Citizen AI

Project Documentation

**1.Introduction**

Project Title: Citizen AI

Team Member: Akshaya G

Team Member: Bhavadharani S

Team Member: Boomika dharshini I

Team Member: Deepika S

**2.Project Overview**

This project introduces an innovative solution for evaluating city performance indicators, including crime data, accident reports, and service requests. By leveraging advanced AI-driven models, the platform delivers actionable insights and provides precise responses to citizen queries. Its primary mission is to support local governments and communities through accurate, real-time data interpretation and service recommendations.

• Interactive Chat Interface: Facilitates natural communication between citizens and city services.

• Comprehensive City Analysis: Generates detailed reports on safety metrics and accident data.

• Public Query Handler: Delivers instant, accurate responses regarding public services.

• Policy Summary Generator: Condenses lengthy policies into easy-to-understand summaries.

• Eco-Friendly Tips: Suggests daily actions to encourage sustainable practices.

• PDF Export: Provides downloadable reports for offline review.

**3.System Architecture**

User Interface: Built with Gradio for a user-friendly, browser-based experience.

Core Engine: Developed using Python with Hugging Face for natural language processing.

Language Model: Uses IBM Granite 3.2 for generating responses and analysis.

**4.Setup Instructions**

• Ensure Python version 3.9 or higher is installed.

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

• Run the application using the main script provided.

• Access the web interface via the generated link or localhost.

• For faster processing, enable GPU if available.

**5.Folder Structure**

city\_analysis\_app/ main\_app.py # Entry point of the Gradio application requirements.txt # List of required dependencies configurations models/ # Model files and utils/ # Helper functions and utilities

**6.Running the Application**

• Start the program with: python main\_app.py

• Open your browser and go to http://localhost:7860 or the generated public URL.

• Switch between City Analysis and Citizen Services using the provided tabs.

• Enter data or queries to receive real-time AI-powered responses.

**7.API Documentation**

Currently, the application runs locally without external APIs. In future releases, FastAPI will be integrated to expose RESTful endpoints for advanced data sharing and third-party integration.

**8.Authentication**

Planned authentication features include:

• Secure JWT-based token authentication.

• Role-specific access for different user types (e.g., admin, citizen).

• Session management and user activity logging.

**9. User Interface Design**

• Tabbed navigation for modular workflows.

• Clean input forms for user queries and city data.

• Instant AI-driven analytics displayed in real time.

• Minimalist layout prioritizing ease of use and accessibility.

**10. Testing**

• Unit tests to validate core application logic.

• Manual testing of the Gradio interface and outputs.

• Edge case verification using malformed or missing inputs.

**11. Future Enhancements**

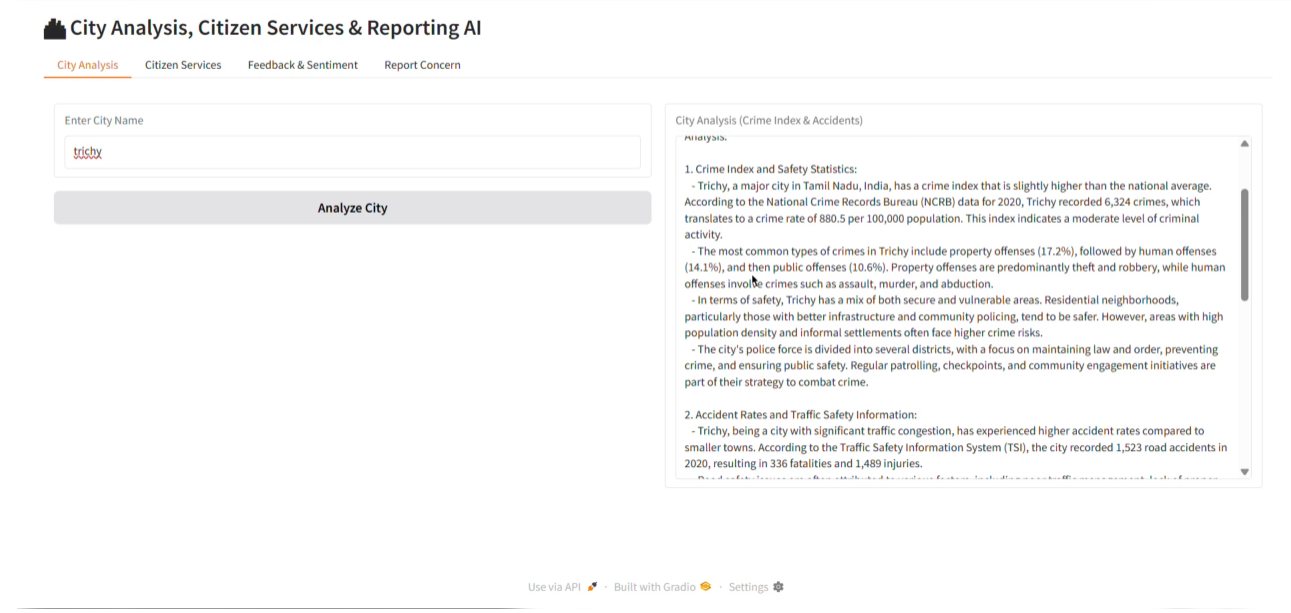
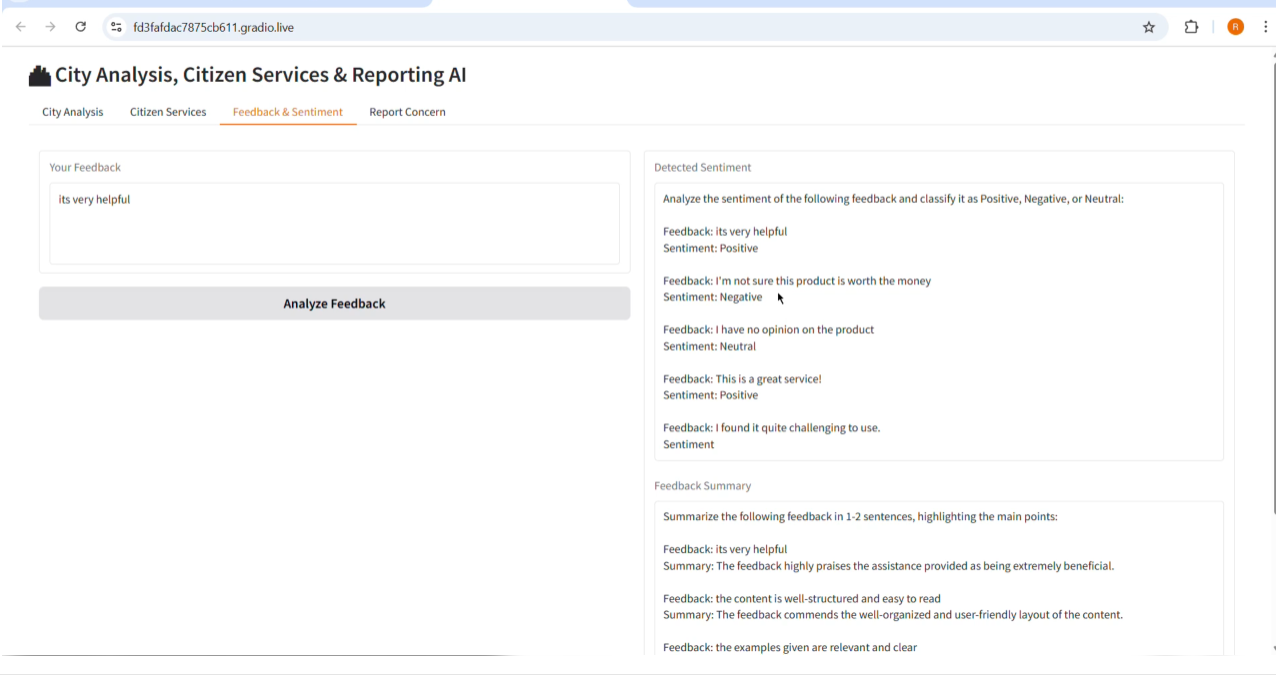
• Integrating live data from public city databases.

