Project Design Phase
Solution Architecture

Date	5 April 2025
Team ID	SWTID1742575574
Project Name	Grocery webapp
Maximum Marks	4 Marks

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Our Solution: Grocery WebApp – Blinkit Clone

We are building an online grocery delivery platform inspired by Blinkit using the MERN Stack (MongoDB, Express.js, React.js, Node.js). The web app provides functionality for users to browse products, manage carts, place orders, and securely complete payments. It also includes dedicated panels for sellers to manage listings and admins to oversee platform activity.

Architecture Overview:

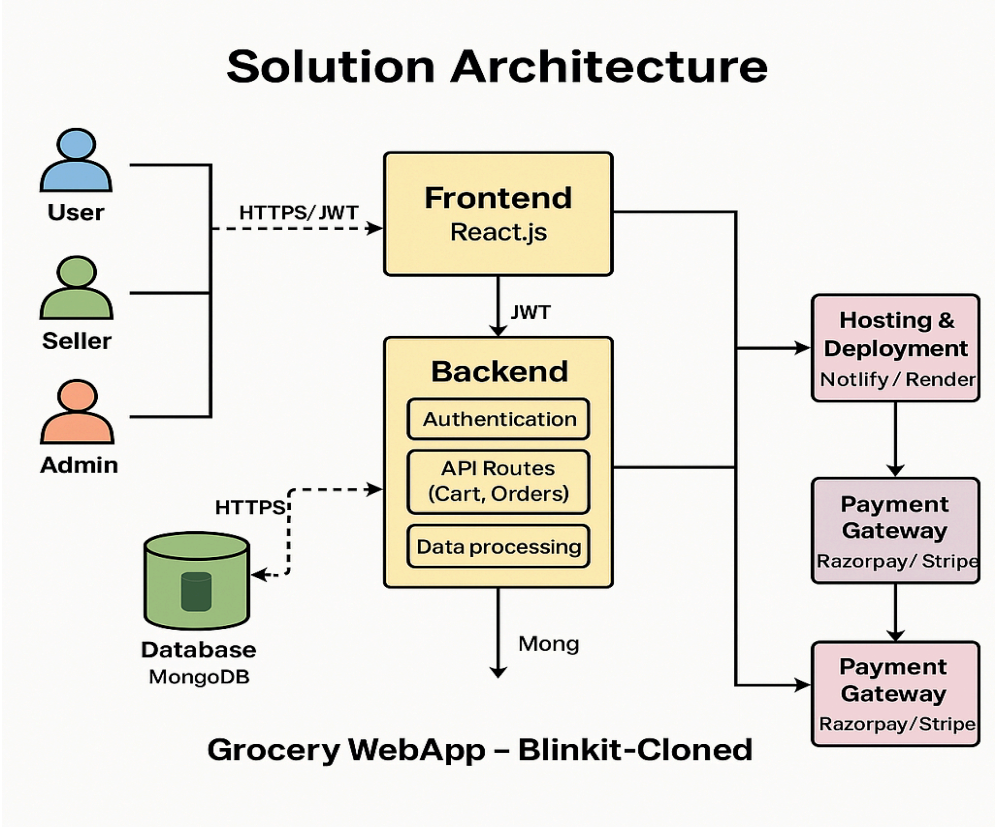
- **Frontend:** React.js with Redux Toolkit (State Management)
- **Backend:** Node.js with Express.js (REST API Development)
- **Database:** MongoDB (NoSQL database to store users, products, orders)
- **Authentication:** JWT (Secure session management)
- **Payment Integration:** Razorpay (Online payment gateway)
- **Hosting:** Netlify (frontend), Render/Heroku (backend)

Key Modules:

1. **User Module:**
    - Registration/Login
    - Product Search & Filters
    - Add to Cart, Wishlist
    - Address Book & Payment
    - Order History
2. **Seller Module:**
    - Product Upload
    - Inventory Management
    - Order Management
3. **Admin Module:**
    - User & Seller Verification
    - Order & Transaction Monitoring
    - Support/Complaints Management

Architecture Diagram:

- The React frontend sends requests to the backend.
- Backend handles routing, processing, authentication (JWT), and communicates with MongoDB and Razorpay API.
- MongoDB stores user/product/order data; Razorpay handles secure payment processing.



Development Phases:

Phases	Deliverables
Phase 1	Set up environment, GitHub repo, and install dependencies
Phase 2	Design frontend UI and backend routes
Phase 3	Implement core user features: login, product listing, cart
Phase 4	Seller dashboard, order APIs, Razorpay integration
Phase 5	Admin panel and analytics
Phase 6	Testing, deployment, and documentation