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

Date	11 April 2025
Team ID	SWTID1742575574
Project Name	Grocery webapp
Maximum Marks	4 Marks

Technical Architecture:

The technical architecture of the Grocery WebApp replicates a real-time grocery ordering and delivery system inspired by Blinkit, built using the MERN stack (MongoDB, Express.js, React.js, Node.js). This architecture follows a modular and scalable approach, ensuring flexibility, maintainability, and efficient handling of concurrent user interactions.

The frontend is developed using React.js and hosted on Netlify, enabling fast, secure, and continuous deployment with seamless user interactions. It offers a responsive and intuitive UI that supports real-time updates and dynamic rendering of components such as product listings, carts, and checkout pages.

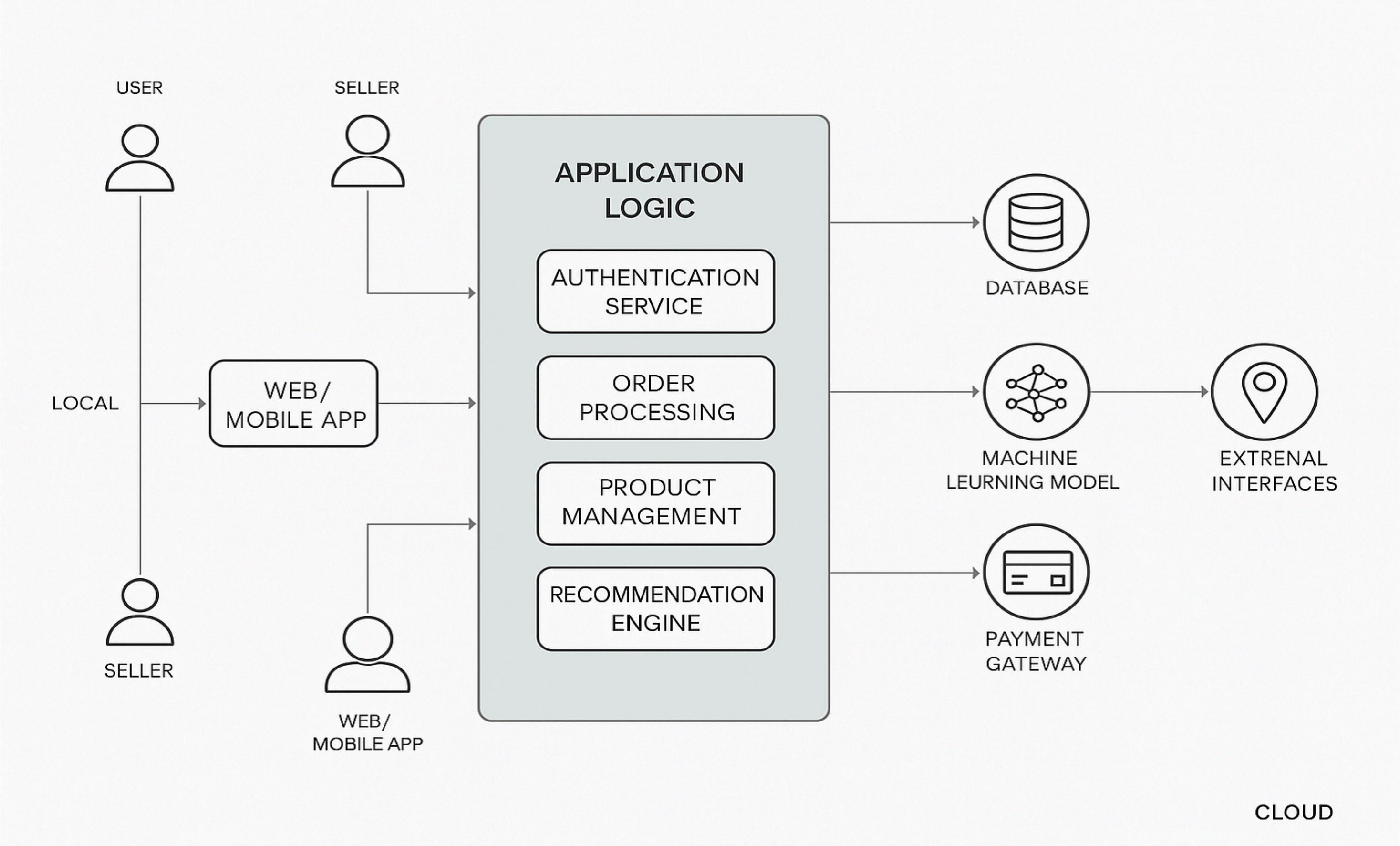
The backend is built with Node.js and Express.js, and deployed on Render, ensuring scalability and reliability. The backend handles all business logic, including user management, order processing, product inventory handling, and communication with external services.

