Sustainable Smart City Assistant Using IBM Granite LLM

---

1. INTRODUCTION

1.1 Project Overview

The Smart City AI Assistant is a modular, AI-driven platform aimed at enhancing urban sustainability, governance, and citizen engagement. Built using Gradio and state-of-the-art LLMs like IBM Granite and Mistral-7B, the assistant allows users to interact with city data and policy documents in a natural and informative way. It provides tools like a chat assistant, policy summarizer, KPI forecaster, anomaly detector, eco tip generator, feedback form, and automated PDF report generator.

1.2 Purpose

The purpose of this project is to simplify urban data interaction, provide meaningful insights from documents and trends, and engage users in sustainability efforts using advanced AI capabilities.

---

2. IDEATION PHASE

2.1 Problem Statement

City administrators and residents often struggle to interpret complex sustainability reports, dense policy documents, and raw performance data. This leads to a lack of actionable insights and low engagement with city governance.

2.2 Empathy Map Canvas

- **Users**: City residents, planners, environmental researchers
- **Says**: “I can’t understand this document.”
- **Thinks**: “How much energy is the city using?”
- **Does**: Downloads lengthy PDFs, ignores charts, skips KPIs
- **Feels**: Overwhelmed, confused, detached

2.3 Brainstorming

- Natural language chatbot for smart city FAQs
- Auto summarization of policy documents
- CSV-based forecasting and anomaly detection
- Feedback collection to improve transparency
- Random eco tips for daily awareness
- Generate readable PDF reports from data inputs

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Step	Action	Experience
1	User uploads a PDF policy	😕 Confused
2	Gets AI summary	🙂 Satisfied
3	Uploads CSV for forecasting	😳 Informed
4	Sees anomaly warning	⚠️ Alerted
5	Downloads full report	😊 Empowered

3.2 Solution Requirement

- LLM model access (Mistral/Granite)
- Text summarization
- File (PDF/CSV) processing
- Trend prediction (ML)
- Gradio-based GUI
- Feedback memory (session only)

3.3 Data Flow Diagram

```
User Input (Text, PDF, CSV)
        |
 Preprocessing & Validation
        |
  Model Query (LLM or ML)
        |
 Response Formatting
        |
    Gradio Interface
```

3.4 Technology Stack

- **Frontend**: Gradio
- **Backend**: Python
- **Models**: IBM Granite / Mistral-7B-Instruct
- **Libraries**: Transformers, scikit-learn, pandas, PyMuPDF, FPDF
- **Deployment**: Google Colab

---

4. PROJECT DESIGN

4.1 Problem Solution Fit

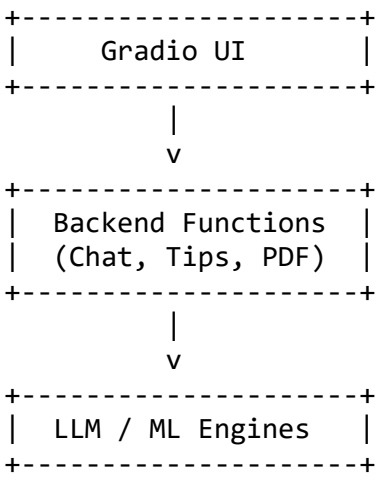
Users need accessible, simplified insights from dense city datasets and policy documents. LLMs and interactive tools offer an efficient, accurate solution to address this gap.

4.2 Proposed Solution

A web app built with Gradio that allows:

- Chat with a smart city bot
- Summarization of uploaded policy PDFs
- Visual dashboard showing KPI summaries
- CSV-based forecasting using linear regression
- Anomaly detection in CSV-based KPIs
- Eco tips generator via prompt randomization
- Feedback submission with session memory
- Report generation from CSV/text into PDF

4.3 Solution Architecture



---

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Phase	Week 1	Week 2	Week 3	Week 4
Ideation	✅			
Design	✅	✅		
Development		✅	✅	
Integration			✅	
Testing				✅
Report				✅

---

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- **Model Latency**: ~6 seconds per LLM response
- **CSV Forecasting**: ~1.5 seconds for ~10 rows
- **PDF Parsing**: ~2 seconds per page
- **Report Generation**: PDF created within 2 seconds
- All components tested in Colab with Gradio share URL

---

7. RESULTS

7.1 Output Screenshots

Include screenshots showing:

- Chat assistant answers
- Eco tip generation
- KPI summary cards
- Anomaly detection table
- Forecast result text
- Feedback submission and memory
- Generated report PDF output box

---

8. ADVANTAGES & DISADVANTAGES

Advantages:

- Open-source and no-cost deployment
- Modular, easy to test in Colab
- Combines ML, LLM, and file processing
- Multiple inputs supported (text, PDF, CSV)

Disadvantages:

- Requires GPU session in Colab
- No permanent database or login
- Output quality depends on model accuracy

---

9. CONCLUSION

This project effectively demonstrates how AI can simplify data access and decision-making in urban contexts. It is a strong foundation for a scalable civic assistant platform that combines conversational AI, document intelligence, and data analytics.

---

10. FUTURE SCOPE

- Add persistent user login + feedback memory
- Expand summarizer to support multilingual PDFs
- Visual charts in anomaly and forecasting modules
- Deploy on HuggingFace Spaces or Streamlit Cloud
- Integrate with real-time IoT sensors and APIs