

Combee – E-commerce Website (MERN Stack) Project Report :

Project Title: Combee – E-commerce Website

Team Members:

- AVANTIKA DHABUGADE – Backend Developer
- SUYASH DANOLE – Frontend Developer
- SANIKA HAVALE – UI Designer
- RAHUL THIKANE – Frontend Developer

Project Overview

Purpose:

The purpose of Combee is to provide a seamless online shopping experience, enabling users to browse products, add them to a cart, and make secure purchases. The platform caters to both customers and administrators, offering a complete solution for managing products, orders, and payments.

Features:

- User authentication & authorization (JWT-based)
- Product listing with search & category filters
- Add to cart & wishlist functionality
- Secure checkout with payment integration
- Order tracking & history
- Admin dashboard for product & order management
- Responsive design for mobile & desktop
- Real-time updates for cart and stock status

Architecture

Component Structure:

- Header & Navigation – Displays logo, categories, search bar, and user menu
- Home Page – Featured products and promotions
- Product Page – Product details, reviews, and related items
- Cart Page – View, update, and checkout items
- Admin Panel – Manage products, orders, and users
- Footer – Contact info, social links, and policy pages

State Management:

- Redux Toolkit used for global state management (cart, user, and product data).

Routing:

- React Router v6 for client-side navigation (Public & Private Routes).

Setup Instructions**Prerequisites:**

- Node.js (v16 or higher)
- MongoDB (local or cloud)

Installation:

1. Clone the repository:

2. **Install dependencies:**

```
cd client
npm install
cd ../server
npm install
```

3. **Configure environment variables in .env file:**

```
MONGO_URI=your_mongodb_uri
JWT_SECRET=your_secret_key
PAYMENT_API_KEY=your_payment_gateway_key
```

4. **Start development servers:**

```
cd server
npm run dev
cd ../client
npm start
```

Folder Structure**Client:**

- src/components – Reusable UI components (Navbar, ProductCard, etc.)
- src/pages – Page components (Home, Product, Cart, Checkout, Admin, etc.)
- src/redux – Redux slices for state management
- src/assets – Images, icons, and styles

Utilities:

- Helper functions for API calls, formatting prices, and authentication checks

Running the Application

Frontend:

cd client

npm start

Backend:

cd server

npm run dev

Component Documentation

Key Components:

- Navbar.jsx – Navigation with search and user options
- ProductCard.jsx – Displays product info
- Cart.jsx – Cart management
- CheckoutForm.jsx – Payment handling

Reusable Components:

- Button.jsx – Styled button component
- Loader.jsx – Loading spinner
- Modal.jsx – Popup for confirmations

State Management

Global State:

Managed with Redux Toolkit for cart, authentication, and product data.

Local State:

Used in components for form inputs, modals, and UI states.

User Interface

Fully responsive with Bootstrap and custom CSS

Dark & Light theme support

Clean, minimalistic product presentation

Styling

CSS Frameworks/Libraries:

- Bootstrap 5
- Styled Components for dynamic styling

Theming:

- Custom theme colors defined in theme.js

Testing

Testing Strategy:

- Unit testing with Jest
- Integration tests with React Testing Library

Known Issues

- Payment API sandbox mode only
- Product search case sensitivity issue

Future Enhancements

- AI-based product recommendations
- Multi-language support
- Progressive Web App (PWA) features