1.Introduction

Project Title: Citizen AI – Intelligent Civic Engagement Assistant

* Team Leader: Mabrooka Nibras K M

* Team Member : Mufeequa Mehnaz P S

* Team Member : Zuwairiya Banu A

* Team Member : Arshiya Thahasin M S

2. Project overview:

Purpose:

Citizen AI is designed to enhance civic engagement by providing an intelligent platform that empowers citizens to interact with government services, access public information, and participate in decision-making processes. Leveraging AI technologies, it enables efficient communication, transparent governance, and active citizen participation in a smart, connected society.

Features:

Conversational Government Interface

Key Point: Natural language interaction

Functionality: Citizens can ask questions, report issues, and receive guidance about public services using simple, conversational language.

Government Document Summarization

Key Point: Easy access to policies

Functionality: Converts complex government documents, laws, and regulations into understandable summaries.

Service Request Automation

Key Point: Streamlined service processes

Functionality: Automates submission of requests such as utility complaints, permit applications, and public grievance reporting.

Public Opinion Collection

Key Point: Citizen feedback loop

Functionality: Collects and analyzes citizen opinions, suggestions, and surveys to inform policy decisions

Anomaly Detection in Civic Data

Key Point: Early issue identification

Functionality: Detects anomalies in city service usage patterns, such as unusual utility consumption or traffic congestion, to alert officials.

Multimodal Input Support

Key Point: Versatile interaction modes

Functionality: Accepts text, images (photos of issues), PDFs, and structured data for efficient reporting and analysis.

KPI Forecasting for Public Services

Key Point: Data-driven governance

Functionality: Forecasts key performance indicators for public services, enabling data-driven resource allocation and planning.

Secure Access & Authentication

Key Point: Data protection

Functionality: Ensures secure interactions between citizens and the government through token-based authentication and role-based access.

3. Architecture

Frontend (React or Streamlit):

User-friendly web interface allowing citizens to interact with the AI, submit service requests, access reports, and receive notifications

Backend (FastAPI or Node.js):

Manages APIs for document summarization, service request handling, anomaly detection, and feedback processing.

LLM Integration (OpenAI GPT or IBM Watsonx Granite):

Provides natural language understanding and generation for answering questions, summarizing policies, and generating responses.

Vector Search (Pinecone):

Indexes government documents, service records, and citizen reports for semantic search capabilities.

Database (PostgreSQL or MongoDB):

Stores citizen profiles, service requests, feedback history, and generated content.

4. Setup Instructions

Python 3.9+ or Node.js environment

Install dependencies from requirements.txt or package.json

API keys for OpenAI GPT or IBM Watsonx Granite

Configure .env with credentials

Start backend and frontend services

Register citizen accounts and integrate with government portals

5. Running the Application

Start backend API server

Launch frontend web interface

Citizens submit service requests or ask questions

Government officials monitor and respond via dashboard

Automated summaries and reports are generated

KPI forecasting visualizes service efficiency

6. API Documentation

POST /ask-citizen-ai: Citizens submit inquiries and get AI-generated responses

POST /submit-service-request: Automate public service applications and complaints

POST /upload-doc: Upload supporting documents or images

GET /search-docs: Search policies or service guides

GET /generate-kpi-report: Generate KPI forecasting reports

POST /submit-feedback: Submit citizen feedback for review

7. Authentication

JWT Authentication

OAuth2 integration for government portals

Role-based access: Citizen, Official, Admin

8. User Interface

Clear and accessible design:

Sidebar for navigation (Ask AI, Service Requests, Feedback, Reports)

Conversational chatbot interface

Service request forms with file upload

KPI dashboards for officials

Summary view of citizen interactions

9. Testing

Unit testing for service request handling and document summarization

API testing using Swagger and Postman

Manual testing for file uploads, chat accuracy, and KPI generation

Edge case handling: Malformed files, invalid input, API key error