CITIZEN AI

Project Documentation

1. INTRODUCTION

Project Title : Citizen AI : Intelligent Citizen Engagement Platform.

- Team member : Varshani.V
- Team member : Ishwariya.R
- Team member : Varsha.B
- Team member : jayadharshini. J

2. PROJECT OVERVIEW

- **Purpose :**

Citizen AI is an AI-powered assistant created to provide comprehensive city safety analysis and responsive citizen services related to government and civic issues. The platform uses advanced language models to provide clear insights into urban safety data, such as crime rates and traffic incidents. It also offers accurate and relevant answers to questions from citizens. This helps people access up to date information about their local areas. The goal is to improve public engagement and support better decision-making by authorities.

- **Features :**

Natural Language Interaction
Key point : User-Friendly Communication.

Functionality: Allows users to interact naturally using simple language inputs for queries and requests.

Citizen Query Assistance
Key Point: Responsive Government Support.

Functionality: Answers citizen queries about public services, policies, and civic issues properly.

City Safety Analysis
*Key Poin*t:Urban Safety Insights.

Functionality: Provides detailed reports on crime statistics, accident rates, and overall city safety.

Dual-Tab User Interface
Key Point: Clear Functional Separation.

Functionality: Separates city analysis and citizen service tools into distinct tabs for ease of navigation.

Device-Aware Model Loading
Key Point: Optimized inference.

Functionality: Automatically detects and utilizes GPU if available for faster processing; otherwise defaults to CPU.

Shareable Web App
Key Point: Easy deployment and sharing.

Functionality: Uses Gradio's share feature to enable easy hosting and sharing of the application interface publicly.

3. ARCHITECTURE

Frontend (Gradio):

The frontend is built using Gradio, which provides an easy-to-use, interactive web interface. The interface consists of multiple tabs enabling users to either receive detailed city safety analyses or get responses to citizen queries related to government services. Gradio's Blocks API is used to organize the UI into rows and columns for clear layout, with input text boxes, buttons to trigger AI processing, and output areas to display generated responses. Gradio also supports sharing the app publicly via a generated URL.

Backend (Transformer Model with PyTorch):

The backend is powered by a pretrained language model, IBM Granite 3.2 instruct, integrated using Hugging Face's Transformers library. Model loading dynamically chooses hardware acceleration based on availability (GPU or CPU). Core backend functions handle prompt tokenization, inference with sampling for natural language generation, and response decoding. The backend logic is embedded within the same Python script that runs the Gradio frontend, keeping a lightweight deployment without a separate API layer.

Language Model Integration (IBM Watsonx Granite):

The IBM Granite model is fine-tuned for instruction-following, enabling task-specific prompt engineering to generate accurate and context-aware text responses. It provides two main capabilities — detailed city analysis based on input city names and responsive, helpful answers to citizen government queries.

4. SETUP INSTRUCTIONS

Prerequisites

- Python 3.9 or later
- Pip and virtual environment tools
- GPU is recommended
- Internet access to access pretrained model weights and dependencies

Installation Process:

- Clone the repository.
- Create and activate a virtual environment for dependency isolation.
- Install dependencies from requirements.txt
- Run the application script to start the Gradio web interface on the computer.

5. FOLDER STRUCTURE

- app/ – Contains the main backend logic, including model loading, tokenization, prompt handling, and AI response generation functions.
- ui/ – Holds all frontend components developed using Gradio, including the multi-tab user interface with city analysis and citizen services features.
- model_integration.py – Manages loading and interfacing with the IBM Watsonx Granite language model, performing prompt construction, tokenization, and decoding of responses.
- app_launcher.py – The entry script that launches the Gradio web app, sets up the UI components, and initiates interaction event handlers.
- auth.py – Contains simple user authentication logic used to control access to the application interface.

6. RUNNING THE APPLICATION

To start the project:

- Run the application script to launch the Gradio web interface on your computer.
- Once launched, the interface will open in a web browser, displaying two tabs for City Analysis and Citizen Services.
- Use the City Analysis tab to enter a city name and get detailed safety reports.
- Use the Citizen Services tab to input citizen queries and receive government-related information.
- All interactions happen in real-time, with the backend language model generating responses dynamically.
- If enabled, authentication is required to access the main app functionality.

