Citizen AI – City Analysis and Citizen Services Assistant

Project Documentation

Prepared by:

Team Leader : KALAIMARAN M

Team member : SIVA A

Team member : IMRAN ARSATH E

Team member : KALAIYARASAN M

Team member : MURUGAPANDI B

1.1 Introduction

In today's world, smart cities need technology to help their residents. People want quick access to information about safety, accidents, and government services. Officials also need tools to process data easily.

The **Citizen AI** project is designed to solve this problem by providing real-time information through an AI assistant. It helps citizens with safety details and answers their questions about city services.

1.2 Project Objective

The main goals of this project are:

- Provide safety information like crime and accident data.
- Help citizens by answering questions about government services.
- Summarize government policies for easy understanding.
- Give users a simple interface to interact with AI.

1.3 Project Overview

Purpose:
The Citizen AI project aims to assist both citizens and city officials by giving instant responses, safety alerts, and helpful information through a user-friendly platform.

Key Features:

- Natural conversation interface using AI.
- City analysis with crime and accident statistics.
- Policy summarization for government rules.
- Query response for public service questions.
- Interactive dashboard with Gradio.

1.4 Technology Architecture

Frontend: Built using Gradio to create an easy-to-use interface.

Backend:

- Uses Python libraries like Transformers and PyTorch.
- Integrated with IBM's Granite model for text generation.

How it Works:

- Users ask questions or request information.
- The AI processes the query and gives responses.
- City analysis is shown with data and statistics.

Main Functions:

- `generate_response(prompt)` – Answers user questions.
- `city_analysis(city_name)` – Gives safety-related data.
- `citizen_interaction(query)` – Responds to queries on services.

1.5 Setup Instructions

Requirements:

- Python 3.8 or above
- Google Colab with T4 GPU (preferred)
- Libraries: `transformers`, `torch`, `gradio`

Steps to Setup:

1. Open Google Colab.
2. Set the runtime to GPU (T4 preferred).
3. Install necessary libraries using `!pip install transformers torch gradio -q`.

4. Paste the project code into the notebook.
5. Run all cells to launch the app.
6. Access the link provided to use the interface.

1.6 Folder Structure

```
app/                 – Backend logic
app/api/             – API routes
ui/                  – Frontend components
smart_dashboard.py   – Main script
granite_llm.py      – Model integration
document_embedder.py – Embedding and document search
```

1.7 Running the Application

1. Launch Google Colab.
2. Run the installation commands.
3. Start the Gradio dashboard.
4. Use the **City Analysis** tab to explore safety data.
5. Use the **Citizen Services** tab to ask questions.

1.8 API Documentation

Available Functions:

- **generate_response(prompt):** Generates replies based on user input.
- **city_analysis(city_name):** Provides crime and safety data.
- **citizen_interaction(query):** Answers service-related questions.

1.9 Authentication & Security

Future versions can include:

- Token-based authentication (JWT, API keys).
- OAuth2 integration.
- Role-based permissions for admins and users.

1.10 User Interface Description

- **City Analysis Tab:** Users enter a city name and view safety data.
- **Citizen Services Tab:** Users ask questions about services and receive answers.
- **Output:** Results are shown in clear text fields.

1.11 Testing Process

- **Unit Testing:** Test each function separately.
- **API Testing:** Check responses for sample inputs.
- **Manual Testing:** Validate features by using the interface.
- **Edge Cases:** Handle wrong city names or empty queries.

1.12 Known Issues

- Requires stable internet connection.
- AI results may not always be accurate.
- Works better with real-time data integration.

1.13 Future Enhancements

- Add real-time city data feeds.
- Improve AI with forecasting and analytics.
- Support multiple languages.
- Create a mobile application.

1.14 Conclusion

The **Citizen AI** project shows how artificial intelligence can make city services more accessible and safer. It helps both citizens and officials by providing quick and accurate information. This project is a step toward smarter cities in the future.