Sustainable Smart City Assistant Using IBM Granite LLM

Project Documentation

1.Introduction

- Project title : Sustainable Smart City Assistant Using IBM Granite LLM
- Team member : Swathi S
- Team member : Shofia H
- Team member : Sreemayee J
- Team member : Sowmiya R

2. Project Overview

The **Sustainable Smart City Assistant** is an AI-powered application designed to support individuals, communities, and policymakers in promoting sustainability and environmental awareness. Built using **IBM Granite LLM**, the system focuses on two core functions:

1. **Eco Tips Generation** – Users can input keywords related to environmental issues such as *plastic pollution, water conservation, renewable energy, or waste management*. The assistant then generates **practical, actionable, and eco-friendly tips** to encourage sustainable living practices.
2. **Policy Summarization** – Government policies, environmental reports, or legal documents are often complex and lengthy. The system allows users to **upload PDF files** or paste text directly, after which the AI summarizes the content, highlighting **key points, provisions, and implications** in a simplified manner.

The project bridges the gap between **technology and sustainability**, making environmental knowledge **accessible, understandable, and actionable**. It leverages **Gradio** for an interactive user interface, **PyPDF2** for PDF extraction, and the **IBM Granite LLM** for natural language understanding and generation.

This solution not only raises awareness but also empowers citizens to **take small, impactful steps toward building greener cities**, while assisting policymakers and researchers in **quickly analyzing critical documents**.

3.Architecture

The architecture of the **Sustainable Smart City Assistant** is designed to ensure efficiency, scalability, and user-friendliness. It integrates **Natural Language Processing (NLP)** with an interactive interface to deliver sustainable solutions and policy insights.

3.1 System Components

1. **User Interface (UI)**
    - Built with **Gradio**, providing a simple and interactive platform.
    - Users can input keywords for eco-tips or upload/paste policy documents.
2. **AI Model Layer**
    - Powered by **IBM Granite LLM** for natural language understanding and generation.
    - Handles two primary tasks:
        - Generating eco-friendly tips.
        - Summarizing environmental policies.
3. **Data Processing Layer**
    - **PyPDF2** is used to extract text from uploaded PDF documents.
    - Input text is preprocessed and tokenized using **Hugging Face Transformers**.
4. **Computation Layer**
    - Model is executed using **PyTorch**, optimized for CPU/GPU.
    - Ensures efficient inference for both text generation and summarization.
5. **Output Layer**
    - Presents **eco tips** or **policy summaries** in a user-friendly text format.
    - Results are displayed in structured, readable paragraphs.

4. Setup Instructions

The following steps explain how to set up and run the **Sustainable Smart City Assistant**:

Prerequisites

- A computer or laptop with **Windows, Linux, or MacOS**.
- **Python (version 3.9 or higher)** installed on the system.
- Internet connection for downloading libraries and running the application.
- (Optional) A **GPU-enabled system** for faster performance.

Required Tools and Libraries

The project requires the following tools and libraries:

- **Torch (PyTorch)** – for handling AI model operations.
- **Transformers** – for using the IBM Granite LLM.
- **Gradio** – for building the interactive user interface.
- **PyPDF2** – for extracting text from PDF files.

Installation Process

1. Install Python on your system.
2. Install the required libraries using Python's package manager.
3. Download or copy the project files to your computer.

Running the Application

- Open the project folder and run the application file.
- The system will generate a **local link** and a **public link**.
- Open the link in a web browser to access the assistant.

Using the Application

1. **Eco Tips Generator**
   - Enter environmental problems or keywords such as *plastic, water saving, solar energy*.
   - Click the button to generate **eco-friendly tips**.
2. **Policy Summarization**
   - Upload a **policy document (PDF)** or paste text into the text box.

    - Click the button to get a **simplified summary** with the main points.

5. Folder Structure

The project files are organized into the following structure for easy navigation and management:

Explanation

- **app.py** → The entry point for running the assistant.
- **requirements.txt** → Contains all dependencies to install.
- **assets/** → Stores logo and other visuals.
- **data/** → Contains sample policy PDFs or input files.
- **modules/** → Organized Python scripts for each feature.
- **outputs/** → Stores generated eco-tips or summaries.
- **docs/** → Project documentation and report files.

6. Running the Application

Once the setup is complete, follow these steps to run and use the **Sustainable Smart City Assistant**:

Step 1: Start the Application

- Open the project folder on your computer.
- Run the application file (app.py).
- The system will automatically launch a **local server** and display a URL link in the terminal.
- Click or copy the URL into your browser to access the assistant interface.

Step 2: Access the Interface

The interface will open in your default web browser with two main tabs:

1. **Eco Tips Generator**
    - Enter keywords such as *plastic*, *renewable energy*, *water waste, solar panels*.

