

Sustainable Smart City Assistant

Project Title: Sustainable Smart City Assistant Using IBM Granite LLM

Team Information:

- **Team ID:** LTVIP2025TMID21183
 - **Team Size:** 4 members
 - **Team Leader:** Thonduru Saisree
 - **Team members:** Polimetla Naveen Kumar, Perecharla Vijay, Pothana Bala Manohar
-

1. Project Overview

The Smart City Assistant is an AI-powered platform designed to support urban sustainability, governance, and citizen engagement. It integrates multiple intelligent modules such as summarization, semantic search, eco-advice, anomaly detection, and forecasting, providing real-time insights using modern NLP and ML models.

2. System Architecture

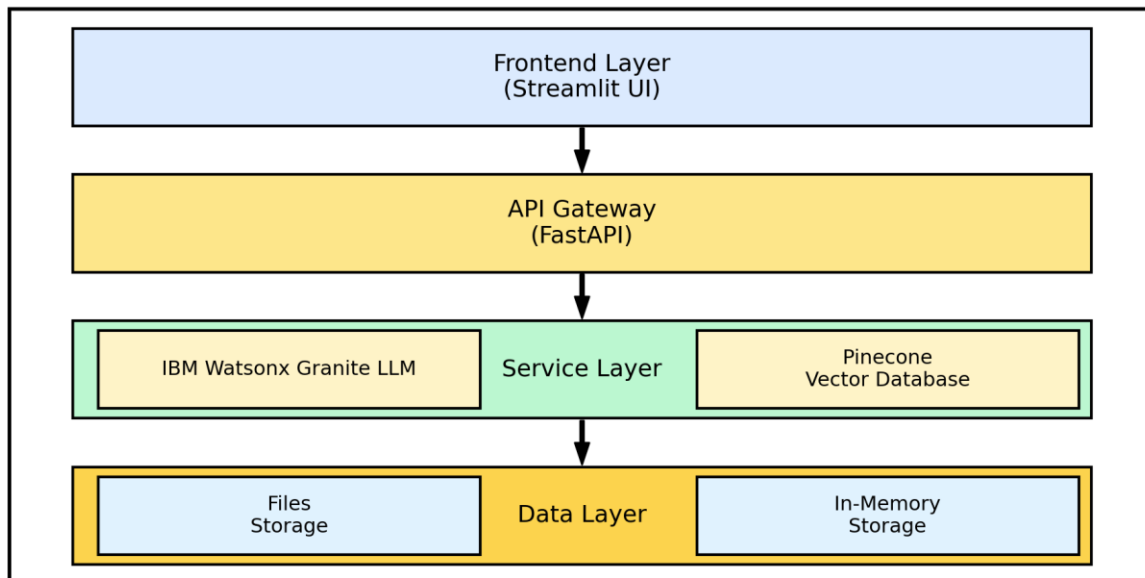
Frontend: Streamlit dashboard for interactive interface

Backend: FastAPI for handling requests and processing data

Model Inference: Hugging Face Transformers for NLP tasks

Database & Embeddings: Pinecone for semantic search, local CSV/JSON for static datasets

User → Streamlit (UI) → FastAPI (Backend) → NLP/ML Modules → Response
→ Pinecone DB / Local Dataset



3. Technology Stack

- **Frontend:** Streamlit
- **Backend:** FastAPI
- **Model Server:** Hugging Face Transformers, SentenceTransformers
- **Database:** Pinecone (Vector DB)
- **Visualization:** Matplotlib
- **Libraries:** Requests, pydantic-settings, scikit-learn, pandas, dotenv

4. Project Structure

smart-city-assistant/

├── backend/

| ├── main.py

| ├── summarizer.py

| ├── search.py

| └── utils.py

```
├── frontend/
|   └── app.py
├── data/
|   └── sample_data.csv
├── .env
├── requirements.txt
└── README.md
```

5. Implementation Details

- **Summarizer:** Uses facebook/bart-large-cnn to summarize long texts.
 - **Semantic Search:** Embeds documents using sentence-transformers and stores in Pinecone.
 - **Eco-Advice:** Template-based NLP for sustainability tips.
 - **Forecasting/Anomaly Detection:** Uses pandas & scikit-learn for trend prediction.
-

6. Development Workflow







1. Backend with FastAPI is started using uvicorn.
 2. Frontend Streamlit app connects to backend endpoints.
 3. User input is processed and routed to summarizer, search or forecasting engine.
 4. Output is returned to frontend in real-time.
-

7. Setup and Installation

1. Clone the repository
2. Create virtual environment: `python -m venv venv`

3. Activate venv and run: `pip install -r requirements.txt`
 4. Add `.env` file with Pinecone and model keys
 5. Start backend: `uvicorn backend.main:app --reload`
 6. Start frontend: `streamlit run frontend/app.py`
-

8. Features and Functionality

-  Text Summarization
 -  Document Semantic Search
 -  Chatbot for FAQs
 -  KPI Forecasting
 -  Anomaly Detection
 -  Eco-Friendly Advice
-

9. API Documentation

- `POST /summarize/?text=<text>`
 - `POST /search/?query=<question>`
 - `POST /forecast/?metric=air_quality`
 - `POST /anomaly/?type=water_usage`
-

10. Screenshots and Results

- **Summary output with input field**
 - **Search interface results with semantic matching**
 - **Graphs from forecasting module**
 - **Anomaly reports with threshold analysis**
-

11. Challenges and Solutions

- Large Model Load: Delayed initial load → Fixed using lazy loading
 - Pinecone Access: Region/env setup confusion → Clear guide and screenshots added
 - Slow inference: Used distilled/smaller models where possible
-

12. Future Enhancements

- Add user login and role management
 - Integrate real-time sensor data (IoT)
 - Use IBM Watsonx Granite model for scalable LLMs
 - Add multilingual support
-

13. Conclusion

The Smart City Assistant is a modular, scalable solution for intelligent urban management. It leverages AI/ML to transform raw data into actionable insights for decision-makers and citizens alike.

Demo link: [city-spark-granite - Personal - Microsoft Edge 2025-06-28 14-30-02.mp4](#)

GITHUB link:

https://github.com/Saisree2005/city_assistant/tree/main/document