**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

| Date | 31 January 3035 |
| --- | --- |
| Team ID | LTVIP2025TMID32175 |
| 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** |
| --- | --- | --- | --- |
|  | User Interface | How user interacts with application e.g.  Web UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript (for Web UI), Flask (for serving UI/chatbot interface) |
|  | Application Logic-1 | Logic for a process in the application | Python (Flask Framework) |
|  | Application Logic-2 | Logic for a process in the application | IBM Watson |
|  | Application Logic-3 | Logic for a process in the application | IBM Watson Assistant |
|  | Database | Data Type, Configurations etc. | Postgresql,Mongo DB. |
|  | Cloud Database | Database Service on Cloud | IBM DB2, IBM Cloudant etc. |
|  | File Storage | File storage requirements | IBM Block Storage or Other Storage Service or Local Filesystem |
|  | External API-1 | Purpose of External API used in the application | IBM Weather API, etc. |
|  | External API-2 | Purpose of External API used in the application | IBM Granite models |
|  | Machine Learning Model | Purpose of Machine Learning Model | IBM Granite Models, IBM Watson (Pre-trained and fine-tuned models) |
|  | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud  Local Server Configuration:  Cloud Server Configuration : | IBM Cloud Foundry / Kubernetes (on IBM Cloud) |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | List the open-source frameworks used | Flask (Python Web Framework), D3.js (for dashboard visualization) |
|  | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | e.gEncryption (data in transit/at rest), IAM Controls (Identity and Access Management), OAuth 2.0 (for social logins), API Key Management for external services |
|  | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Microservices Architecture (using Flask services), Containerization (e.g., Docker/Kubernetes) |
|  | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Load Balancers, Distributed Services across multiple availability zones, Automated Failover |
|  | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | Caching mechanisms (e.g., Redis), Asynchronous processing, Efficient API integrations, Content Delivery Networks (CDNs) for static assets |

**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)