**Full Stack Development with MERN**

**Project Documentation format**

**1. Introduction**

* **Project Title:** Ecomern - E-commerce Application
* **Team Members:**
* K.Hina Fathima - au412721205016
* K.V.V.Mathimithra - au412721205030
* R.Avanthiga-au412721205004
* V.Merlin Vensiya - au412721205031
* D.Janani - au412721205021

**2. Project Overview**

* **Purpose:** Ecomern aims to make online shopping easy and enjoyable by offering a centralized platform with a comprehensive catalog and personalized experience. Shoppers can effortlessly find, compare, and buy products, while sellers benefit from an efficient order management system and business insights to grow their sales.
* **Features**:
* **Effortless Product Discovery** - Intuitive search and filter options allow users to find products quickly.
* **Personalized Shopping Experience** - Product recommendations based on user history and preferences.
* **Seamless Checkout Process** - Simple and secure checkout with various payment methods.
* **Efficient Order Management for Sellers** - Sellers can manage orders, track inventory, and update listings.
* **Insightful Analytics for Business Growth** - Sellers access analytics on sales trends, popular products, and customer behavior.

**3. Architecture**

**Frontend**

Built using **React** for dynamic and responsive web pages, the frontend includes:

* + **User Authentication**: Secure login and registration for users and admins.
  + **Product Listing and Search**: Browse and filter products by category, price, brand, and rating.
  + **Shopping Cart**: Manages items selected for purchase with quantity adjustments.
  + **Order History and Tracking**: View past orders, track shipment status, and access receipts.
  + **Admin Dashboard**: Tools for sellers to manage products, view orders, and analyze sales data.

**Backend**:

* Developed with **Node.js** and **Express.js**, the backend API includes endpoints for:
  + **User Management**: Handles registration, login, and profile updates.
  + **Product Management**: Allows adding, updating, deleting, and retrieving product details.
  + **Order Management**: Processes orders from creation to shipment.
  + **Authentication & Authorization**: Secure token-based user sessions for both customers and admins.
* **Admin Dashboard**: Provides advanced order and inventory management, with restricted access through authentication.

**Database**:

* **MongoDB** is used for flexible, scalable data storage.
  + **Users**: Stores user information, purchase history, and preferences.
  + **Products**: Stores product details, categories, and availability.
  + **Orders**: Records transaction details, delivery status, and payment confirmation.
  + **Cart**: Temporarily stores user selections for seamless checkout.

**4. Setup Instructions**

* **Prerequisites:** Hardware: Windows 8 machine with two web browsers.
* Network: Minimum 30 Mbps internet speed.
* Software Dependencies: Node.js, MongoDB, React, Express.js, and Postman (for API testing).
* **Installation**:
* **Clone the Repository**: git clone [repository URL]
* **Install Dependencies**: Run npm install in both client and server directories.
* **Environment Variables**:
* Set up .env files in the backend for sensitive data such as database URIs, JWT secrets, etc.
* Example:

MONGO\_URI=mongodb://localhost:27017/ecomern

JWT\_SECRET=your\_jwt\_secret

**5. Folder Structure**

* **Client** (React):
* src/components: Contains UI components like ProductCard, Cart, and UserProfile.
* src/pages: Pages like Home, ProductDetails, Checkout, and AdminDashboard.
* src/utils: Helper functions, constants, and API calls.
* **Server** (Node.js/Express):
* routes: API endpoints for users, products, orders.
* models: MongoDB schemas for User, Product, and Order.
* controllers: Business logic for user authentication, order processing, and product retrieval.
* middleware: Custom middleware for authentication and error handling.

**6. Running the Application**

**Frontend**: Run npm start from the client directory to launch the React app at http://localhost:3000.

**Backend**: Run npm start from the server directory to start the API server at <http://localhost:5000>.

**7. API Documentation**

**User Endpoints**:

* **POST /api/auth/register**: Registers a new user.
* **POST /api/auth/login**: Logs in an existing user and issues a JWT.
* **GET /api/user/profile**: Retrieves the user's profile (requires authentication).

