

Name : Hency Depani

Enroll. No. : 92200133014

## System Design and Architecture

**Project Title : Online Grocery Delivery Web App Blink-Shop-Now**

### **1. Modular Design**

The Blinkit Clone project uses a design where the system is built from separate, smaller parts. Each part has its own job and can connect with other parts in specific ways. This helps make the application easier to build, take care of, and improve later on.

#### **The main parts include:**

User Interface (Front-End): Created with React.js, this is what the user sees, like looking at products, searching, putting items in a cart, and paying.

Application Logic (Back-End): Made with Node.js and Express.js, this takes care of important tasks such as processing orders, handling payments, and talking to the database.

Database Layer: Uses MongoDB to keep track of product information, customer profiles, orders, and past transactions.

API Layer: Works as a connection between what the user sees and the behind-the-scenes operations, making sure information moves easily.

Admin Panel: Lets administrators add or delete products, keep up with sales, handle stock, and check data.

## **2. Technology Stack**

Picking the right tech setup is key for speed, safety, and growing later on. The setup used for this work is made up of:

- Front-End: React.js to make a quick, easy to use, and good looking UI.
- Back-End: Node.js with Express.js to deal with requests and things that happen on the server.
- Database: MongoDB as a NoSQL database, which is great for holding many product lists and changing user info.
- Authentication & Security: JWT (JSON Web Tokens) and bcrypt to keep login info and passwords safe.
- Payment Gateway: Razorpay or Stripe working together to make sure payments are safe and trustworthy.
- Hosting & Deployment: AWS or Google Cloud to put the app out there and help it grow when needed.
- Version Control: GitHub for the team to work together and keep track of the code.

## **3. Scalability Planning**

Because the app might get bigger as more people use it and traffic increases, planning for growth is really important. The plan for growing includes:

- Horizontal Scaling: Putting in more servers to handle when there are more people using the app.
- Load Balancing: Sending user requests to different servers so none of them get too busy.
- Database Scaling (Sharding): Dividing MongoDB into smaller

pieces to deal with lots of info and make searches quick.

- Caching (Redis): Keeping data used often (like lists of products) in the computer's memory to respond faster.
- Content Delivery Network (CDN): Getting product pictures and files that don't change to people quickly based on where they are.
- Monitoring & Alerts: Using tools in the cloud to watch how things are working and stop the app from going down.