SUSTAINABLE SMARTCITY ASSISTANT USING IBM GRANITE LLM

PROJECT DOCUMENT

1. INTRODUCTION

✦ PEOJECT TITLE : SUSTAINABLE SMARTCITY ASSISTANT USING IBM GRANITE LLM

✦TEAM MEMBER : HASHINI S

✦TEAM MEMBER : DHARSHINI R

✦TEAM MEMBER : ASHWINI M

✦TEAM MEMBER : JANANIDEEPA S

✦ TEAM MEMBER : JAYASRI R

2. PROJECT OVERVIEW

PURPOSE :

The Sustainable Smart City Assistant is an AI-powered platform that leverages IBM Watsonx's Granite LLM and modern data pipelines to support urban sustainability, governance, and citizen engagement. It integrates several modules like City Health Dashboard, Citizen Feedback, Document Summarization, Eco-Advice, Anomaly Detection, KPI forecasting and Chat Assistant through a modular FastAPI backend and a Streamlit- based frontend dashboard.

FEATURES :

*/ Conversational Assistant: A chatbot powered by IBM Granite LLM that answers sustainability-related queries and provides eco-tips based on user input topics.

*/ Document Summarization: Summarizes lengthy government documents into simple bullet points, enhancing transparency and accessibility.

*/ Resource Consumption Prediction: Predicts resource usage based on

user-provided data, enabling proactive resource management.

*/ Citizen Feedback Handling: Allows citizens to submit feedback and generates polite acknowledgment messages using the LLM.

*/ KPI Report Generator: Provides summary reports and trend insights based on user-provided KPI data.

*/ Anomaly Detection: Identifies anomalies in urban data, enabling swift action to mitigate issues.

*/ Eco Tips: Suggests sustainability tips based on user input topics, such as water, plastic, or energy.

*/ KPI Forecast: Forecasts next month's usage based on historical data uploaded by users.

*/ Policy Summarizer: Accepts long policy text and summarizes it to make it citizen-friend

3.Architecture

I. Frontend (Streamlit) :

*/ Interactive Web UI with dashboards, file uploads, chat interface, feedback forms, and report viewers.

*/ Sidebar Navigation handled using the streamlit-option-menu library.

*/ Modular Design ensures scalability and maintainability.

II. Backend (Fast API) :

*/ Acts as the REST framework serving API endpoints for:

*/ Document processing

*/ Chat interactions

*/ Eco-tip generation

*/ Report creation

*/ Vector embedding

*/ Optimized for asynchronous performance with built-in Swagger API docs.

III. LLM Integration (IBM Watsonx Granite) :

*/ Granite LLM models power natural language understanding & generation.

*/ Prompt engineering used for:

*/ Policy summarization

*/ Sustainability tips

*/ Report generation

IV. Vector Search (Pinecone) :

*/ Sentence Transformers generate embeddings for uploaded policy documents.

*/ Pinecone Vector DB stores embeddings for semantic search.

*/ Supports cosine similarity search with natural language queries.

V. ML Modules (Forecasting & Anomaly Detection) :

*/ Built using Scikit-learn for lightweight modeling.

*/ Handles time-series forecasting (e.g., energy demand, water usage).

*/ Performs anomaly detection for unusual consumption or pollution patterns.

*/ Results visualized with pandas & matplotlib

4. SETUP INSTRUCTIONS

Prerequisites :

1. GitHub Repository: Access the repository (e.g., Bhargav5452/smart-city-assistant or Akhileshp123/Sustainable-Smart-City-Assistant-Using-IBM-Granite-LLM).

2. Python Environment: Ensure Python 3.8 or later is installed.

3. Hugging Face Account: Create a Hugging Face account and obtain an API token.

Setup Steps :

1. Clone the Repository: Clone the GitHub repository using git clone.

2. Install Dependencies: Run pip install -r requirements.txt to install required libraries.

3. Set Hugging Face API Token: Securely set your Hugging Face API token (replace "your_token_here" with your actual token).

