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

***Project Documentation***

******

**College**: Government Arts College Trichy -22

**Team Leader:** M. Mahalakshmi

**Email:** mahamani1406@gmail.com

**Team Member 1:** G. Ramya

**Email:** ramyakrish177@gmail.com

**Team Member 2:** R. Latchaya

**Email:** latchayarajavadivel@gmail.com

**Team Member 3:** A. Kanmani

**Email:** kanmaniarulsevan@gmail.com

**Introduction :**

The Sustainable Smart City Assistant Using IBM Granite LLM is an AI-powered solution designed to help cities and their residents move toward a greener, more efficient, and inclusive urban future.

By integrating IBM’s Granite Large Language Model with real-time city data—such as energy consumption, water usage, traffic patterns, and environmental metrics—the assistant provides:

Natural language interaction so citizens and officials can ask questions or request reports in plain English (or local languages).

Policy summarization to quickly condense lengthy government or regulatory documents into clear, actionable insights.

Predictive analytics to forecast energy, water, and waste needs, supporting proactive resource management.

Personalized sustainability tips to encourage eco-friendly daily habits among residents.

Decision-support dashboards for city planners, enabling data-driven governance and smarter infrastructure investment.

This introduction positions the project as a bridge between advanced AI technology and everyday civic engagement, ensuring that sustainability goals are practical, measurable, and citizen-friendly.

**Project Overview**

**Purpose:**

The Sustainable Smart City Assistant empowers cities and residents to thrive in an eco-friendly and connected urban environment.  
 By using IBM Granite LLM and real-time data, it optimizes energy, water, and waste resources while providing citizens with personalized sustainability tips.  
 City officials gain decision-making support through insights, forecasting tools, and concise policy summaries, promoting greener, more efficient, and inclusive cities.

**Features:**

Conversational Interface – Natural language Q&A and guidance for citizens & officials

Policy Summarization – Converts lengthy government docs into simple summaries

Resource Forecasting – Predicts energy/water/waste usage from historical + live data

Eco-Tip Generator – Personalized daily sustainability advice

Citizen Feedback Loop – Collects/analyzes public input for planning

KPI Forecasting – Projects city performance indicators for strategic planning

Anomaly Detection – Flags unusual sensor/usage data early

Multimodal Input Support – Handles text, PDFs, CSVs for analysis and forecasting

Streamlit/Gradio UI – Interactive dashboards and chat interfaces.

**Architecture:**

**Frontend (Streamlit):**

Interactive web UI with dashboards, file uploads, chat, feedback forms, and report viewers.  
 Navigation via sidebar using streamlit-option-menu; each page is modular for scalability.

Backend (FastAPI):  
 REST framework powering API endpoints for document processing, chat, eco-tip generation, report creation, and vector embedding.  
 Optimized for asynchronous performance with Swagger integration.

LLM Integration (IBM Watson Granite):  
 IBM Granite models handle natural language understanding and generation, creating summaries, sustainability tips, and reports.

Vector Search (Pinecone):  
 Policy documents embedded using Sentence Transformers and stored in Pinecone for semantic search.

ML Modules (Forecasting & Anomaly Detection):  
 Lightweight ML models (Scikit-learn) for time-series forecasting and anomaly detection.

**Setup Instructions:**

**Prerequisites**:

Python 3.9+

pip & virtualenv

IBM Watson & Pinecone API keys

Internet access for cloud services

**Installation Process:**

1. Clone repository
2. pip install -r requirements.txt
3. Create .env with API credentials
4. Run FastAPI backend
5. Launch Streamlit frontend
6. Upload data & interact with modules

**Folder structure:**

app/ # FastAPI backend logic  
 app/api/ # API routes (chat, feedback, report, vectorization)  
 ui/ # Streamlit frontend components  
 smart\_dashboard.py # Launches main Streamlit dashboard  
 granite\_llm.py # IBM Granite LLM interactions  
 document\_embedder.py# Embeds documents & stores in Pinecone  
 kpi\_file\_forecaster.py # Forecasts energy/water trends  
 anomaly\_file\_checker.py # Detects anomalies in KPI data  
 report\_generator.py # AI-generated sustainability reports

**Running the Application:**

Start FastAPI server to expose backend endpoints.

Run Streamlit dashboard for the web UI.

Use sidebar navigation to upload documents, chat with the assistant, and view reports/predictions.

All interactions update in real-time through backend APIs.

**API Documentation:**

POST /chat/ask – AI-generated response to user query

POST /upload-doc – Upload & embed documents in Pinecone

GET /search-docs – Semantic policy search

GET /get-eco-tips – Sustainability tips (energy, water, waste)

POST /submit-feedback – Store citizen feedback

**Authentication:**

Demo runs open, but production can include:

Token-based (JWT/API key)

OAuth2 with IBM Cloud

Role-based access (admin/citizen)

Planned session & history tracking

**User Interface:**

Sidebar navigation

KPI visualizations & summary cards

Tabs for chat, eco tips, forecasting

Real-time forms and PDF report downloads

Clean, accessible design for all users

**Testing:**

Unit testing for prompts & utilities

API testing via Swagger/Postman

Manual testing for uploads, chat, output consistency

Edge cases: malformed input, large files, invalid keys

**Known Issues:**

Dependent on stable internet for Granite & Pinecone

High-load scaling not yet benchmarked

**Future Enhancements:**

IoT sensor integration for real-time city-wide updates

Mobile app with push notifications

Voice-based assistant features

