Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	7 February 2026
Team ID	LTVIP2026TMIDS79872
Project Name	OrderOnTheGo: Your On-Demand Food Ordering Solution
Maximum Marks	4 Marks

1. Technical Architecture (Table 1)

This table defines the core components, their purpose within SB Foods, and the specific MERN stack technologies used.

S.No	Component	Description	Technology
1.	**User Interface**	Responsive web application for customers and restaurant owners.	**React.js**, HTML5, CSS3, JavaScript.
2.	**Application Logic-1**	User and Restaurant Authentication (Login/Register) and Profile Management.	**Node.js, Express.js**.
3.	**Application Logic-2**	Order Processing: Logic for cart management, menu filtering, and order placement.	**Node.js**.
4.	**Application Logic-3**	Restaurant & Admin Dashboards: Logic for product CRUD operations and order monitoring.	**Node.js**.
5.	**Database**	Storage for User, Restaurant, Product, Cart, and Order collections.	**MongoDB** (NoSQL).

S.No	Component	Description	Technology
6.	**Cloud Database**	Cloud-based managed database service for data persistence.	**MongoDB Atlas**.
7.	**File Storage**	Storage for food item images and restaurant logos.	Local Filesystem / Cloudinary.
8.	**External API-1**	Payment gateway integration for processing food orders.	Stripe API / Razorpay API.
9.	**Infrastructure**	Environment where the MERN application is deployed.	**Local:** Node Runtime; **Cloud:** Render / Vercel.

2. Application Characteristics (Table 2)

This section highlights the architectural strengths and security measures of SB Foods.

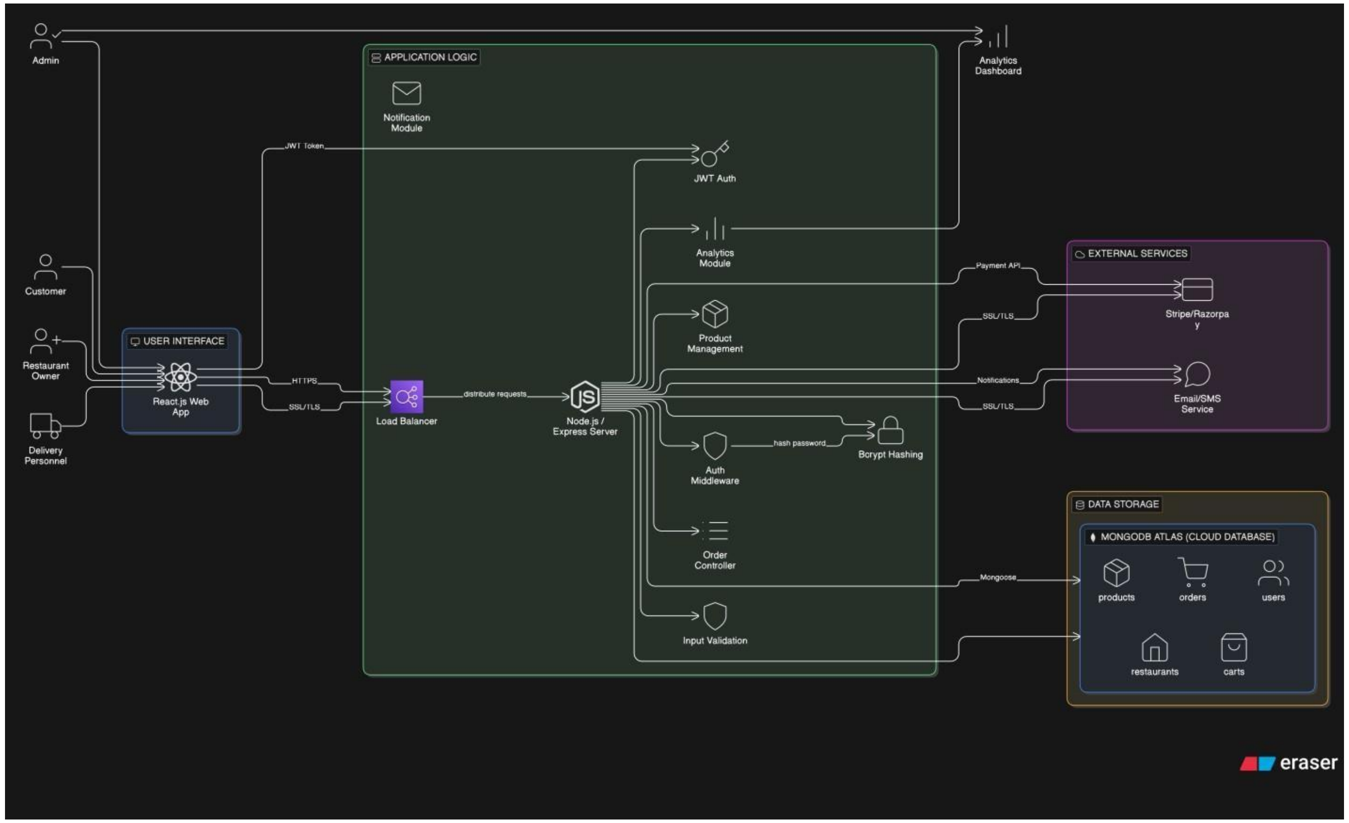
S.No	Characteristics	Description	Technology
1.	**Open-Source Frameworks**	**Primary libraries used for rapid full-stack development.**	**MERN Stack (MongoDB, Express, React, Node).**
2.	**Security Implementations**	**Access control for Users, Restaurants, and Admins; password protection.**	**JWT (JSON Web Tokens), Bcrypt for hashing, Middleware guards.**
3.	**Scalable Architecture**	**Decoupled frontend and backend allowing independent scaling.**	**3-Tier Architecture (Client-Server-Database).**
4.	**Availability**	**Ensuring the app remains accessible for late-night cravings.**	**Distributed Cloud Hosting (Render/Atlas).**
5.	**Performance**	**Fast retrieval of menu items and optimized UI rendering.**	**React Virtual DOM, Mongoose Indexing, Image Optimization.**

3. Architectural Guidelines

- **Infrastructural Demarcation: The application uses a Local development environment (Node.js/Compass) and transitions to Cloud (Atlas/Render) for production.**

- **External Interfaces: Integrated via RESTful APIs for payments and map/location services.**

- **Data Storage: All relational-style data (Users linked to Orders) is handled via MongoDB Collections.**


Admin
Customer
Restaurant Owner
Delivery Personnel
USER INTERFACE
React.js Web App
JWT Token
HTTPS
SSL/TLS
APPLICATION LOGIC
Notification Module
Load Balancer
distribute requests
Node.js / Express Server
JWT Auth
Analytics Module
Product Management
Auth Middleware
hash password
Bcrypt Hashing
Order Controller
Input Validation
Analytics Dashboard
Payment API
SSL/TLS
Notifications
SSL/TLS
EXTERNAL SERVICES
Stripe/Razorpa y
Email/SMS Service
Mongoose
DATA STORAGE
MONGODB ATLAS (CLOUD DATABASE)
products
orders
users
restaurants
carts
eraser