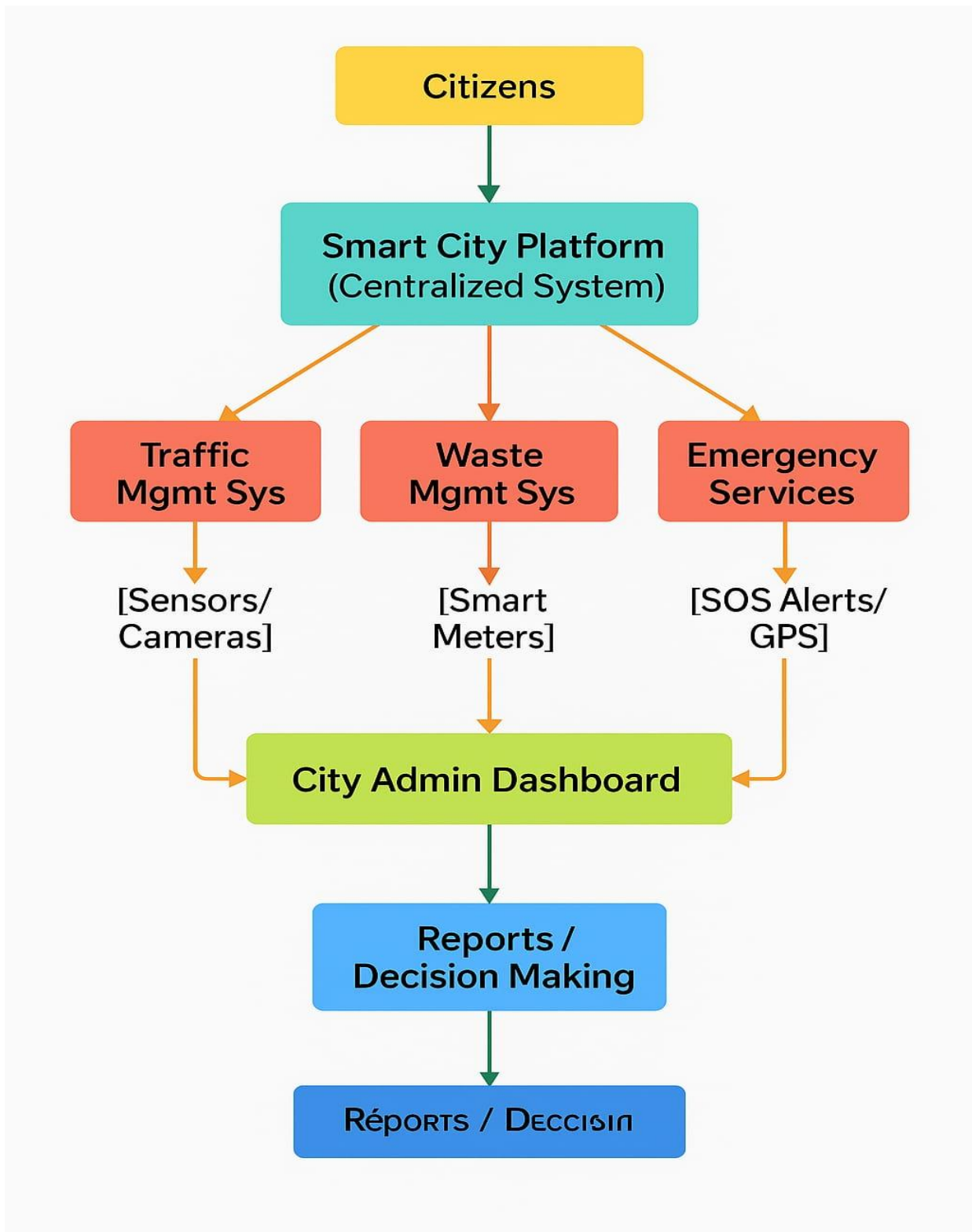
6.End/Next Action – The user can log a new complaint, share feedback, or explore sustainable living tips suggested by the assistant.

