Sustainable Smart City Assistant Using IBM Granite LLM

Project Documentation

1.Introduction

- Project title : Sustainable Smart City Assistant Using IBM Granite LLM
- Team member : KESHAVA PRIYA M
- Team member : HARINI S
- Team member : KAVYA M
- Team member : FIRTHOUS FATHIMA M

2.Project overview

- Purpose:

The Sustainable Smart City Assistant Using IBM Granite LLM is designed to create a smarter and greener urban environment by using advanced AI. It leverages IBM Granite LLM to analyze real-time city data such as energy usage, waste management, and traffic flow, and provides accurate, sustainable solutions for both citizens and administrators. The assistant simplifies complex processes, promotes eco-friendly practices, and ensures inclusivity through multilingual support. With a focus on transparency, data security, and efficiency, it reduces manual work, supports informed decision-making, and enhances citizen engagement. Ultimately, the system strengthens the relationship between people and government, driving sustainable growth and improving overall city life.

- Features:

Conversational Interface

Key Point: A user-friendly chatbot powered by IBM Granite LLM that enables natural and human-like interactions.

Functionality: Citizens can ask questions in simple language (text or voice), and the system responds instantly with accurate and context-aware information.

Policy Summarization

Key Point: AI-driven summaries of sustainability and urban policies for easy understanding.

Functionality: The system condenses lengthy policy and environmental documents into clear summaries, helping citizens and officials quickly grasp rules, benefits, and eco-guidelines.

Eco-Tip Generator

Key Point: AI-powered suggestions to promote green and sustainable practices.

Functionality: Based on user habits and city data, the system provides personalized eco-tips such as energy conservation, waste reduction, and eco-friendly transport.

Citizen Feedback Loop

Key Point: A system for collecting and analyzing citizen feedback to improve sustainability initiatives.

Functionality: Citizens can share opinions or concerns, and AI processes this input to give valuable insights for better city services and environmental policies.

KPI Forecasting

Key Point: Predictive analysis of sustainability and service efficiency indicators.

Functionality: AI models forecast trends in resource usage, pollution levels, traffic, and citizen engagement, enabling proactive planning and data-driven decision-making.

Anomaly Detection

Key Point: AI-powered identification of unusual patterns in city operations.

Functionality: The system detects anomalies like sudden spikes in pollution, irregular energy consumption, or unexpected traffic congestion, allowing early intervention and preventive measures.

Multimodal Input Support

Supports text, voice, and image queries for easy citizen interaction.

3. Architecture

Frontend (Streamlit)

Provides simple dashboards, forms, and visualizations for user access.

Backend (FastAPI)

Handles requests, connects with AI models, and ensures scalability.

LLM Integration (IBM Watsonx Granite)

Powers natural language understanding, policy summaries, and smart responses.

Vector Search (Pinecone)

Enables fast, context-aware retrieval of sustainability data.

ML Modules (Forecasting and Anomaly Detection)

Forecast KPIs like energy/waste usage and detect anomalies for proactive city management.

4. Setup Instructions

Key Point: Step-by-step process to install and configure the Smart City Assistant.

Functionality: Ensures smooth installation of required tools, libraries, and dependencies.

Install Python 3.9 or above.

Install required libraries using pip install -r requirements.txt.

Set up FastAPI backend and configure environment variables for APIs and databases.

Initialize Pinecone for vector search and integrate with IBM Granite LLM API.

Run Streamlit frontend for user interaction.

5. Running the Application

Key Point: Guide to start and access the system.

Functionality: Helps users and developers run the application seamlessly.

Start the FastAPI server: uvicorn main:app --reload.

Launch the Streamlit UI: streamlit run app.py.

Access the application on http://localhost:8501.

The system connects frontend, backend, LLM, and vector search for full functionality.

6. API Documentation

Key Point: Provides details of backend API endpoints for smooth integration.

Functionality: Enables developers to test and extend services easily.

/query – Handles citizen queries via text, voice, or image.

/summarize – Generates AI-driven policy and document summaries.

/eco-tips – Provides personalized eco-friendly suggestions.

/feedback – Collects and analyzes citizen feedback.

/forecast – Predicts KPIs like energy, traffic, and waste trends.

