

Requirement Gathering and Analysis Phase Technology Stack (Architecture & Stack)

Date	29-June-24
Team ID	SWTID1720199985
Project Name	Grocery Webapp
Maximum Marks	4 Marks

Technical Architecture:

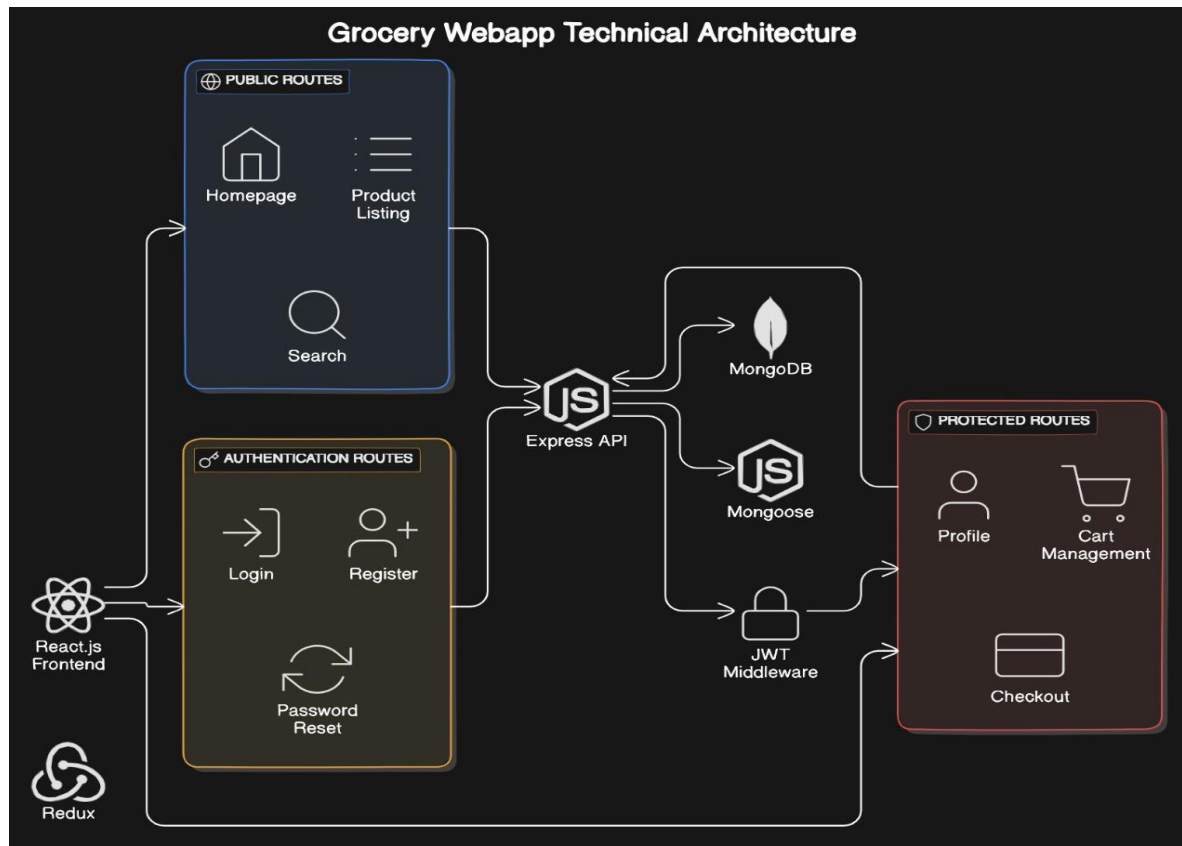


Table-1 : Components & Technologies:

<u>S.no</u>	Component	Description	Technology
1	Customer Support	System for managing customer inquiries and requests. This could be a separate web application or integrated into the main web app.	Web application framework (e.g., Django, Ruby on Rails)
2	Product Service	Backend service that manages product data (add, edit, delete products), retrieves product information based on user searches, and handles product filtering and sorting.	Node.js with MongoDB
3	API Gateway	Single entry point for API requests from the UI, routes requests to appropriate backend services.	Cloud-based API Gateway service
4	User Interface	Web application for users to browse products, register, login, and manage their shopping cart.	HTML, SCSS, JavaScript / React.js
5	Cart Service	Backend service that manages user shopping carts, including adding, removing, and updating cart items, and calculating total cost.	Node.js with Redis (for fast cart updates)
6	Order Service	Backend service that processes checkout requests, manages order creation and payment, and sends order confirmation emails.	Node.js with a payment gateway
7	User Service	Backend service that handles user registration, login, and profile management, including order history and saved items.	Node.js with MongoDB
8	Search Service	Backend service that processes user search queries and retrieves relevant product listings from the database.	Node.js with Elasticsearch (for powerful product search)
9	Payment Gateway	Service that facilitates secure online payments for groceries.	Third-party payment gateway service

Table-2: Application Characteristics:

<u>S.No</u>	Characteristics	Description	Technology	
1	Performance	Delivers fast response times for product searches, details, and other functionalities, resulting in a smooth user experience.	Prisma ORM, MongoDB, Node.js, Express	
2	Open-Source Frameworks	Open-source frameworks enable dynamic user interfaces, robust backend support, scalable data storage, and a seamless user experience.	Prisma, Express.js, React.js	
3	Scalable Architecture	Allows the application to adapt to a growing user base and data volume, ensuring smooth operation even with high traffic.	MongoDB, RESTful API, Node.js	
4	Security Implementations	Safeguard user data and control access to various features like product listings and user profiles.	bcrypt, JSON web tokens (JWT), Middleware-Token Verification, Role Based Access	
5	Availability	Minimizes downtime and guarantees continuous access to grocery shopping features for users.	Multiple Node.js instances	