**Citizen AI – Intelligent Citizen Engagement Platform**

**1. Introduction:**

**Project Title:**

Citizen AI – Intelligent Citizen Engagement Platform

**Team Members:**

* Nithyasri B(Team leader)
* Nandhini R
* Neelufar A
* Nethra Shree B

**2. Project Overview:**

**Purpose**:  
 Citizen AI is designed to empower citizens and policymakers by providing sustainable living tips and summarizing lengthy government policy documents. By leveraging AI-driven natural language processing, it simplifies complex policy texts and generates practical eco-friendly guidance.

**Features**:  
 - Conversational Eco-Tips Generator: Generates personalized, actionable eco-friendly lifestyle suggestions.  
 - Policy Summarization: Extracts key points, provisions, and implications from uploaded PDFs or pasted text.  
 - PDF Document Handling: Uses PyPDF2 to read and process uploaded policy files.  
 - Gradio Web UI: Provides a simple, user-friendly interface with tab-based navigation.

**3. Architecture:**

**Frontend** (Gradio):

Provides a minimal web interface with two main tabs:  
- Eco Tips Generator (input keywords → get sustainability suggestions)  
- Policy Summarization (upload PDFs or paste text → get concise summary).

**Backend** (Transformers + PyTorch):

Uses Hugging Face’s ibm-granite/granite-3.2-2b-instruct model for natural language understanding and response generation. Deployed via Gradio Blocks with tabbed layouts.

**Modules**:  
- Text Generation: Generates eco tips and summaries.  
- PDF Processing: Extracts text from uploaded files.  
- Policy Summarization: Converts documents into actionable summaries.

**4. Setup Instructions:**

**Prerequisites**:  
 - Python 3.9+  
 - pip and virtual environment tools  
 - Internet access for model download

**Installation** **Steps**:  
 1. Clone the repository.  
 2. Install dependencies:  
 pip install transformers torch gradio PyPDF2  
 3. Run the app:  
 python citizen\_ai.py  
 4. Access the Gradio web interface in your browser.

**5. Folder Structure:**

citizen\_ai.py # Main application file  
 requirements.txt # Dependencies (to be created)  
 README.md # Project documentation (optional)

**6. Running the Application:**

- Launch the Gradio app by running citizen\_ai.py.  
 -Navigate to the interface in your browser.  
 -Select Eco Tips Generator or Policy Summarization tabs.  
 -Upload documents or input keywords.  
 -View generated outputs in real-time.

**7. API Documentation:**

Currently, the project runs as a Gradio UI app and does not expose separate REST APIs. Planned enhancements may include:  
 - REST endpoints for eco tips generation.  
 - Policy summarization API for external integration.

**8. Authentication:**

Current version is open-access for demo purposes. Future versions may include:  
 - Token-based authentication.  
 - Role-based access (citizen, policymaker, admin).

**9. User Interface:**

**Tabs**:  
 - Eco Tips Generator  
 - Policy Summarization  
 **Inputs**:  
 - Textbox for keywords or policy text  
 - File upload for PDFs  
 **Outputs**:  
 - Textbox showing AI-generated eco tips or policy summary

**10. Testing**

**Unit** **Testing**:  
 - Verified text extraction from PDFs.  
 - Tested model inference for eco tips and summarization.

**Manual** **Testing**:  
 - Checked Gradio UI functionality.  
 - Validated edge cases (empty input, corrupted PDFs).

**11. Known Issues:**

- Long documents may exceed model context length.  
 - First-time model download may take time.  
 - No offline mode (requires internet for model).

**13. Future Enhancements:**

- REST API integration with FastAPI.  
 - Database support for saving user queries and summaries.  
 - Improved UI with dashboards and charts.  
 - Multi-language support for policy summarization.