**Citizen AI: Intelligent Citizen Engagement Platform**

**Project Documentation**

1. **Introduction:**

* Project Title: Citizen AI: Intelligent Citizen Engagement Platform.
* Team Member: SATHYA B
* Team Member: NISHANTHI S
* Team Member: ABINAYA P
* Team Member: AREEFA SHAHANA S

1. **Project Overview:**

* **Purpose:**  
  To create a centralized AI-powered platform that simplifies citizen–government interaction
* **Features:**
  + AI-powered chatbot for query handling
  + Personalized citizen dashboard
  + Request/complaint submission and tracking
  + Real-time notifications
  + Data analytics for government agencies

1. **Architecture**

* **Frontend:** React / Angular (user interface for citizens and admins)
* **Backend:** FastAPI / Node.js (handles business logic, API services)
* **Database:** PostgreSQL / MongoDB (stores citizen data, requests, and logs)
* **AI Integration:** NLP models (for chatbot and sentiment analysis)
* **Authentication:** OAuth 2.0 / JWT-based authentication
* **Deployment:** Docker + Kubernetes on cloud (AWS / Azure / GCP)

1. **Setup Instruction:**

 Clone the repository from GitHub

 Install required dependencies

 Configure .env file with database credentials and API Keys

 Start backend and frontend servers

1. **Folder Structure:**

 The **folder structure** in a project defines how different files and modules are organized.

 A well-structured project improves **readability, maintainability, scalability, and collaboration**.

 In Edutor AI, the folder structure is divided into logical parts such as frontend, backend, AI models, database, and configuration files.

1. **Running Applications:**

* Start backend server
* Start frontend
* Access the app at: http://localhost:3000

1. **Folder Structure:**

* API documentation is available via **Swagger UI** or **Postman Collection**.
* Example endpoints:
* POST /api/auth/login → User login
* POST /api/request → Submit new request
* GET /api/request/{id} → Track request status
* GET /api/notifications → Fetch user notifications

1. **Authentication:**

* **User Authentication:** JWT-based login for citizens and admin roles.
* **Role Management:** Citizens, Admins, and Super Admins.
* **Data Security:** Encrypted credentials, secure token storage.

1. **Testing:**

* **Unit Testing:** Validate API endpoints and business logic
* **Integration Testing:** Ensure smooth communication between frontend and backend
* **UI Testing:** Check responsiveness and accessibility
* **Load Testing:** Measure platform performance under heavy user load

1. **Known Issues:**

* AI chatbot accuracy may vary with regional language inputs
* Limited offline support in low-network areas
* Delay in notifications under high traffic load

1. Future Enhancement:

* Multilingual support for better inclusivity

 Mobile application for Android/iOS

 AI-driven predictive analytics for government agencies

 Integration with digital payment gateways for service fees

 Voice assistant support for accessibility