**Citizen AI- intelligent Citezen Engagement Platform**

|  |  |
| --- | --- |
| Date | 30 june 2025 |
| Team ID | LTVIP2025TMID60842 |
| Project Name | Citizen AI- intelligent Citezen Engagement Platform |
| Maximum Marks | 4 Marks |

## ****Citizen AI – Intelligent Citizen Engagement Platform****

**Technical Architecture**

The **Citizen AI** platform enables smart and secure citizen engagement with local and national government bodies, powered by conversational AI, feedback systems, and analytics dashboards.

A high-level **architecture diagram** for the platform would include:

* **Frontend Interfaces (Web, Mobile, Chatbot)**
* **Application Layer** (Microservices for registration, feedback, notifications, etc.)
* **AI Services** (Speech-to-text, NLP, Assistant)
* **Backend Systems** (Database, Storage, External APIs)
* **Infrastructure & DevOps Layer** (Cloud-native infrastructure with CI/CD)

🔗 **Reference**: Based on IBM's AI-powered backend system patterns during pandemics – [IBM Developer](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)

### ****Table-1: Components & Technologies****

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1 | User Interface | Web Portal, Mobile App, and AI-powered Chatbot | HTML, CSS, JavaScript, ReactJS, Flutter |
| 2 | Application Logic-1 | Core registration, feedback & issue tracking logic | Node.js / Java / Python |
| 3 | Application Logic-2 | Voice to text conversion for citizen audio messages | IBM Watson Speech-to-Text, Google STT |
| 4 | Application Logic-3 | Conversational AI for citizen engagement | IBM Watson Assistant, Rasa NLU |
| 5 | Database | Stores structured citizen data, issue logs | PostgreSQL, MongoDB |
| 6 | Cloud Database | Cloud-native database for scalability | IBM Cloudant, Amazon RDS |
| 7 | File Storage | Stores multimedia uploads (images, documents, video evidence) | IBM Cloud Object Storage, AWS S3 |
| 8 | External API-1 | Integration for real-time weather/emergency updates | IBM Weather API, OpenWeatherMap |
| 9 | External API-2 | Identity verification via national digital identity | Aadhaar API, DigiLocker API |
| 10 | Machine Learning Model | NLP-based model to classify feedback and recommend services | BERT, Custom NLP classifier, Scikit-learn |
| 11 | Infrastructure | Application hosting and container orchestration | Kubernetes, IBM Cloud Foundry, AWS EKS |

### ****Table-2: Application Characteristics****

| **S.No** | **Characteristics** | **Description** | **Technology Used** |
| --- | --- | --- | --- |
| 1 | Open-Source Frameworks | Frameworks used for UI, backend and AI | ReactJS, Node.js, TensorFlow, Rasa |
| 2 | Security Implementations | Identity/access control, data protection, API security | OAuth 2.0, SHA-256, JWT, OWASP Controls |
| 3 | Scalable Architecture | Microservices-based modular, cloud-native deployment | Kubernetes, Docker, API Gateway |
| 4 | Availability | High availability using load balancers, active-active cloud setup | HAProxy, NGINX, AWS ALB, IBM Load Balancer |
| 5 | Performance | Use of CDN, caching, and auto-scaling to optimize performance | Cloudflare CDN, Redis Cache, Auto Scaling |

If you’d like, I can also:

* Create a **C4-style diagram** (Context, Container, Component, Code level)
* Generate a **visual architecture diagram** using tools like Mermaid, PlantUML, or draw.io
* Export this as a formatted document (Word, Excel, or PDF)