# Full Stack Development with MERN
# ShopSmart – Project Documentation

## 1. Introduction

**Project Title:** *ShopSmart – A Digital Grocery Store*

**Team Members:**
- Bathini Aneela– Full Stack Developer
- Bellamkonda Homa shankar – Frontend Developer
- Badugu Deepthi – Backend Developer
- Buraga Naga raja triveni – UI/UX Designer

## 2. Project Overview

**Purpose:**
ShopSmart is an online grocery store platform built using the MERN stack, aimed at offering users a smooth and interactive grocery shopping experience.

**Features:**
- Product Listing & Categorization
- Add to Cart & Quantity Update
- Order Placement (Simulation)
- Admin Dashboard
- Search & Filter Functionality
- Fully Responsive UI

## 3. Architecture

**Frontend:**
Built with React functional components and hooks. Uses Axios for API calls and React Router for navigation.

**Backend:**
Node.js and Express.js with RESTful API architecture. JWT-based authentication and role-based access control.

**Database:**
MongoDB with Mongoose for schema modeling and CRUD operations. Collections: users, products, orders.

## 4. Setup Instructions

**Prerequisites:**

- Node.js >= 14
- MongoDB (local or cloud)
- Git

**Installation:**

1. Clone repo:

```
 git clone https://github.com/your-username/shopsmart.git
```

2. Install dependencies:

```
 cd client && npm install
 cd server && npm install
```

3. Set environment variables in server/.env

## 5. Folder Structure

**Client:**

```
client/
├── public/
├── src/
│   ├── components/
│   ├── pages/
│   ├── context/
│   ├── App.js
│   └── index.js
```

**Server:**

```
server/
├── controllers/
├── models/
├── routes/
├── middleware/
├── config/
└── server.js
```

## 6. Running the Application

**Frontend:**

```
Npm install
```

```
set NODE_OPTIONS=--openssl-legacy-provider
npm start
```

**Backend:**

```
cd backend
npm start
```

## 7. API Documentation

**Endpoints:**

- POST /api/users/register: Register a new user
- POST /api/users/login: Authenticate user and return token
- GET /api/products: Fetch all products
- POST /api/cart: Add to cart (Auth required)
- GET /api/cart: View cart (Auth required)
- POST /api/admin/products: Add/edit product (Admin only)

## 8. Authentication

JWT-based authentication.
Tokens are issued on login and stored in localStorage. Middleware protects routes based on user role.

## 9. User Interface

Include screenshots or animated GIFs of:

- Landing Page
- Product List
- Cart Page
- Login/Register
- Admin Dashboard

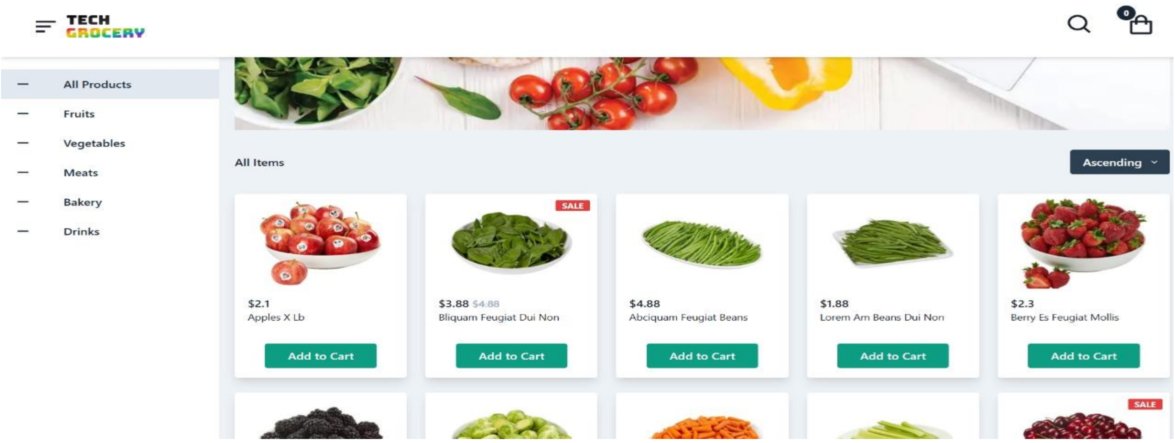
## 10. Testing

Manual testing using browser and Postman for APIs.
Unit tests with Jest.

## 11. Screenshots or Demo

### Admin Dashboard Overview

**Figure 1: Dashboard Overview**
This screen displays a categorized product listing under sections like Fruits, Vegetables, Meats, Bakery, and Drinks. Users can browse, sort, and add items to the cart directly from this view. It serves as the central control panel for managing products and viewing user interactions.



