CITIZEN AI: INTELLIGENT CITIZEN

ENGAGEMENT PLATFORM

**1.Introduction**

* Project Title: Citizen AI – Intelligent Citizen Engagement Platform
  + Team member:Sanjana.R
  + Team Members: Sandhiya.T
  + Team members:Srilekha.A
  + Team member:Eshwari.k

**2. Project Overview**

* Purpose

The purpose of this application is to provide an AI-powered assistant that supports sustainability and policy understanding through two core functionalities: eco-advice generation and policy summarization. By IBM Granite’s advanced language model, the system helps citizens adopt sustainable practices by offering practical and actionable eco-friendly tips tailored to specific environmental issues such as waste management, energy saving, and water conservation. At the same time, it assists policymakers, researchers, and the public by simplifying complex policy documents, extracting the most important points, provisions, and implications to make them easier to understand and apply. Through its simple Gradio-based interface, the application bridges the gap between citizens and sustainability knowledge, promoting informed decision- making and active participation in building smarter and greener communit

**Features**

**1. City Analysis Module**

Generates safety insights for any city.

Covers crime index, accident rates, and overall safety assessment.

**2. Citizen Services Module**

Responds to queries on public services, policies, and civic issues.

Provides structured, context-aware responses.

**3. AI-Powered Insights**

Built on IBM Granite LLM for natural language understanding.

Delivers context-aware and high-quality responses.

**4. Interactive Interface**

Developed with Gradio Blocks.

Provides tab-based navigation for easy usage.

**5. Flexible Deployment**

Runs locally or can be shared publicly (share=True).

**3.Architecture**

Frontend (Gradio)

**Two tabs:**

1. City Analysis (textbox + button + output area)

2. Citizen Services (textbox + button + output area)

**Simple UI with instant results.**

Backend (Python)

**Contains core functions:**

* generate\_response() → Generates AI-powered answers.
* city\_analysis() → Provides city-specific safety analysis.
* citizen\_interaction() → Handles citizen service queries.

**LLM Integration (IBM Granite LLM)**

Uses ibm-granite/granite-3.2-2b-instruct.

Tokenization, prompt construction, and safe max token handling.

Supports GPU acceleration for efficiency.

**Deployment Layer**

* Hosted via Gradio.
* Supports local + public links for demonstrations.

**4. Setup Instructions**

* Prerequisites
* Python 3.9 or later
* pip & virtualenv
* Internet access (to fetch model & run Gradio)

**Installation**

1. Clone the repository.

2. Install dependencies:

pip install -r requirements.txt

3. Run the script:

python citizen\_ai.py

**5. Folder Structure**

citizen-ai/

│

├── citizen\_ai.py # Main application file

├── requirements.txt # Dependencies (transformers, torch, gradio)

├── README.md # Project documentation

├── data/ # (Optional) City datasets

├── outputs/ # (Optional) Logs or saved responses

**6. Running the Application**

* Run python citizen\_ai.py
* Gradio launches at http://127.0.0.1:7860

**Features:**

* City Analysis Tab → Enter a city name and get crime/accident data.
* Citizen Services Tab → Enter a query and get AI-powered civic response.

**7. API Documentation**

Future integration (example):

* POST /city-analysis → Accepts city name, returns crime & accident stats.
* POST /citizen-query → Accepts query, returns civic response.
* (Currently handled via Gradio UI).

**8. User Interface**

City Analysis Tab

Input: City name

Output: Safety statistics

Citizen Services Tab

Input: Free-text query

Output: Government-style response

**9. Testing**

* Unit Testing → Tested generate\_response, city\_analysis, citizen\_interaction.
* Manual Testing → Validated inputs/outputs on Gradio UI.
* Edge Cases → Empty queries, invalid city names, long prompts.

**10. Known Issues**

* Long prompts may cause slow responses.
* City statistics depend on AI-generated context (not real-time).
* First-time model load may take time.
* Currently no authentication or database integration.

**11. Future Enhancements**

* Add real-time integration with open data APIs (crime, traffic).
* Authentication layer for secure citizen access.
* Multi-language support for global usage.
* Mobile-friendly dashboards.
* Analytics & visualization of city trends.