E-Commerce Platform with MERN Stack

Project Title: E-Commerce Platform with MERN Stack

Problem Statement:

Design and develop a comprehensive E-Commerce platform using the MERN (MongoDB, Express.js, React, Node.js) stack, aimed at providing users with a seamless shopping experience. The platform will facilitate user registration, product selection, cart management, order placement, and administration functionalities.

Functional Requirements:

1. User Authentication:

- Users can register for an account with their email and password.
- Registered users can log in to the platform using their credentials.
- Passwords will be securely hashed and stored in the database.

2. Homepage:

- The homepage will present users with a login option and a registration option for new users.

- After logging in, users will be directed to their dashboard.

3. Dashboard:

- The user dashboard will include a navigation bar and an offcanvas for filtering products.
- The navigation bar will offer links to different product categories and user-specific options.
- The offcanvas will allow users to filter products based on gender (mens, womens, kids) and apply filters.

4. Product Listings:

- Products will be categorized into household items, electronic devices, and fashion.
- Users can view product details and add them to their cart.

5. User Management:

- Users can update their profile picture and change their password.
- Users can manage their shipping addresses and add new addresses.
- Users can view their order history.

6. Cart Management:

- Users can add products to their cart.
- Users can view and manage the contents of their cart.

- Users can proceed to checkout from the cart.

7. Order Placement:

- Users can place orders for products in their cart.
- Users will provide shipping address details and select a payment method.
- Orders will be stored in the database with status and payment details.

8. Admin Panel:

- Admin users will have access to a drop-down menu for managing categories, products, and orders.
- Admins can add new categories and subcategories.
- Admins can manage product details and availability.
- Admins can view and manage user orders.

Non-Functional Requirements:

1. Security:

- User passwords will be securely hashed and stored.
- JWT authentication and authorization will be implemented for user access control.

2. Performance:

- The application should load products quickly and efficiently.

- The database queries should be optimized for fast retrieval.

3. User Experience:

   - The user interface should be intuitive and responsive.
   - Filter options should be user-friendly and easy to apply.

4. Scalability:

   - The application should be designed to accommodate future scalability needs.

5. Reliability:

   - Data integrity and reliability will be maintained during transactions.

Technology Stack:

- Frontend: React, React Router, Bootstrap, Redux Toolkit
- Backend: Node.js, Express.js, MongoDB
- User Authentication: JWT (JSON Web Tokens)
- State Management: Redux Toolkit
- UI Components: React Bootstrap
- API Communication: Axios

Database Tables:

- Users
- Addresses
- Categories
- Subcategories
- Products
- Carts
- Orders