CitizenAI - Al-Powered Citizen Service Platform Introduction

Project Title: CitizenAI - Al-Powered Citizen Service Platform Team Members:

Ramesh P, Prasanna R, Shyam S,

Sakthi Murugan P

Project Overview

Purpose:

The goal of CitizenAl is to enhance citizen services and governance using Al technologies such as IBM Watsonx Granite LLM, FastAPI, LangChain, and Streamlit.

The platform automates citizen query handling, service automation, grievance redressal, and documentation, reducing manual effort and improving efficiency.

Key Features:

Citizen Request Upload C Categorization - Citizens upload requests or grievances in PDF, classify them into service categories, and convert into structured service tickets.

Al Service Response Generator - Convert natural language queries into automated responses and service actions.

Issue Resolution Assistant - Identify and provide solutions for citizen issues automatically. Citizen Support Chatbot - Real-time assistance for public queries and services.

Modular C Secure Backend - Built using FastAPI, scalable and API-documented with Swagger Ul.

Citizen Service Phases with Examples

Phase 1: Citizen Request Analysis

Citizens upload PDFs or text with unstructured requests.

Al processes requests and transforms them into structured service tickets. Example Citizen Request (Input):

"The citizen wants to update their residential address in government records." Structured Service Ticket (Al Output):

As a citizen, I want to update my residential address so that my government records remain accurate.

Phase 2: Service Design

Backend: FastAPI for secure citizen data management, LangChain for workflow integration.

Frontend: Streamlit dashboard for citizen service interaction and visualizations. Authentication: JWT-based role authentication (future enhancement).

Example Design Output:

Entity Diagram: Citizen Request → Al Model →

Response Generator

API Endpoints:

POST /analyze\_request Process uploaded requests

POST /generate\_response Convert citizen query to response POST /resolve\_issue Resolve provided issue

Phase 3: Service Automation

Al-powered service generator creates functional responses and workflows from natural prompts.

Example Prompt:

"Generate a confirmation response for an address update request." Al-Generated Output:

Your residential address has been successfully updated in the records."

Phase 4: Validation C Feedback

Generated responses are automatically validated. Errors or incomplete cases are corrected by Al.

Test Case Example:

Input: Citizen requests a birth certificate.

Output: Service confirmation with a downloadable certificate link.

Phase 5: Implementation C Continuous Support

Backend deployed with FastAPI + Uvicorn Frontend hosted with Streamlit

APIs documented via Swagger UI

Future Enhancements:

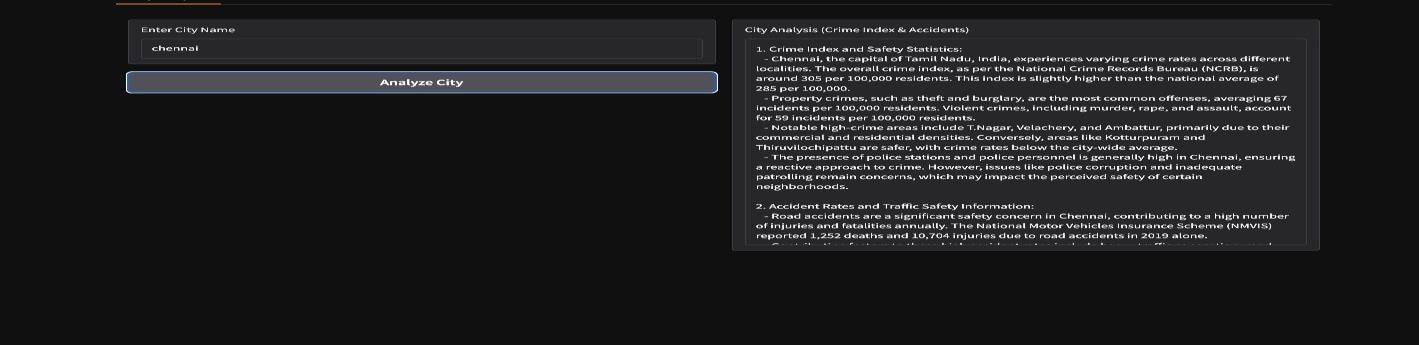
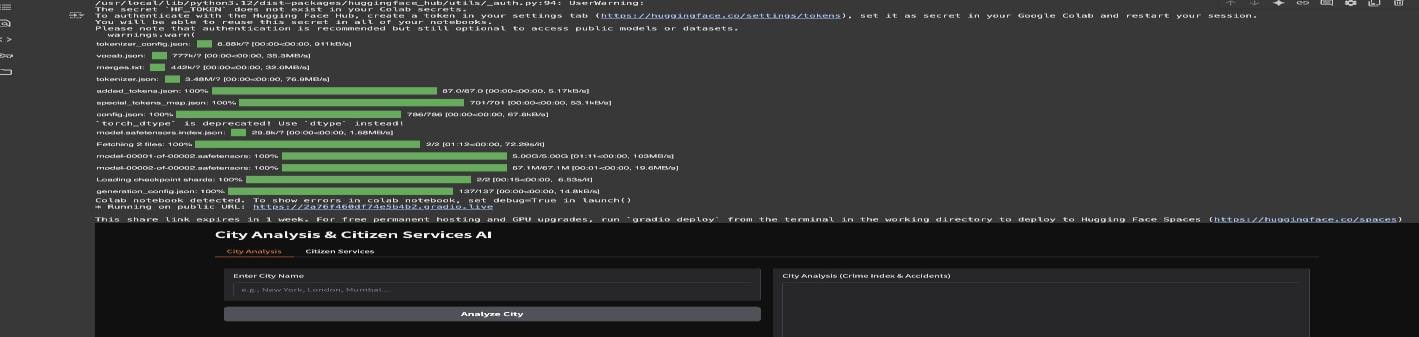
Real-time database integration (citizen records, logs)

Role-based access control (admin, officer, citizen) Multi-language support for inclusivity

Example Workflow (End-to-End)

1. Upload a citizen request document (e.g., "Residential address update").
2. Al classifies the request generates a service ticket.
3. Select Response Generator enter prompt "Generate confirmation response."
4. Al outputs official confirmation.
5. Paste unresolved issue into Issue Resolution Assistant → Al returns corrected guidance.
6. Results shown in Streamlit dashboard → exportable to PDF.

Output



Conclusion

CitizenAl demonstrates how Al can transform citizen services and governance - from request analysis to service delivery - making public services faster, more accurate, and more efficient. This reduces manual effort, improves service quality, and helps governments focus on innovation and inclusivity.