# PROJECT REPORT

TITLE**: Sustainable Smart City Assistant Using IBM Granite LLM**

**Team ID:** LTVIP2025TMID32000  
**Team Leader:** K Sai Charan  
**Team Members:** Inkollu Sai Bhanu Koushik, Gunisetti Rakesh, Gousia Begum

### 1. INTRODUCTION

**1.1 PROJECT OVERVIEW**

The **Sustainable Smart City Assistant** is a Streamlit-based AI application powered by **IBM Watsonx Granite LLM**, designed to support city administrators and citizens by offering intelligent tools for policy summarization, sustainability forecasting, citizen feedback processing, anomaly detection, and automated report generation. It brings together Natural Language Processing, real-time data insights, and semantic search to enhance urban governance and civic engagement.

**1.2 PURPOSE**

The purpose of this system is to **accelerate sustainable city planning and public decision-making** by reducing manual work, enhancing transparency, and providing AI-generated insights from city data. It enables municipalities to **analyze policy documents, track key performance indicators (KPIs)**, and **respond to citizen input efficiently** — all within a unified digital platform.

### 2. IDEATION PHASE

**2.1 PROBLEM STATEMENT**

City departments often deal with **fragmented tools and manual processes** for analyzing long policy documents, forecasting resource usage, detecting anomalies in environmental KPIs, and managing public feedback. This leads to **delayed responses**, **missed insights**, and **low citizen engagement**. A unified, AI-powered assistant can **simplify analysis**, **centralize feedback**, and **generate actionable summaries and predictions**, improving governance and sustainability outcomes.

**2.2 EMPATHY MAP CANVAS**

**SAYS**

* “How can I simplify policy reviews?”
* “Can the assistant generate insights from city data?”
* “Is there a platform where I can submit issues and get feedback?”

**THINKS**

* “Will AI help reduce planning and reporting delays?”
* “Is this system secure and reliable for official use?”
* “Can citizens understand the data and use the platform too?”

**DOES**

* Uploads policy documents and KPI reports
* Submits structured or unstructured feedback
* Uses the assistant to ask about eco practices or anomalies
* Reviews AI-generated reports and eco tips

**FEELS**

* Curious about AI support in smart governance
* Relieved when summaries and insights save time
* Empowered by data-driven tools and suggestions
* Confident when decisions are backed by trends and analysis

**PAINS**

* Manual drafting of sustainability reports
* Slow and tedious analysis of documents and feedback
* Lack of real-time alerts for anomalies in city performance
* Poor integration between various governance tools

**GAINS**

* Faster review of lengthy policies
* High citizen satisfaction through feedback loop
* Better predictions of consumption and planning needs
* Effective visual reports and data-driven suggestion

### 3. REQUIREMENT ANALYSIS

**3.1 CUSTOMER JOURNEY MAP**

1. The user opens the **Sustainable Smart City Assistant** via the Streamlit web dashboard.
2. From the sidebar, they choose a module:
   * *Policy Summarization*
   * *Feedback Submission*
   * *KPI Forecasting*
   * *Anomaly Detection*
   * *Eco Tips*
   * *Semantic Search*
   * *Sustainability Report Generator*
   * *AI Chat Assistant*
3. They upload a .txt policy document, .csv KPI file, or type a query/prompt.
4. The system sends the input to the appropriate FastAPI backend and IBM Granite LLM model.
5. AI generates the output: a summary, prediction graph, feedback classification, anomaly alert, tip, or report.
6. The result is displayed in real-time on the UI.
7. The user reviews, copies, or downloads the results as needed.
8. They can switch between modules or start a new session.

**3.2 SESSION REQUIREMENTS**

* **a)** Users must be able to **upload policy (.txt) and KPI (.csv) files** easily.
* **b)** All interactions (queries, uploads) should receive **real-time AI responses**.
* **c)** Users must have the ability to **download AI-generated outputs** (summaries, graphs, reports).
* **d)** The application should **retain session state**, preserving chat interactions and user input during the session (Streamlit session state).