4. Run the Application: Execute streamlit run app.py to start the Streamlit app.

5. FOLDER STRUCTURE

app/ — Contains all Fast API backend logic including routers, models, and

integration modules.

app/api/ — Subdirectory for modular API routes like chat, feedback, report, and

document vectorization.

ui/ — Contains frontend components for Stream lit pages, card layouts, and

form Uls.

smart_dashboard.py — Entry script for launching the main Stream lit

dashboard.

granite_llm.py — Handles all communication with IBM Watsonx Granite model

including summarization and chat.

document_embedder.py — Converts documents to embeddings and stores in

Pinecone.

kpi_file_forecaster.py — Forecasts future energy/water trends using regression.

anomaly_file_checker.py — Flags unusual values in uploaded KPI data.

report_generator.py — Constructs AI-generated sustainability reports.

6. RUNNING THE APPLICATION

1. Start Backend (FastAPI)

*/ Launches the backend server to provide API endpoints.

2. Start Frontend (Streamlit Dashboard)

*/ Opens the Streamlit dashboard for the interactive web interface.

3. Navigate the Application

*/ Use the sidebar to move between:

*/ Dashboards

*/ File uploads

*/ Chat assistant

*/ Feedback form

*/ Report viewer

4. Interact with Modules

*/ Upload policy documents or CSV datasets.

*/ Ask questions through the conversational assistant.

*/ View outputs such as:

*/ Policy summaries

*/ Sustainability tips

*/ AI-generated sustainability reports

*/ Forecasts and anomaly detection insights

5. Real-Time Updates

*/ All user actions are processed instantly by backend APIs.

*/ The frontend dynamically refreshes outputs for a seamless experience.

7. API Documentation

The backend provides a set of RESTful APIs to support document processing, chat interactions, eco-advice, and feedback collection. All endpoints are tested and auto-documented using Swagger UI for easy inspection.

Available Endpoints :

1. POST /chat/ask

Accepts a user query and returns an AI-generated response using IBM Granite LLM.

2. POST /upload-doc

Uploads documents and converts them into vector embeddings for storage in Pinecone.

3. GET /search-docs

Performs semantic search and returns policy documents most similar to the input query.

4. GET /get-eco-tips

Provides sustainability tips for selected domains (e.g., energy, water, waste).

5. POST /submit-feedback

Collects and stores citizen feedback for future analytics and system improvement.

8. Authentication

Currently, all endpoints are tested and documented in Swagger UI for easy inspection and trial during development. This demonstration version runs in an open environment for simplicity.

Secure Deployment Options :

For production-ready systems, authentication and access control can be integrated:

Token-Based Authentication — Using JWTs or API keys for secure API access.

OAuth2 Integration — With IBM Cloud credentials for enterprise-level security.

Role-Based Access Control (RBAC) — Different permissions for administrators, citizens, and researchers.

Planned Enhancements — User session management and history tracking for personalized interactions.

9.USER INTERFACE

The interface is minimalist and functional, focusing on accessibility for non-technical users. It includes:

=> Sidebar with navigation

=> KPI visualizations with summary cards

=> Tabbed layouts for chat, eco tips, and forecasting

=> Real-time form handling

=> PDF report download capability

The design prioritizes clarity, speed, and user guidance with help texts and intuitive flows.

10. TESTING

Testing was carried out in multiple phases to ensure stability, accuracy, and robustness of the application.

Testing Phases:

*/ Unit Testing – Validated prompt engineering functions, utility scripts, and core logic.

