Citizen AI – Intelligent Citizen Engagement Platform

Project Documentation

1. **Introduction**

Project Title: Citizen AI – Intelligent Citizen Engagement Platform
Team Leader: Ramya M
Team Members: Nidhi Pandey A
Team Members: Shahanaz Banu J
Team Members: Roofiya Yasmeen N

2) **Project Overview:**

Purpose:

- Empower citizens with simplified summaries of lengthy government policies.
- Provide actionable eco-friendly lifestyle tips.
- Collect structured citizen feedback for officials.
- Build a scalable and accessible AI-driven interface.
- Increase public awareness about sustainable living.

Features:

- Eco-Tips Generator: Generates practical suggestions for sustainable living.
- Policy Summarization: Transforms long documents into concise, actionable insights.
- Conversational Interface: Accepts natural language inputs from citizens.
- Citizen Feedback Loop: Enables citizens to submit feedback.
- Data Reports (future): Supports strategic planning for officials.
- Custom Alerts: Notify users about updates in policies or new eco-friendly initiatives.

3. **Architecture**

Frontend (Gradio):

- User-friendly dashboard with tabs for Eco Tips and Policy Summarization.
- Intuitive design with minimal learning curve for citizens.

Backend (FastAPI/Functions):

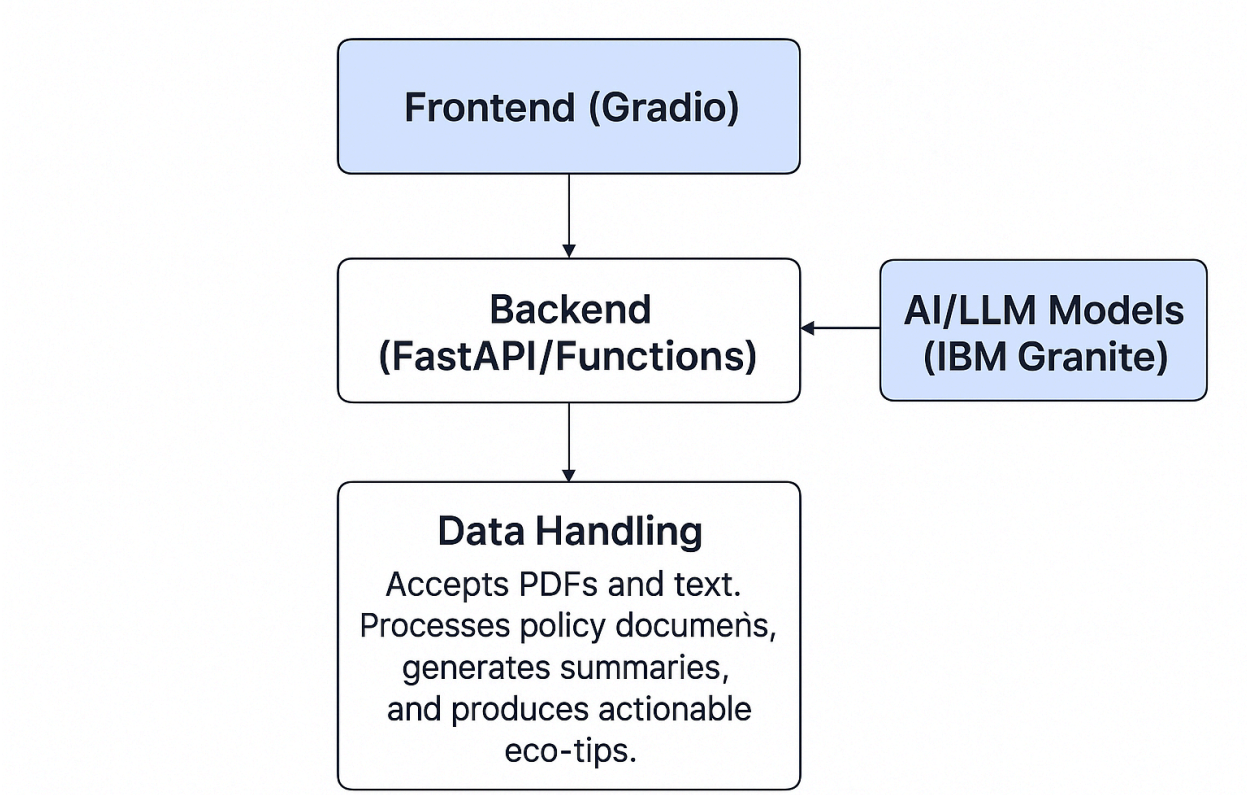
- Handles document parsing, summarization, and eco-tip generation.
- Ensures fast response times and smooth user experience.

