**Sustainable Smart City Assistant**

**1. Introduction**

* **Project Title:** Sustainable Smart City Assistant
* **Team ID:**
* **Team Leader:**
* **Team Members:**

**2. Project Overview**

**Purpose:**  
This project empowers citizens and officials with AI-powered tools to promote sustainability. It delivers practical eco-friendly tips, simplifies lengthy policies, and is designed to expand into forecasting, anomaly detection, and citizen engagement.

Current Features:

* Eco-Tips Generator – Provides actionable advice for sustainable living.
* Policy Summarization – Simplifies lengthy policy documents into concise summaries.

**Planned Features:**

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

**3. Architecture**

* Frontend: Built with Gradio, using tabs for Eco Tips and Policy Summarization.
* Backend: Uses Hugging Face Transformers and IBM Granite LLM for text generation.
* Planned Enhancements: FastAPI backend, vector databases (Pinecone/FAISS), and machine learning modules for forecasting and anomaly detection.

**4. Setup Instructions**

**Install dependencies:**

pip install transformers torch gradio PyPDF2 -q

Run the Python script to launch the app. Gradio will provide a local/shareable link to access the UI in your browser.

**5. Folder 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 Gradio app entry

│── requirements.txt # Dependencies

**6. Running the Application**

1. Install dependencies.
2. Run janani\_nm\_project.py.
3. Open the Gradio link in your browser.
4. Use the Eco Tips tab or Policy Summarization tab.
5. *(Planned)* Upload CSVs for forecasting, submit feedback, and download sustainability reports.

**7. API Documentation (Planned with FastAPI)**

* POST /eco-tips – Generate eco-friendly tips
* POST /summarize-policy – Summarize text or uploaded PDFs
* POST /forecast-kpi – Predict energy/water/waste usage
* POST /detect-anomaly – Identify unusual patterns
* POST /submit-feedback – Collect citizen input

**8. Authentication**

Currently runs in an open environment. Planned enhancements:

* Token-based authentication (JWT/API keys)
* OAuth2 with IBM Cloud
* Role-based access (Citizen, Official, Researcher)

**9. User Interface**

* Eco-Tips tab: Input keywords → get sustainability tips.
* Policy Summarization tab: Upload PDF or paste text → get a summary.
* *(Planned)* Additional tabs for Forecasting, Feedback, and Reports.

**10. Testing**

* Unit tests for text extraction and prompts.
* Manual testing: eco tips generation and PDF summarization.
* *(Planned)* API testing with Swagger/Postman.
* Edge case handling: empty files, unreadable PDFs, invalid text input.

**11. Known Issues**

* Works only with text/PDF input.
* Forecasting and anomaly detection not yet available.
* No authentication in demo version.

**12.Screen Short**

**13. Future Enhancements**

* Add forecasting and anomaly detection.
* Expand inputs (CSV, Excel).
* Role-based authentication.
* Visualization dashboards.
* Downloadable sustainability reports.
* Chatbot-style assistant for real-time help.