**PROJECT REPORT**

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

**1. INTRODUCTION**

**1.1 Project Overview**

The Sustainable Smart City Assistant is an AI-powered platform designed to address key challenges in modern urban governance. Using IBM Watsonx Granite LLM and a modular frontend in Streamlit, this assistant enables city officials, sustainability planners, and citizens to interact with city data and services more intelligently. It offers features such as policy summarization, KPI forecasting, anomaly detection, citizen feedback logging, eco-friendly guidance, and conversational AI.

**1.2 Purpose**

To automate and simplify complex urban decision-making processes by leveraging Generative AI for real-time analysis, summarization, and engagement. The project provides a centralized assistant for data-driven urban management and sustainable living support.

**2. IDEATION PHASE**

**2.1 Problem Statement**

Urban stakeholders struggle to access insights from long policy documents, forecast civic trends, and process large datasets manually. This project proposes an LLM-integrated dashboard that simplifies interactions, increases transparency, and encourages citizen engagement.

**2.2 Empathy Map Canvas**

* **Says:** “I want quicker access to key points from policy documents.”
* **Thinks:** “Can this tool help me find patterns in my city’s data?”
* **Does:** Uploads PDFs/CSVs, submits feedback, queries chatBot.
* **Feels:** Relieved when AI handles tedious tasks.
* **Pains:** Manual data handling, lack of real-time insights.
* **Gains:** Faster decision-making, better public communication.

**2.3 Brainstorming**

Ideas included:

1. AI-powered summarization of city policy documents
2. Upload-based CSV forecasting and anomaly detection
3. Structured feedback reporting interface for citizens
4. Eco Tips Generator using LLM based on user keywords
5. Chat assistant for answering sustainability or governance queries
6. Modular layout for switching between modules
7. Loading animations and success confirmation for better UX
8. One-click PDF export of summaries and forecasts (future scope)

**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey Map**

1. Launch the web app
2. Choose module (e.g., Policy Summarizer, Feedback Reporter)
3. Upload file / input text
4. Receive AI-generated results
5. Take action (e.g., share summary, plan based on forecast)

**3.2 FunctionalRequirements**

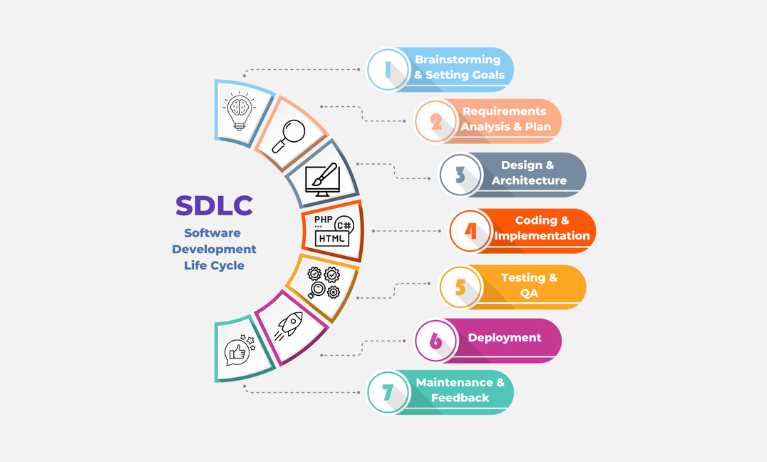
· PDF and CSV input handling

· Text input for chat and eco modules

· AI response display with loading state

· Session management via Streamlit

**3.3 Data Flow Diagram**



**3.4 Technology Stack**

* **Frontend:** Streamlit
* **Backend:** Python + Pandas + PyMuPDF
* **AI Service:** IBM Watson LLM
* **Storage :** Streamlit Session State
* **Deployment:** Localhost / Streamlit Cloud

**4. PROJECT DESIGN**

**4.1 Problem-Solution Fit**

City officials and citizens lack access to real-time, easy-to-understand summaries and insights. This assistant automates these tasks using LLMs and turns raw data into actionable knowledge.

**4.2 Proposed Solution**

An AI assistant that allows users to upload PDFs, input text or data, and receive structured summaries, forecasts, or alerts—reducing reliance on technical staff.

