1. project overview

2. technology stack

3. modules and it's details

4. overall functionality of your application

**Project Overview**

* A basic ecommerce website in which a customer can order products and vendor can host their products on the site.
* Customers can add & remove products to cart, place orders, give reviews, ratings.
* Admin can add new Products and from which Vendor is providing it.

**Technology Stack**

* FRONT END • HTML • CSS • JavaScript • Bootstrap • React • Axios
* BACK END • Spring Boot • Spring Data JPA • Hibernate • REST • JWT Based Security
* DATABASE MANAGEMENT SYSTEM • MySQL
* DevOps : Docker Git

**Modules and it’s details**

a. User Authentication and Authorization Module:

• Responsible for user registration, login, and access control.

• Handles user profiles and security•related functions.

b. Dashboard Module:

• Displays personalized content based on user roles and preferences.

• Includes widgets for quick access to key features.

c. Product Catalog Module:

• Manages the product inventory, categories, and details.

• Allows users to browse and search for products.

d. Shopping Cart and Checkout Module:

• Handles shopping cart functionality.

• Enables users to add, edit, and remove items from the cart and complete the checkout process.

e. Payment Gateway Integration Module:

• Integrates with payment gateways to process transactions securely.

f. Order Management Module:

• Manages customer orders, order history, and status updates.

h. Search and Filtering Module:

• Implements search functionality and advanced filtering options.

k. Settings and Configuration Module:

• Allows administrators to configure site settings, preferences, and customization options.

**4. Overall functionality of your application**

**FUNCTIONAL REQUIREMENTS**

* Customers can browse through all available products .
* Customers can place orders for products and view their order history.
* Admins can manage various product details like inventory, price, adding new
* products and customers, etc.
* Admins can delete customers if the need arises.

**NON FUNCTIONAL REQUIREMENTS**

* Since the application uses lightweight and established software components that are also cross platform, it is remarkably performant and has good support for every operating system.
* The use of React for front end and Spring Boot, Spring Data JPA and Hibernate for back end delivers quick response times to admins and customers alike.
* Card style UI and well known icons and symbols used throughout the application provides a consistent theme and user friendly interface that anyone can grasp easily, even without a technical background