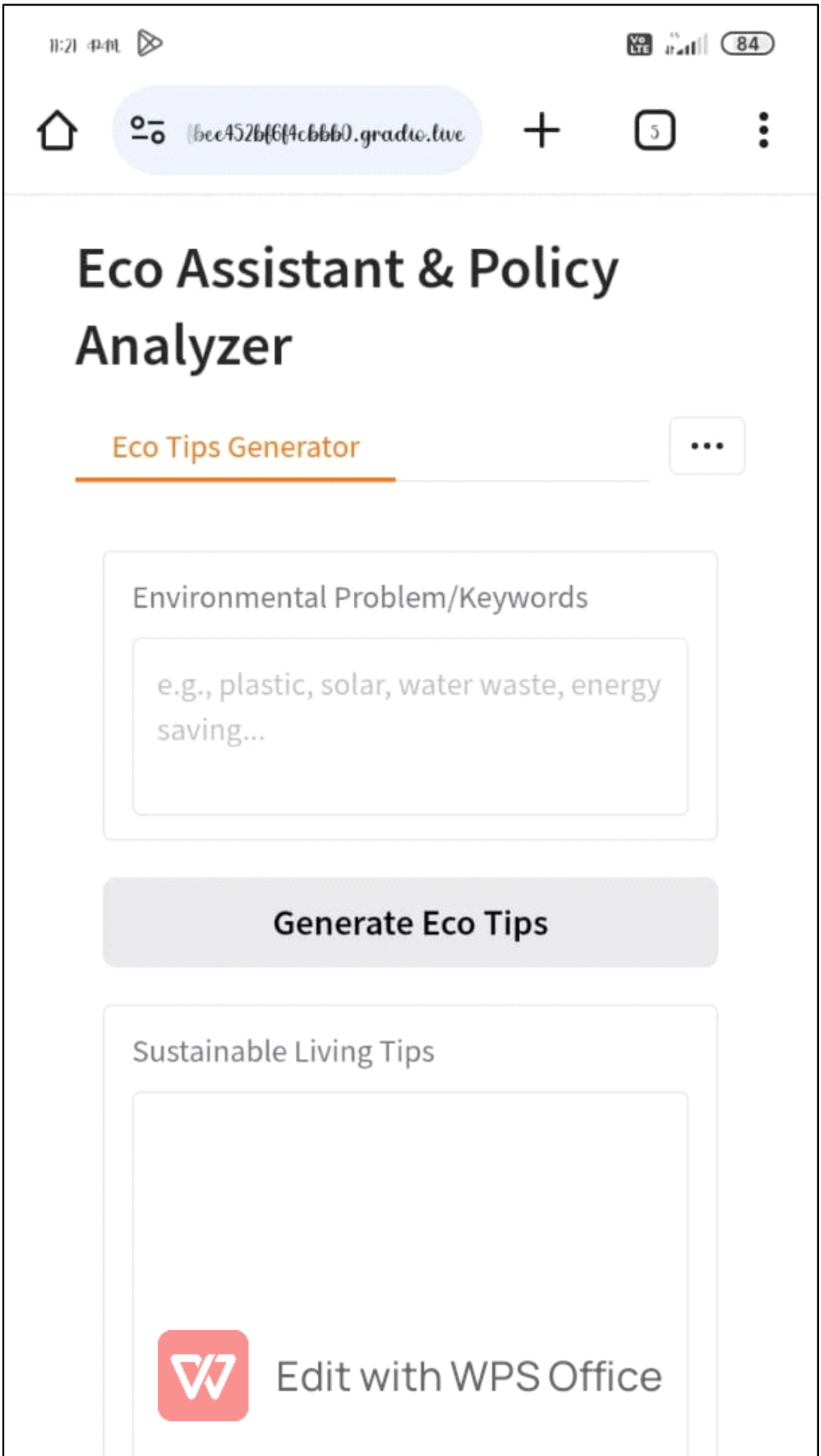
*/ API Testing – Performed using Swagger UI, Postman, and automated test scripts to verify endpoint functionality.

*/ Manual Testing – Checked file uploads, chat responses, forecasting outputs, and report generation for usability and correctness.

*/ Edge Case Handling – Tested against malformed inputs, large files, and invalid API keys to confirm graceful error handling.

Each function was validated for reliability in both offline mode and API-connected mode, ensuring consistent performance across environments.

11. Screenshot


11:21 PM
84
bee452bf6f4cbbb0.gradio.live
5
Eco Assistant & Policy Analyzer
Eco Tips Generator
Environmental Problem/Keywords
e.g., plastic, solar, water waste, energy saving...
Generate Eco Tips
Sustainable Living Tips
Edit with WPS Office

12.FUTURE ENHANCEMENT:

1. Predictive Environmental Monitoring & Optimization
2. Intelligent Energy ManagemenT
3. Proactive Infrastructure Maintenance Alert
4. Citizen-Centric Dialogue Interface
5. Policy Impact Simulation
6. Transparent Data Insights Dashboard
7. Privacy-Aware Data HandlingS