

Project Documentation

1. Introduction

Project Title: City Insights & Citizen Services AI

Team ID : NM2025TMID03748

Team Member: Daniel Raj A

Team Member: Dhanush V

Team Member: Sanjay Raj S

Team Member: Manimaran K

2. Project Overview

Purpose:

The purpose of the City Insights & Citizen Services AI is to empower citizens and city officials with accessible, data-driven insights into the urban environment. By leveraging the IBM Granite Large Language Model, the application provides a dual-function interface to enhance both safety awareness and civic engagement.

For citizens, it offers a simple way to get detailed safety and crime analyses for any city, complete with data visualizations. It also provides a conversational assistant, specifically tailored to answer questions about Indian public services and government procedures. For officials, it serves as a rapid information retrieval tool, capable of summarizing urban data trends and responding to civic inquiries.

Ultimately, this assistant bridges the gap between complex urban data and the public, fostering safer, more informed, and more engaged communities.

Features:

- AI-Powered Urban Analysis: On-demand safety reports.
- Dynamic Data Visualization: Instant visual insights with bar charts.
- Conversational Interface for Citizen Services: Natural language interaction.
- India-Specific Context: Localized and relevant information.
- Engaging Follow-Up Suggestions: Guided user experience.
- Gradio User Interface: User-friendly and accessible.

3. Architecture

The current project utilizes a monolithic architecture where the frontend and backend are tightly integrated within a single application, which is ideal for rapid prototyping and demonstration.

Frontend & Backend (Gradio):

The application is built entirely with Gradio. Gradio serves as both the web UI framework and backend server. User interactions trigger Python functions directly.

LLM Integration (IBM Granite on Hugging Face):

Powered by the `ibm-granite/granite-3.2-2b-instruct` model via Hugging Face transformers.

Data Visualization (Matplotlib):

Generates bar charts dynamically by parsing numerical data from the LLM output.

Model Caching (Google Drive):

Models are cached in Google Drive to avoid repeated downloads.

4. Setup Instructions

The project is designed for easy setup and execution in Google Colab.

Prerequisites:

- A Google Account with Google Drive.
- A web browser.

Steps:

1. Open Colab and load the notebook file.
2. Configure runtime → select T4 GPU.
3. Run Cell 1: Install dependencies.
4. Run Cell 2: Mount Google Drive.
5. Run Cell 3: Launch app. First run downloads models, later runs are faster.
6. Access UI via the `.gradio.live` link provided.

5. Folder Structure

- Cell 1: Dependencies installation.

- Cell 2: Google Drive mounting.
- Cell 3: Main Application Logic (model loading, helper functions, UI definition, event listeners, app launch).

6. Running the Application

1. Follow setup instructions.
2. Run cells in order.
3. Access application via .gradio.live URL.

7. API Documentation

No REST APIs are exposed. Backend is integrated with Gradio. Future improvement could involve FastAPI for public APIs.

8. Authentication

Currently runs in an open environment for demo. Production deployment should add authentication.

9. User Interface

- Dark theme.
- Tabbed layout for City Analysis & Citizen Services.
- Textboxes, plots, chatbot interface.
- Responsive design.

10. Testing

Manual testing conducted:

- City Analysis with various cities.
- Citizen Services with public queries.
- Edge cases for empty/ambiguous inputs.

11. Screenshots

(Insert screenshots of running application here)

- City Analysis Interface [Placeholder].
- Citizen Services Chatbot Interface [Placeholder].

12. Known Issues

- Chart depends on expected response format.
- LLM knowledge cutoff (no real-time info).
- Colab performance may vary.

13. Future Enhancements

- RAG for real-time data.
- Forecasting modules for urban metrics.
- Multimodal input support (PDFs, CSVs).
- Citizen feedback mechanism.
- Deploy to Hugging Face Spaces.