**Product Endpoints**:

* **GET /api/products**: Retrieves a list of all products.
* **GET /api/products/**

: Fetches details of a specific product.

* **POST /api/products**: Adds a new product (admin only).

**Order Endpoints**:

* **POST /api/orders**: Creates a new order.
* **GET /api/orders/**

: Retrieves details of a specific order.

* **PATCH /api/orders/**

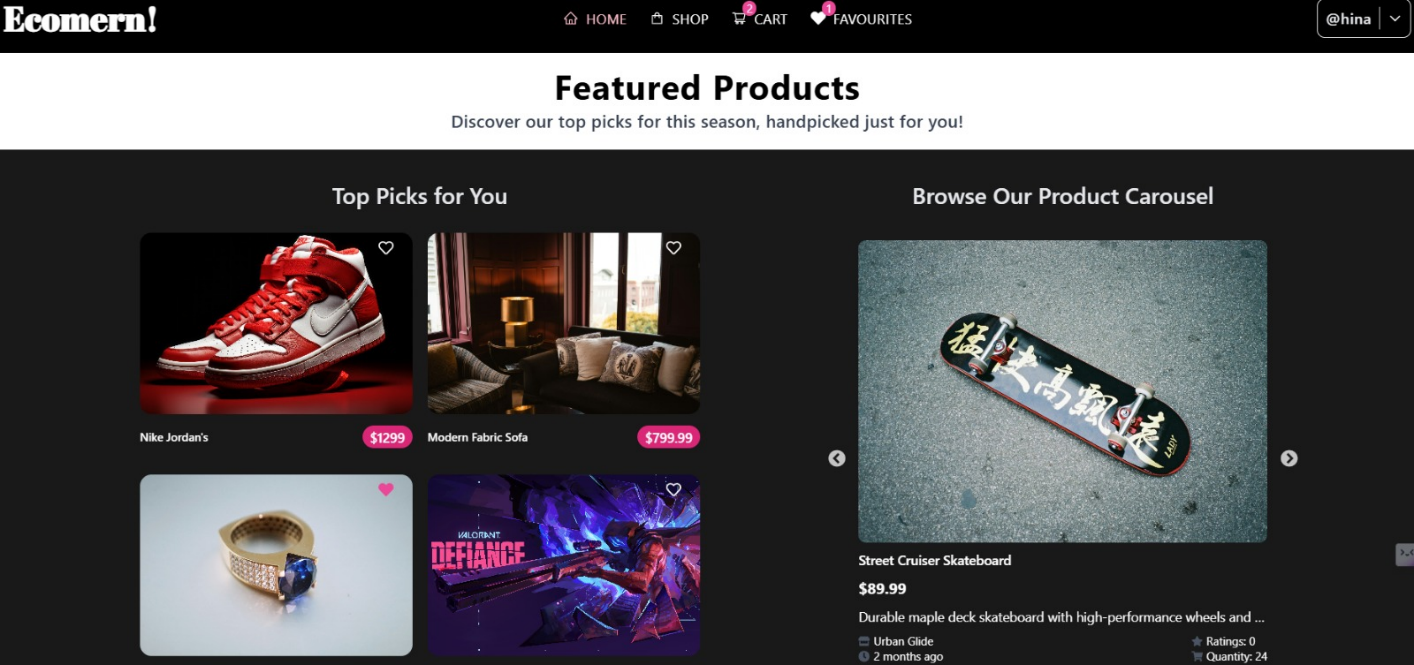
: Updates order status (admin only).

