**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Project Name | **Snack Squad: A Customizable Snack Ordering and Delivery App** |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The architecture includes the following logical and infrastructural components. It is designed to optimize scalability, maintainability, and performance in a cloud-enabled environment. The architecture supports both local and cloud deployment.

Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API’s etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1 | User Interface | Web and mobile-based interface for interaction | HTML, CSS, JavaScript, React Js |
| 2 | Application Logic-1 | Core business logic (order handling, cart, etc.) | Python (Flask/Django) |
| 3 | Application Logic-2 | Voice-to-text input for order placement | IBM Watson STT service |
| 4 | Application Logic-3 | Virtual assistant for help and suggestions | IBM Watson Assistant |
| 5 | Database | Stores user, order, and product information | MySQL |
| 6 | Cloud Database | Cloud-based replica for scale and backup | IBM Cloudant |
| 7 | File Storage | Snack images, receipts, static content | IBM Block Storage / Local Filesystem |
| 8 | External API-1 | Weather-aware delivery suggestions | IBM Weather API |
| 9 | External API-2 | Identity verification and age check | Aadhar API |
| 10 | Machine Learning Model | Personalized recommendations and order prediction | Object Recognition Model (Custom Model) |
| 11 | Infrastructure | Deployment environment | Cloud Foundry / Kubernetes / Local Server |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1 | Open-Source Frameworks | Frameworks and libraries used in development | React, Flask, Django, Bootstrap |
| 2 | Security Implementations | User data protection, secure APIs, hashed passwords | SHA-256, SSL, JWT, IAM Controls |
| 3 | Scalable Architecture | Modular and scalable deployment (3-tier, microservices) | Docker, Kubernetes, NGINX |
| 4 | Availability | Ensures uptime with load balancers and multi-zone servers | AWS Load Balancer, Auto-Scaling |
| 5 | Performance | Fast content delivery, caching, optimized API calls | Redis Cache, CDN, API Gateway |

### References

* <https://c4model.com/>
* <https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>
* <https://www.ibm.com/cloud/architecture>
* <https://aws.amazon.com/architecture>
* <https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>