CITIZEN AI - INTELLIGENT CITIZEN ENGAGEMENT PLATFORM

# 1. Introduction

• Project Title: CITIZEN AI - INTELLIGENT CITIZEN ENGAGEMENT PLATFORM

• Team Members:

1. Priyadharshini P.C (Team Lead)

2. Preethi B

3. Preethiyanga P

4. Pushpaja G

# 2. Project Overview

• Purpose:  
The purpose of Citizen AI is to serve as an intelligent citizen engagement platform that enables governments to provide quick, reliable, and AI-powered responses to citizens’ queries. It helps improve transparency, accessibility, and efficiency in governance by leveraging natural language models for civic issue resolution, policy information, and city safety analysis.

• Features:

- City Analysis: Provides detailed safety and accident insights of a given city.

- Citizen Services: Answers queries related to public services, policies, and civic issues.

- Conversational Interface: Natural language interaction for accessibility.

- AI-Powered Insights: Uses IBM Granite model for intelligent responses.

- Gradio Web Interface: Simple and user-friendly dashboard with tabs for services.

# 3. Architecture

Frontend (Gradio):  
The frontend is designed with Gradio, providing an interactive tab-based UI for city analysis and citizen queries. It includes input textboxes, buttons, and output sections to display results in real-time.

Backend (Transformers + PyTorch):  
The backend uses Hugging Face Transformers and PyTorch to load IBM Granite LLM for natural language processing tasks such as query answering and city safety analysis.

LLM Integration (IBM Granite):  
Granite model processes prompts and generates reliable responses to both city analysis and citizen queries.

# 4. Setup Instructions

Prerequisites:  
• Python 3.9 or later  
• Google Colab or local Python environment  
• Installed packages: transformers, torch, gradio

Installation Process:  
• Clone the project or upload the file citizen\_ai.py  
• Install dependencies using pip  
• Run the project in Google Colab with T4 GPU enabled for faster performance  
• Execute the script to launch the Gradio web app

# 5. Folder Structure

citizen\_ai.py – Main project file containing frontend (Gradio) and backend logic  
No additional folders are required since this is a single-file implementation.

# 6. Running the Application

To start the project:  
➢ Run citizen\_ai.py in Google Colab or your local environment  
➢ Launch the Gradio web app (a public shareable link is generated)  
➢ Use the 'City Analysis' tab to analyze crime and accident data of a city  
➢ Use the 'Citizen Services' tab to ask questions related to government policies or services

# 7. API Documentation

This project does not expose traditional REST APIs. Instead, it provides AI responses through the Gradio interface. Functions include:  
- city\_analysis(city\_name): Generates crime and safety insights for a city  
- citizen\_interaction(query): Provides government-related information for a citizen query

# 8. Authentication

Currently, the project runs in an open environment without authentication. Future versions may include secure access control, API keys, or role-based restrictions for government officials and citizens.

# 9. User Interface

The interface is created using Gradio with two main tabs:  
• City Analysis Tab – Accepts a city name input and returns crime and accident analysis  
• Citizen Services Tab – Accepts a query and provides government policy/service responses

# 10. Testing

Testing was performed in the following ways:  
- Functional Testing: Verified city analysis and citizen interaction outputs  
- Manual Testing: Checked Gradio UI flow and response accuracy  
- Edge Case Handling: Tested with empty inputs and uncommon city names

# 11. Screenshots

[Add screenshots of the Gradio interface showing City Analysis and Citizen Services tabs]

# 12. Known Issues

- Responses may vary depending on the LLM output  
- No persistent storage or database integration  
- Requires GPU for faster inference

# 13. Future Enhancements

- Integration with government databases for real data  
- Role-based access control  
- Support for multiple languages  
- Mobile app interface