    - Click on **Generate Eco Tips**.
    - The system will display **practical and eco-friendly suggestions**.
2. **Policy Summarization**
    - Upload a policy PDF document **or** paste the policy text into the text box.
    - Click on **Summarize Policy**.
    - The system will provide a **clear, simplified summary with key points and implications**.

Step 3: Interact and Explore

- You can run both modules independently.
- Try different environmental issues or policy documents for diverse outputs.
- Summaries and tips can be copied and saved for future use.

Step 4: Closing the Application

- To stop the assistant, go back to the terminal window.
- Press **Ctrl + C** to shut down the local server.

7. API Documentation

The system exposes functional APIs through its backend modules. These APIs allow interaction with the **IBM Granite LLM** for generating eco-tips and summarizing policy documents. Each function processes input and returns structured output for the user interface.

Available Endpoints / Functions

- **POST /generate-eco-tips**
    - Description: Generates sustainability tips based on given environmental keywords.
    - Input: JSON with "keywords": "plastic waste, water saving"
    - Output: A list of actionable eco-friendly suggestions.
- **POST /summarize-policy**
    - Description: Summarizes uploaded policy PDF or pasted policy text.
    - Input:
        - "pdf_file" → PDF document (optional)

        - "policy_text" → Raw policy text (optional)
    - Output: Concise summary including **Overview, Key Provisions, and Implications**.
- **POST /extract-pdf-text**
    - Description: Extracts raw text from an uploaded PDF file.
    - Input: PDF file.
    - Output: Plain extracted text from the document.
- **POST /generate-response**
    - Description: Core API for interacting with the IBM Granite LLM.
    - Input:
        - "prompt" → User query or instruction.
        - "max_length" (optional) → Maximum response length.
    - Output: AI-generated natural language response.

API Notes

- All functions return responses in **plain text** for easy integration with the Gradio UI.
- Errors such as unreadable PDF files return a descriptive error message.
- The APIs are lightweight and optimized for both **CPU and GPU environments**.

8. Authentication

The current version of the **Sustainable Smart City Assistant** runs in an **open environment** for demonstration purposes. This means that users can directly access the system without the need for login credentials or tokens.

For a **secure deployment**, the following authentication mechanisms can be integrated:

- **Token-Based Authentication (API Keys or JWT)**
    - Protects access to the AI model and ensures only authorized users can interact with the backend APIs.
- **OAuth2 Integration**
    - Can be connected with **IBM Cloud** or third-party identity providers for secure login.
- **Role-Based Access Control (RBAC)**
    - Assigns roles such as *Administrator, Citizen, Researcher, or Policymaker* to restrict access to specific features.

- **Session Management**
  - Maintains user history, preferences, and past queries securely.

Planned Enhancements

- Multi-user support with personalized dashboards.
- User-specific eco-tip recommendations based on previous interactions.
- Secure storage and encryption for uploaded policy documents.

9. User Interface

The **Sustainable Smart City Assistant** provides a simple and user-friendly interface built using **Gradio**. The design prioritizes accessibility for non-technical users, ensuring smooth interaction with the system.

Key Elements of the Interface

- **Tabbed Layout**
  - Two main tabs:
    1. **Eco Tips Generator** – for generating sustainability tips.
    2. **Policy Summarization** – for analyzing and summarizing documents.
- **Eco Tips Generator Tab**
  - Input: Textbox to enter environmental problems or keywords (e.g., plastic, water saving, solar energy).
  - Action: Button to generate eco-friendly tips.
  - Output: Large textbox displaying practical sustainability suggestions.
- **Policy Summarization Tab**
  - Input: Option to upload a PDF policy document or paste policy text directly.
  - Action: Button to summarize the policy.
  - Output: Large textbox showing **concise summaries, key provisions, and implications**.
- **File Upload Support**
  - Allows users to upload PDF files for automatic text extraction and summarization.
- **Minimalist Design**
  - Clear labels and structured layout for better navigation.
  - Large output areas for easy readability.

User Experience (UX) Considerations

- Designed to be **intuitive**, requiring no technical expertise.
- Works in both **local environments** and **shareable web links**.
- Supports **real-time interaction**, providing instant results for user queries.

10. Testing

The **Sustainable Smart City Assistant** was tested in multiple phases to ensure correctness, reliability, and usability. Different levels of testing were carried out on both the **Eco Tips Generator** and **Policy Summarization** modules.