MongoDB Atlas, a cloud-hosted NoSQL database, serves as the central data store. It ensures high availability, secure access, and scalable data management for product catalogs, user data, orders, and delivery statuses.

This architecture models a real-time grocery ordering and delivery system cloned from Blinkit. The Grocery WebApp is built on a **MERN stack** with external integrations. It includes secure and scalable services deployed using cloud-based hosting solutions.

- Modular architecture with frontend, backend, database, and third-party integrations

- Hosted frontend on **Netlify**, backend on **Render**, database on **MongoDB Atlas (Cloud)**
- Payment integration via **Razorpay**
- User authentication with **JWT**



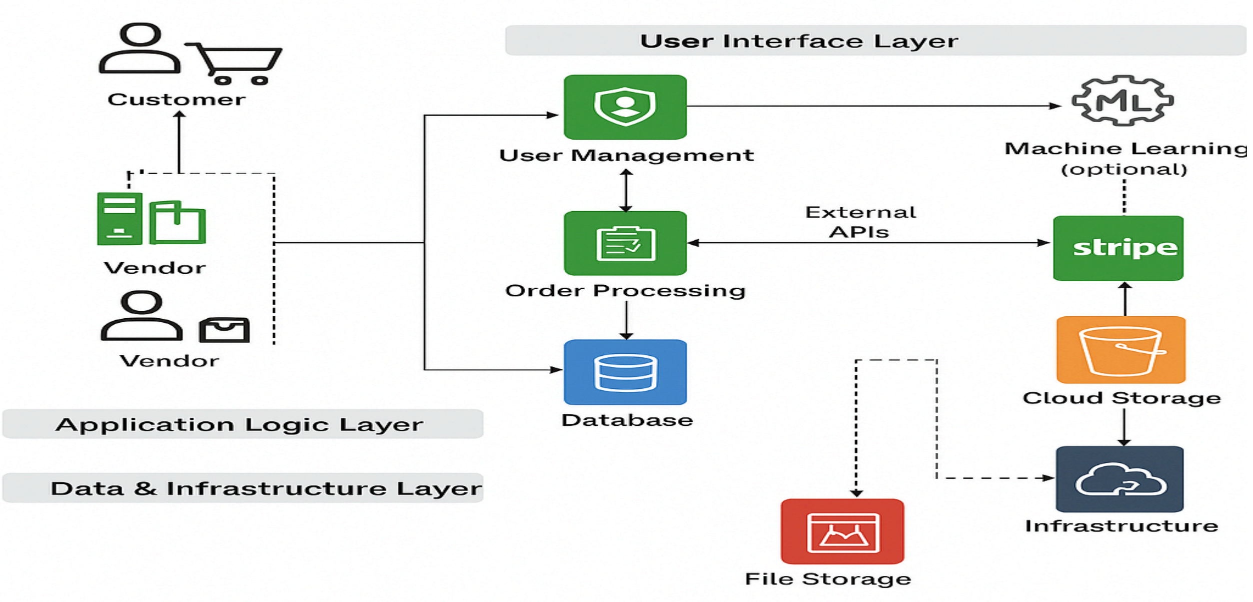
Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web-based UI for customers	HTML, CSS, Tailwind CSS, React.js
2.	Application Logic- User Management	User authentication, cart logic, role management	Node.js, Express.js
3.	Application Logic- Order Processing	Inventory management, product listing for sellers	Node.js, Express.js
4.	Application Logic-3	Admin panel logic, order tracking, verification controls	Node.js, Express.js
5.	Database	Storage of users, products, orders, seller details	MongoDB (NoSQL)
6.	Cloud Database	Scalable cloud-hosted database	MongoDB Atlas
7.	File Storage	Stores product images and user files.	Cloudinary
8.	External API- Payment Gateway	Payment gateway API integration	Razorpay or Stripe API
9.	External API- Location Services	Fetches vendor locations for delivery optimization	Google Maps API

Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	React.js for frontend, Node.js and Express.js for backend	Mern stack
2.	Security Implementations	Ensures data protection, encryption, and secure access.	JWT, Razorpay, HTTPS

S.No	Characteristics	Description	Technology
3.	Scalable Architecture	Built with microservices to handle high traffic.	Cloud-hosted Node + MongoDB
4.	Availability	Hosted using cloud services ensuring 24x7 uptime and redundancy	Netlify, Render, MongoDB Atlas
5.	Performance	Optimized for fast response time & caching	Redis, CloudFront CDN, IndexedDB for client-side caching


User Interface Layer
Customer
User Management
Machine Learning (optional)
Vendor
External APIs
Order Processing
stripe
Vendor
Database
Cloud Storage
Application Logic Layer
Data & Infrastructure Layer
Infrastructure
File Storage