Citizen AI – Intelligent Citizen Engagement Platform

1. Introduction

Project Title: Citizen AI – Intelligent Citizen Engagement Platform

Team Leader: Madhumitha A

Team Members: Rabiyath Mushitha Y

Fathima A

2. Project Overview

Purpose:
Citizen AI is an AI-powered platform designed to improve interactions between citizens and government or civic organizations. It focuses on making public services more accessible, transparent, and personalized.

Core Functionalities:

1. **Citizen Query Resolution** – AI-powered chat bot to handle citizen queries on government services, schemes, and policies.
2. **Service Request Management** – Allows users to raise service requests (e.g., waste management, complaints) and track their status
3. **Feedback & Sentiment Analysis** – Collects and analyzes citizen feedback to improve governance quality.
4. **Community Engagement** – Provides announcements, polls, and updates from local authorities.

Key Features:

Multi-channel Support (Web, Mobile, WhatsApp, Voice Assistants)

Multilingual Conversations

AI-driven Insights & Prioritization of Issues

Personalized Notifications

Transparent Tracking of Requests

3. Architecture

Frontend:

Built with Gradio or React-based UI.

Provides chat interface, dashboards, and request tracking view.

Backend (Python):

Handles natural language understanding (NLU), classification, and response generation.

Uses rules + AI model for accurate information retrieval.

LLM Integration:

Uses ibm-granite/granite-3.2-2b-instruct for conversational AI.

Integrates with government service APIs for real-time data.

Deployment Layer:

Can be deployed on local servers, cloud platforms, or public domain with secure access.

4. Setup Instructions

Prerequisites:

Python 3.9+

pip & virtual environment tools

Hugging Face Transformers & Torch

REST API access to government datasets (optional)

Installation:

1. Clone repository
2. Install dependencies from requirements.txt
3. Run: python app.py
4. Access the platform via browser

5. Folder Structure

app/ – Main application folder
├── citizen_query.py – AI-driven query handling
├── service_request.py – Service request management
├── feedback_analysis.py – Sentiment & feedback module
├── model_loader.py – Loads IBM Granite LLM
├── interface.py – Defines UI (Gradio/React)main.py – Entry script
requirements.txt – Dependencies
README.md – Documentation

