Sustainable Smart City Assistant using IBM Granite LLM

1. INTRODUCTION

1.1 Project Overview :
The "Sustainable Smart City Assistant using IBM Granite LLM" is a comprehensive AI-powered web application aimed at transforming urban governance through intelligent automation, citizen-centric services, and sustainable living support. It integrates various smart modules including chat-based assistance, eco tips, KPI forecasting, anomaly detection, policy summarization, and sustainability report generation. Built using Streamlit (frontend), FastAPI (backend), and IBM Granite LLM, the system leverages advanced NLP and ML capabilities to deliver a greener, smarter, and citizen-friendly urban experience.

1.2 Purpose:

The purpose of this project is to build an interactive and intelligent platform that empowers citizens and city administrators with real-time sustainability insights, AI-driven assistance, and actionable recommendations to foster smarter and more eco-conscious urban living.

2. IDEATION PHASE

2.1 Problem Statement:

Urban citizens lack a unified and intelligent platform to access real-time sustainability data, personalized eco-tips, and smart policy assistance. Current systems are fragmented and lack AI integration for forecasting, anomaly detection, and natural language-based interactions.

2.2 Empathy Map Canvas

- **Say:** "I want to know how my city is performing sustainably."
- **Think:** "I wish there was one platform to answer all my city-related queries."
- **Do:** Search multiple websites for city policies and sustainability metrics.
- **Feel:** Confused, frustrated, and disconnected from governance decisions.

2.3 Brainstorming

- Use of AI to answer citizen questions.
- Dashboard showing clean energy usage, recycling rates, and pollution levels.
- Forecast and detect anomalies in city sustainability metrics.
- Generate eco tips and sustainability reports.
- Chatbot and semantic search for policies.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. User visits Smart City Assistant.
2. Explores dashboard insights.
3. Uploads data for forecasting.
4. Asks chat assistant questions.
5. Receives eco tips and policy summaries.
6. Submits feedback.

3.2 Solution Requirement

- Natural language-based chatbot using IBM Granite LLM.
- Forecasting module for KPIs.
- Anomaly detection for threshold breaches.
- Dashboard for sustainability metrics.
- Semantic search using Pinecone.

3.3 Data Flow Diagram User (Input) -> Streamlit UI -> FastAPI Backend -> AI/NLP Layer (IBM Granite) + ML Modules -> Output (Forecasts, Reports, Responses)

3.4 Technology Stack

- **Frontend:** Streamlit
- **Backend:** FastAPI
- **AI/NLP:** IBM Granite LLM
- **Visualization:** Matplotlib, Plotly
- **Vector DB:** Pinecone
- **Modeling:** Scikit-learn (Linear Regression)

4. PROJECT DESIGN

4.1 Problem-Solution:

Fit Lack of centralized smart city services is addressed by combining AI-driven chat, analytics, and eco modules in a single assistant.

4.2 Proposed Solution:

A modular, AI-integrated dashboard that offers chat interaction, eco advice, real-time insights, policy summarization, anomaly detection, and forecasting via a friendly Streamlit UI and FastAPI backend powered by IBM Granite LLM.

4.3 Solution Architecture

- Streamlit UI (Chat, Eco Tips, Dashboard)
- FastAPI (Routing, APIs)
- IBM Granite LLM (Chat, Summarization)
- KPI Forecasting (Linear Regression)

- Anomaly Detection (Threshold)
- Pinecone (Semantic Search, optional)

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning **Sprint 1: Setup & Architecture (13 SP)**

- Set up FastAPI and Streamlit UI
- Integrate IBM Granite LLM
- Deploy chatbot interface

Sprint 2: Core Features (21 SP)

- Eco-tip generation
- Feedback form
- Document upload & policy summarization

Sprint 3: Intelligence & Forecasting (21 SP)

- Linear regression for KPI forecasting
- Threshold-based anomaly detection
- Sustainability report generation

Sprint 4: Optimization (13 SP)