AI/LLM Models (IBM Granite):

- Provides natural language understanding and summarization.
- Capable of contextual responses based on citizen queries.

Data Handling:

- Accepts PDFs, text files, and keyword inputs.
- Processes policy documents, generates summaries, and produces actionable eco-tips.
- Stores user feedback securely for analysis and reporting.



4. **Prerequisites**

- Python 3.9 or later
- pip and venv tools
- Libraries: Transformers, Torch, Gradio, PyPDF2
- Internet access for AI integration
- Recommended: 8GB RAM or higher for optimal performance

5. **Setup Instructions**

1.Install required dependencies:

pip install transformers torch gradio PyPDF2

2. Download the IBM Granite model:

model_name = "ibm-granite/granite-3.2-2b-instruct"

3. Run the backend functions.

4. Launch Gradio interface:

app.launch(share=True)

5. Upload a PDF or enter policy text.

6. Generate eco-tips or summaries.

7. Optional: Configure local storage paths for outputs in the outputs/ folder.

6. **Folder Structure**

citizen_ai_project/
app.py → Main Gradio application
eco_tips_generator.py → Eco tips logic
policy_summarizer.py → Policy summarization logic
utils/ → Helper functions
file_handler.py → Handles PDF/text input
logger.py → Logs system events
requirements.txt → Project dependencies
outputs/ → Stores saved outputs/screenshots
README.md → Project overview and usage instructions

7. **Running the Application**

Launch the Gradio interface (app.py).

Choose between two tabs:

1. Eco Tips Generator → Enter environmental keywords.

2. Policy Summarization → Upload PDF or paste text.

Receive summarized outputs or eco-tips in real-time.

8. **API Documentation**

Even though the app is built with Gradio UI, backend functions mimic API behavior:

POST /eco-tips – Generates eco-tips based on keywords.

POST /summarize-policy – Summarizes uploaded documents or text.

Example: Input: { "topic": "plastic waste" } Output: { "eco_tip": "Switch to reusable bottles and reduce single-use plastics." }

Error Handling:

Returns descriptive messages for invalid inputs.

Handles large PDF files gracefully with partial processing.

9. **Authentication**

(Current Version)

Open demo (no authentication).

(Future Version)

JWT token-based access.

Role-based authentication for Citizens vs Officials.