Types of Testing Performed

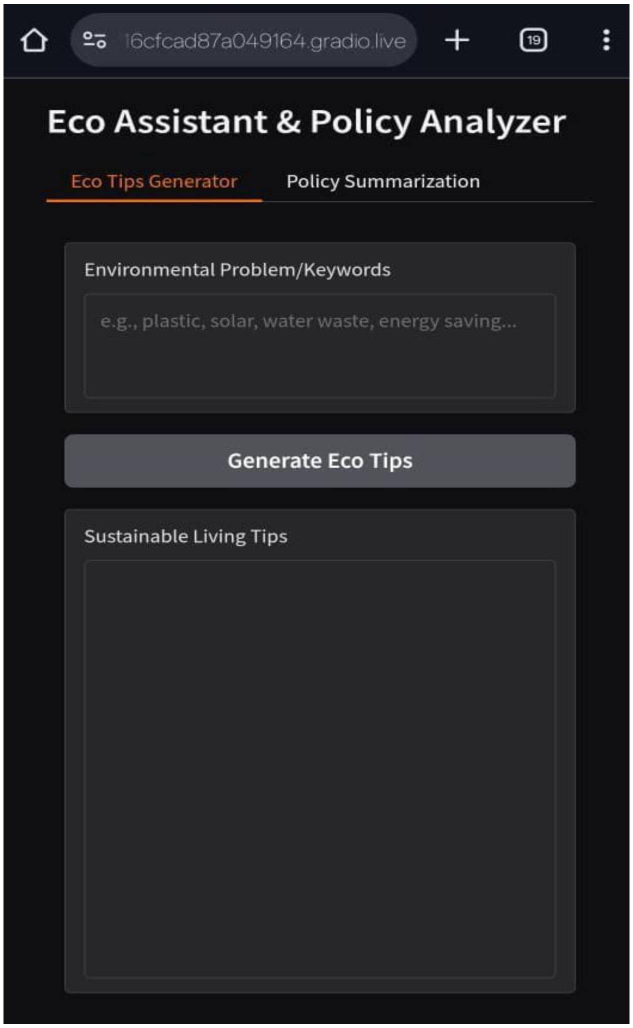
- **Unit Testing**
    - Verified individual functions such as generate_response, extract_text_from_pdf, eco_tips_generator, and policy_summarization.
    - Ensured that inputs and outputs were handled correctly.
- **Integration Testing**
    - Confirmed smooth interaction between the Gradio frontend, AI model backend, and PDF extraction module.
    - Verified that extracted text was correctly passed to the summarization function.
- **Functional Testing**
    - Tested the Eco Tips Generator with various environmental keywords to validate that results were relevant and actionable.
    - Checked Policy Summarization with multiple PDF files and pasted texts to confirm accuracy of generated summaries.
- **Performance Testing**
    - Measured response time on both CPU and GPU setups.
    - Verified that the system could handle large documents within reasonable time limits.
- **User Acceptance Testing (UAT)**
    - Gathered feedback from sample users (students and general users).
    - Confirmed that the system was easy to use, and outputs were understandable and practical.
- **Edge Case Handling**

  - Tested with empty inputs, corrupted PDFs, and very long texts to ensure error handling was appropriate.

Results

- The system generated **relevant eco-tips** and **concise summaries** consistently.
- Users found the **Gradio interface** easy to navigate.
- Minor issues were observed with scanned PDFs (non-text data), which will be addressed in future enhancements.

11.Screen shots


16cfcad87a049164.gradio.live
19
Eco Assistant & Policy Analyzer
Eco Tips Generator
Policy Summarization
Environmental Problem/Keywords
e.g., plastic, solar, water waste, energy saving...
Generate Eco Tips
Sustainable Living Tips

12. Known Issues

While the system performs effectively in generating eco tips and summarizing policies, a few limitations and issues were identified during testing:

- **PDF Extraction Limitations** – Scanned or image-based PDFs are not fully supported since text cannot be extracted without OCR.
- **AI Output Variability** – Responses may occasionally be generic or inconsistent depending on input phrasing.
- **Performance Constraints** – Running the IBM Granite LLM on lower-end systems without GPU can cause slow response times.
- **No Offline Mode** – The system requires internet access for model loading and dependencies, limiting offline usability.
- **Limited Personalization** – Eco tips are general and not yet fully customized to user profiles or real-time urban data.

13. Future Enhancement

To improve functionality and broaden applicability, the following enhancements are planned:

- **OCR Support** – Integrate OCR tools (e.g., Tesseract) to handle scanned and image-based PDFs.
- **Advanced Personalization** – Provide user-specific eco tips based on lifestyle, location, or preferences.
- **Mobile Application** – Develop Android and iOS apps for easier citizen engagement.
- **Multilingual Support** – Extend outputs to multiple regional and global languages for wider adoption.
- **Cloud Deployment** – Host the assistant on cloud platforms (IBM Cloud, AWS, or Azure) for scalability.
- **Data Visualization** – Introduce dashboards with charts/graphs to represent policy insights and sustainability metrics visually.
- **Integration with IoT and Smart Sensors** – Connect with real-time city data (energy, water, waste) for dynamic recommendations.
- **Enhanced Security** – Add stronger authentication and encryption for document handling in enterprise/government use cases.