**8. Authentication**

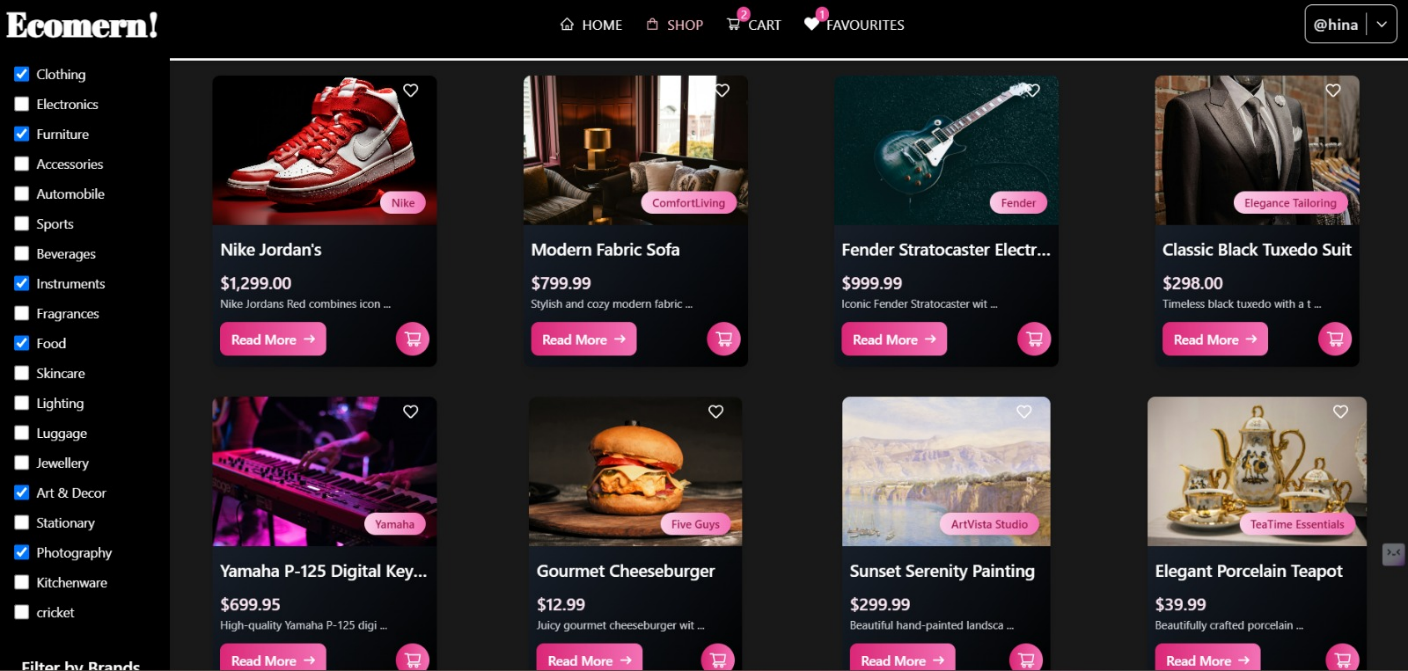
* **Token-Based Authentication**: Secure sessions are created using JWTs, stored in cookies or local storage.
* **Role-Based Access Control**: Admin privileges are required for certain actions (e.g., adding products, managing orders).