**3.3 TECHNOLOGY STACK**

| **Component** | **Technology Used** |
| --- | --- |
| Frontend | Streamlit (Python Web UI) |
| Backend | FastAPI with Python 3.11 |
| AI Service Layer | IBM Watsonx Granite 13B Instruct v2 |
| Vector Search Engine | Pinecone (semantic document search) |
| PDF/Text Processing | PyMuPDF (fitz), pandas for CSV handling |
| Environment Management | virtualenv, .env (with python-dotenv) |
| Deployment | Streamlit Cloud / IBM Cloud |

### **4. PROJECT DESIGN**

**4.1 PROBLEM–SOLUTION FIT**

City governments and planners require real-time analysis of lengthy policy documents, accurate forecasts of sustainability KPIs, and tools to process citizen feedback at scale. Integrating IBM Granite LLMs into a modular assistant provides AI-driven summaries, anomaly alerts, and eco-insights—bridging the gap between governance needs and intelligent automation.

**4.2 PROPOSED SOLUTION**

* **Layer 1: User Interface**  
  Interactive Streamlit dashboard with individual pages for policy upload, forecasting, feedback, eco tips, and chatbot.
* **Layer 2: Core Logic**  
  Python-based backend modules that handle file parsing, feedback classification, data cleaning, anomaly detection, and AI query routing.
* **Layer 3: AI Layer**  
  Securely integrated IBM Watsonx Granite API used for summarization, forecasting, Q&A, and eco tips, supported with retry and error handling.

**4.3 SOLUTION ARCHITECTURE**

| **Component** | **Description** |
| --- | --- |
| UI Layer | Streamlit sidebar, file uploaders, chat window, KPI widgets |
| Application Logic | app.py and pages/ route requests to backend helpers |
| Helper Layer | granite.py, forecast\_utils.py, feedback\_handler.py |
| AI Layer | Watsonx Granite LLM service with throttling + retry logic |
| Semantic Search | Pinecone for vector-based search of uploaded policy docs |

### **5. PROJECT PLANNING AND SCHEDULING**

| **Week** | **Milestones / Deliverables** |
| --- | --- |
| Week 1 | Project scope finalization, Streamlit UI skeleton, policy upload |
| Week 2 | Granite LLM integration, feedback and forecast modules, basic testing |
| Week 3 | Semantic search, anomaly detection, chat assistant, UI enhancements |
| Week 4 | Final testing, report generation, PDF export, deployment & demo prep |