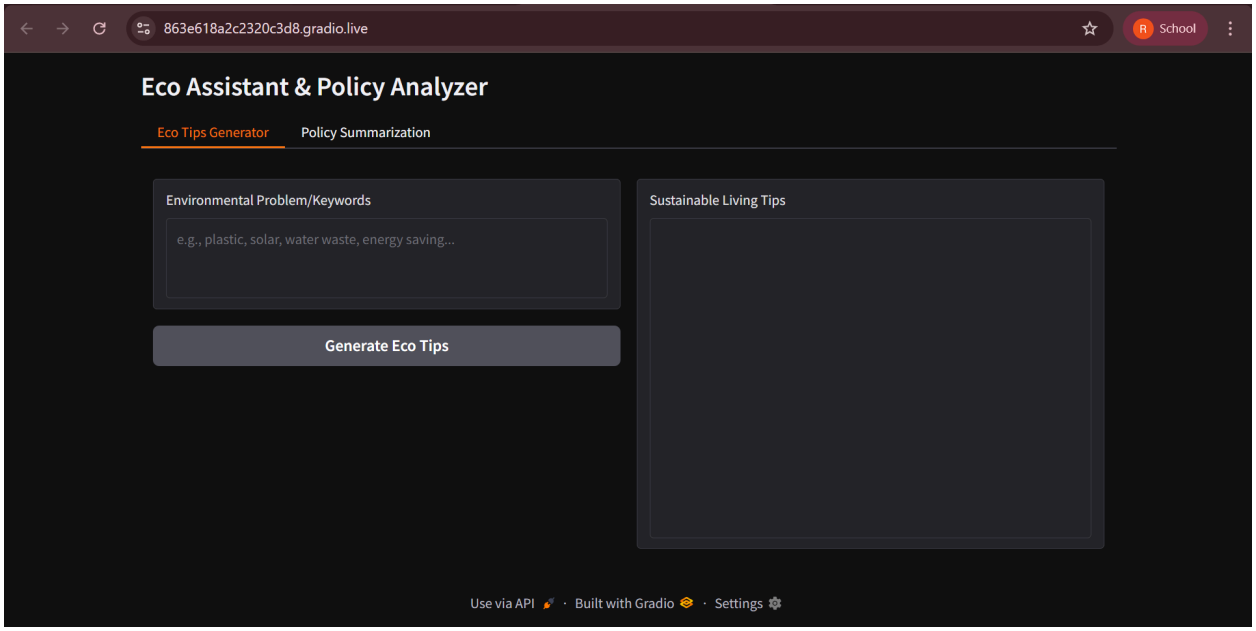
OAuth2 integration for secure access.

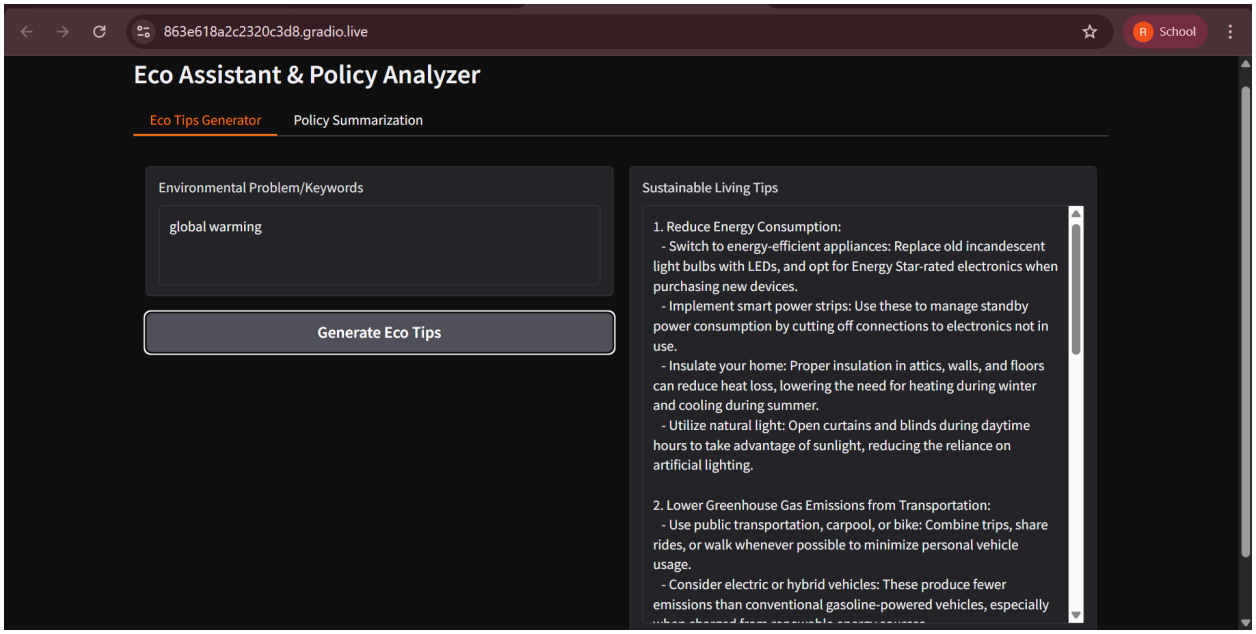
Password encryption for stored accounts.