Frontend (Gradio):

The frontend is built entirely with Gradio, offering an interactive, easy-to-use interface. The app includes multiple tabs, with text input boxes, action buttons, and text output areas arranged for clarity. The design allows users to easily switch between city analysis and citizen service functions.

Backend (Hugging Face Transformers & IBM Granite):

The backend resides within the same script, utilizing the IBM Granite 3.2 instruction-following language model loaded via Hugging Face Transformers. It performs prompt processing, model inference, and response generation, all triggered by frontend inputs.

7. API DOCUMENTATION

- POST /city_analysis — Accepts a city name as input and returns a detailed safety analysis including crime statistics and traffic accident rates.

- POST /citizen_query — Accepts a citizen's query related to government services or civic issues and responds with an AI-generated answer.

- POST /login — Handles user authentication with username and password verification (simple login system).

8. AUTHENTICATION

The application includes a simple username and password authentication system built directly into the Gradio interface. This basic login mechanism restricts access to the main functionality of the app and is designed primarily for demonstration purposes.

- Users must enter valid credentials to gain access to the City Analysis and Citizen Services features.
- Unlike Swagger-based APIs, authentication here is part of the frontend interface flow rather than exposed API endpoints.
- The authentication is implemented within the Gradio Blocks UI using a straightforward login form and credential check.
- For secure deployment, more advanced authentication methods should be integrated
    - Token-based authentication (JWT or API keys)
    - OAuth2 with IBM Cloud credentials
    - Role-based access (admin, citizen, researcher)

9. USER INTERFACE

The interface of CITIZEN AI is minimalist and functional, designed to be easy to use for non-technical users. It focuses on clear interaction and accessibility.

Tabbed Layout

Text Input Boxes

Action Buttons

Text Output Areas

Immediate Feedback

Shareable Interface

Responsive Layout

The design emphasizes clarity, speed, and ease of use, guiding users easily through labels and placeholder texts without overwhelming them with technical details.

10. TESTING

Testing was done in multiple phases:

Unit Testing: For prompt generation functions and utility scripts.

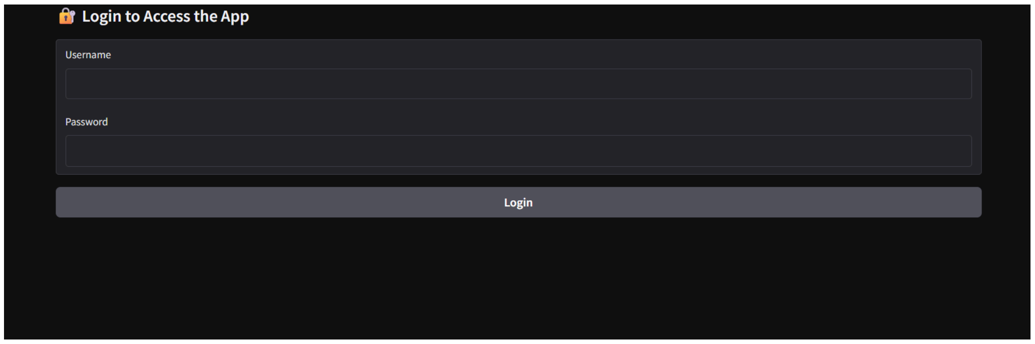
API Testing: Using Swagger UI, Postman, and test scripts.

Manual Testing: Chat responses, and output consistency.