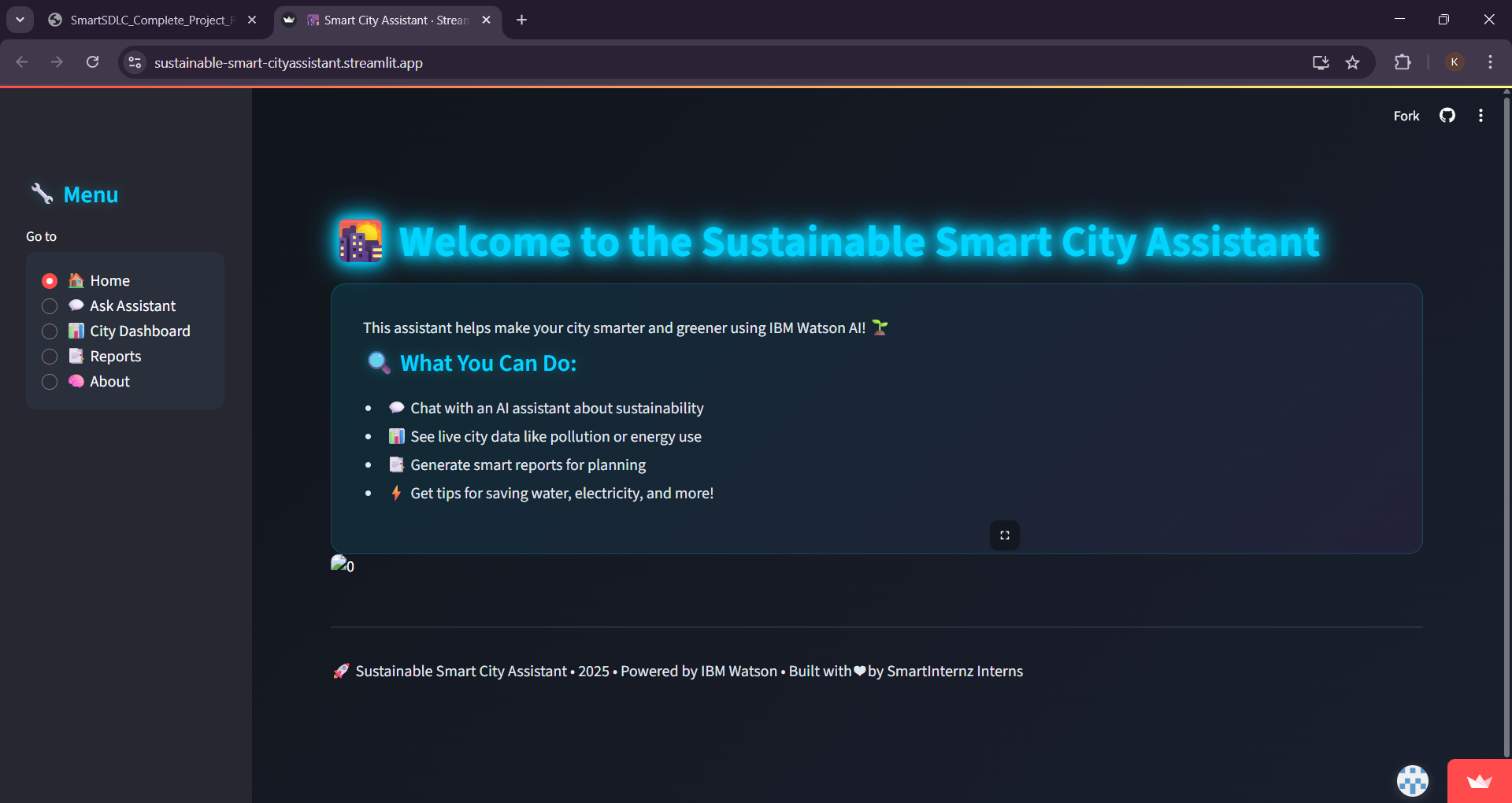
### **6. FUNCTIONAL AND PERFORMANCE TESTING**

* **Unit Testing**
  + Testing PDF/CSV ingestion, feedback classifier, and KPI forecast logic
* **Integration Testing**
  + Full data flow from UI → FastAPI → Watsonx Granite → UI response
* **Manual Testing**
  + Real .txt policies and .csv KPIs across city categories (water, energy)
* **Error Handling Tests**
  + Network interruptions, file size limits, API rate-limit failures, empty inputs
* **Performance**
  + Average response time per module < 5 seconds
  + Sustains concurrent use by up to 10 simulated users in test environment