- Prompt refinement
- Testing and deployment
- Usability improvements

Velocity: 68/4 = 17 SP per sprint

6. FUNCTIONAL AND PERFORMANCE TESTING

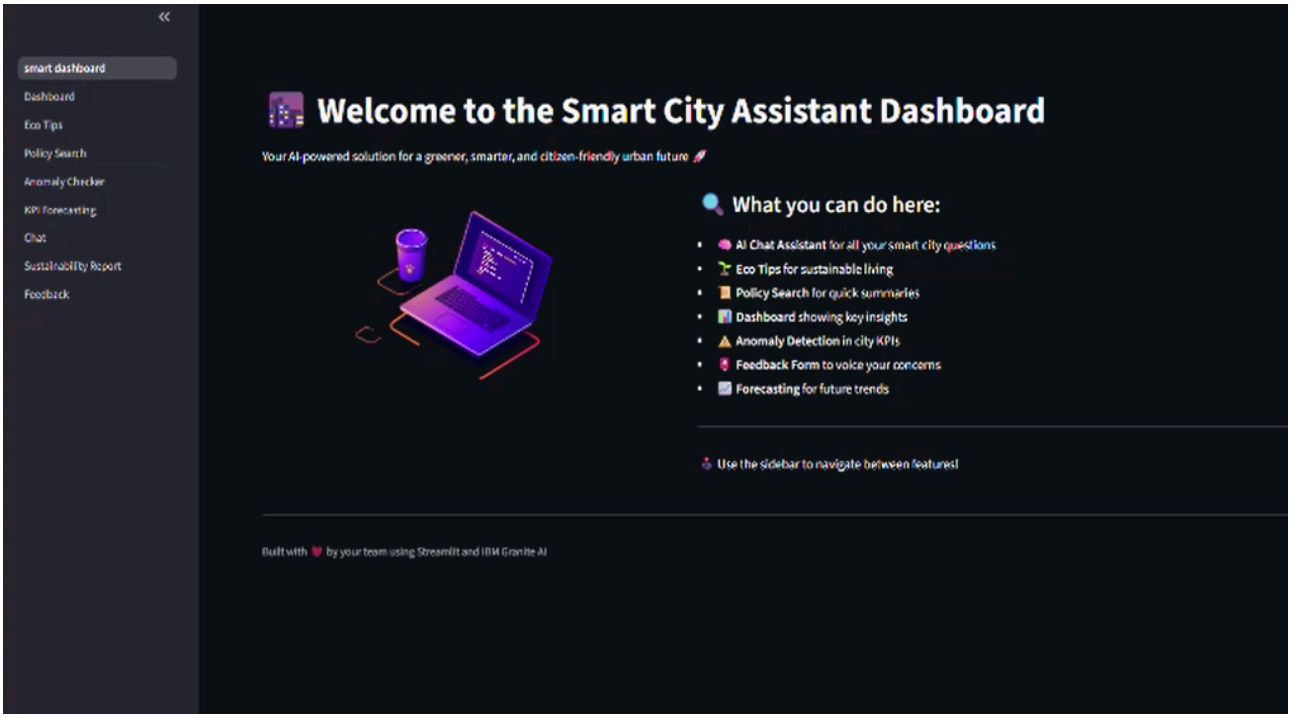
6.1 Performance Testing

- Forecasting tested with time-series KPI CSVs.
- Chat latency tested for prompt response.
- Edge cases validated for empty data, corrupted files, and invalid inputs.