**9. User Interface**

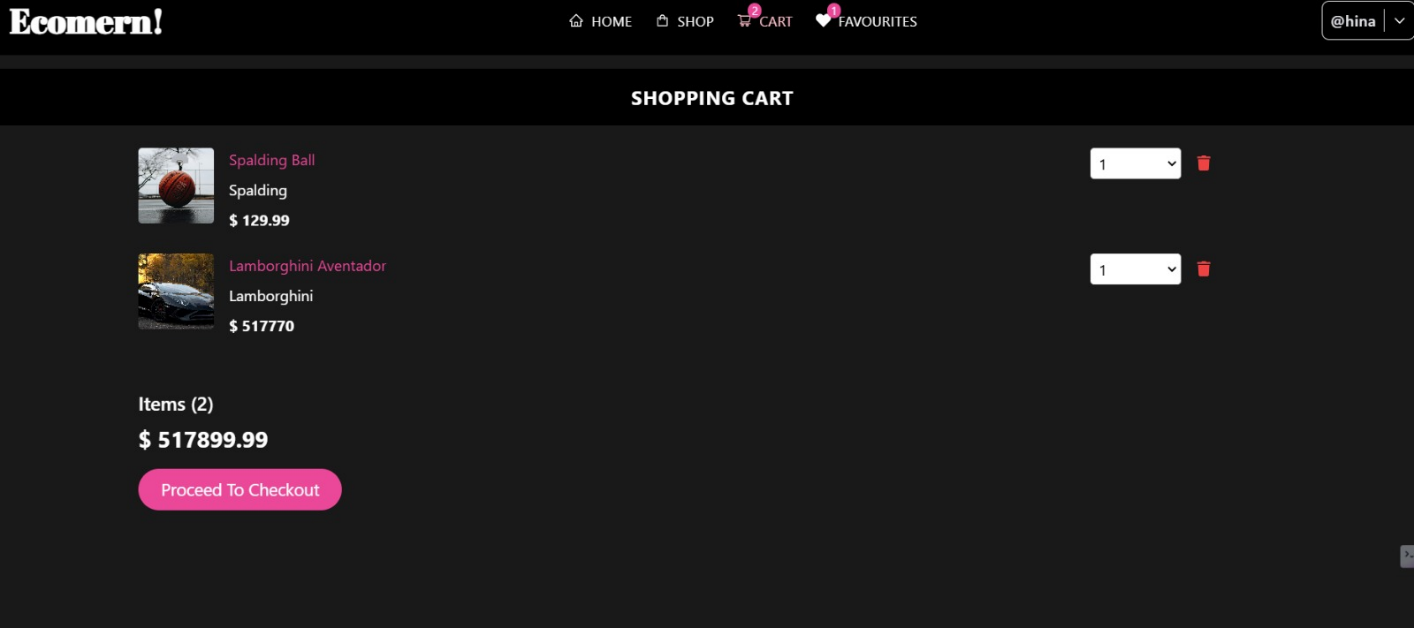
**Homepage**: Featuring categories, search bar, and recommendations.

