**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

| Date | 31 January 3035 |
| --- | --- |
| Team ID | LTVIP2025TMID55575 |
| Project Name | SHOPSMART-DIGITAL GROCERY STORE EXPERIENCE |
| 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 | Web application interface for browsing and ordering groceries. | HTML, CSS, JavaScript / Angular Js / React Js etc. |
|  | Application Logic-1 | Handles product browsing, cart, and checkout logic. | Node.js ( Express ) |
|  | Application Logic-2 | Handles voice-to-text search for groceries. | Google Speech-to-Text API |
|  | Application Logic-3 | Customer support chatbot | Diagflow or IBM Watson Assistant |
|  | Database | Stores product, order, and user data. | MongoDB |
|  | Cloud Database | Cloud-hosted database for scalability. | MongoDB Atlas |
|  | File Storage | Stores product images and customer uploads. | Firebase Storage / AWS S3 |
|  | External API-1 | Weather-based delivery time adjustment. | OpenWeather API |
|  | External API-2 | Address verification through Aadhaar. | UIDAI Aadhaar API |
|  | Machine Learning Model | Product recommendation system. | Custom ML model hosted on Google Colab/Cloud |
|  | Infrastructure (Server / Cloud) | Deployment environment for the entire app. | FRONTEND : Vercel /  BACKEND : Heroku or Render / Cloud:  Firebase Hosting. |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | React, Node.js, Express, Bootstrap | JavaScript, React.js, Node.js |
|  | Security Implementations | User authentication, encryption, secure APIs | JWT, HTTPS, bcrypt, OAuth2 |
|  | Scalable Architecture | 3-tier architecture with separate frontend, backend, and database layers. | Micro-services (optional ) , REST APIs |
|  | Availability | Cloud hosting with automatic scaling, CDN for static files. | Firebase Hosting, Vercel, Load balancer. |
|  | Performance | Use of cache for frequently accessed data, CDN for faster loading. | Redis (optional), Cloudflare CDN |

**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)