Sustainable Smart City Assistant

1. Introduction

- **Project Title**: Sustainable Smart City Assistant
- **Team ID:** NM2025TMID03914
- **Team Leader**: Janani S
- **Team Member**: Santhiya S
- **Team Member**: Thaisha Shree D
- **Team Member**: Vijayadharshini G
- **Team Member**: Mohammed Umar H

2. Project Overview

Purpose:
This project is designed to empower citizens and officials with AI-powered tools for sustainability. It provides practical eco-friendly tips, simplifies lengthy policies, and aims to extend into forecasting, anomaly detection, and citizen feedback.

Existing Features:

- Eco-Tip Generator – Gives actionable advice for sustainable living.
- Policy Summarization – Converts long documents into concise summaries.

Planned Features:

- KPI Forecasting (energy, water, waste).
- Anomaly Detection in data.
- Citizen Feedback collection.
- Multimodal Input (PDF, text, CSV).
- Sustainability Report Generator.

3. Architecture

- **Frontend**: Built using Gradio UI with tabs for Eco Tips and Policy Summarization.
- **Backend**: Uses PyTorch and Hugging Face Transformers with IBM Granite LLM for text processing.
- **Planned Enhancements**: Integration with FastAPI backend, Pinecone/FAISS vector database, and ML modules for forecasting.

4. Setup Instructions

Run the following command to install required libraries:

!pip install transformers torch gradio PyPDF2 -q

Gradio will provide a link to access the UI in your browser after launching the app.

5. Folder Structure

Suggested project structure:

```
├── app/                # Backend logic (future FastAPI integration)
│   ├── granite_llm.py     # Handles model communication
│   ├── document_tools.py  # PDF/text extraction helpers
│   ├── forecast.py        # KPI forecasting (planned)
│   ├── anomaly.py         # Anomaly detection (planned)
├── ui/                 # Frontend components
│   ├── eco_tips_tab.py
│   ├── policy_summary_tab.py
├── janani_nm_project.py   # Main entry file with Gradio UI
├── requirements.txt       # Dependencies
```

6. Running the Application

1. Install dependencies.
2. Launch the Gradio app.
3. Open the provided link in browser.
4. Use the “Eco Tips” tab or “Policy Summarization” tab to interact.
5. Planned: Upload CSV for forecasting, submit feedback, and download reports.

7. API Documentation (Planned with FastAPI)

- POST /eco-tips – Generate eco-friendly tips.
- POST /summarize-policy – Summarize uploaded text/PDF.
- POST /forecast-kpi – Predict energy/water/waste usage.
- POST /detect-anomaly – Detect unusual data patterns.
- POST /submit-feedback – Collect citizen input.

8. Authentication

Currently runs in open environment. Future secure deployments will support:

- Token-based authentication (JWT or API keys).
- OAuth2 with IBM Cloud credentials.
- Role-based access (Citizen, Official, Researcher).

9. User Interface

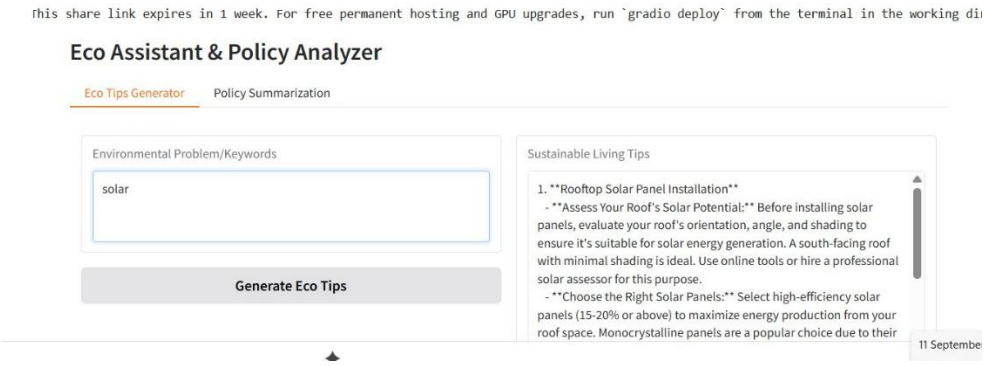
- Gradio UI with tabs.
- Eco-Tips tab: Input keywords, get tips.

- Policy Summarization tab: Upload PDF or paste text, get summary.
- Planned tabs: Forecasting, Feedback, and Reports.

10. Testing

- **Unit Testing**: For prompt and text extraction functions.
- **Manual Testing**: File uploads, summarization, and eco tips validation.
- **Planned**: API testing with Swagger/Postman.
- **Edge Cases**: Handling empty files, unreadable PDFs, invalid text inputs.

11. Screen Shots



12. Known Issues

- Limited to text and PDF inputs currently.
- No forecasting or anomaly detection module yet.
- No authentication in demo version.

13. Future Enhancement

- Add forecasting and anomaly detection modules.
- Expand input formats (CSV, Excel).
- Implement role-based authentication.
- Integrate with Streamlit dashboards for visualization.
- Generate downloadable sustainability reports.
- Add chatbot-style assistant for real-time queries.

PROJECT DEMO VIDEO LINK: https://drive.google.com/file/d/14_zLa-b1cPe585MlfpfNWaqdLCk5tZ8i/view?usp=drive_link