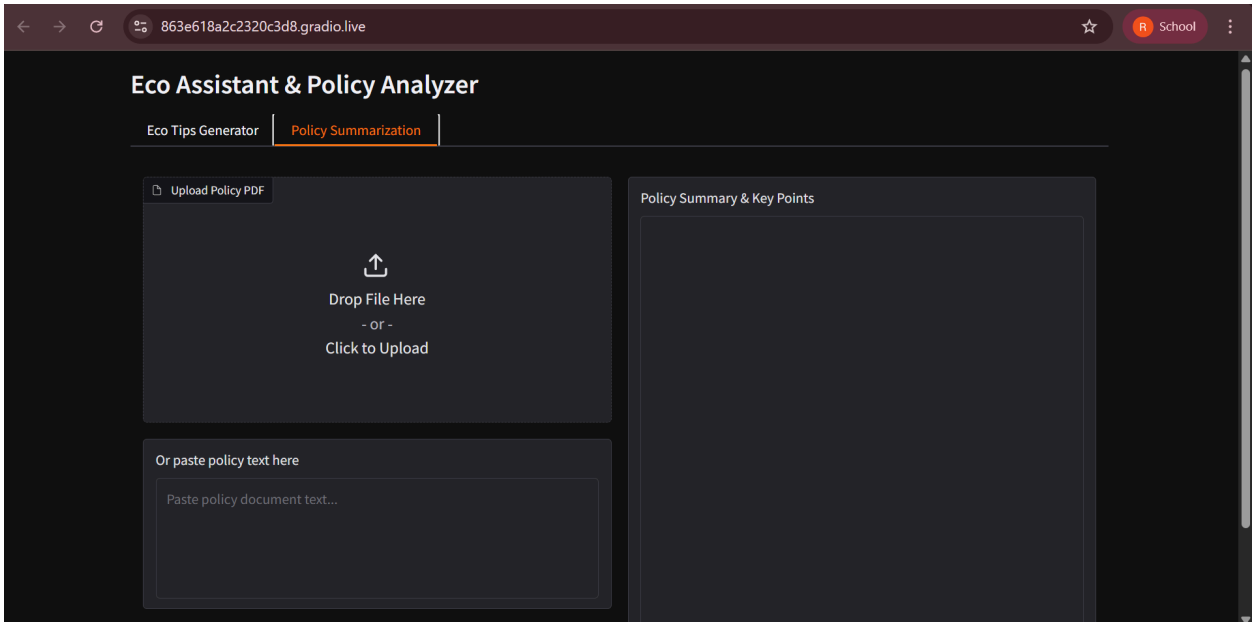
10. **User Interface**

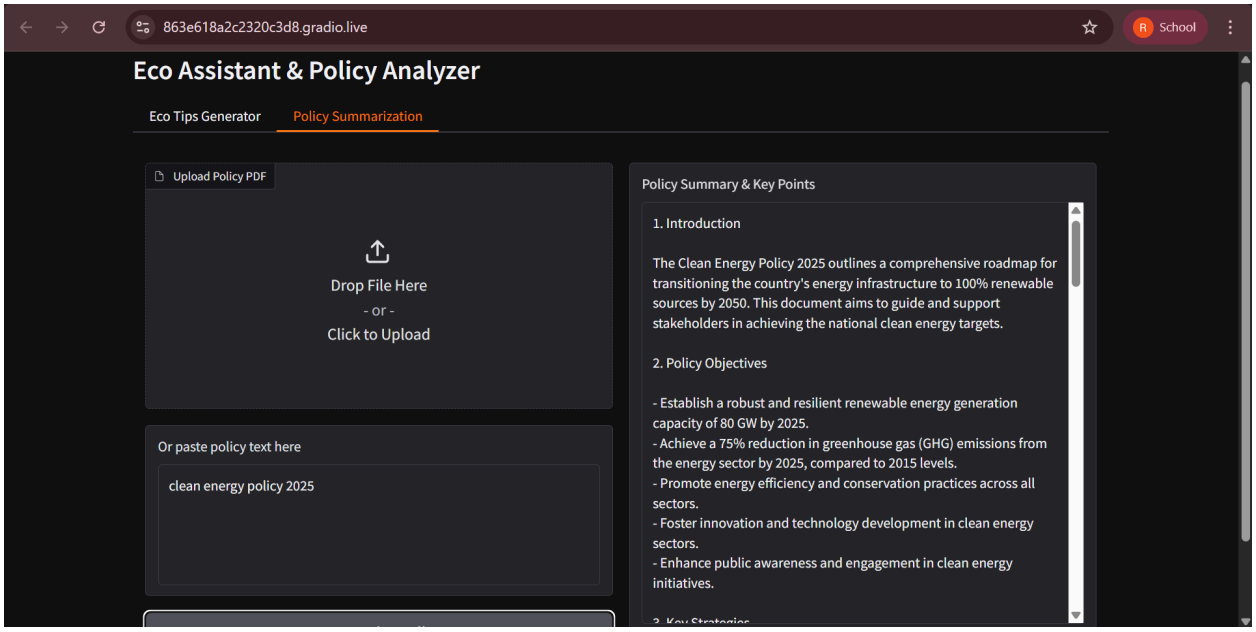
- Eco Tips Tab: Users input environmental keywords.
- Policy Summarization Tab: Users upload PDFs or enter text.
- Feedback Tab: Citizens can submit suggestions or report issues.
- Outputs displayed in a simple, readable format.
- Shareable link available for remote access.

11. **Screenshots**




863e618a2c2320c3d8.gradio.live
School
Eco Assistant & Policy Analyzer
Eco Tips Generator
Policy Summarization
Environmental Problem/Keywords
global warming
Generate Eco Tips
Sustainable Living Tips
1. Reduce Energy Consumption:
- Switch to energy-efficient appliances: Replace old incandescent light bulbs with LEDs, and opt for Energy Star-rated electronics when purchasing new devices.
- Implement smart power strips: Use these to manage standby power consumption by cutting off connections to electronics not in use.
- Insulate your home: Proper insulation in attics, walls, and floors can reduce heat loss, lowering the need for heating during winter and cooling during summer.
- Utilize natural light: Open curtains and blinds during daytime hours to take advantage of sunlight, reducing the reliance on artificial lighting.
2. Lower Greenhouse Gas Emissions from Transportation:
- Use public transportation, carpool, or bike: Combine trips, share rides, or walk whenever possible to minimize personal vehicle usage.
- Consider electric or hybrid vehicles: These produce fewer emissions than conventional gasoline-powered vehicles, especially





11. Testing

Unit Testing:

Each module (eco-tips, summarizer, feedback) tested with sample inputs.

Edge cases like empty inputs and large files verified.

API Testing:

POST requests tested for consistency.

Response validation performed for JSON output.

Manual Testing:

Uploaded various policy PDFs and observed summaries.

Tested eco-tip generation with keywords like energy saving, water usage, solar power.

Checked UI responsiveness and usability.

Performance Testing:

Verified processing time for large PDFs.

Assessed memory and CPU usage for backend functions.

13. **Known Issues**

Large PDF processing can be slow.

Current model only supports English text.

UI not fully mobile-optimized.

Some rare keywords may return generic eco-tips.

Feedback data not yet fully integrated with analytics.

14. **Future Enhancements**

- Add multilingual support (regional languages).
- Expand to mobile-friendly responsive UI.
- Integrate IoT/sensor data for real-time eco-monitoring.
- Enhance authentication (JWT/OAuth2).
- Build advanced analytics dashboards for officials.
- Enable push notifications for policy updates.
- Incorporate AI-driven sentiment analysis for citizen feedback.
- Support downloadable reports and eco-tips in PDF format.