

ShopSmart: Digital Grocery Web App

Project Documentation:-

1. Frontend Development

Overview

Frontend development involves building the **user interface (UI)** and implementing visual elements for the web app. It ensures a seamless and engaging user experience.

Setup & Node.js + MongoDB Connection

- Video tutorial: [Frontend Setup & Database Connection](#)

UI Design

- Create visually appealing and consistent design using modern principles.
- Use tools like **Figma, Adobe XD, Sketch, or InVision** for wireframes and mockups.
- Pay attention to **typography, color schemes, spacing, visual hierarchy**.
- Implement **responsive design** using **Bootstrap, Tailwind CSS**, and media queries.
- Test across multiple devices for seamless experience.

Product Catalog

- Display product images, titles, descriptions, prices, and other details.
- Implement **search, filtering, and sorting** functionalities.

Shopping Cart & Checkout

- Add products to the cart, update quantities, remove items.
- Multi-step checkout: shipping info → payment → order review.

User Authentication & Account Management

- Registration, login, and profile pages.
- Manage personal info, addresses, payment methods, and order history.
- Implement authentication guards for restricted access.

Payment Integration

- Integrate **Stripe, PayPal, or Braintree**.

- Handle secure payment flow, success/failure feedback.

2. Backend Development

Setup

- Create project folder and package.json using npm init.
- Install dependencies: **Express.js, Mongoose, dotenv, cors, etc.**

Database Configuration

- Use **MongoDB** (local or cloud).
- Create collections: users, products, bookings, orders, etc.

Express.js Server

- Configure middleware: body-parser, cors.
- Create API endpoints for **users, products, bookings, authentication**.

Data Models

- Use **Mongoose schemas** for data entities.
- Implement CRUD operations.

API Design

- Design RESTful APIs for **user management, product catalog, cart, orders, payment, shipping**.
- Ensure **security, validation, and proper responses**.

User Management

- Registration/login using **JWT or session-based authentication**.
- Hash passwords and secure credentials.
- Protect sensitive routes with middleware.

Product & Inventory Management

- Maintain product details, pricing, availability.
- Handle search, filters, and inventory updates.

Shopping Cart & Orders

- Add/modify cart items, place orders.
- Calculate totals, apply discounts, manage order lifecycle.

Payment Gateway Integration

- Integrate payment SDK/API.
- Handle transactions and callback endpoints.

Shipping & Logistics

- Integrate APIs for shipping cost, labels, and tracking.

External Services

- Email notifications, analytics, CRM integrations as needed.

Security & Data Protection

- Input validation and sanitization.
- Encryption and access control.
- Error handling and logging for monitoring.

3. Database Integration

- Connect MongoDB to backend using **Mongoose**.
- Ensure CRUD operations and data consistency.

4. User Roles & Permissions

Admin

- Full control over system.
- Manage shops, products, users, and bookings.
- Generate reports & analytics.

User

- Browse and search products.
- Manage bookings and cart.
- View order history and update profile.