**4.3 Solution Architecture**

* **UI Layer:** Streamlit Forms, Chat Input , Uploaders
* **Logic Layer :** Prompt generation, data handling
* **AI Layer:** IBM Watsonx LLM (Granite)
* **Output Layer:** Summary, forecast, chat, anomaly report

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

|  |  |  |
| --- | --- | --- |
| **Week** | **Dates** | **Activites** |
| Week 1 | June 1 – June 7 | Ideation, Phase planning, UI layout design |
| Week 2 | June 8 – June 14 | Backend phase logic, UI coding, form setup |
| Week 3 | June 15 – June 21 | Testing, debugging, final PDF integration |
| Week 4 | June 22 – June 26 | |  | | --- | | Report generation, documentation, submission | |

**6. FUNCTIONAL AND PERFORMANCE TESTING**

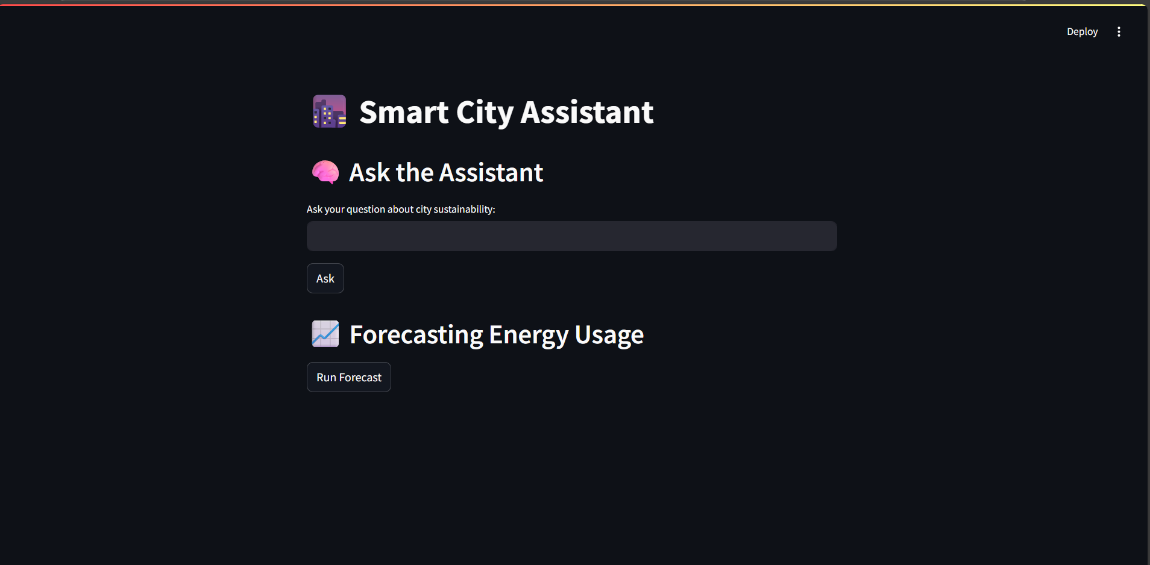
**6.1 Performance Testing**

* **UI Speed:** Instantaneous response to phase selections
* **Load Handling:** Able to process long inputs without lag
* **PDF Accuracy:** Consistent formatting across all phase outputs
* **Cross-Browser:** Tested on Chrome, Firefox, Edge