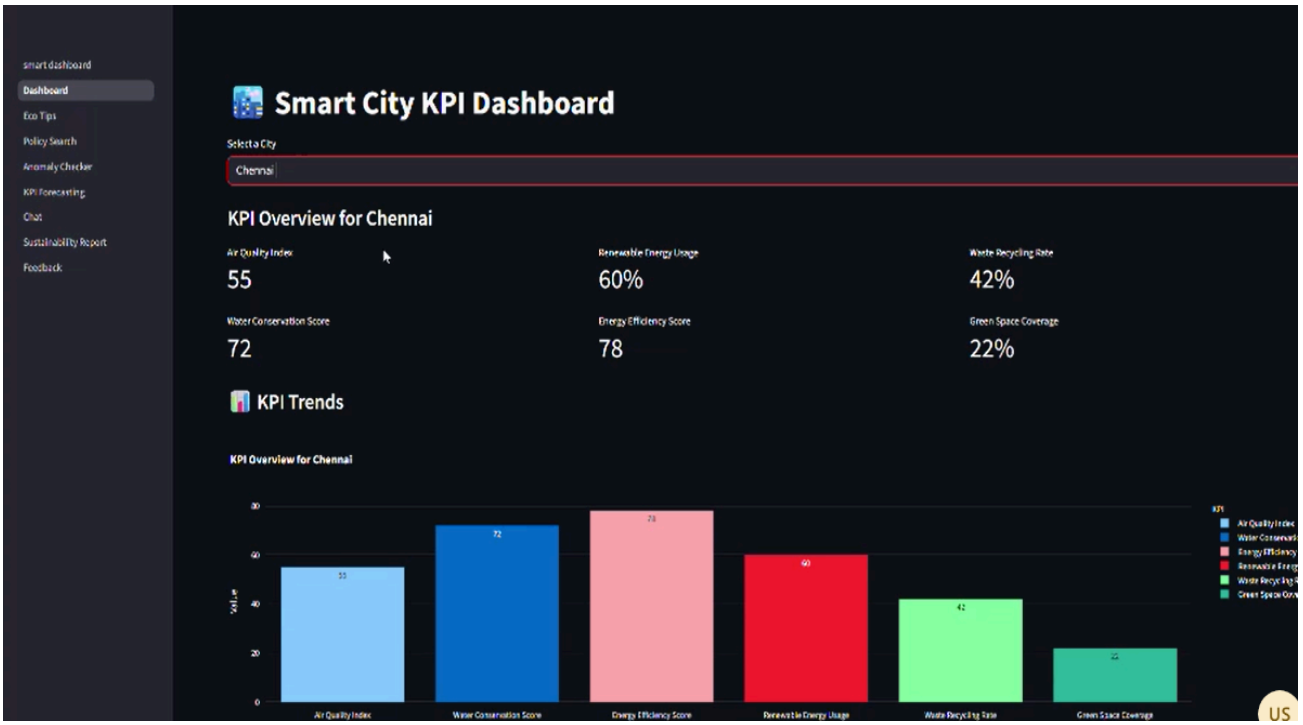
7. RESULTS

7.1 Output Screenshots:

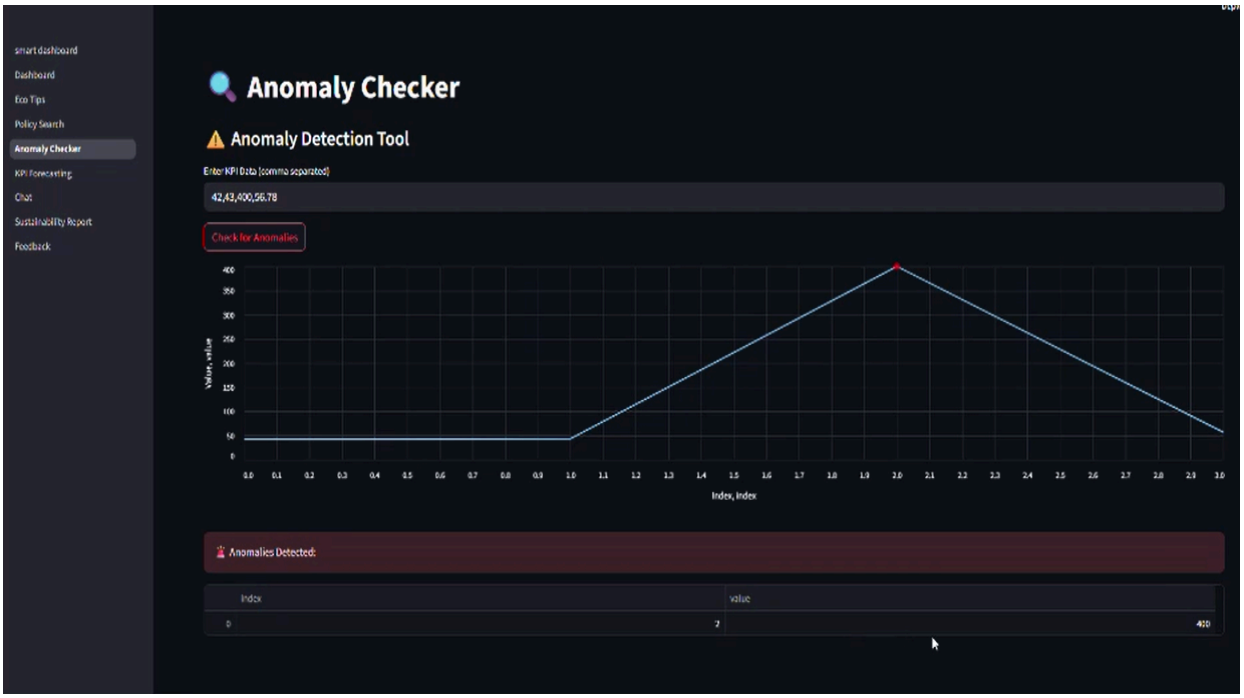
- Chat Assistant Interface



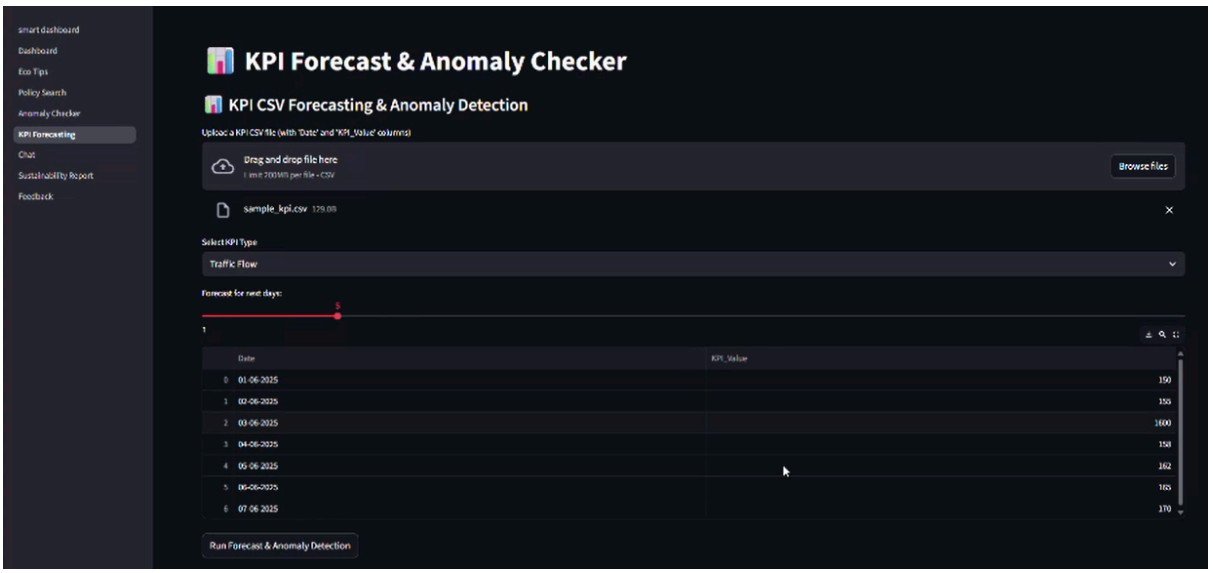
- Sustainability Dashboard



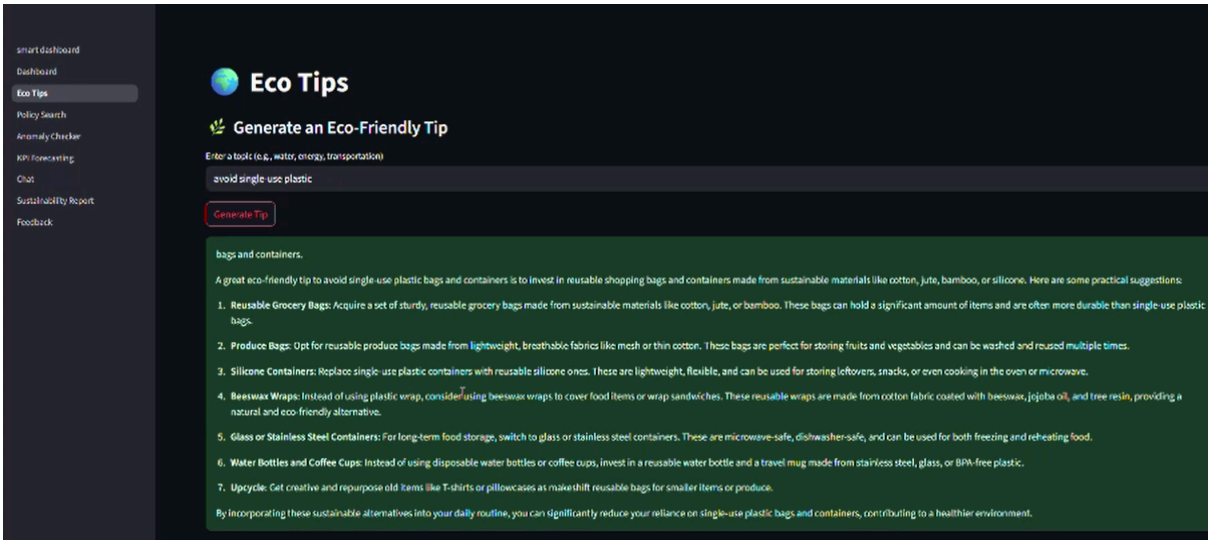
- Anomaly Alerts



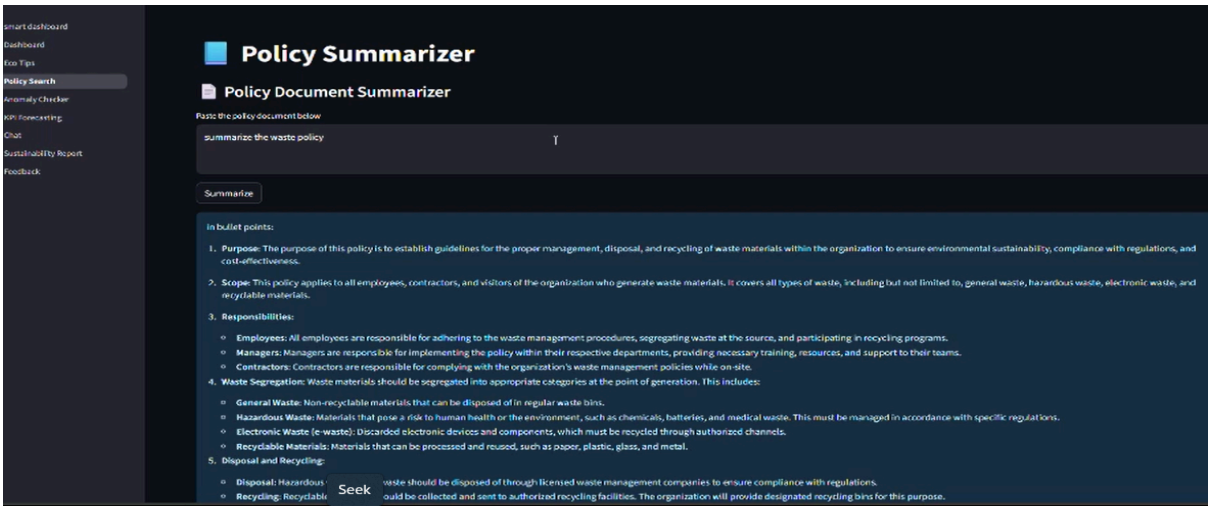
- Forecast Graphs



- Eco Tips Page



- Policy Summarization Output



- Feedback Form Submission

8. ADVANTAGES & DISADVANTAGES

Advantages:

- Centralized smart city services
- NLP-based citizen interaction
- Visual insights and trend forecasting
- Modular and scalable design

Disadvantages:

- Requires constant data updates
- Limited to structured KPI input
- Lacks real-time sensor integration

9. CONCLUSION

The Sustainable Smart City Assistant successfully combines AI with sustainability data, creating a useful and accessible platform for both citizens and city planners. It showcases how LLMs can power intelligent services for real-time urban management.

10. FUTURE SCOPE

- Integration with real-time IoT data from smart sensors
- Support for more cities and languages
- Deployment on cloud infrastructure for scalability
- Deep learning models for improved forecasting

11. APPENDIX

- **GitHub/Project Demo:** https://github.com/SrikariSadvi/smart-city-assistant