Planning and Logic for ShopSmart Project

**Project Name:** ShopSmart – Your Digital Grocery Store Experience

**Team ID:**  LTVIP2025TMID54978

**Date:** 26 June 2025

# Planning and Development Logic

The **ShopSmart** project is a MERN-stack based full-stack web application aimed at digitizing grocery shopping. It provides users with a seamless experience of browsing, selecting, and purchasing grocery items with secure login, intuitive UI, and efficient backend logic.

# Frontend (React.js + CSS/Bootstrap)

## Login / Registration Page:

* + Users can register using their name, email, and password.
  + JWT-based login and logout functionalities are implemented.

## Homepage:

* + Displays all available grocery items.
  + Includes search bar and category filters (e.g., fruits, vegetables, dairy, snacks).

## Product Details:

* + Clicking a product shows detailed view with price, quantity selector, description, brand, and Add to Cart button.

## Cart Page:

* + Shows items added to cart.
  + Quantity can be updated or items removed.
  + Total amount auto-updated.

## Order Summary / Checkout:

* + Displays selected items, total amount.
  + Place Order button triggers backend API to store order details.

## Order History:

* + Displays past orders with date, status (Pending, Delivered), and total price.

# Backend (Node.js + Express.js)

## Routes Created:

* + /api/users: For registration, login, profile.
  + /api/products: CRUD operations for grocery items (admin only), and user- facing product fetch.
  + /api/cart: Add/remove/update items in cart.
  + /api/orders: Place order, fetch order history.

## Controllers:

* + Each route has a controller handling the business logic.
  + Middleware for error handling and authentication is applied globally.

## Security:

* + Passwords hashed using bcrypt.
  + JWT issued on login; verified for protected routes.

# Database (MongoDB with Mongoose)

## Collections:

* + Users: name, email, password, role
  + Products: name, price, category, brand, image, stock
  + CartItems: userId, productId, quantity
  + Orders: userId, products[], totalPrice, orderDate, status

## Mongoose Models:

* + Defined for each collection with appropriate validations and relationships.

# Authentication (JWT)

* Tokens are generated on successful login.
* Stored in browser localStorage or cookies.
* Middleware ensures protected routes like checkout and order history.

# Admin Panel Features

* Product management (add/edit/delete).
* View all user orders.
* Update order status (e.g., Delivered, Cancelled).

**Tools & Technologies**

|  |  |
| --- | --- |
| **Stack Component** | **Technology Used** |
| Frontend | React.js, Bootstrap, CSS |
| Backend | Node.js, Express.js |
| Database | MongoDB + Mongoose |
| Authentication | JWT, bcrypt |
| Hosting | (Planned) Render/ Netlify |