**Citizen AI**

**Project Documentation**

**1. Introduction**

* **Team Members:**
* Ansar Ali.A
* Buvanesh.S
* Shatrock.L
* Ashwin Kumar.J

**2. Project Overview**

Purpose: Citizen AI is an intelligent citizen engagement platform designed to enhance interaction between governments and the public. Using AI and real-time data, the assistant provides city safety insights, government service query resolution, and sentiment analysis of citizen feedback.

**Features:**

* Conversational Interface – Natural language interaction for citizens.
* City Safety Analysis – Summarized crime and accident statistics.
* Sentiment Analysis – Classifies citizen feedback.
* Citizen Feedback Dashboard – Visual analytics of sentiment distribution.
* Gradio UI – Interactive multi-tab dashboard.

**3. Architecture**

* Frontend: Built with Gradio (interactive UI).
* Backend: Python + Transformers + Torch.
* LLM Integration: IBM Granite model.
* Visualization: Matplotlib for charts.

**4. Setup Instructions**

* Python 3.9+ required.
* Install dependencies: transformers, torch, gradio, textblob, matplotlib.
* Run: python app.py
* Access Gradio dashboard in browser.

1. **Folder Structure**

project/

■■■ app.py # Main script

■■■ requirements.txt # Dependencies

■■■ models/ # Model configs

■■■ ui/ # Gradio layouts

■■■ utils/ # Utility functions

1. **Running the Application**

* Launch Gradio app.
* Use City Analysis, Citizen Assistant, Sentiment Analysis, and Dashboard tabs.

**7. API Documentation**

* POST /city-analysis – Returns AI-generated safety report.
* POST /citizen-assist – Handles citizen queries.
* POST /analyse-sentiment – Performs sentiment analysis.

**8.Authentication**

Currently open for demo use. Future enhancements include token-based authentication, role-based access (citizens, officials, admins).

**9.User Interface**

Tabbed layout for each feature, dashboard visualization of sentiment, and clean accessible design for non-technical users.

A screenshot of a computer

AI-generated content may be incorrect.

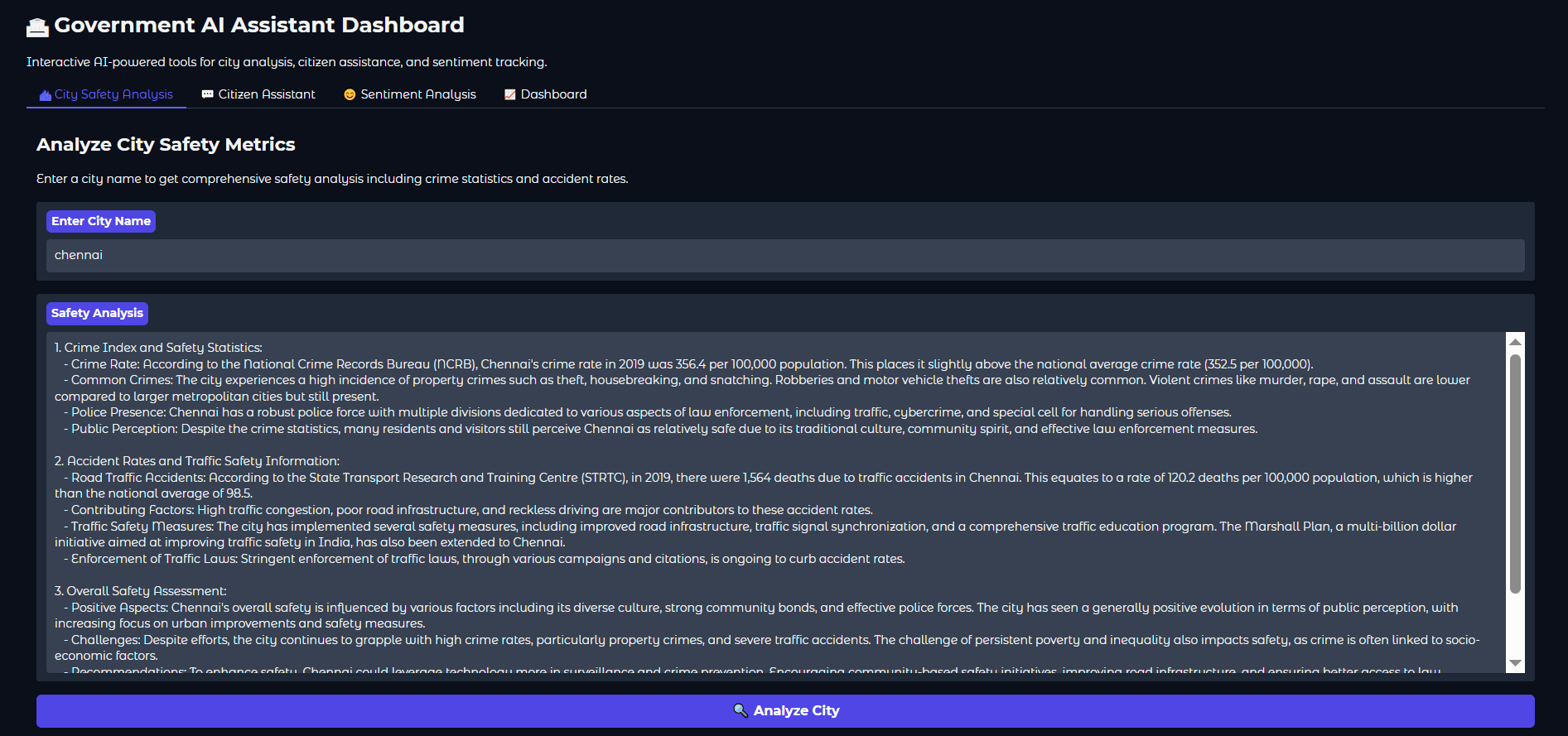
**10.Testing**

* Unit Testing – for AI response functions.
* Manual Testing – UI flows.
* Edge Cases – large queries, empty input.

**11.Screenshots**

A screenshot of a computer

AI-generated content may be incorrect.

****

**12.Conclusion**

This program demonstrates a Citizen AI Assistant Dashboard that integrates AI-powered analysis, citizen engagement, and data visualization into one interactive platform. Using transformers for natural language generation, TextBlob for sentiment analysis, and Matplotlib for visualization, it provides:

City Safety Analysis – Delivers crime and accident statistics through AI-driven reports.

Citizen Assistant – Acts as a conversational government helpdesk, answering policy and service-related queries.

Sentiment Analysis – Classifies citizen feedback as Positive, Neutral, or Negative.

Interactive Dashboard – Tracks and visualizes collective sentiment trends for better decision-making.

By combining AI models, real-time interactions, and visual analytics, this system can help governments, civic bodies, or organizations understand public safety concerns, citizen queries, and overall public sentiment.

**13. Known Issues**

* AI sometimes repeats prompt text.
* Model slower on CPU.

**14. Future Enhancements**

* Integration with real-time city data APIs.
* Multi-language support.
* Role-based authentication.
* Advanced analytics and forecasting.