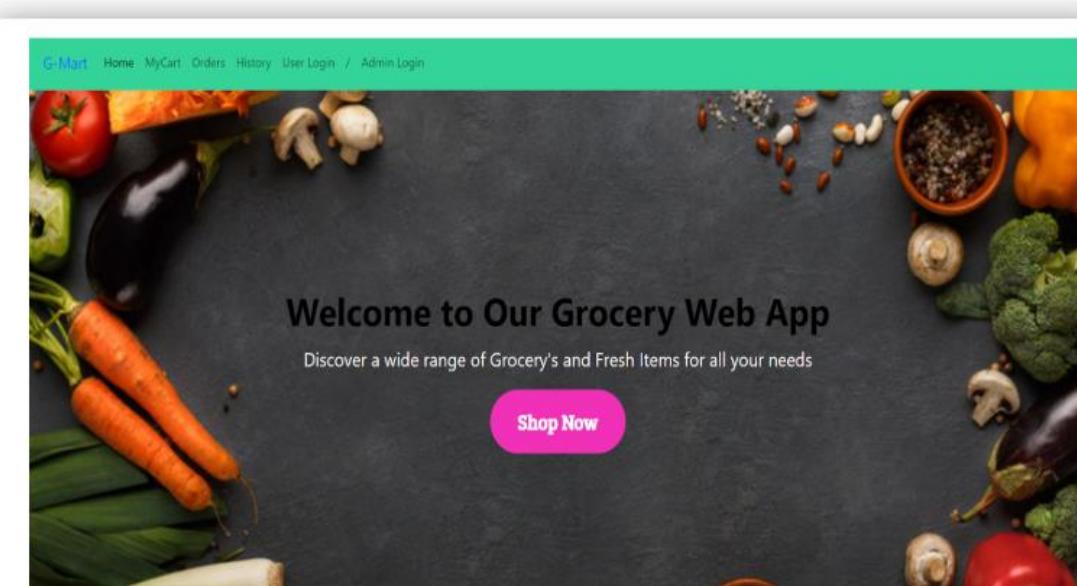
5. Project Flow

1. **Frontend Development:** Angular-based UI development.
2. **Backend Development:** Node.js + Express.js + MongoDB.
3. **Integration:** Connect frontend with backend and database.

Integration

Integration is the process of combining and connecting the frontend and backend components of the online flower shop web application to create a unified and fully functional system. It involves establishing communication channels, exchanging data, and ensuring seamless interaction between the frontend UI and backend APIs. The following activities are part of the integration process:

Landing page:-

A screenshot of the user login interface. It consists of a white rectangular box with rounded corners. At the top center, the word "Login" is displayed in a bold, black font. Below this, there are two input fields: one for "Email" containing the placeholder "Enter email" and another for "Password" containing the placeholder "Enter password". Both input fields have a thin gray border. At the bottom of the box is a blue rectangular button with the word "Login" in white. Below the button, a small line of text reads "Don't have an account? [Sign Up](#)". The entire login form is set against a white background.

Items Page:-

G-Mart Home MyCart Orders History Logout

Search By Product Name

Filter By Category

Products



Apple
\$200

[Buy Now](#) [Add to Cart](#)



orange
\$120

[Buy Now](#) [Add to Cart](#)



Milk
\$80

[Buy Now](#) [Add to Cart](#)



Cashew
\$800

[Buy Now](#) [Add to Cart](#)



Chicken
\$250

[Buy Now](#) [Add to Get](#)

My Cart:-

G-Mart Home MyCart Orders History Logout

My Cart



Apple
\$200

[Remove from Cart](#) [Buy this](#)



Milk
\$80

[Remove from Cart](#) [Buy this](#)



Chicken
\$250

[Remove from Cart](#) [Buy this](#)

My Orders Page:-

The screenshot shows a web application interface for managing orders. At the top, there is a navigation bar with links for Home, MyCart, Orders, History, and Logout. The main content area is titled "My Orders". Below the title, there is a detailed view of a single order, enclosed in a light gray box. The order information includes:

- Order ID:** 6614b085c30b51d3c700f0a
- Name:** syed arshad
- Phone:** 9505221870
- Date:** 2024-04-09T03:05:41.563Z
- Price:** 400
- Status:** Pending
- Payment Method:** credit

My History Page:-

The screenshot shows a web application interface for viewing order history. At the top, there is a navigation bar with links for Home, MyCart, Orders, History, and Logout. The main content area is titled "My History". Below the title, there is a detailed view of a single order, enclosed in a light gray box. The order information includes:

- Order ID:** 6614afddc30b51d3c700fb4
- Name:** syed arshad
- Phone:** 9505221870
- Date:** 2024-04-09T02:36:47.498Z
- Price:** 400
- Status:** Delivered
- Payment Method:** debit

Add Product page:-

The screenshot shows a web application interface titled "Grocery Web App". At the top, there is a navigation bar with links: Dashboard, Users, Products, Add product, Orders, and Logout. Below the navigation bar, the title "Add Product" is centered. The main form consists of several input fields and dropdown menus:

Product Name	Rating	Price
Enter product name	Enter product rating	Enter product price

Image URL	Category	Count in Stock
Enter image URL	Select Category	Enter count in stock

Description
Enter product description

Add Product

Admin Orders Page:-

This screenshot is identical to the one above, showing the "Add Product" page of the "Grocery Web App". It features the same navigation bar, title, and form structure for adding a new product.

Admin Orders Page:-

The screenshot shows the "Orders" page of the "Grocery Web App". The top navigation bar includes links for Dashboard, Users, Products, Add product, Orders, and Logout. The main content area is currently empty, indicating no orders are present.