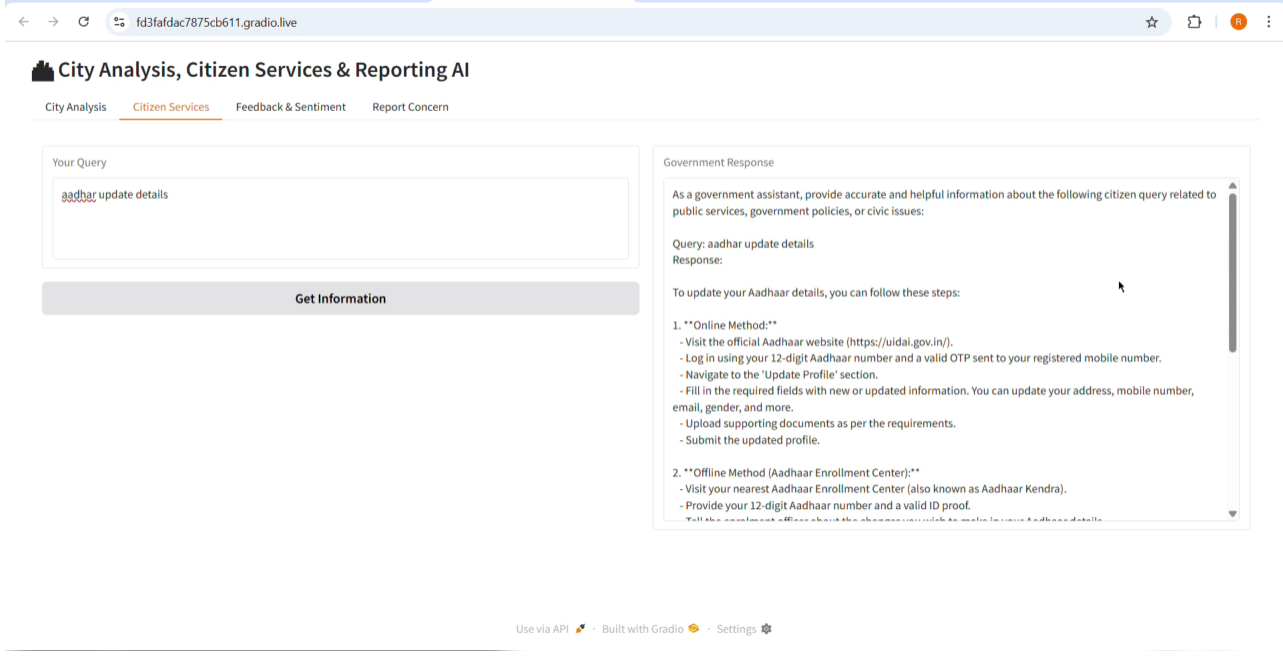
• Deploying secure user authentication and permissions.

• Hosting the solution on cloud platforms for scalability.

• Generating downloadable reports directly through the interface.

**12.Screenshot**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.