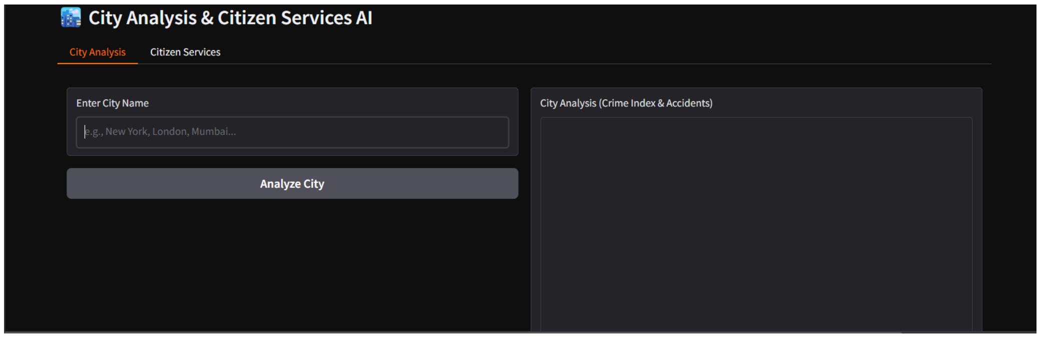
Edge Case Handling: Including malformed inputs, large files, and invalid API keys.

11. SCREENSHOTS

LOGIN PAGE:

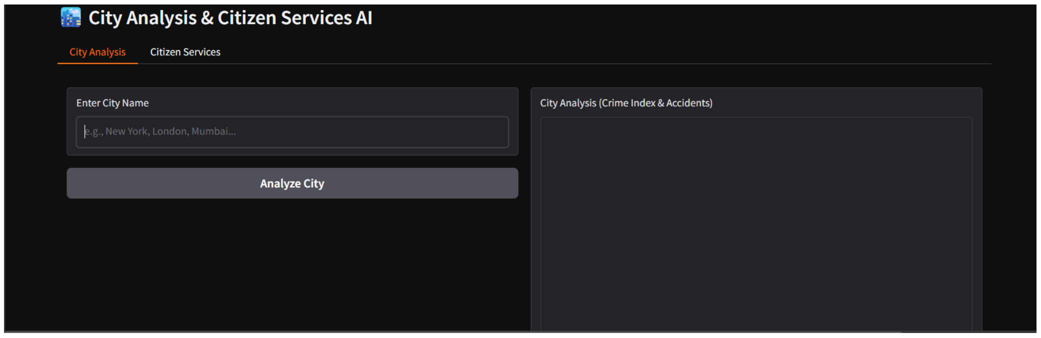


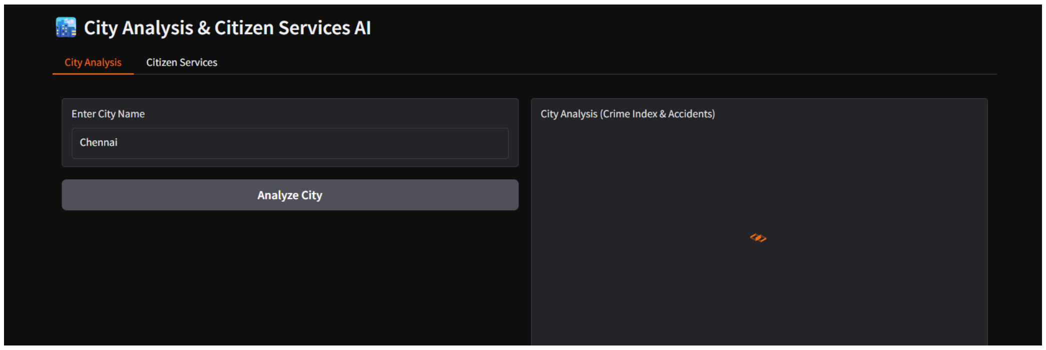
CITY ANALYSIS & CITIZEN SERVICES AI PAGE :