### 7. RESULTS

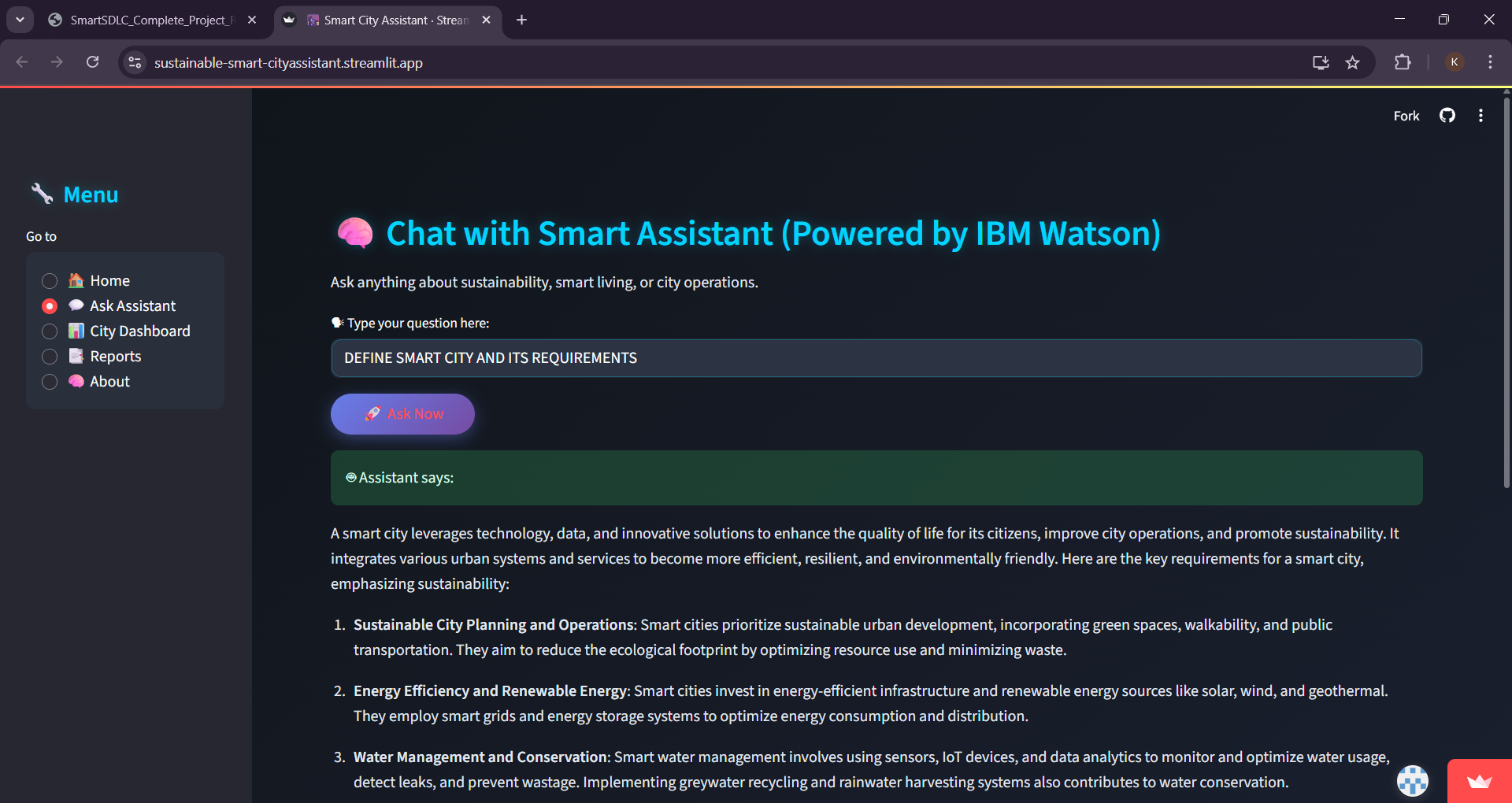
## 1. Home Page Interface

The homepage introduces users to key functionalities such as chatting with an assistant, viewing city KPIs, and generating reports.



