**Citizen AI - Intelligent Citizen Engagement Platform**

**Project Documentation**

**1. Introduction**

**Project Title:** Citizen AI - Intelligent Citizen Engagement Platform  
**Team Members:**

* Team member: S. Afrin
* Team member: A. Arockia Handalisa
* Team member: H. J. Amrees Jafin
* Team member: A. Aarthi

Citizen AI enhances citizen-government interactions with AI-driven, real-time responses and detailed city safety analytics.

**2. Project Overview**

**Purpose:**  
Empower citizens and officials with accurate, accessible info on public services, policies, and city safety using advanced AI.

**Key Features:**

* City safety analysis (crime, accidents).
* Citizen query handling.
* AI chatbot with Gradio UI.
* GPU-optimized IBM Granite LLM.

**3. Architecture**

**Frontend:**

* Gradio-based, tabbed UI for City Analysis and Citizen Services.

**Backend:**

* IBM Granite LLM via PyTorch for NLP.
* Prompt engineering for generating responses.

**4. Setup Instructions**

* Python 3.7+ with GPU (recommended).
* Install dependencies (transformers, torch, gradio).
* Run main script (e.g., citizen\_ai.py) in GPU-enabled environment (Google Colab recommended).

**5. Folder Structure**

* Root folder: project source and configs
* citizen\_ai.py: main app script
* requirements.txt: dependencies
* README.md: documentation
* utils/: helper scripts
* models/: stored ML models
* data/: sample datasets
* docs/: additional documentation
* scripts/: optional helpers
* tests/: automated tests

**6. Running the Application**

Run on GPU (e.g., Google Colab), install dependencies, launch citizen\_ai.py to open the Gradio UI.

**7. API Documentation**

* generate\_response(prompt, max\_length) — core response generator
* city\_analysis(city\_name) — city safety report generator
* citizen\_interaction(query) — citizen query handler

Potential REST endpoints (if extended):

* POST /analyze-city
* POST /citizen-query

**8. Authentication**

* Open environment (demo)
* Future: JWT, OAuth2, role-based access

**9. User Interface**

* Tabbed layout for City Analysis and Citizen Services
* Input/output textboxes with buttons
* Minimalist and responsive design

**10. Testing**

* Unit, manual UI, environment, and edge case testing

**11. Known Issues**

Slow responses, limited context, data quality dependence, no voice support, no security, English-only, limited scalability.

**12. Future Enhancements**

Better context, voice & multilingual support, live data, security, mobile app, user feedback, scalable cloud deployment and dashboards.

**13.screen shots:**

