E-Commerce Web Application Project Report

1. Introduction

This project is a React-based e-commerce application that simulates an online store. It includes product listings, product detail views, a cart management system, user authentication interfaces, and contact functionality. The backend API used for product data is https://fakestoreapi.com.

Technologies used:

- React.js (with Hooks)

- Redux (for cart management)

- React Router

- Bootstrap for UI styling

2. Home Page

The `Home.jsx` component renders the main homepage using subcomponents: Navbar, Main, Product, and Footer. It serves as the landing page of the application, displaying a featured product section and navigation options.

3. Product Display

The `Product.jsx` component fetches data for a single product from the Fake Store API and displays it with details such as image, title, category, price, rating, and description. It includes an 'Add to Cart' button and a recommendation section for similar products using a Marquee effect.

4. Products Page

The `Products.jsx` component is a simple wrapper that renders the Navbar, Product component, and Footer.

E-Commerce Web Application Project Report

This creates a product listing page or a single entry point to display a product.

5. Cart Page

The `Cart.jsx` component handles the user's shopping cart. It allows users to increase or decrease item quantity, view the cart summary, and proceed to checkout. Redux is used to manage the cart state.

6. User Authentication

The `Register.jsx` and `Login.jsx` components provide a frontend form for user registration and login. While no backend is integrated, forms exist with placeholders and routing logic. Links are provided to switch between login and registration views.

7. Contact Page

The `ContactPage.jsx` provides a contact form with fields for name, email, and message. It is styled using Bootstrap and includes a submit button (disabled by default), along with Navbar and Footer components.

8. Conclusion

This React project simulates an end-to-end e-commerce application with dynamic product fetching, cart functionality, and user interfaces. While some components like login and register are placeholders, the application demonstrates practical use of React hooks, component-based design, routing, and API integration.