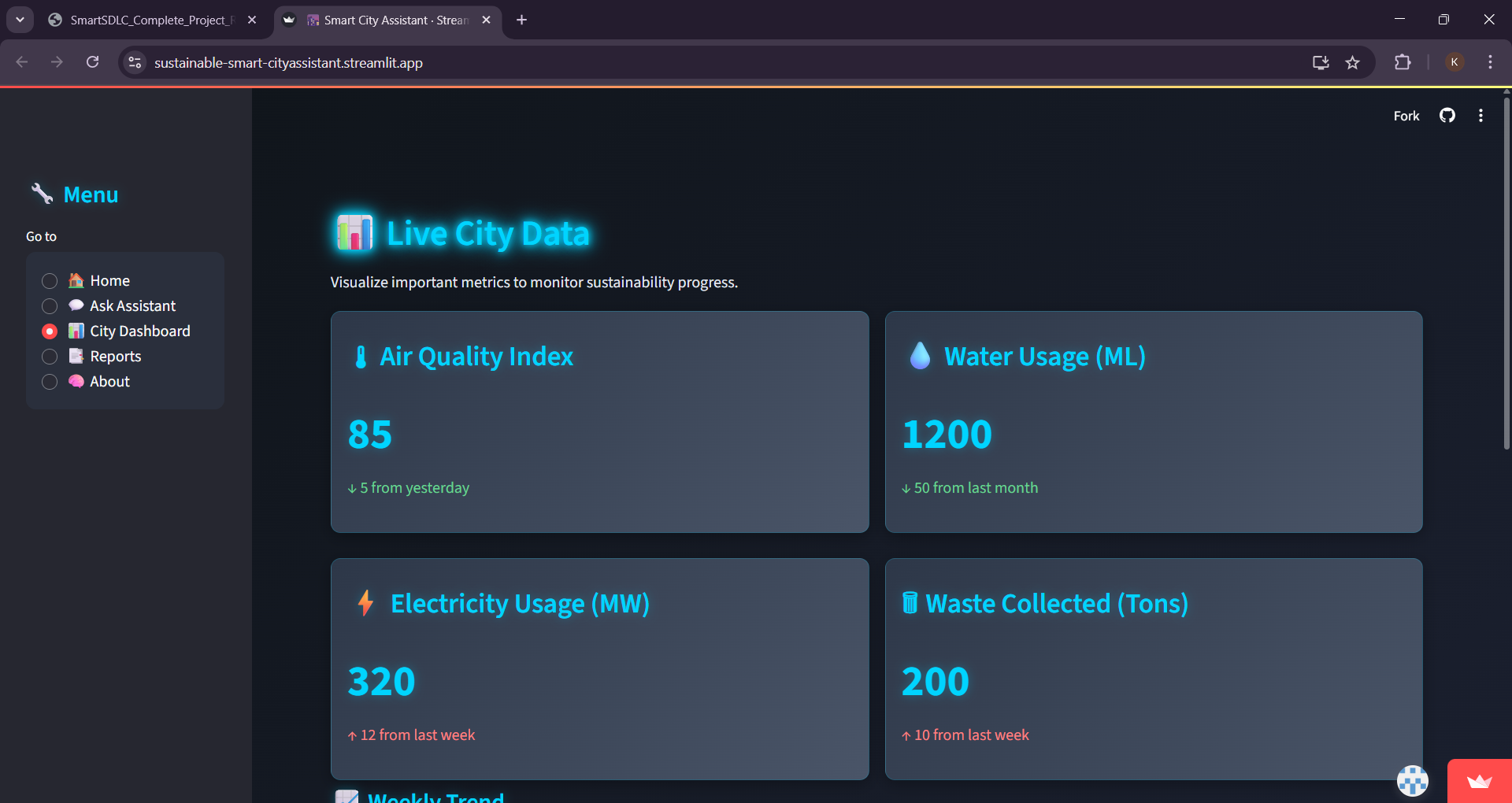
## 2. Chat with Assistant

Citizens can query the AI assistant about sustainability and receive instant contextual replies powered by IBM Watsonx.



