CitizenAI – Intelligent Citizen Engagement Platform

Generative AI with IBM – Project Report

Submitted by: **Team Leader**- Abinaya T

Team Members- Abinaya T, Abinaya R, Aishwarya S,Akalya I

TABLE OF CONTENTS

1.ABSTRACT:

This project presents CitizenAI – an Intelligent Citizen Engagement Platform designed to bridge the gap between citizens and government services. The system leverages IBM Granite AI, Hugging Face Transformers, and Gradio interface to provide quick, accessible answers about civic issues, government schemes, and public rights.

The primary goal of this project is to empower individuals to better engage with governance by receiving AI-driven insights, civic guidance, and feedback dashboards. It also tracks public sentiment and presents structured information to both citizens and officials. CitizenAI emphasizes responsible AI usage with disclaimers and directs users to verify with official government sources.

2.OBJECTIVES:

- To create an AI-powered citizen assistant with a simple UI.
- To answer questions about government schemes, services, and civic issues.
- To provide personalized civic suggestions based on demographics and needs.
- To generate sentiment dashboards for public officials.
- To ensure safe and reliable AI-driven recommendations.

3.SYSTEM REQUIREMENTS:

Software Requirements:

- Python 3.x
- Gradio
- Transformers
- Torch
- IBM Granite AI model

Hardware Requirements:

- Minimum 4GB RAM
- Dual-core processor
- Stable Internet connection

Deployment Requirements:

- Google Colab (T4 GPU) for low-cost, reliable execution.
- GitHub for version control and project storage.

4. PROJECT WORKFLOW

CitizenAI follows a modular workflow for extensibility and ease of deployment.

Workflow Steps:

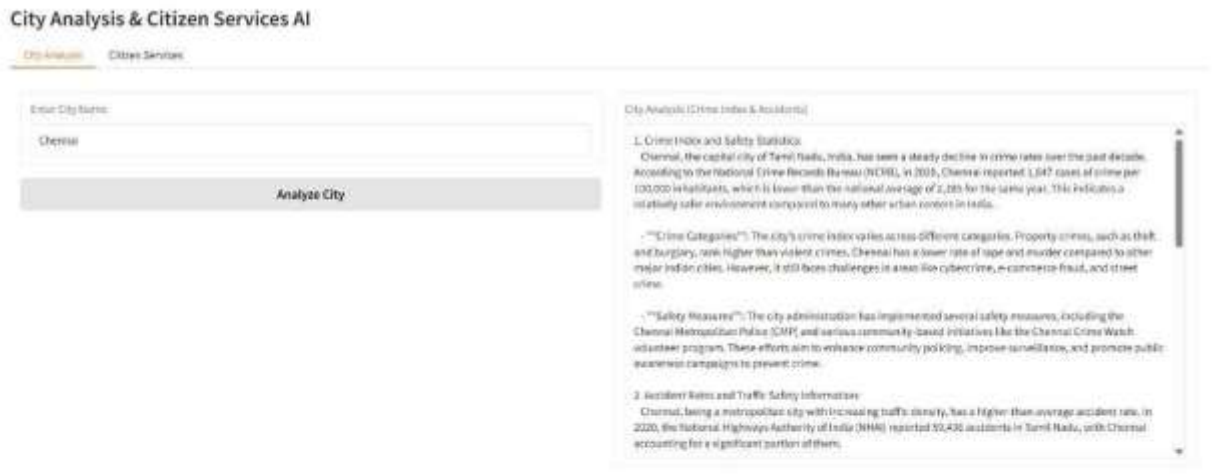
1. Exploring Smart Interz Portal: Access project workspace via Naan Mudhalvan Smart Internz portal and review guided project resources.
2. Model Selection: Choose IBM Granite models from Hugging Face (e.g., granite-3.2-2b-instruct) for lightweight, fast performance.
3. Running Application in Google Colab: Configure runtime with T4 GPU, install dependencies (transformers, torch, gradio), and run the application.
4. Deployment: Launch Gradio-based interface, access the live app through shared links, and visualize outputs.
5. Version Control: Upload code and resources to GitHub for project management and future updates.

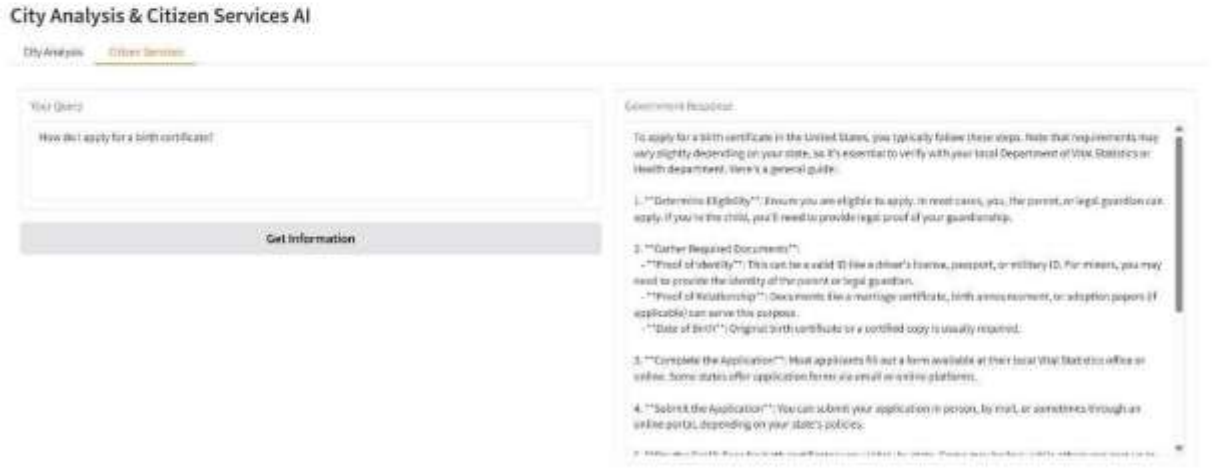
5.PROJECT FILES:

- .env – Stores API keys and configurations
- App.py – Main application file with Gradio blocks
- requirements.txt – List of required dependencies
- CitizenAI.ipynb – Google Colab notebook for model execution

6. APPENDIX: SOURCE CODE:

7.OUTPUT SCREENSHOT:





The City Analysis tab provides details such as crime index, safety measures, and accident statistics for a selected city (example: Chennai). The Citizen Services tab answers user queries with step-by-step government procedures, such as applying for a birth certificate.

8.CONCLUSION:

The CitizenAI application demonstrates how **Generative AI** can be applied to improve civic engagement and public awareness. By

combining **city analysis** and **citizen services assistance**, the system helps individuals access reliable information about safety, traffic, and government procedures in a simple and interactive way.

Key Outcomes:

- Provides easy-to-understand insights on city statistics such as crime and accident rates.
- Delivers step-by-step guidance for citizen services (e.g., applying for certificates, accessing schemes).
- Empowers citizens to stay informed and engaged with governance.
- Encourages safe, responsible use of AI by including disclaimers and pointing to official sources.

Future Scope:

- Integration with official e-governance portals for real-time updates.
- Multilingual support for wider accessibility.
- Enhanced dashboards for public officials to monitor feedback.
- Cloud deployment for scalability and secure access.

Overall, CitizenAI acts as a bridge between citizens and government services, promoting transparency, awareness, and smarter civic participation.
