**Project Documentation**

**Citizen AI Project – City Analysis & Citizen Services**

**1. Introduction**

**Project Title:** Citizen AI – City Analysis & Citizen Services

**Team Members:**

* THIRUMURUGAN S
* SANTHOSH KUMAR M
* VIGNESH M
* SANTHURU M
* YAKHESHWAR D

The **Citizen AI** project is a modern artificial intelligence system designed to act as a **virtual civic assistant**. It analyzes safety conditions in cities and provides **direct citizen services** by answering queries related to government policies, civic issues, and public services.

In today’s world, access to trustworthy and clear civic data is often difficult for citizens. At the same time, city administrators require analytical insights into safety trends to make better decisions. **Citizen AI bridges this gap** by using AI-driven analysis and conversational interactions to serve both citizens and government bodies.

**2. Project Overview**

**Purpose**

The purpose of Citizen AI is to combine **natural language understanding** with **urban safety analysis** into one accessible platform. The system runs on IBM Granite LLM and is deployed with Gradio for an interactive experience.

**Objectives**

1. **City Analysis** – Generate crime and accident analysis reports.
2. **Citizen Interaction** – Provide helpful responses to governance-related queries.
3. **Accessibility** – Offer a user-friendly interface usable by citizens, policymakers, and researchers.
4. **Scalability** – Enable future integration of more datasets such as healthcare, transport, and education.

**Features**

* **City Analysis** – Crime index, accident statistics, and overall safety assessments.
* **Citizen Services** – Query handling for government policies and public services.
* **GPU Acceleration** – Optimized to run on Google Colab with T4 GPU.
* **Simple UI** – Clean tab-based interface for easy navigation.
* **Modularity** – Functions can be extended for new domains and services.

**3. Architecture**

Citizen AI follows a **modular client-server design**:

* **Frontend (Gradio):**
  + Interactive UI with two tabs: *City Analysis* and *Citizen Services*.
  + Inputs for city names or queries, outputs in neatly formatted textboxes.
* **Backend (Python + IBM Granite LLM):**
  + Natural language processing and inference.
  + Custom prompt engineering ensures responses remain relevant to civic and governance topics.
* **Key Libraries:**
  + transformers – Load and run IBM Granite model.
  + torch – GPU acceleration for deep learning computations.
  + gradio – Build user-facing web interface.

**Workflow**

1. User provides input (city name or query).
2. System tokenizes the input and passes it to IBM Granite model.
3. Model generates structured responses.
4. Output is displayed instantly in Gradio UI.

**4. Setup Instructions**

**Prerequisites**

* Python 3.9+
* pip package manager
* Internet connectivity
* Google Colab (preferred environment)

**Installation Process**

1. Open the notebook file in Google Colab.
2. Change runtime type → GPU (T4 GPU recommended).
3. Install dependencies:
4. !pip install transformers torch gradio -q
5. Load the IBM Granite model and tokenizer.
6. Launch the Gradio application.
7. Access the UI via the public link generated.

This makes deployment seamless without requiring heavy local setups.

**5. Folder Structure**

* **CitizenAI\_Project.ipynb** – Jupyter notebook containing complete code.
* **Functions:**
  + generate\_response – Generic AI prompt-response handler.
  + city\_analysis – Provides safety and accident details for a given city.
  + citizen\_interaction – Handles citizen queries about civic and government services.
* **Gradio Interface:**
  + *City Analysis Tab* – Enter city name, get AI-generated report.
  + *Citizen Services Tab* – Ask governance-related queries, get structured responses.

**6. Running the Application**

1. Open the notebook in Google Colab.
2. Install dependencies and initialize the model.
3. Launch the Gradio interface.
4. Navigate between two main tabs:
   * **City Analysis Tab** – Input city name and retrieve analysis report.
   * **Citizen Services Tab** – Ask questions on public services or government issues.
5. Review AI-generated outputs.
6. Adapt prompts or extend functionality for specific use cases.

**7. API Documentation**

* **generate\_response(prompt)**
  + Input: Prompt string
  + Output: AI-generated text
* **city\_analysis(city\_name)**
  + Input: City name string
  + Output: Safety analysis report (crime index, accident stats, overall safety)
* **citizen\_interaction(query)**
  + Input: User query string
  + Output: AI-generated explanation tailored to governance context

**8. Authentication**

Currently runs in **open access** mode. For production:

* Implement **OAuth authentication** for citizen accounts.
* Use **role-based access control** for administrators vs. citizens.
* Add **API keys** for third-party integrations.

This ensures security and accountability in real-world deployment.

**9. Use Cases**

* **City Administrators** – Safety and accident insights for planning.
* **Citizens** – Easy access to government-related information.
* **Researchers & NGOs** – Analyzing AI-driven insights for policy recommendations.
* **Academia** – Teaching applications of AI in governance.
* **Developers** – Extending system by integrating new civic datasets.

**10. Testing**

* **Unit Testing:** Validated functions with controlled inputs.
* **Integration Testing:** Verified smooth interaction between AI and Gradio.
* **Edge Case Testing:** Empty queries, invalid city names.
* **Performance Testing:** Compared CPU vs. GPU runtimes.

Findings confirmed stable functionality across scenarios.

**11. Known Issues**

* City analysis reports are AI-generated, not real-time official stats.
* Limited customization due to Gradio constraints.
* Slow performance when executed on CPU instead of GPU.

**12. Future Enhancements**

* Real-time integration with government crime & safety APIs.
* Visualization dashboards (charts, graphs, maps).
* Multilingual query support.
* Feedback system for citizens to rate responses.
* Deployment as standalone mobile/web app.
* Fine-tuned AI model trained on civic datasets.