## 3. Live City Data Dashboard

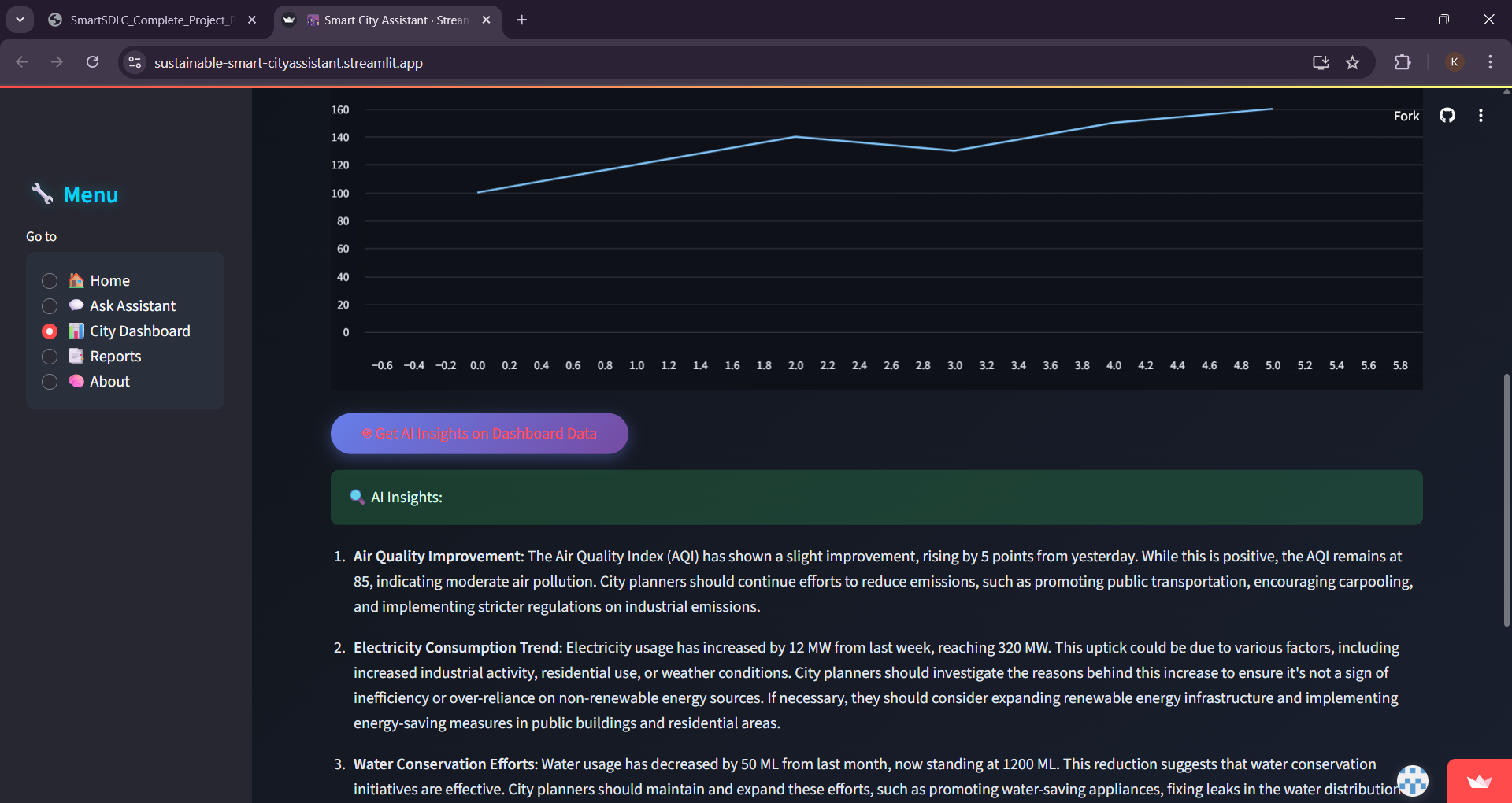
Displays real-time metrics such as air quality, water usage, electricity usage, and waste collected.