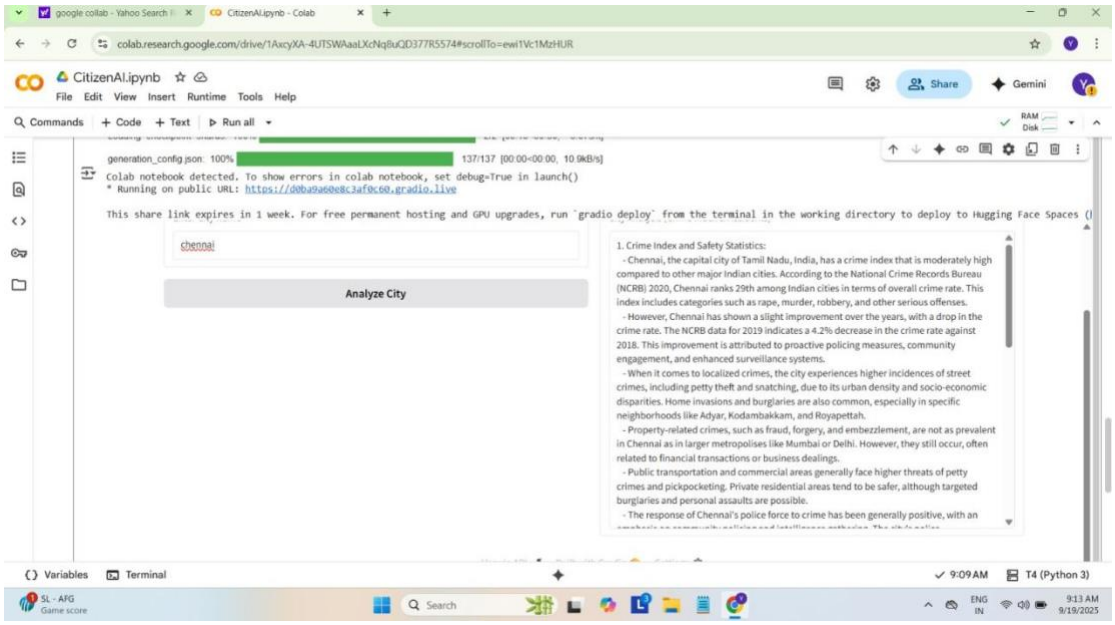
6. Running the Application

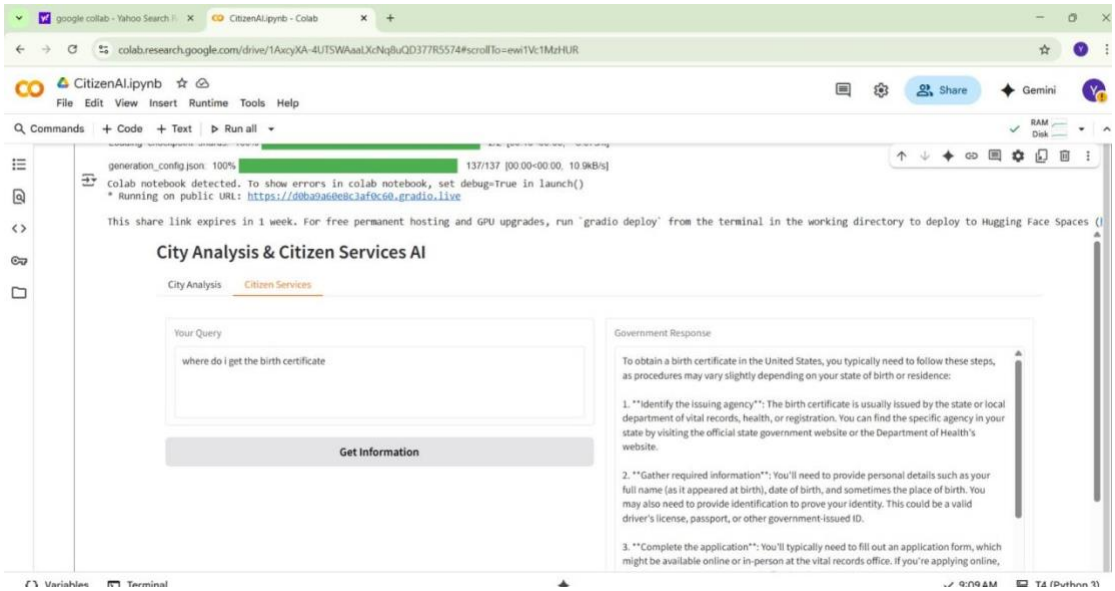
Run:

python main.py

Platform launches at: http://127.0.0.1:7860
Public link available with share=True

7. Screenshots





8.API Documentation (Future Integration)

POST /citizen-query → Returns response to user query

POST /service-request → Creates and tracks service requests

GET /feedback → Retrieves citizen feedback & analysis results

Authentication

Role-based access (Citizen, Admin, Officer)

Secure authentication with JWT or OAuth2

9. User Interface

Modules:

1. **Chatbot Interface** – Ask queries about services
2. **Service Request Panel** – Submit, view, and track issues
3. **Feedback Dashboard** – Provide feedback, see community ratings
4. **Announcements Board** – Get latest civic updates

10.Testing

Unit tests for query classification & response accuracy

Manual testing with multiple citizen scenarios

Load testing for concurrent users

11. Known Issues

AI may sometimes return generic responses

Latency in fetching real-time government data

Requires multilingual fine-tuning for some dialects

Future Enhancements

Integration with voice assistants (Alexa, Google Assistant)

Predictive analytics for issue prioritization

Mobile app deployment

Automated grievance redressal system

Community participation gamification (badges, points)

12.Conclusion

Citizen AI is designed to bridge the gap between citizens and civic authorities by providing a seamless, AI-powered communication channel. Through intelligent query resolution, transparent service request tracking, and actionable feedback insights, the platform empowers citizens and promotes better governance.