**Project Title:**

Citizen AI - Intelligent Citizen Engagement Platform

**1.INTRODUCTION**

• **Project title**: Citizen AI Intelligent Citizen Engagement Platform

• Team member 1: Abinaya.M

• Team member 2: Aruna Elize.Y

• Team member 3: Arya.N

• Team member 4: Hasvanthini.A

**3. PROJECT OVERVIEW**

• Purpose: Citizen AI is designed to transform the way governments, municipalities, and communities interact with their citizens. By leveraging Artificial Intelligence, the platform provides real-time communication, automated policy summarization, sentiment analysis, and personalized engagement. It empowers citizens with easy access to services and information while enabling governments to receive structured feedback, identify trends, and enhance service delivery. The platform ultimately fosters transparency, inclusivity, and trust between authorities and the community.

**• Features:**

Conversational Interface

Key Point: Seamless citizen communication

Functionality: Natural language chatbot for inquiries, complaints, and service requests.

**Policy Summarization**

Key Point: Clear policy understanding

Functionality: Converts lengthy government documents into concise summaries accessible to all citizens.

**Sentiment Analysis**

Key Point: Gauge public mood

Functionality: Analyzes citizen feedback and social media data to measure community sentiment on various issues.

**Citizen Feedback Loop Key Point:**

Structured engagementFunctionality: Collects, organizes, and prioritizes citizen concerns for government review.

**Service Automation:**

Key Point: Efficient service delivery

Functionality: AI-driven ticketing system for issue tracking, complaint resolution, and updates. **Predictive Analytics:**

Key Point: Anticipating needs

Functionality: Uses historical data to forecast community needs and optimize public resource allocation.

**Accessibility Support:**

Key Point: Inclusivity

Functionality: Multilingual support, voice-enabled interaction, and accessibility features for differently-abled citizens.

**User Dashboard:**

Key Point: Transparency

Functionality: Provides citizens with updates on requests, local initiatives, and real-time data visualizations.

**3. Architecture**

Frontend (React/Streamlit):

Interactive citizen dashboard with sections for service requests, policy updates, and feedback submission. Accessibility-focused design for diverse user groups.

Backend (FastAPI):

Handles API endpoints for chat, policy summarization, sentiment analysis, and service automation. Ensures scalability and integrates securely with government databases.

AI Integration (LLM Models):

Large Language Models used for summarization, conversational assistance, and predictive insights.

Database & Vector Search (PostgreSQL + Pinecone):

Stores citizen records, requests, and embedded government documents for semantic search.

ML Modules (Forecasting & Sentiment Analysis):

Implements predictive models for demand forecasting and NLP-driven sentiment classification. **4. Setup Instructions**

Prerequisites: • Python 3.9+ • Node.js for frontend (if React) • API keys for LLM and vector database • Secure access credentials for government services

Installation: • Clone repository • Install dependencies from requirements.txt and package.json • Configure .env file with credentials • Launch backend (FastAPI) • Start frontend (React/Streamlit) • Access dashboard via browser

**5. Folder Structure**

app/ – FastAPI backend logic app/api/ – API routes for citizen services, chat, feedback, and document processing ui/ – Frontend UI components models/ – ML models for summarization and sentiment analysis document\_embedder.py – Converts government docs into searchable embeddings citizen\_feedback.py – Collects and processes structured citizen feedback policy\_summarizer.py – Generates concise summaries of policy documents report\_generator.py – Creates citizen engagement and service performance reports

**6. Running the Application**

• Start backend server with FastAPI • Run frontend UI (React/Streamlit) • Citizens log in to dashboard • Submit queries, requests, or feedback • AI modules process and respond in real-time • Authorities monitor citizen feedback and service requests through the admin panel

**7. API Documentation**

POST /chat/query – Citizen queries answered by AI

POST /upload-policy – Uploads and summarizes documents

GET /search-policies – Returns similar documents to query

POST /feedback – Submits citizen feedback

GET /sentiment – Returns community sentiment insights

POST /service-request – Creates service request ticket

**8. Authentication**

• Role-based authentication (citizen, official, admin)

• OAuth2 with government credentials

• JWT-based session handling

• Optional biometric login for citizens

**9. User Interface**

• Citizen DashboardChat assistant, service request tracking, policy updates • Government Dashboard: Analytics on citizen engagement, sentiment reports, issue tracking • Accessibility Features: Multilingual support, text-to-speech, high-contrast mode

**10. Testing:**

 • Unit Testing: For NLP summarization, feedback classification • API Testing: Swagger, Postman, automated test scripts • Integration Testing: End-to-end service request flows • Stress Testing: Large-scale citizen queries and document uploads

**11.Known Issues**

• Limited multilingual support in beta version

• Sentiment analysis accuracy varies with informal language

**12. Future Enhancements**

• Expanded regional language support

• Integration with IoT data for smart city management • Blockchain-enabled citizen identity verification

• AI-driven participatory budgeting recommendations