## 4. AI-Generated Insights from KPI Trends

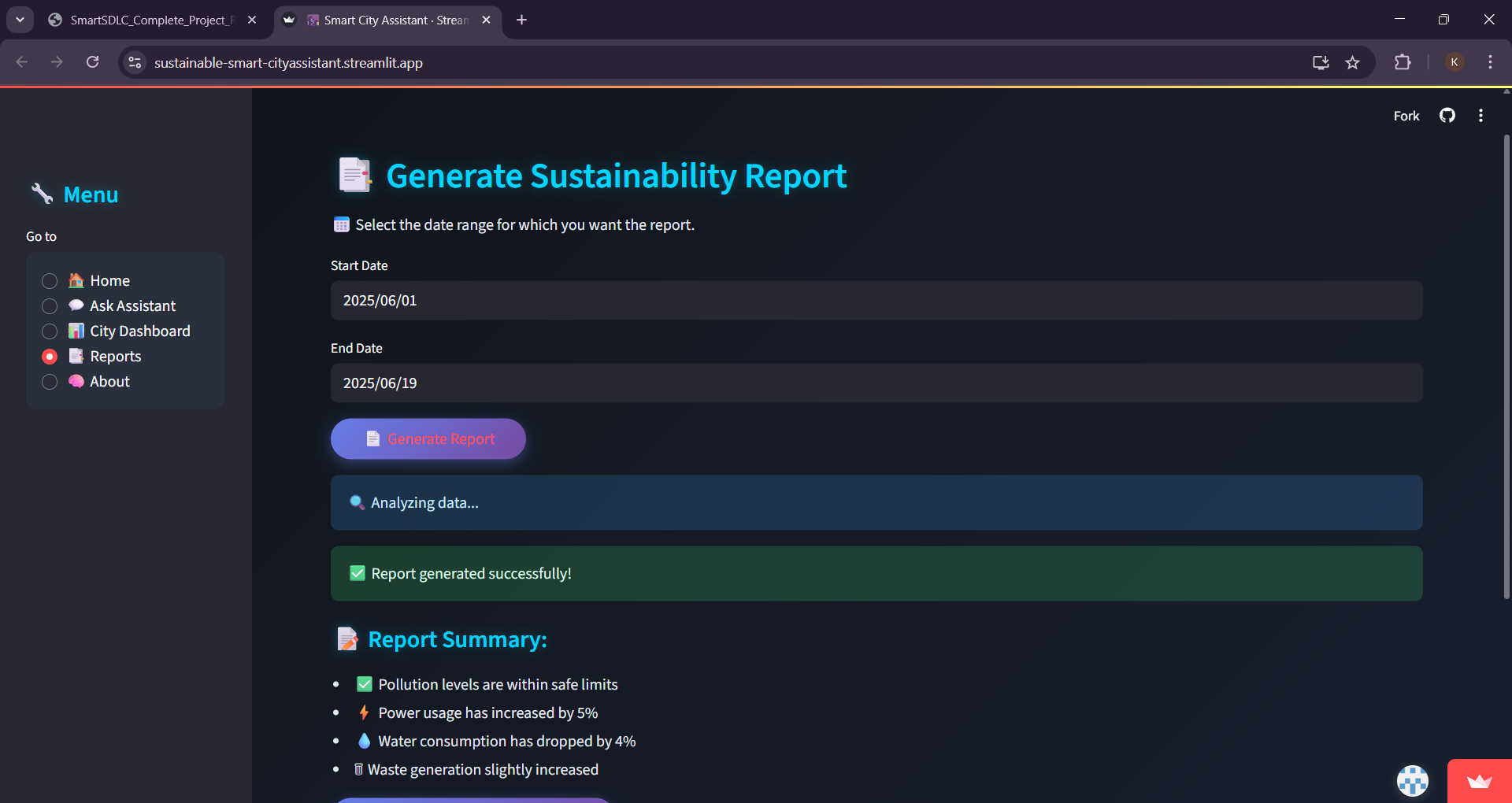
AI analyzes usage trends and anomalies to suggest actions on improving air quality, power usage, and water conservation.





## 5. Sustainability Report Generation

Users can select a date range and automatically generate a sustainability summary highlighting progress and problem areas.



### 8. ADVANTAGES AND DISADVANTAGES

**ADVANTAGES**

* Unified platform for policy summarization, citizen feedback, forecasting, anomaly detection, and reporting
* Real-time AI responses powered by IBM Watsonx Granite LLM
* User-friendly, modular interface with low entry barrier (Streamlit)
* Open-source, extensible architecture with support for Pinecone search and .csv/.txt inputs
* Rapid turnaround for sustainability analysis and decision-making

**DISADVANTAGES**

* No user login or authentication implemented yet
* Supports English only (multilingual capabilities planned)
* Requires internet connection for accessing Watsonx Granite LLM API
* Currently no integration with persistent storage (e.g., databases)

### **9. CONCLUSION**

The **Sustainable Smart City Assistant** demonstrates how **generative AI can power the next generation of smart governance**. By combining Streamlit’s ease of use with the analytical capabilities of **IBM Watsonx Granite LLM**, the assistant reduces manual overhead in urban planning and reporting.

The tool showcases how policy summaries, predictive analytics, eco guidance, and real-time citizen engagement can be streamlined through AI. It sets a strong foundation for scaling smart city tools that are **modular, AI-enabled, and citizen-inclusive**.

### **10. APPENDIX**

* **GitHub Repository:**  
  <https://github.com/InkolluKoushik/Sustainable-Smart-City-Assistant>
* **Key Files:**  
  app.py, pages/, modules/, .env, requirements.txt
* **LLM Model Used:**  
  ibm/granite-13b-instruct-v2 via Watsonx API
* **License:**  
  MIT License