3.2

3.3 *Session Requirements*

- Real-time issue input via chat or voice
- Response generation based on complaint type and location

Data Flow Diagram:



3.4 Technology Stack

- **Frontend:** html,css,javascript
- **Backend:** Python

- **AI Service:** IBM granite Model
 - **Visualization:** Leaflet.js, chart.js
 - **Environment Management:** virtualenv
-

4. PROJECT DESIGN

4.1 Problem-Solution Fit

People need quick, understandable, and trustworthy civic support. SmartCityAI fulfills this by using AI foundation models for accurate and timely responses.

4.2 Proposed Solution

- A layered app with UI , core logic (Python functions), and AI service (IBM Granite). It guides users from issue reporting to intelligent civic responses and sustainability tips.
 - **UI Layer:** Voice/chat input, complaint forms, and interactive chart
 - **Application Logic:** main.py manages user interactions
-
- **AI Layer:** IBM Granite (3.3-2 b instruct model)

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Week Duration	Dates	Activities
Week 1	June 12 – June 19	Idea finalization, architecture planning, frontend UI with javascript
Week 2	June 20 – June 26	Backend AI integration, testing, debugging, and documentation

This two-week schedule allowed the team to focus on clear milestones and complete the HealthAI project within the planned timeline.

6. FUNCTIONAL AND PERFORMANCE TESTING

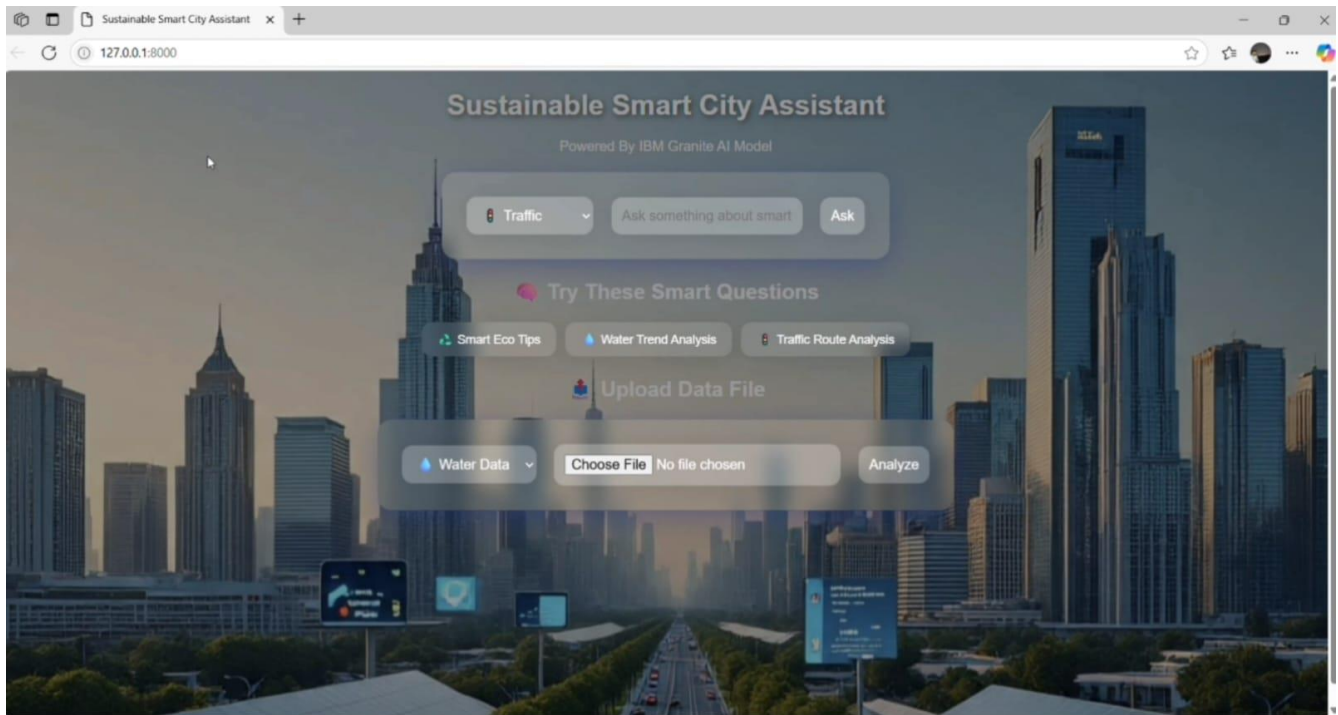
6.1 Performance Testing

- Unit Testing: Model init, text generation
 - Integration Testing: Chat to AI flow
 - Manual Testing: Verifying each feature with sample users
 - Error Handling: Invalid API or missing input cases handled
-