$2.1
Apples X Lb

$3.88 ~~$4.88~~
Bliquam Feugiat Dui Non

$4.88
Abciquam Feugiat Beans

$1.88
Lorem Am Beans Dui Non

$2.3
Berry Es Feugiat Mollis

## Sorting Products by Priority (Ascending Order)

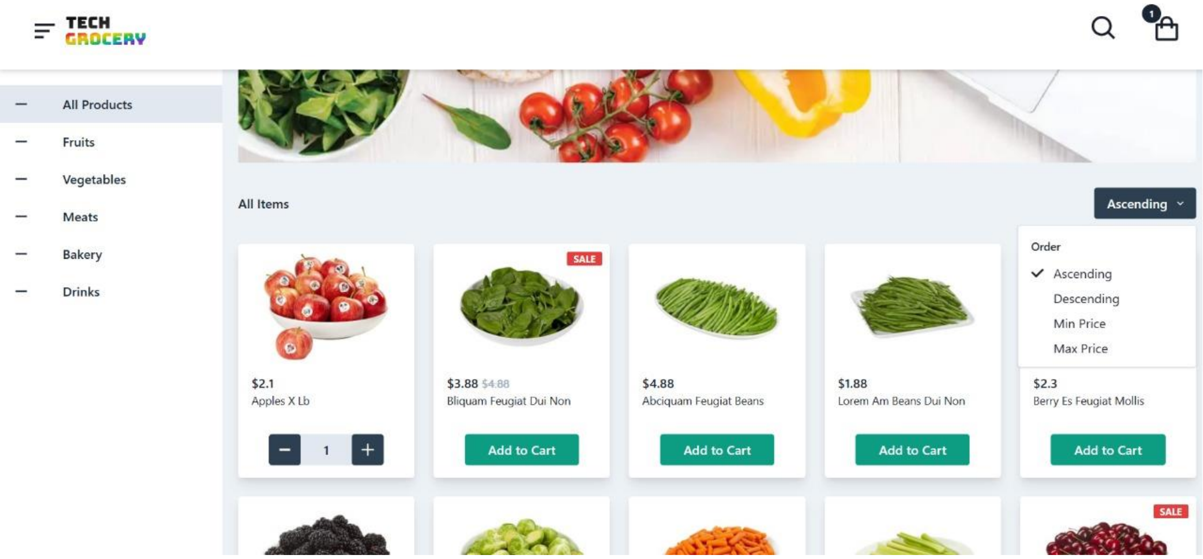
**Figure 2: Ascending Order Product Sorting**

The user interface allows customers to sort products using a dropdown menu, enhancing the shopping experience by enabling personalized product views. As shown in the image, users can sort the items in **Ascending Order**, which arranges products from **lowest to highest price**. This is particularly useful for budget-conscious shoppers who wish to view the most affordable items first.

Other available sorting options include:

- **Descending** – Displays products from highest to lowest price.
- **Min Price** – Highlights products with the minimum price.
- **Max Price** – Highlights the most expensive items.

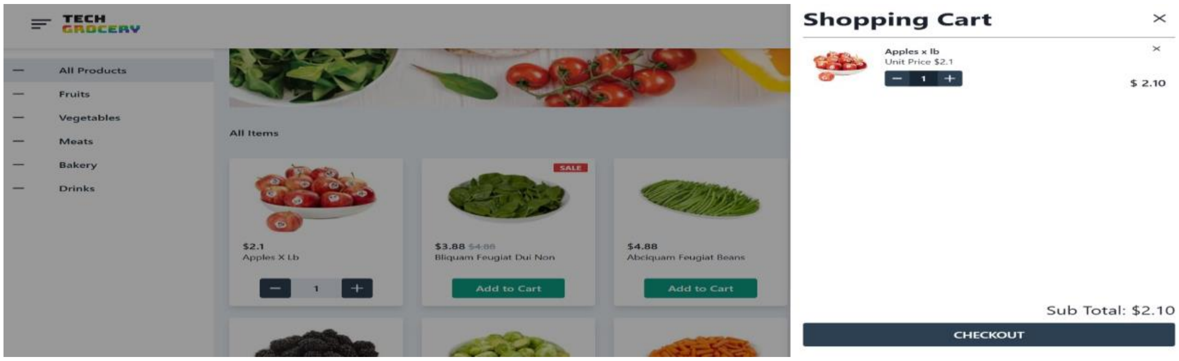
This feature helps users **purchase products based on their price priority**, ensuring a more efficient and tailored shopping experience.



