

# **Citizen AI: Intelligent Citizen Engagement Platform**

## **1. Introduction**

Team leader :[ Vivek vasudevan.R ]

Team member :[ Sham.LV ]

Team member :[ Subramani.R ]

Team member :[ Vijay.M ]

Team member :[ Vishal prabhu.P ]

The City Analysis & Citizen Services AI is an intelligent platform designed to analyze city-level safety data and provide accurate, conversational responses to citizen queries regarding public services, government policies, and civic engagement.

Built with Gradio + FastAPI + LLM (IBM Watsonx Granite), it integrates advanced AI modules such as forecasting, anomaly detection, and multimodal input support, ensuring transparency, efficiency, and ease of citizen interaction.

## **2. Project Overview**

This project enables two main functionalities:

City Analysis – Provides reports on crime index, accident statistics, and overall safety.

Citizen Services – Answers citizen queries about government policies, public services, and civic issues.

Additional modules expand the platform into a citizen engagement ecosystem with forecasting, eco-tips, KPI monitoring, and feedback loops.

## **3. Conversation Interface**

Built using Gradio Blocks with two main tabs:

City Analysis Tab: Input city name → Output detailed analysis.

Citizen Services Tab: Input query → Output government-style response.

Simple, interactive UI with markdown, textboxes, and buttons.

## **4. Policy Summarisation**

AI generates concise summaries of lengthy policy documents.

Helps citizens understand the essence of laws/regulations in plain language.

## **5. Resource Forecasting**

Predicts demand for public services (transport, healthcare, electricity).

Supports data-driven governance by preempting shortages.

## **6. Eco-Tip Generator**

Suggests personalized eco-friendly lifestyle tips (waste management, water conservation, carbon reduction).

Encourages sustainability awareness among citizens.

## **7. Citizen Feedback Loop**

Collects and processes citizen inputs.

AI clusters common complaints and sends them as insightful dashboards to authorities.

## **8. KPI Forecasting**

Tracks and forecasts Key Performance Indicators (KPIs) for city governance (pollution, traffic flow, healthcare access).

Generates trend analysis for better decision-making.

## **9. Anomaly Detection**

Detects unusual spikes in crime, accidents, or civic complaints.

Triggers alerts for faster government intervention.

## **10. Multimodal Input Support**

Future-ready expansion:

Text input (queries, policies).

Image input (accident reports, civic photos).

Audio input (voice-based citizen complaints).

## **11. Architecture**

Frontend (Gradio UI)

Simple, interactive conversational interface.

Tabs for different citizen engagement modules.

Backend (FastAPI)

API layer to handle requests between UI, LLM, and ML modules.

Ensures scalability and modular integration.

LLM Integration (IBM Watsonx Granite)

Provides natural language responses.

Used for policy summarisation, query handling, and city reports.

Vector Search (Pinecone)

Stores policy documents, FAQs, civic datasets.

Enables semantic retrieval for accurate responses.

ML Modules (Forecasting & Anomaly Detection)

Uses PyTorch/Sklearn models for time-series forecasting.

Anomaly detection algorithms flag irregular patterns.

## 12. Setup Instructions

Prerequisites

Python 3.9+

GPU (optional, for faster LLM inference)

Accounts for: Hugging Face, Pinecone, IBM Watsonx

Installation Process

# Clone repository

```
git clone https://github.com/your-repo/city-ai
```

```
cd city-ai
```

# Create virtual environment

```
python -m venv venv
```

```
source venv/bin/activate # Linux/Mac
```

```
venv\Scripts\activate # Windows
```

# Install dependencies

```
pip install -r requirements.txt
```

## 13. Folder Structure

city-ai/

```
| — app.py          # Main Gradio app
| — backend/        # FastAPI backend
|   | — api.py
|   | — utils.py
| — models/         # ML models (forecasting, anomaly detection)
| — data/           # Datasets & vector DB storage
| — docs/           # Documentation & API references
| — requirements.txt # Dependencies
| — README.md       # Project overview
```

## 14. Running the Application

Step 1: Start Backend

```
uvicorn backend.api:app --reload
```

Step 2: Run Gradio App

```
python app.py
```

Step 3: Access Application

Open browser → <http://127.0.0.1:7860>

Use city analysis or citizen services tabs.

## 15.screen

short:

The screenshot shows a web application titled "City Analysis & Citizen Services AI". The "City Analysis" tab is selected. It features a form with "Enter City Name" containing the text "trichy" and a button labeled "Analyze City". To the right, the "City Analysis (Crime Index & Accidents)" section displays the following information:

- 1. Crime Index and Safety Statistics:
  - Trichy, a significant city in Tamil Nadu, India, is part of the Greater Chennai Metropolitan Region. According to the National Crime Records Bureau (NCRB) data for 2019, it recorded a Crime Index of 107.4, which is slightly higher than the national average and the state of Tamil Nadu. This index is a composite measure of crimes per 100,000 population.
  - The city's crime rate varies across different types of offenses:
    - Index for Index Crimes (181 crimes): 107.4
    - Index for Non-Index Crimes (120 crimes): 93.1

The interface includes a top navigation bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A "Share" button is visible in the top right. The bottom status bar shows "Variables", "Terminal", "2:02 PM", and "T4 (Python 3)".

## 16.screen short:

The screenshot shows the same web application, but the "Citizen Services" tab is selected. It features a form with "Your Query" containing the text "road safety" and a button labeled "Get Information". To the right, the "Government Response" section displays the following information:

Title: Road Safety: A Comprehensive Guide for Citizens

Introduction:  
Road safety is a critical concern for every citizen. As a government assistant, I'm here to provide you with essential information, guidelines, and resources to promote safe road practices and advocate for improved infrastructure.

1. General Road Safety Tips:

- Always wear a seatbelt: This is the most effective way to protect

A black overlay with the text "To exit full screen, press and hold Esc" is visible in the top center. The interface includes a top navigation bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A "Share" button is visible in the top right. The bottom status bar shows "Variables", "Terminal", "2:02 PM", and "T4 (Python 3)".