# 

**Citizen AI – Intelligence & Citizen Engagement Platform**

*Project Documentation*

## 

**College: Government Arts College Trichy -22**

**Team Leader: Parameswari G**

**Email** : [kk7289273@gamil.com](mailto:kk7289273@gamil.com)

**Team Member 1: Charumathi M**

**Email:** [charumathim1771@gamil.com](mailto:charumathim1771@gamil.com)

**Team Member 2: Jereen Anjalina S**

**Email:** [jereenanjalina@gmail.com](mailto:jereenanjalina@gmail.com)

**Team Member 3: Navyapriya M**

**Email**:[navyamurugan006@gamil.com](mailto:navyamurugan006@gamil.com)

**Team Member 4: Kokila T**

**Email:** [tkokila16012006@gmail.com](mailto:tkokila16012006@gmail.com)

# **1. Introduction**

Citizens AI is an innovative platform designed to transform the way governments and citizens interact.

In today’s fast-moving world, citizens often face challenges in accessing services, understanding policies, and participating in governance.

At the same time, government officials require efficient tools to manage public needs, analyze feedback, and make informed decisions.

The Citizens AI platform bridges this gap by combining artificial intelligence, data analytics, and multilingual support to create a seamless channel of communication.

It empowers citizens with easy access to government schemes, policies, and services through natural language interaction, while also enabling governments to gather insights, track grievances, and improve policy-making.

By serving as both a digital assistant for citizens and an intelligence dashboard for officials, Citizens AI promotes transparency, inclusivity, and smarter governance.

This results in a more engaged community, faster service delivery, and better decision-making for sustainable development.

## **2. Project Overview**

**Purpose:** Citizens AI is an AI-powered engagement platform that bridges the gap between governments and citizens by simplifying access to services, policies, and civic participation. It leverages artificial intelligence, real-time analytics, and multilingual support ensure that governance is transparent, inclusive, and responsive.

**Features:**

* **Conversational AI Assistant**
  + Natural language interaction for citizens to ask questions, access services, and get updates.
* **Policy Intelligence**
  + Summarizes complex government policies into citizen-friendly explanations.
* **Digital Service Hub**
  + One-stop access to government schemes, licenses, applications, and certificates.
* **Feedback & Grievance Management**
  + Collects complaints and feedback, tracks resolution progress, and ensures accountability.
* **Community Engagement & Alerts**
  + Provides updates on local events, emergencies, and public decisions.
* **Multilingual & Inclusive Access**
  + Supports multiple regional languages and accessibility features (voice/text).
* **Analytics Dashboard for Officials**
  + Real-time insights on citizen sentiment, service usage, and policy impact.
* **AI-Powered Forecasting**
  + Predicts trends in citizen demands, service loads, and community issues.

## **3. Architecture**

* **Frontend (Web & Mobile):** Interactive citizen dashboard with chatbot, service directory, feedback forms, and policy summaries.
* **Backend (FastAPI / Node.js):** Handles service requests, feedback processing, policy intelligence, and real-time notifications.
* **AI & LLM Integration:** Large Language Models (LLMs) for conversation, translation, summarization, and predictions.
* **Database (SQL/NoSQL):** Stores service records, policy documents, citizen feedback, and analytics data.
* **Analytics & Visualization:** Dashboards for government officials using data insights and forecasting tools.
* **Security Layer:** Role-based authentication (citizen, official, admin), encrypted communication, OAuth2/JWT support.

## **4. Setup Instructions**

**Prerequisites:**

* Python 3.9+ / Node.js (for backend)
* Database (PostgreSQL / MongoDB)
* API Keys for LLM services
* Internet connectivity

**Installation Process:**

1. Clone repository
2. Install dependencies (requirements.txt or package.json)
3. Configure .env with API keys and database credentials
4. Run backend server
5. Launch frontend (web or mobile)
6. Interact with the citizen dashboard

## **5. Folder Structure**

* **backend/** – APIs for chat, services, feedback, policy intelligence
* **frontend/** – UI components for citizens and officials
* **models/** – AI/ML models for summarization & forecasting
* **database/** – Schemas and data storage logic
* **utils/** – Utility scripts for notifications, reports, analytics

## **6. Running the Application**

1. Start backend server (FastAPI/Node.js)
2. Launch frontend (Web/Mobile App)
3. Citizens interact via chatbot, feedback forms, and service hub
4. Officials monitor analytics dashboard for insights

## **7. API Documentation**

* **POST /chat/query** – Ask questions & receive AI responses
* **GET /policy-summary** – Get simplified version of policies
* **POST /submit-feedback** – Citizens submit feedback or complaints
* **GET /service-info** – Retrieve information on schemes/services
* **GET /analytics-dashboard** – Officials view real-time insights

## **8. Authentication**

* OAuth2.0 for secure citizen login
* Role-based access: **Citizen / Government Official / Admin**
* Token-based authentication (JWT)
* Planned addition: Biometric/ID verification

## **9. User Interface**

* **Citizen Side:**
  + Chatbot for queries
  + Service catalog with step-by-step guidance
  + Feedback/grievance tracking
  + Policy summaries in simple language
  + Multilingual chat/voice support
* **Official Side:**
  + Analytics dashboard with charts & KPIs
  + Feedback monitoring system
  + AI-driven forecasting insights
  + Reports exportable in PDF/Excel

**10. Testing**

* **Unit Testing:** Service modules, chatbot responses, API endpoints
* **API Testing:** Swagger UI, Postman
* **Manual Testing:** Multilingual queries, policy summaries, feedback submission
* **Stress Testing:** High-volume citizen queries and complaints

## 

## **11. Known Issues**

* Requires constant internet connectivity
* Limited offline features
* Initial dependency on external LLM APIs

## **12. Future Enhancements**

* AI-powered **voice assistant** for citizens
* Mobile-first citizen engagement app
* Predictive models for **urban planning & crisis management**
* Integration with **IoT devices** for smart city data
* Blockchain for transparent citizen feedback tracking

