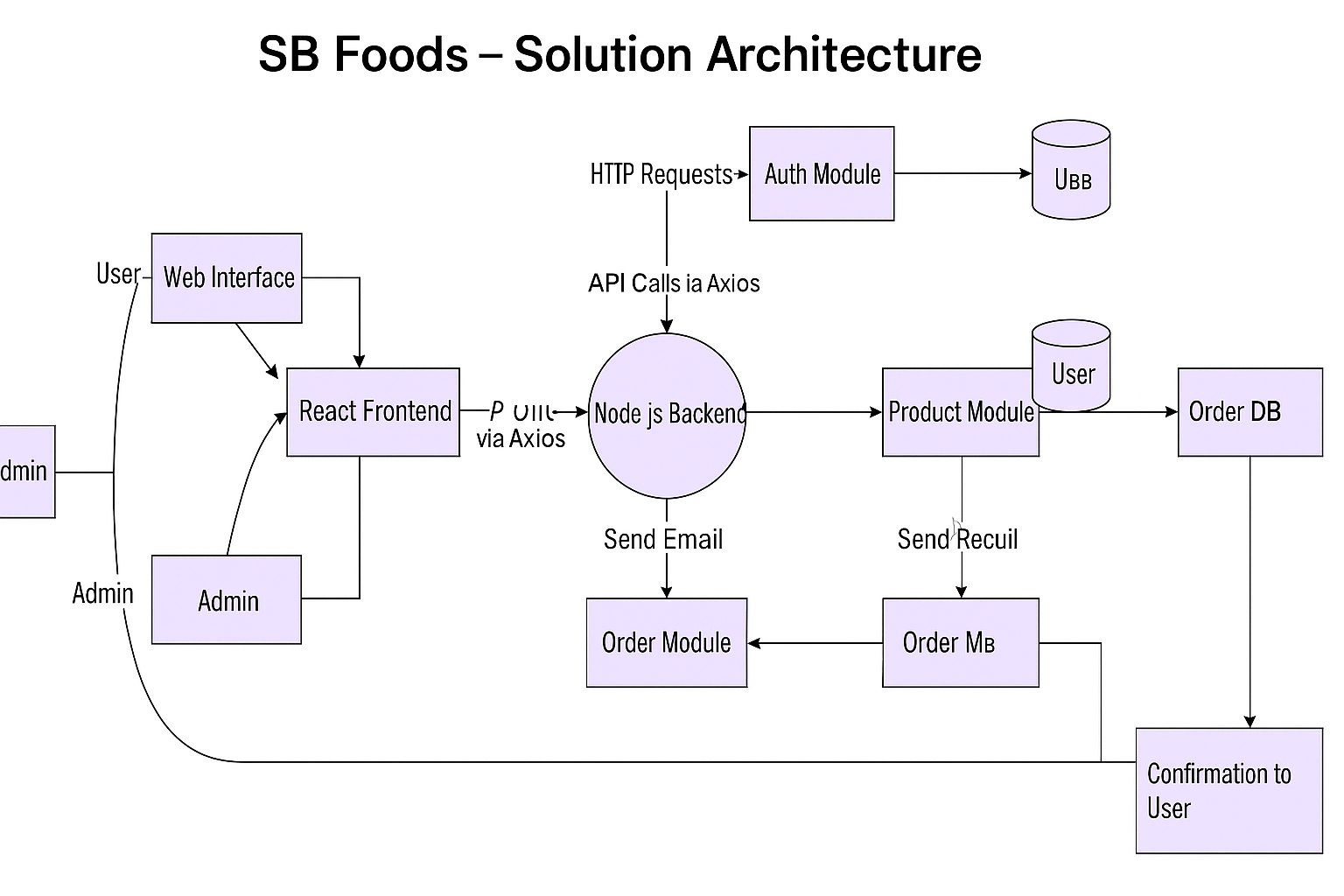
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 5 February 2025 |
| Team ID | LTVIP2025TMID55302 |
| Project Name | OderGo – Food Ordering MERN App |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

OderGo is a full-stack cloud-based food ordering system that supports user authentication, restaurant listings, item filtering, cart management, and order processing. It follows a three-tier architecture with a responsive frontend, an Express.js-based backend API, and a MongoDB NoSQL database hosted on the cloud. External APIs such as Google Maps are integrated for enhanced location-based services.

**Architecture Diagram:**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web UI for users, restaurants, and admins | React.js, HTML5, CSS3, Bootstrap |
|  | Application Logic-1 | Backend logic for authentication and order flow | Node.js, Express.js |
|  | Application Logic-2 | Cart and checkout logic, token-based middleware | Express.js with JWT |
|  | Application Logic-3 | Admin dashboard for managing restaurants and menus | Node.js controllers, Express routing |
|  | Database | NoSQL data store for users, items, orders | MongoDB |
|  | Cloud Database | Cloud-hosted database | |  | | --- | |  |  |  | | --- | | MongoDB Atlas | |
|  | File Storage | Images for food items and restaurant logos | |  | | --- | |  |  |  | | --- | | Cloudinary / Local filesystem | |
|  | External API-1 | Location-based services | Google Maps API |
|  | External API-2 | |  | | --- | |  |  |  | | --- | | Food image optimization | | Cloudinary API |
|  | Machine Learning Model | (Optional future extension) Food recommendation engine | TensorFlow.js / Custom ML model (planned) |
|  | Infrastructure (Server / Cloud) | |  | | --- | |  |  |  | | --- | | Application deployment and environment |   : | Local for dev, Render/Vercel for deployment |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | MERN Stack components | React.js, Node.js, Express.js, MongoDB |
|  | Security Implementations | JWT authentication, password hashing, role-based access | bcrypt, JWT, Helmet.js, CORS, HTTPS. |
|  | Scalable Architecture | Modular design with API-based routing and service separation | 3-tier architecture, Express.js, MongoDB Atlas |
|  | Availability | Frontend/backend deployed on scalable platforms, cloud DB with replicas | Vercel, Render, MongoDB Atlas (multi-region) |
|  | Performance | Backend caching, image CDN, optimized queries, pagination | Cloudinary CDN, Axios, efficient Mongo queries |

**References:**

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

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

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

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