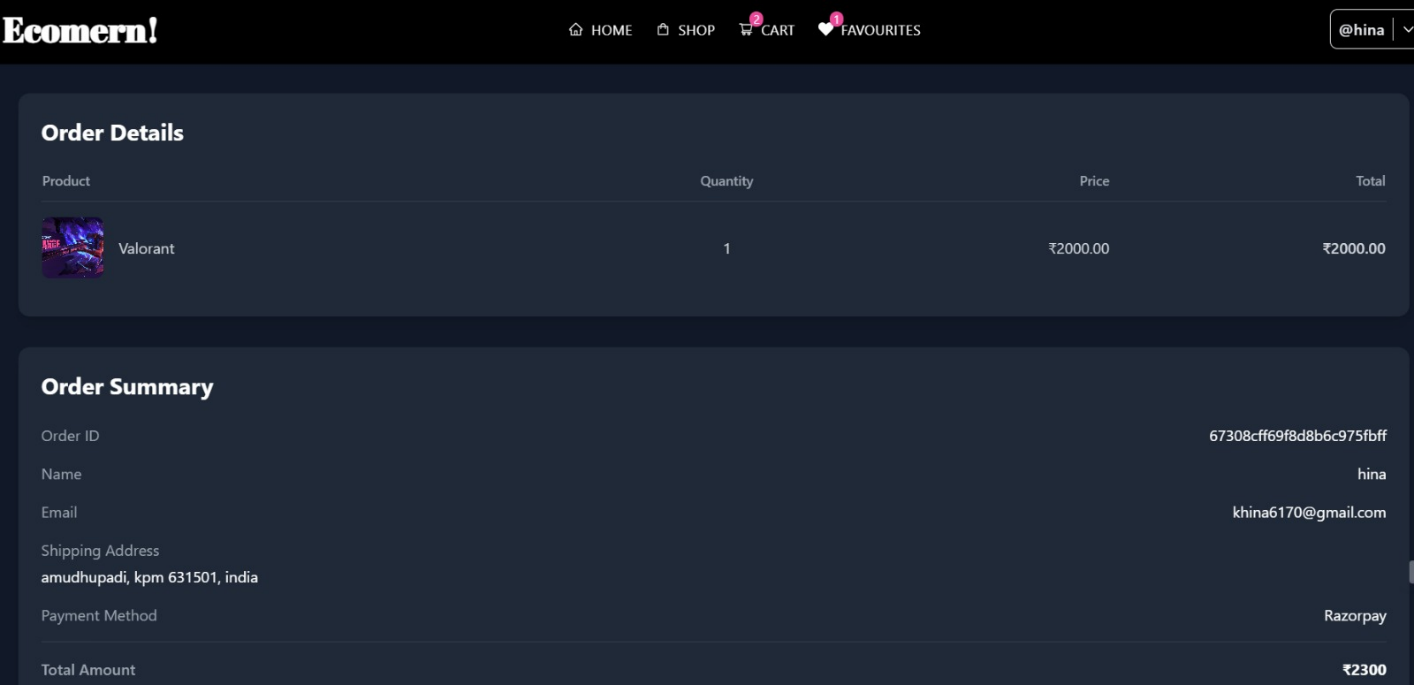


**Product Details Page**: Displaying images, descriptions, and reviews.

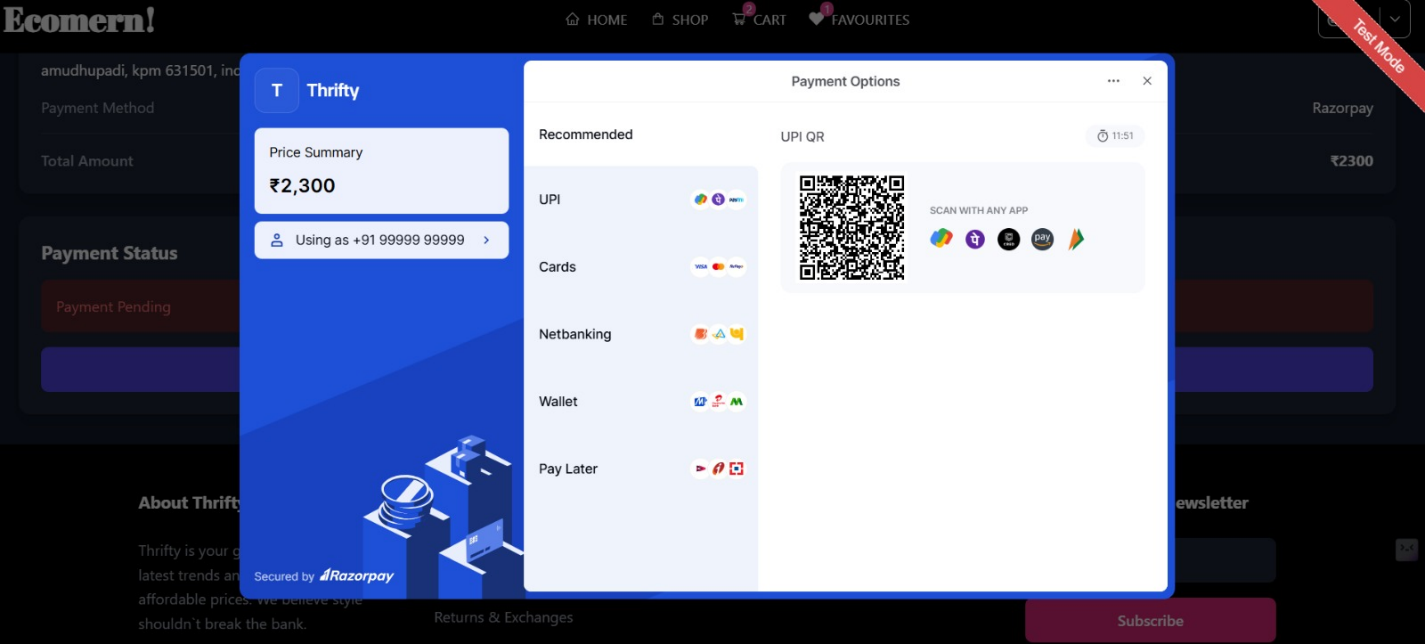


**Cart**: User-selected products with a quick checkout option.

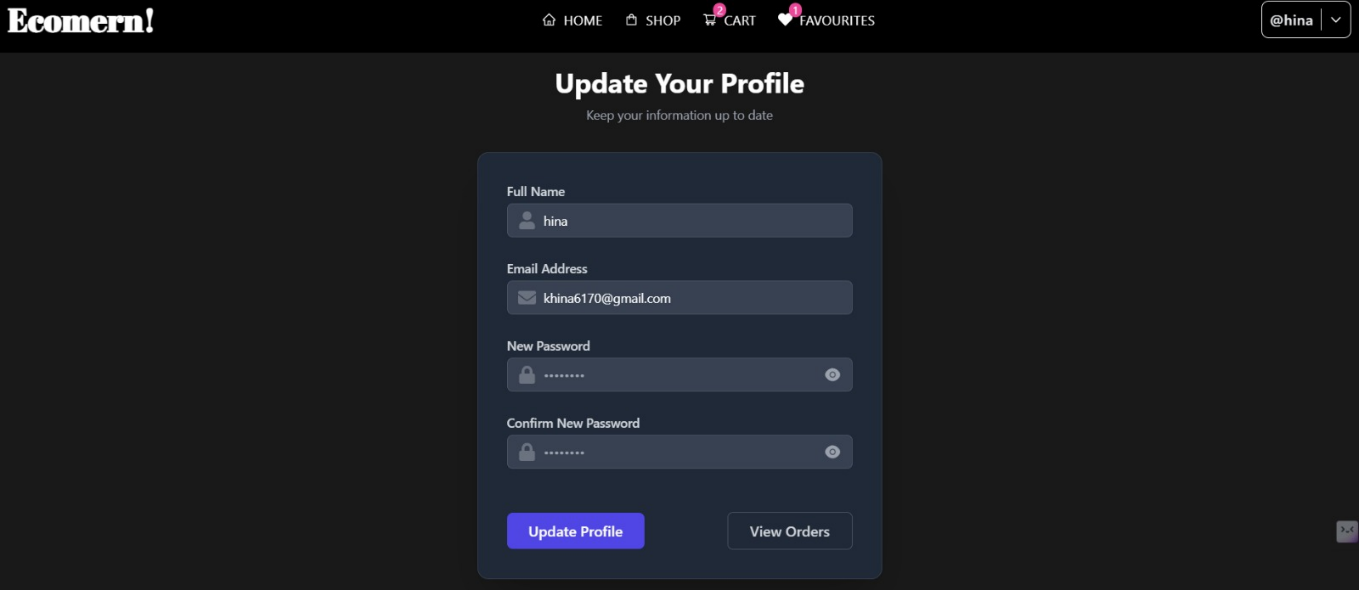




**Payment Gateway:** Order summary and payment page



**Login Page:** User and Admin Login page.



**10. Testing**

* **Unit Testing**: With Jest for component and API function testing.
* **Integration Testing**: Validates interactions between components, ensuring data flows correctly.
* **End-to-End Testing**: Simulates real user scenarios using tools like Selenium or Cypress.

**11. Known Issues**

List of known bugs or areas for improvement:

* **Bug 1**: Occasional delay in product filtering results.
* **Bug 2**: Rare API timeout when the server is under heavy load.

**12. Future Enhancements**

1. **AI-Driven Recommendations**: Implement machine learning to enhance personalized suggestions.
2. **Multi-Language Support**: Broaden accessibility by supporting multiple languages.
3. **Advanced Analytics**: Provide sellers with deeper insights into product performance and user behaviour.