CITIZEN AI

Project Documentation

• Introduction

- Project title : Citizen AI
- Team Head : Vignesh K
- Team member : Aravinth V
- Team member : Sunilkumar K
- Team member : Tamil Mani K E

• Project Overview

- Purpose :

The purpose of Citizen AI is to empower citizens and officials by providing AI-driven insights. It helps analyze city safety (crime and accident statistics) and also assists with public services, government policies, and civic issues. By leveraging AI and natural language models, it makes information accessible and supports effective decision-making.

- Features:

City Analysis - Key Point: Automated reports, Functionality: Provides crime index, accidents, safety assessment.
Citizen Interaction - Key Point: Civic assistance, Functionality: Provides responses for queries about government services.
Conversational Interface - Key Point: Natural language, Functionality: Simplifies citizen interaction. User-Friendly Interface - Key Point: Gradio UI, Functionality: Two-tab interface for analysis and services.

• Architecture

Frontend (Gradio): Provides an interactive web UI with tabs for City Analysis and Citizen Services.
Backend (Transformers + PyTorch): Model inference using Hugging Face Transformers and PyTorch. LLM Integration (Granite 3.2 2B Instruct): IBM Granite model used for generating structured AI responses.

• Setup Instructions

Prerequisites:

- Google Colab with T4 GPU runtime
- Python 3.9 or later
- Internet connection

Installation Process:

- Open project in Colab
- Change runtime to GPU (T4)
- Run: !pip install transformers torch gradio -q
- Execute notebook and launch Gradio app

• Folder Structure

app/ – Backend logic (future FastAPI integration)
ui/ – Frontend Gradio components
models/ – Model artifacts and tokenizer setup
notebooks/ – Google Colab notebook for demo
reports/ – Documentation and screenshots

• Running the Application

- Set runtime to T4 GPU in Colab
- Run installation cell for dependencies
- Load model and tokenizer
- Launch Gradio app (share link generated)
- Navigate to:
- City Analysis Tab → Enter city name for safety analysis
- Citizen Services Tab → Enter query for civic information

• API Documentation

Available functions:

- generate_response(prompt): Core AI text generator
- city_analysis(city_name): Generates safety analysis report
- citizen_interaction(query): Handles citizen service queries

• Authentication

Current version: No authentication (demo only).

Future Enhancements:

- Token-based authentication (JWT or API keys)
- Role-based access (citizen, admin, officials)
- Secure sessions for user history

• User Interface

The interface includes:

- Tabbed layout (City Analysis, Citizen Services)
- Input fields for city names and queries
- Text output for AI-generated responses
- Buttons for triggering analysis
- Shareable Gradio link for demos

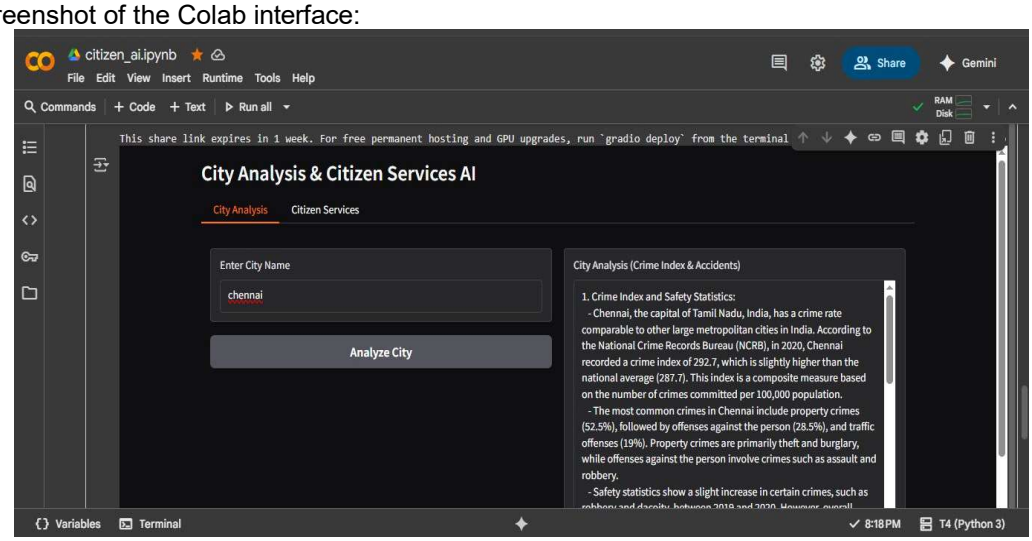
• Testing

Testing includes:

- Unit Testing: Prompt and function checks
- Manual Testing: City queries and policy queries
- Edge Case Handling: Invalid inputs, empty queries

• Screenshots

Screenshot of the Colab interface:



• Known Issues

- AI outputs may not match real statistics
- No real-time government data integration
- Gradio share links expire after Colab session ends

• Future Enhancement

- Integrate live APIs for crime and traffic data

- Add multilingual support (e.g., Tamil, Hindi)
- Implement REST API for broader use
- Add user authentication and dashboards