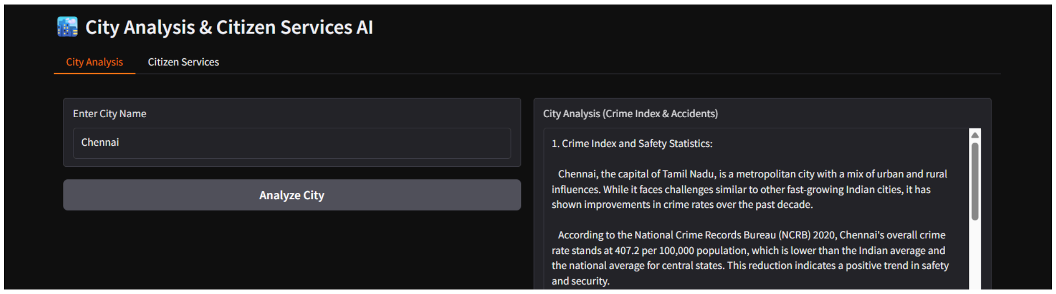

City Analysis & Citizen Services AI
City Analysis
Citizen Services
Enter City Name
e.g., New York, London, Mumbai...
Analyze City
City Analysis (Crime Index & Accidents)

CITY ANALYSIS TAB:

Use the City Analysis tab to enter a city name and get detailed safety reports.


City Analysis & Citizen Services AI
City Analysis
Citizen Services
Enter City Name
e.g., New York, London, Mumbai...
Analyze City
City Analysis (Crime Index & Accidents)


City Analysis & Citizen Services AI
City Analysis
Citizen Services
Enter City Name
Chennai
Analyze City
City Analysis (Crime Index & Accidents)


City Analysis & Citizen Services AI

City Analysis | Citizen Services

Enter City Name: Chennai

Analyze City

City Analysis (Crime Index & Accidents)

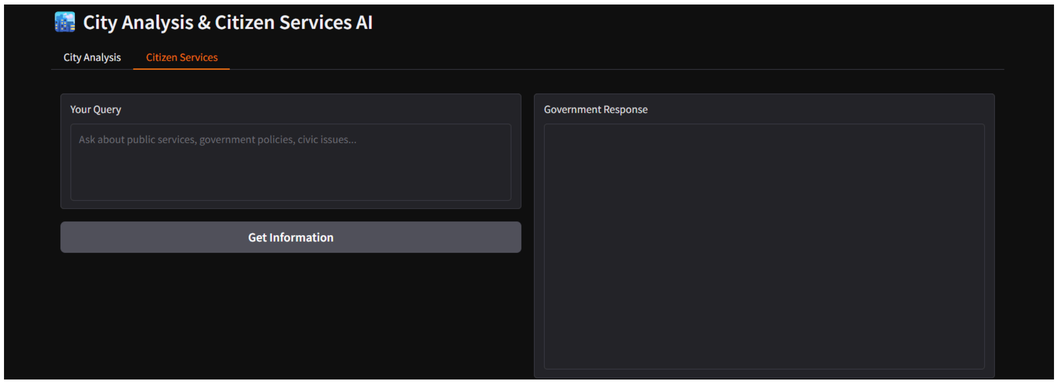
1. Crime Index and Safety Statistics:

Chennai, the capital of Tamil Nadu, is a metropolitan city with a mix of urban and rural influences. While it faces challenges similar to other fast-growing Indian cities, it has shown improvements in crime rates over the past decade.

According to the National Crime Records Bureau (NCRB) 2020, Chennai's overall crime rate stands at 407.2 per 100,000 population, which is lower than the Indian average and the national average for central states. This reduction indicates a positive trend in safety and security.

CITIZEN SERVICES TAB :

Use the Citizen Services tab to input citizen queries and receive government-related information.

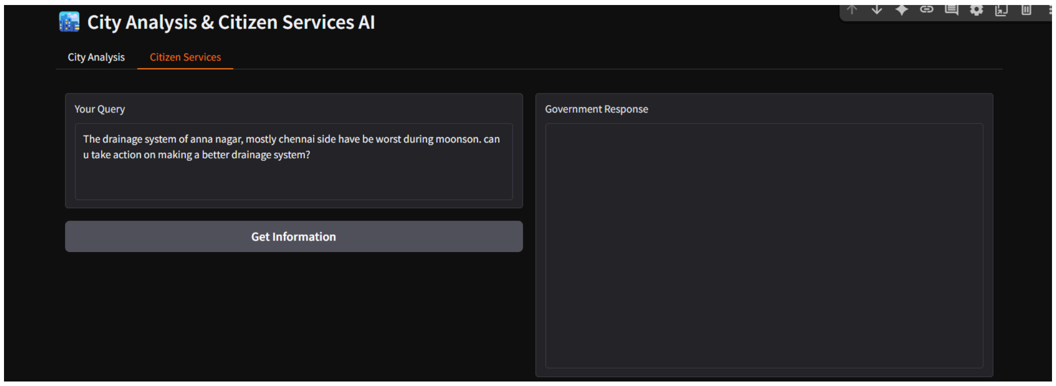

City Analysis & Citizen Services AI

City Analysis | Citizen Services

Your Query: Ask about public services, government policies, civic issues...

