E-Commerce Project with MERN Stack

Overview

This project is an E-Commerce website developed using the MERN (MongoDB, Express.js, React, Node.js) stack. It features a robust frontend built with React, a backend powered by Node.js and Express, and data storage using MongoDB. The project includes essential E-Commerce functionalities such as product fetching, product listing, user authentication (login, logout, signup), and shopping cart management (add to cart, checkout).

Technologies Used

- **Frontend**: React, React Router Dom, Tailwind CSS, react-lazy-load-image-component
- **Backend**: Node.js, Express.js
- **Database**: MongoDB

Features

- **Product Fetching**: Users can browse through a list of available products on the website.
- **Product Listing**: Display of products with details such as name, image, price, and description.
- **Add to Cart**: Users can add products to their shopping cart for later checkout.
- **Checkout**: A seamless checkout process for users to review their cart and complete the purchase.

User Authentication:

Login: Existing users can log in with their credentials.
Signup: New users can create an account by providing necessary information.

User Session Management:
Logout: Users can securely log out of their accounts.

Installation

Prerequisites:

- Node.js and npm installed
- MongoDB installed and running

Image Optimization

This project uses the **react-lazy-load-image-component** library for image optimization, ensuring efficient loading of images and enhancing the overall user experience.

Additional Libraries

- **React Router Dom**: For handling navigation and routing in the React application.
- **Tailwind CSS:** A utility-first CSS framework for styling the frontend.

Steps

Install dependencies for both frontend and backend:

```
cd client
npm install

cd server
npm install
```

Configure Environment Variables:

Create a .env file in the backend directory and set the following variables:

```
PORT=8080
MONGODB_URI=xxxxxxxxxxxxxxxxxxxxxxxxx
JWT_SECRET=your-jwt-secret
```

To Run Application

```
Frontend:
        cd client
        npm start
Backend:
        cd server
        npm start
```