**CITIZEN AI- INTELLIGENT CITIZEN ENGAGEMENT PLATFORM**

**PROJECT DOCUMENTATION**

**1.Introduction:**

* **Project tittle: CITIZEN AI- INTELLIGENT CITIZEN ENGAGEMENT PLATFORM**
* **Team member: S.DHIVYA**
* **Team member: S.PUVISHA**
* **Team member: J.KANISHKA**
* **Team member: R.MAHALAKSHMI**

**2.Project Overview:**

* **PURPOSE:**

**Citizen AI** is an AI-powered platform that helps governments, institutions, or public organizations engage with citizens more effectively. It uses chat bots, data analytics, and feedback systems to improve communication and solve issues faster.

* **KEY FEATURES:**

**1.AI Chat Assistant:**

* Answers citizen questions instantly (24/7)
* Can guide users on services, programs, or processes

**2.** **Feedback & Sentiment Analysis**

* Collects opinions from citizens
* Analyzes emotions (positive/negative) to identify common problems

**3.** **Issue Reporting (Ticketing System)**

* Lets users raise complaints or requests
* Tracks and updates status automatically

**4.** **Real-Time Dashboards**

* Shows data like number of complaints, satisfaction trends, or common queries
* Helps decision-makers respond better and faster

**5.** **Secure and Scalable**

* Works on web or mobile
* Can handle thousands of user

### 3.Architecture:

### 1. ****Frontend Layer (User Interface)****

* Accessible via Web, Mobile, or Chatbot
* Used by citizens to:
  + Ask questions
  + Submit feedback or complaints
  + Track request status

### 2. ****Backend Layer****

* Handles user requests and business logic
* Routes inputs to appropriate modules (AI, Feedback, Ticketing)

### 3. ****AI Engine (NLP / LLMs)****

* Powered by models like GPT, IBM Watson, etc.
* Capable of:
  + Understanding citizen queries
  + Providing instant responses
  + Translating queries into actions (e.g., submit a ticket)

### 4. ****Data Storage****

* Stores:
  + Chat history
  + Feedback logs
  + User data and ticket details
* Uses secure SQL or No SQL database

**4.Setup Instructions:**

**Prerequisites**

Before starting, make sure you have these installed:

* **Python (3.8 or above)**
* **pip** (Python package installer)
* **Git**
* **IBM Cloud account** (for Watson or Granite API keys, if used)
* **Node.js** (if using advance frontend frameworks like React)

**5.Folder Structure:**

* **app.py**: Main application that wires all routes and modules.
* **templates/**: Contains the HTML pages rendered via Flask.
* **routes/**: Modular route definitions for different features.
* **services/**: Contains logic for AI chat bot, feedback analysis, etc.
* **models/**: Python classes that represent the database tables.
* **frontend/**: If you're using a separate React frontend, it goes here.

**6.Running The Application:**

**FRONTEND**

The frontend of Citizen AI is what people see and use, like a website or app. To run it, you go to the frontend folder, install the needed tools, and start it. Then you open a browser at http://localhost:3000 to use the chat bot, send feedback, or report problems. If there’s no separate frontend, the website is shown by the backend itself.

**BACKEND**

The backend of Citizen AI is the part that runs the smart features like the AI chatbot and handles all the data. To run it, you set up a special Python environment, install the needed software, add your secret keys, and then start the server. It usually runs on and works behind the scenes to answer questions, collect feedback, and manage tickets.

**7.API Documentation:**

### **POST** /chat : Send a user message to the AI chatbot and get a response.

### **POST** /feedback : Submit user feedback.

### **POST** /ticket : Submit a complaint or issue ticket.

### **GET**/tickets: Get a lis t of all tickets (admin only).

### 8.Authentication:

### The ****authentication system in Citizen AI**** is designed to keep the platform secure and organized. Citizens can usually access features like chatting with the AI or giving feedback without logging in, making it easy to use.

* Citizens use the platform freely without logging in.
* Admins must log in to access and manage backend data.
* Secure sessions or tokens are used to protect admin actions.

**9.User interface:**

When a user opens the platform, they see a **homepage** with a chatbot where they can type questions or ask for help. There are also easy-to-find sections like:

* Feedback **Form** – where users can rate services and share their experience
* **Report an Issue** – to raise complaints or problems (like water issues, electricity, etc.)
* **Track My Ticket** – to check the status of submitted issues
* **Admin Login** – only for staff or officials to access the dashboard

**10.Testing:**

**Testing** in the **Citizen AI – Intelligent Citizen Engagement Platform** makes sure everything works correctly before it is used by real citizens or admins. It checks if users can chat, submit feedback, raise tickets, and that the AI responds properly.

* Postman – To test APIs
* Browser testing – To check pages manually
* Unit tests – For backend logic (Python)