Get Information

Government Response

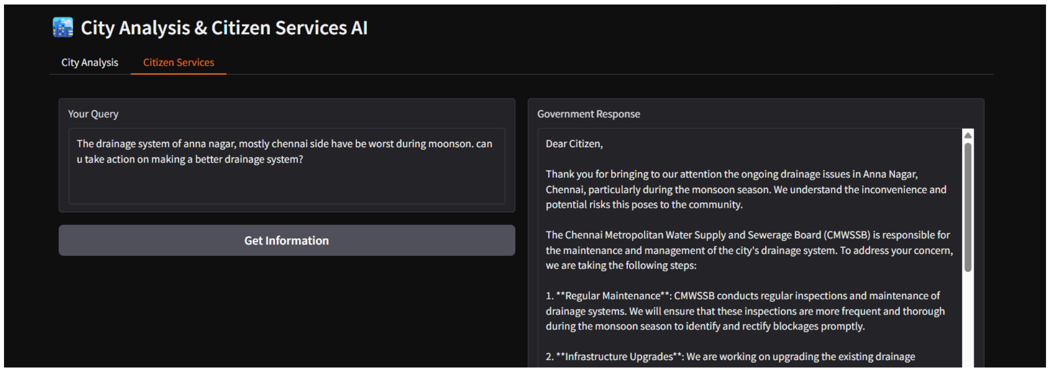

City Analysis & Citizen Services AI

City Analysis | Citizen Services

Your Query: The drainage system of anna nagar, mostly chennai side have be worst during moonson. can u take action on making a better drainage system?

Get Information

Government Response



Your Query

The drainage system of anna nagar, mostly chennai side have be worst during moonson. can u take action on making a better drainage system?

Government Response

Dear Citizen,

Thank you for bringing to our attention the ongoing drainage issues in Anna Nagar, Chennai, particularly during the monsoon season. We understand the inconvenience and potential risks this poses to the community.

The Chennai Metropolitan Water Supply and Sewerage Board (CMWSSB) is responsible for the maintenance and management of the city's drainage system. To address your concern, we are taking the following steps:

1. **Regular Maintenance**: CMWSSB conducts regular inspections and maintenance of drainage systems. We will ensure that these inspections are more frequent and thorough during the monsoon season to identify and rectify blockages promptly.

2. **Infrastructure Upgrades**: We are working on upgrading the existing drainage

12. KNOWN ISSUES

- **Response Specificity:** The AI-generated responses might sometimes be too general or less specific due to the nature of the prompt and model limitations.
- **Performance Variability:** Inference time can be slow on CPU-only machines, causing noticeable delays in response generation.
- **Limited Input Validation**: The app currently has basic input validation; unusual or very long inputs might lead to unexpected responses or truncation.
- **No Multilingual Support** : The interface supports queries and responses only in English, limiting accessibility for non-English speakers.

13. FUTURE ENCHANCMENTS

- **Real-Time Data Integration:** Incorporate live city data feeds for crime statistics, traffic accidents, and safety alerts to provide up-to-date analysis.

- **Multilingual Support:** Extend the interface and AI responses to support multiple languages, increasing accessibility for diverse user groups.

- **Expanded Citizen Services:** Add more detailed government policy information, document search, and FAQs to support a wider range of citizen queries.

- **Visual Analytics and Reports:** Integrate graphical visualizations such as charts and maps to supplement text-based city analysis and safety reports.

- **Offline Mode:** Explore options for limited offline functionality or local model deployments for improved privacy and reliability.