# PROJECT REPORT

TITLE: Sustainable Smart City Assistant Using IBM Granite LLM

Team ID: LTVIP2025TMID32000

Team Size: 4

Team Leader: K Sai Charan

Team Member: Inkollu Sai Bhanu Koushik

Team Member: Gunisetti Rakesh

Team Member: Gousia Begum

## CONTENTS

1. Introduction

1.1 Project Overview

1.2 Purpose

1. Ideation Phase

2.1 Problem Statement

2.2 Empathy Map Canvas

2.3 Brainstorming

1. Requirement Analysis

3.1 Citizen Journey Map

3.2 Session Requirements

3.3 Data Flow Diagram

3.4 Technology Stack

1. Project Design

4.1 Problem-Solution Fit

4.2 Proposed Solution

4.3 Solution Architecture

1. Project Planning and Scheduling
2. Functional and Performance Testing
3. Results
4. Advantages and Disadvantages
5. Conclusion
6. Appendix

### 1. INTRODUCTION

#### 1.1 Project Overview

The Sustainable Smart City Assistant is an intelligent AI-powered system designed to assist citizens and administrators in managing urban sustainability. It leverages IBM Watsonx Granite LLM and a Streamlit-based interface to simplify policy access, generate eco-advice, forecast resource usage, detect anomalies, and answer sustainability-related questions.

#### 1.2 Purpose

The system promotes sustainability awareness, simplifies governance, and enables data-driven decision-making. It provides AI-powered support for policy summarization, resource management, and citizen engagement from a single, user-friendly dashboard.

### 2. IDEATION PHASE

#### 2.1 Problem Statement

Citizens and administrators struggle with fragmented tools for accessing policy information, forecasting resource needs, and promoting sustainable practices. A unified AI-powered assistant can streamline these tasks, making sustainability efforts more effective.

#### 2.2 Empathy Map Canvas

SAYS: "Where can I find simple information on city policies?" "How can I help reduce resource waste?"

THINKS: "Is this AI providing reliable and transparent information?"

DOES: Searches online, submits feedback, consults disconnected platforms

FEELS: Confused by complex documents, eager for actionable guidance

PAINS: Lack of clear information, manual processes for reporting issues

GAINS: Quick policy understanding, easy eco-tips, efficient feedback mechanisms

#### 2.3 Brainstorming

Standalone features such as policy summarization, eco-tip generation, KPI forecasting, and anomaly detection were combined into one end-to-end AI-powered smart city assistant.

### 3. REQUIREMENT ANALYSIS

#### 3.1 Citizen Journey Map

1. Launch the Streamlit-based assistant.
2. Select a module: Policy Summary, Eco Tips, Feedback, KPI Forecast, Anomaly Detection, Chat.
3. Upload documents or input prompts.
4. The AI model provides summaries, eco-advice, forecasts, or chat responses.
5. Review outputs, submit feedback, or take recommended actions.
6. Switch modules or exit.

#### 3.2 Session Requirements

* Upload policy documents and KPI datasets.
* Receive real-time AI responses.
* View eco-tips and sustainability reports.
* Preserve chat history within sessions.

#### 3.3 Data Flow Diagram

The assistant processes user inputs directly within the Streamlit application, interacts with IBM Watsonx, and delivers outputs via the Streamlit interface.

#### 3.4 Technology Stack

* **Frontend:** Streamlit
* **Backend:** Python (within Streamlit application logic)
* **AI Service:** IBM Watsonx Granite LLM
* **Document Processing:** PyMuPDF (If implemented for document reading)
* **Environment Management:** virtualenv, .env secrets

### 4. PROJECT DESIGN

#### 4.1 Problem–Solution Fit

City residents and administrators require faster, AI-supported access to policy information and sustainability tools. Embedding Granite LLM within an accessible dashboard provides intelligent automation that meets these needs.

#### 4.2 Proposed Solution

* **Layer 1:** Streamlit-based modular interface.
* **Layer 2:** Python application logic (within Streamlit) handling file processing, AI prompts.
* **Layer 3:** AI layer utilizing IBM Granite LLM for AI-generated outputs.

#### 4.3 Solution Architecture

* **UI Layer:** Sidebar navigation, interactive cards, chat window.
* **Backend Logic:** Python application logic (within app.py), file processing utilities.
* **AI Layer:** Granite LLM with secure API access.

### 5. PROJECT PLANNING AND SCHEDULING

|  |  |  |
| --- | --- | --- |
| **Week** | **Timeline** | **Activities** |
| Week 1 | 12 Jun – 19 Jun | Idea finalization, Streamlit skeleton, document handling setup |
| Week 2 | 20 Jun – 26 Jun | Granite LLM integration, core AI module development |
| Week 3 | 27 Jun – 03 Jul | Anomaly detection, eco-tip generator, UI polish |
| Week 4 | 04 Jul – 10 Jul | Final testing, documentation, deployment preparation |

### 6. FUNCTIONAL AND PERFORMANCE TESTING

* **Unit Testing:** Prompt generation, output validation functions.
* **Integration Testing:** Streamlit application to Granite LLM communication.
* **Manual Testing:** Real-world policy documents, KPI datasets, feedback modules.
* **Error Handling:** Network issues, oversized files, API rate limits.

### 7. RESULTS

The assistant successfully simplifies policy summarization, provides actionable eco-tips, forecasts resource usage, detects anomalies, and enhances citizen engagement.

### 8. ADVANTAGES AND DISADVANTAGES

**Advantages:**

* Unified platform for policy, sustainability, and citizen engagement.
* Real-time AI responses improve efficiency.
* High-quality summaries, reports, and eco-advice.
* Scalable and modular architecture.

**Disadvantages:**

* No persistent user login system.
* No database for long-term storage (session-based only).
* Limited support for offline operation.
* API usage dependent on cloud subscriptions.

### 9. CONCLUSION

The Sustainable Smart City Assistant demonstrates that AI can enhance urban governance and sustainability efforts. By combining a citizen-friendly dashboard with the power of IBM Watsonx, it supports informed decision-making, resource management, and citizen participation.

### 10. APPENDIX

* **Repository:** [Private Repository - Project Files]
* **Key Files:** app.py, requirements.txt, .env
* **AI Model:** IBM Watsonx Granite-13b-instruct-v2
* **License:** MIT