Citizen AI

Project Documentation

**1.Introduction**

* Project Title: AI-Powered City Analysis and Citizen Service Assistant
* Team Member : Rajitha.Y
* Team Member : Priya.K
* Team Member : Rafiya Khatun.S
* Team Member : Rithanya Shree.B

**2.Project Overview**

The purpose of this project is to provide a comprehensive AI-powered solution for analyzing city metrics such as crime index, accident rates, and citizen service queries. The assistant leverages IBM Granite 3.2 model to generate insights and provide real-time responses. The goal is to help city officials and citizens interact with government data more efficiently.

• Conversational Interface: Natural language interaction for citizens and officials.

• City Analysis: Provides detailed crime, accident, and safety analysis for cities.

• Citizen Services: Answers public queries related to government policies and civic issues.

• Policy Summarization: Simplifies lengthy government documents into actionable summaries.

• Eco-Tip Generator: Recommends daily actions for sustainability and eco-conscious behavior.

• PDF Report Download: Users can download city analysis reports.

**3.System Architecture**

**Frontend**: Built using Gradio for interactive web-based UI.

**Backend**: Python with Hugging Face Transformers for AI-powered generation.

**LLM Integration**: IBM Granite 3.2 model for city analysis and citizen services.

**4.Setup Instructions**

• Install Python 3.9 or later.

• Install dependencies: pip install -r requirements.txt.

• Run the Python script to start the Gradio app.

• Access the application via the provided localhost or public URL.

• Ensure GPU support for faster model inference (optional).

**5.Folder Structure**

project/ app.py # Main Gradio application requirements.txt # Dependencies models/ # Model files utils/ # Utility scripts.

**6.Running the Application**

• Run the command: python app.py requirements.txt # Dependencies

• Open the browser at http://localhost:7860 or the provided shareable link.

• Navigate between tabs: City Analysis and Citizen Services.

• Enter city names or queries to get real-time responses.

**7.API Documentation**

This application currently runs locally without external API endpoints. Future enhancements include FastAPI integration for backend services.

**8.Authentication**

Currently, the application does not include authentication. Future plans include:

• Token-based authentication using JWT or API keys.

• Role-based access control for citizens and administrators.

• User session and history tracking.

**9.User Interface Design**

The interface is built with Gradio featuring:

• Tabbed layout for easy navigation.

• Textboxes for entering city names and user queries.

• Real-time AI-generated responses.

• Clean and minimal design for better usability.

**10.Testing**

• Unit testing for core functions.

• Manual testing for Gradio UI and response quality.

• Edge case testing with invalid inputs and empty fields.

**11.Future Enhancements**

• Integrate external data sources like government APIs for real-time city metrics.

• Add authentication and role-based access.

• Deploy the application on cloud platforms.

• Enable PDF report generation directly from the app.

**12.Screen Shot**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.