

E-commerce Web Application Documentation

Overview

This document serves as documentation for the e-commerce web application developed using Spring Boot for the backend and React for the frontend. The application facilitates online buying and selling of products, providing users with a seamless shopping experience.

Table of Contents

1. Introduction
2. Backend Implementation
3. Frontend Implementation
4. Conclusion

1. Introduction

The e-commerce web application aims to provide users with a platform to browse, purchase, and manage products online. It features a user-friendly interface, secure authentication, product categorization, cart management, checkout functionality, and order tracking.

2. Backend Implementation

Technologies Used:

- Spring Boot
- Spring Data JPA
- Spring Security
- MySQL

Key Components:

Controllers: Define RESTful endpoints for handling HTTP requests.

Services: Implement business logic and interact with repositories.

Repositories: Interface with the database using Spring Data JPA.

Features:

- User authentication and authorization.
- Product management: CRUD operations for products.
- Order management: Creating, updating, and tracking orders.

3. Frontend Implementation

Technologies Used:

- React
- React Router
- Redux (optional, for state management)
- Axios (or any preferred HTTP client for making requests to the backend)

Key Components:

Components: Reusable UI elements for building pages.

Containers: Higher-level components that manage state and pass props to child components.

Features:

- User authentication: Login, registration, and logout.
- Product browsing: Displaying products by category, searching, and filtering.
- Shopping cart management: Adding, updating, and removing items from the cart.
- Checkout process: Address input, payment, and order confirmation.

4. Conclusion

The e-commerce web application provides a robust platform for online shopping, combining the power of Spring Boot for backend development and React for frontend interactivity. Further enhancements can be made to improve performance, add new features, and enhance the user experience.