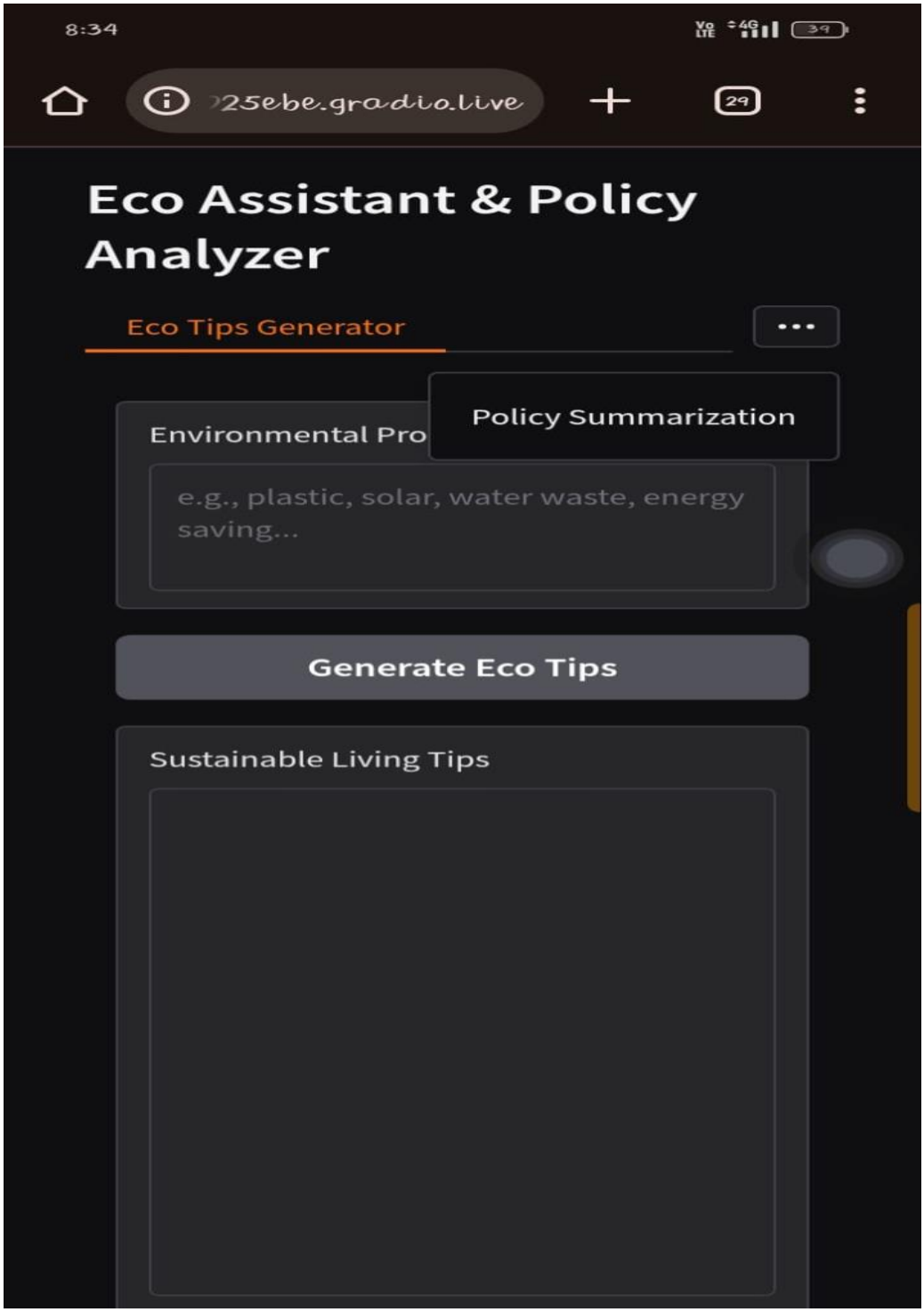
/anomaly – Detects irregularities in sustainability data.

User Interface

The user interface (UI) is the visual and interactive part of the Smart City Assistant through which citizens and administrators access services. It provides dashboards, chat options, and real-time updates in a simple and user-friendly manner.

Testing

Testing is the process of evaluating the system to ensure accuracy, reliability, and smooth performance. It verifies that each module works correctly, integrates seamlessly, and meets the needs of both citizens and administrators.


8:34
25ebe.gradio.live
29
Eco Assistant & Policy Analyzer
Eco Tips Generator
Policy Summarization
Environmental Pro
e.g., plastic, solar, water waste, energy saving...
Generate Eco Tips
Sustainable Living Tips