## Shopping Cart View

**Figure 3: Shopping Cart Sidebar**

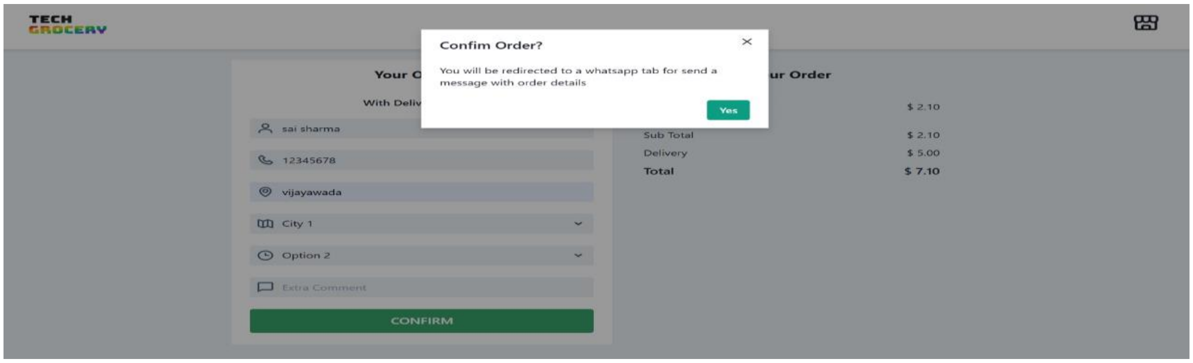
When a user adds items to the cart, a sidebar appears showing selected items with quantity controls, individual pricing, subtotal, and a checkout button.



## Order Confirmation Prompt

**Figure 4: Confirm Order Modal**

Upon clicking checkout, the user is asked to confirm the order. On confirmation, they are redirected to WhatsApp to send the order details.



Confim Order?

You will be redirected to a whatsapp tab for send a message with order details

Yes

sai sharma

12345678

vijayawada

City 1

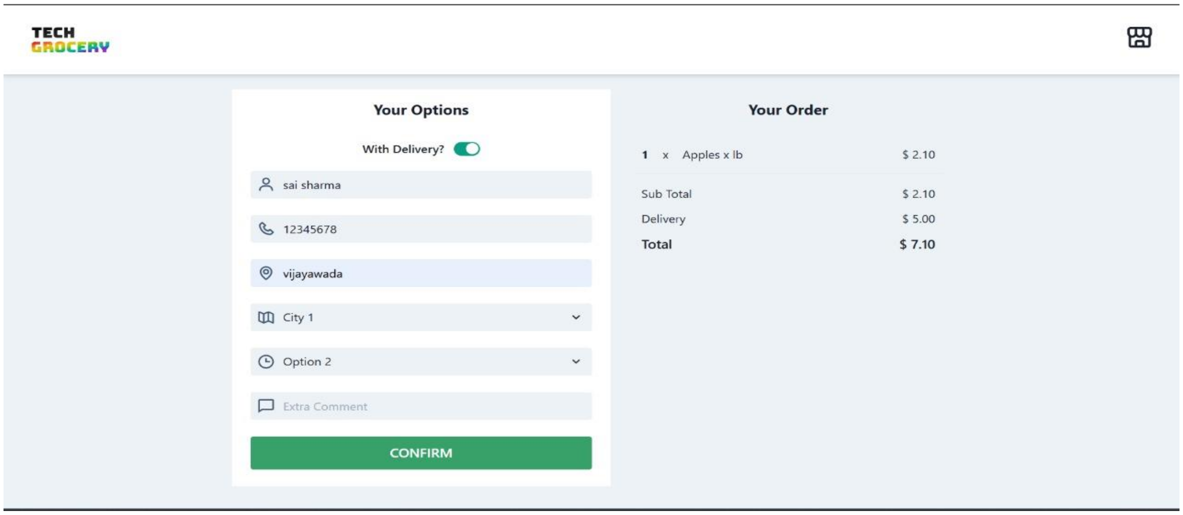
Option 2

Extra Comment

|  |  |
|---|---|
|  | $ 2.10 |
| Sub Total | $ 2.10 |
| Delivery | $ 5.00 |
| Total | $ 7.10 |

## Final Order Summary

**Figure 5: Delivery Details and Order Summary**
This screen collects delivery preferences and shows a final summary of the user's order, including total cost breakdown.



With Delivery? ●

sai sharma

12345678

vijayawada

City 1

Option 2

Extra Comment

|  |  |
|---|---|
| 1 x Apples x lb | $ 2.10 |
| Sub Total | $ 2.10 |
| Delivery | $ 5.00 |
| Total | $ 7.10 |

**GitHub:** https://github.com/BELLAMKONDAHOMASHANKAR/grocory-store

## 12. Known Issues

- Cart does not persist after logout
- No payment integration
- Admin features are basic

## 13. Future Enhancements

- Integrate payment gateway
- Order history and tracking
- Reviews and ratings
- Notifications and email support