**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

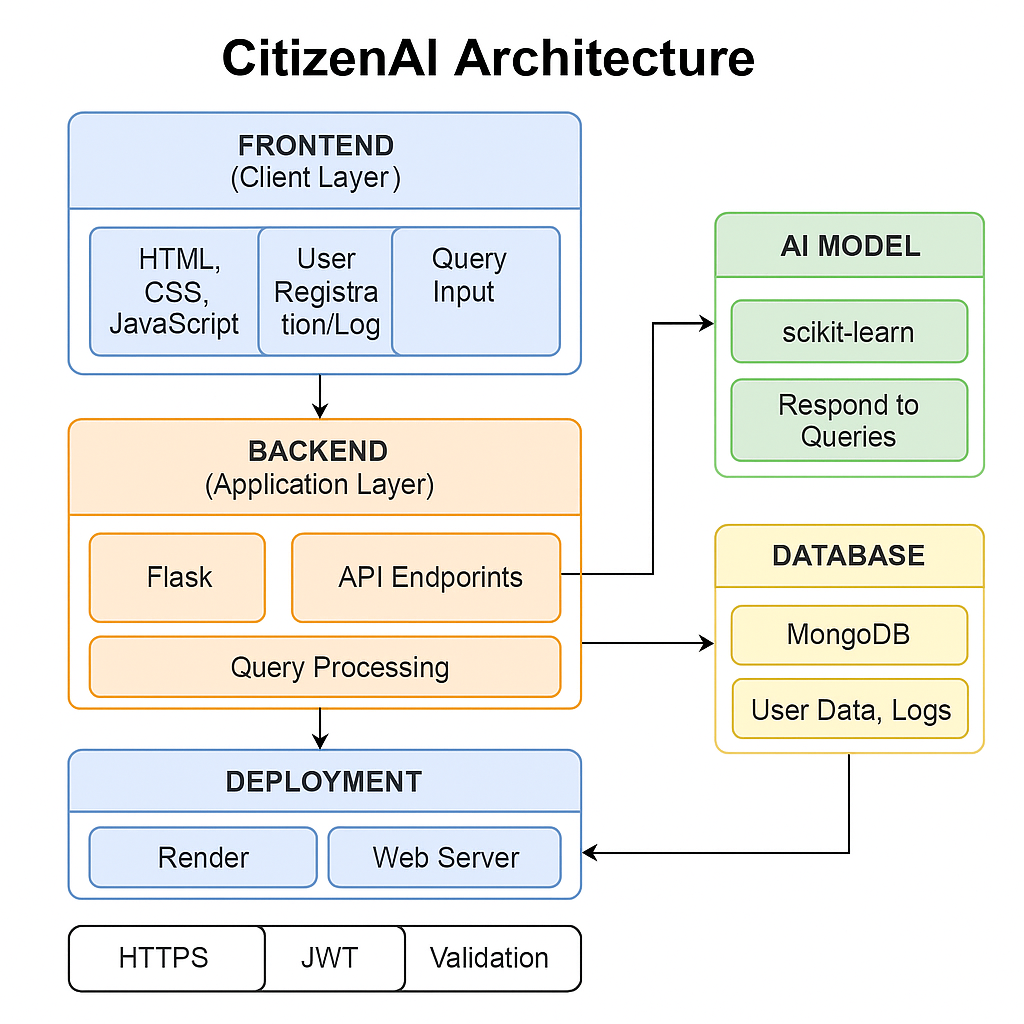
|  |  |
| --- | --- |
| Date | 01 July 2025 |
| Team ID | LTVIP2025TMID37541 |
| Project Name | Citizen AI – Intelligent Citizen Engagement Platform |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example: Order processing during pandemics for offline mode**

**Reference:** [**https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)



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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology (CitizenAI)** |
| 1 | User Interface | How users interact (web/mobile interface, chatbot) | HTML, CSS, JavaScript (optional: React JS) |
| 2 | Application Logic-1 | Business logic for handling requests | Python (Flask or FastAPI) |
| 3 | Application Logic-2 | Conversational logic using NLP | IBM Watson Assistant |
| 4 | Database | Local storage for user data, logs | MongoDB (NoSQL) |
| 5 | Cloud Database | Scalable cloud-based database | IBM Cloudant |
| 6 | File Storage | Store user logs, uploaded files | IBM Block Storage / Local Filesystem |
| 7 | External API-1 | Weather insights or contextual services | IBM Weather API |
| 8 | Machine Learning Model | Analyze user intent, classify requests | scikit-learn or custom-trained ML model |
| 9 | Infrastructure (Server/Cloud) | Hosting/deployment infrastructure | IBM Cloud Foundry, optionally Docker + Kubernetes |

**Table-2: Application Characteristics:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Characteristic** | **Applied Concept / Implementation** |
| **1** | Open-Source Frameworks | Flask, scikit-learn, MongoDB |
| **2** | Security Implementations | HTTPS, IAM controls |
| **3** | Scalable Architecture | 3-Tier Architecture (Frontend, Backend, Database), Flask APIs |
| **4** | Availability | IBM Cloud Foundry |
| **5** | Performance | Async Flask routing |

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)