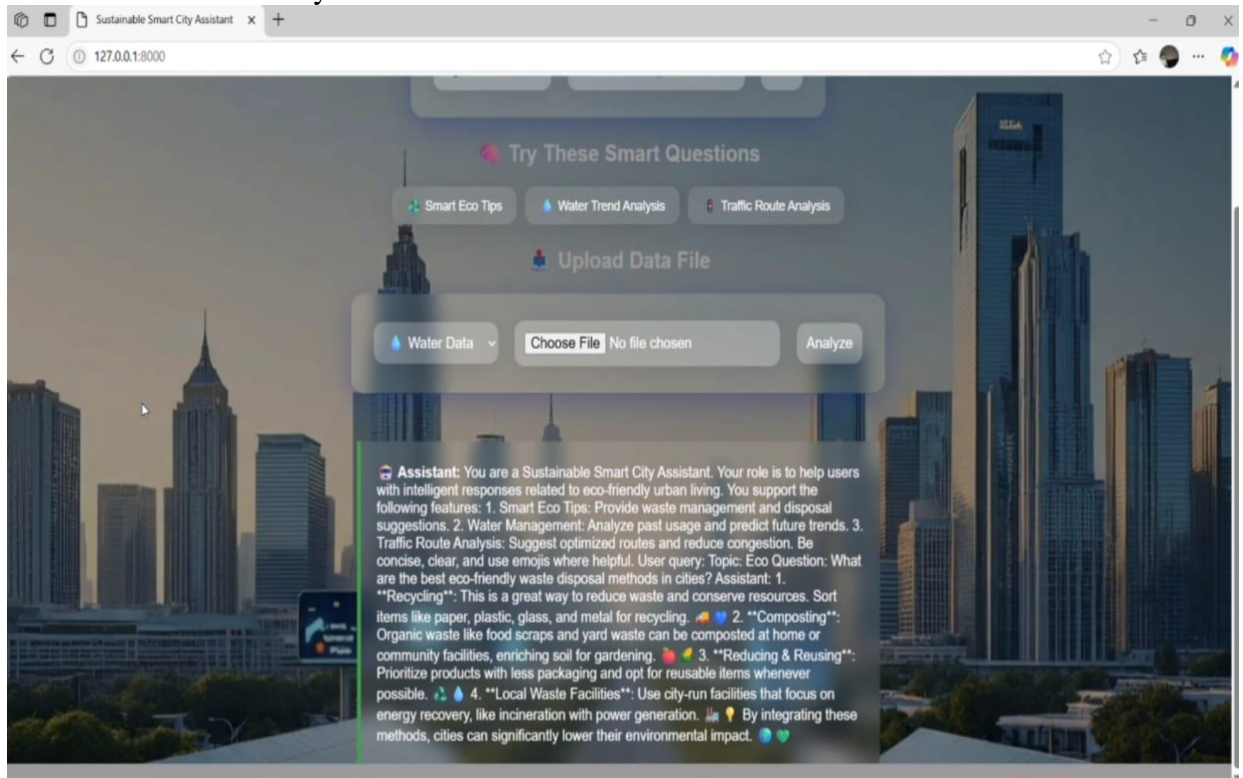
7. RESULTS

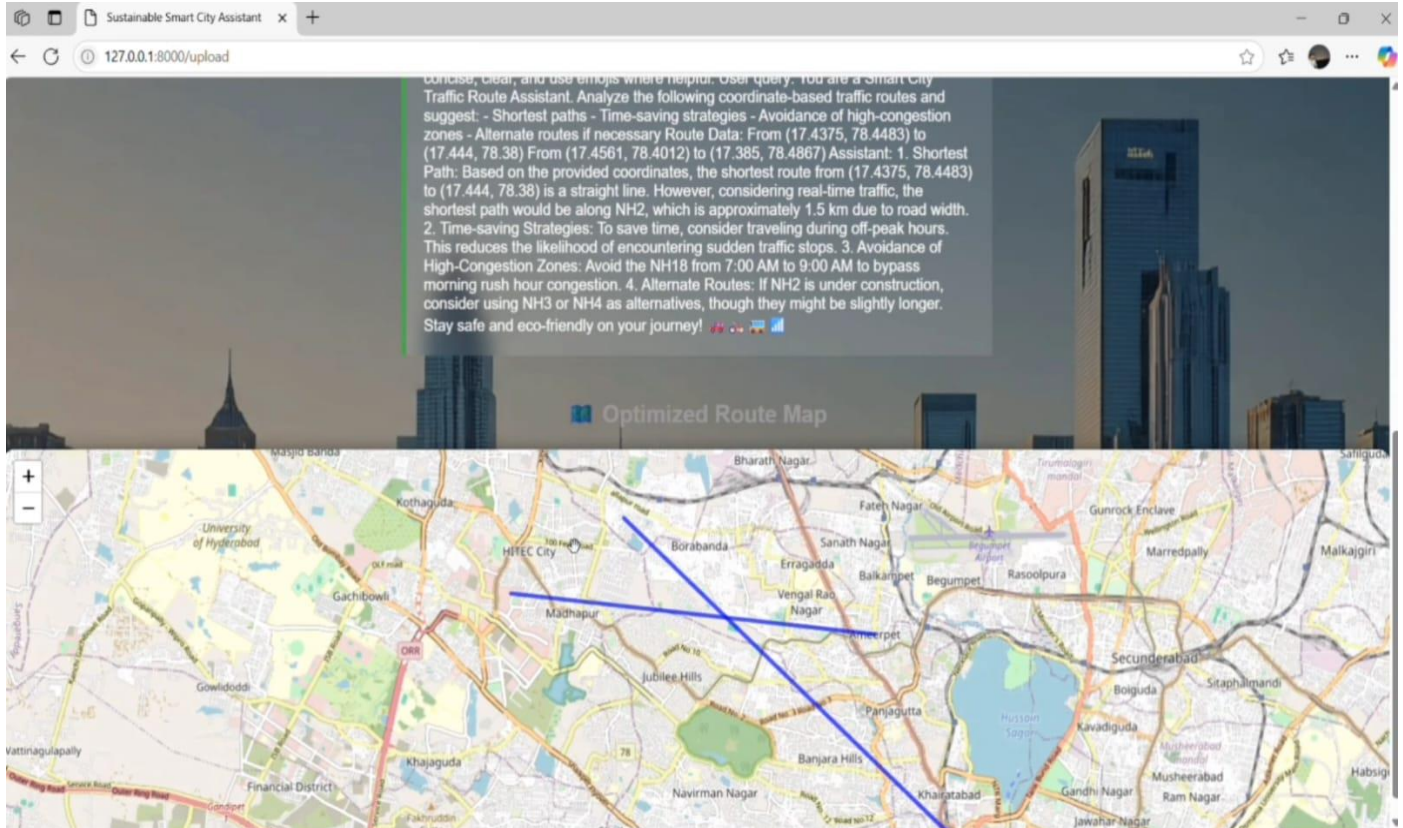
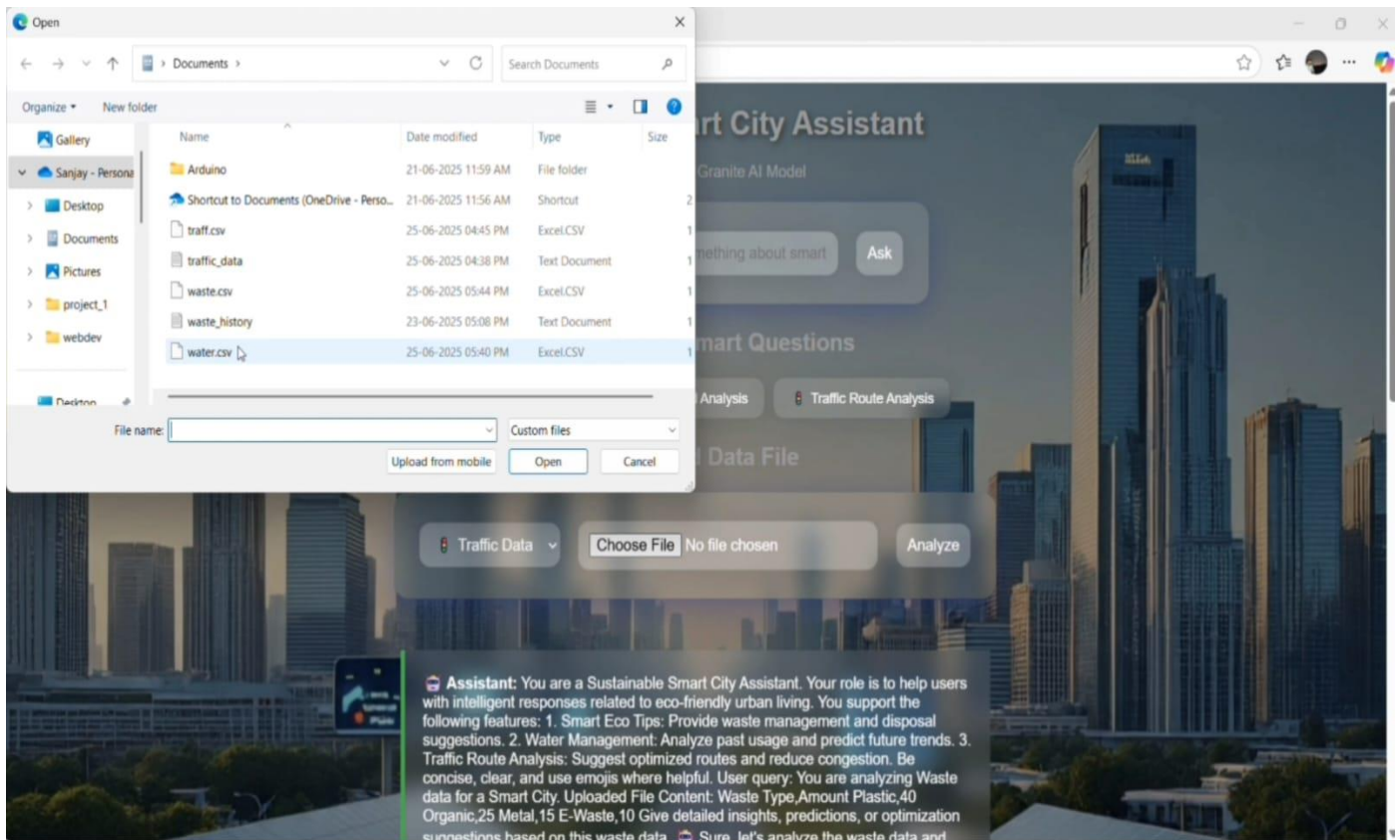
7.1 Output Screenshots

- Screenshot of Introduction



- Screenshot of Smart City





8. ADVANTAGES & DISADVANTAGES

Advantages: - Easy to use - AI-powered civic issue classification - Visualization of environmental and complaint data

Disadvantages: - No real-time municipal data integration - No authentication or user profiles - General- purpose AI model

9. CONCLUSION

SmartCityAI successfully demonstrates the application of AI in civic services by combining a simple, accessible user interface with powerful backend intelligence. While currently a prototype, it holds strong potential for real-world deployment with further enhancements.

10.FUTURE SCOPE

- Add secure user login
 - Use real -time municipal service databases
 - Integrate with IoT sensors and smart city infrastructure
 - Fine-tune Granite AI model on city-specific complaint and service datasets
 - Add features like SMS/email alerts, complaint tracking, and public service notifications
-

1. APPENDIX

- **GitHub Link:** <https://github.com/LukkaSanjay/sustainable-smart-city>
- **Source Code Files:** `main.py`, `index.html`, `style.css`, `map.js`, `chart.js`