**7. RESULTS**

**7.1 Output Screenshots**

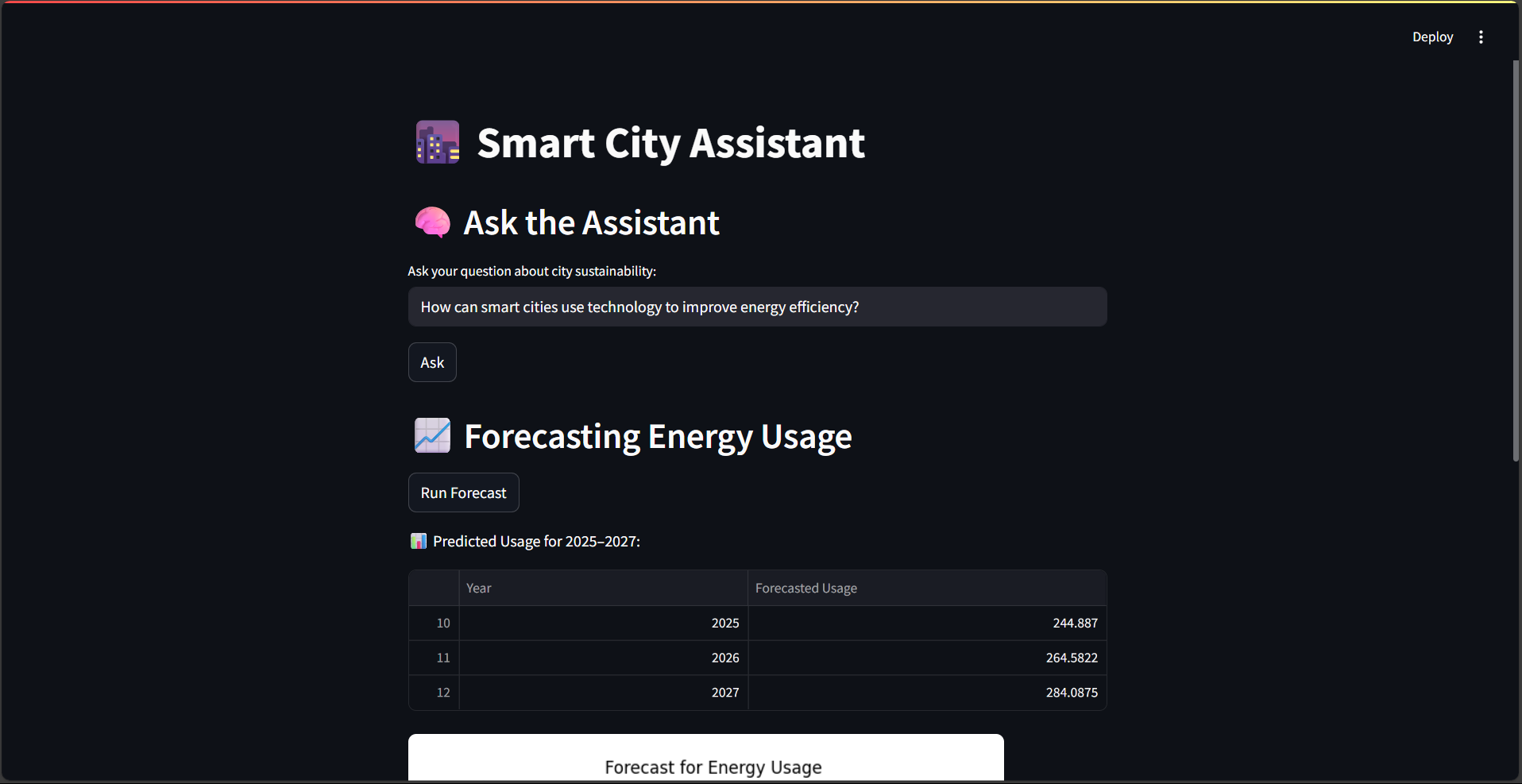
* Screenshot: DashBoard

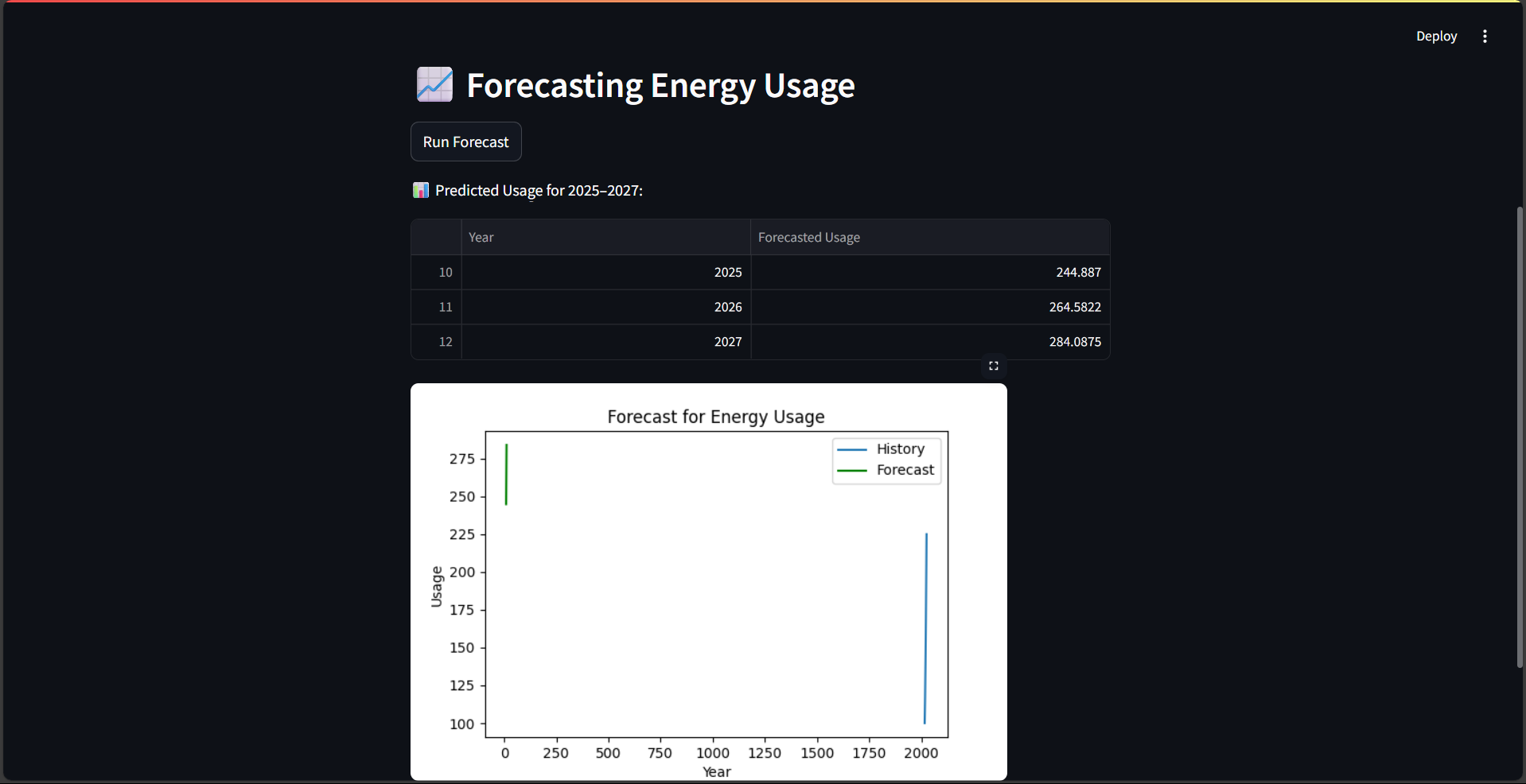


* Screenshot: Dashboard – Assistant Response



* Screenshot: Dashboard – Forecast Result:





**8. ADVANTAGES & DISADVANTAGES**

**Advantages**

* No-login, lightweight UI
* LLM-backed summaries and forecasting
* Citizen-centric interaction model
* Modular and scalable architecture

**Disadvantages**

* No user authentication or saved sessions
* Requires internet for Watsonx API
* Doesn’t support full backend execution or live database

**9. CONCLUSION**

This project demonstrates how LLMs can be embedded in smart city applications to streamline governance, data analysis, and civic communication. It bridges gaps between government data and public understanding using Generative AI in a user-friendly interface.

**10. FUTURE SCOPE**

* Enable user login and feedback tracking
* Integrate GitHub/Gov DBs for real-time document access
* Include data visualization for KPIs
* Deploy to Streamlit Cloud or IBM Cloud
* Add Hindi/Telugu language support via LLM prompts

**11. APPENDIX**

* **GitHub Link:** *(https://github.com/HariJakku/sustainable-smart-city)*
* **Demo Link:** *(e.g.,* [*https://smart-city.streamlit.app*](https://smart-sdlc.streamlit.app/)*)*
* **Source Files:** app.py, requirements.txt