LOW LEVEL DESIGN (LLD)

E-COMMERCE APPLICATION

Name: SAHIL PARAG KINDO

Table of Contents

Abstract

Low-Level Design Documentation of eCommerce MERN App

This document presents a comprehensive low-level design for the eCommerce MERN app, offering detailed insights into the technical components, modules, and interactions that collectively create the functional user interface and backend architecture.

Directory Structure:

The app's directory structure is organized into frontend (client) and backend (server) sections, with separate folders for components, routes, controllers, models, middleware, and static assets. This organization enhances modularity and maintainability.

1. Introduction

This document provides in-depth technical details about the eCommerce MERN app's architecture, modules, components, and interactions.
It serves as a comprehensive guide for developers working on the app's implementation, enhancements, and maintenance.

2. Directory Structure

- client/: Contains the React client-side code.
- server/: Houses the Node.js Express server-side code.
- public/: Holds static assets for the client.
- config/: Stores configuration files and environment variables.
- models/: Defines MongoDB data models.
- routes/: Contains Express route definitions.
- controllers/: Implements route controllers.
- middlewares/: Houses custom middleware functions.

3. Front-End (Client)

Components

- Navbar: Navigation menu with links to home, products, cart, and user profile.
- ProductList: Displays a grid of products with details and a "Add to Cart" button.
- ProductDetail: Shows detailed information about a selected product.
- Cart: Displays items in the user's cart with options to update and proceed to checkout.
- Checkout: Collects user information for shipping and initiates payment processing.
- Redux: Manages global state including cart items, user details, and authentication status.
- React Context: Provides localized state for components that don't require global management.

API Calls

- Axios: Performs HTTP requests to the server API endpoints.
- API Endpoints: Utilizes routes on the server for fetching products, managing the cart, processing orders, and user authentication.

4. Back-End (Server)

Routes and Controllers

- User Routes: /api/auth, /api/users for user registration and authentication.
- Product Routes: /api/products to fetch products and details.
- Cart Routes: /api/cart/add, /api/cart/remove for managing the cart.

Middleware

- Auth Middleware: Validates JWT tokens and authorizes access to protected routes.
- Error Handling Middleware: Centralized error handling for route errors and exceptions.
- Validation Middleware: Validates request data using libraries like express-validator.

Database Schema

- User Model: Contains fields for name, email, password hash, role.
- Product Model: Stores information about products including name, description, price, image URL.
- Cart Model: Tracks user's cart items.

Deployment and Environment Variables

- Deployment Platforms: Can be deployed to platforms like Heroku, AWS, or DigitalOcean.
- Environment Variables: Utilizes environment variables for sensitive information such as database connection strings, API keys, and configuration settings.

Testing

Unit Testing: Uses tools like Jest and React Testing Library for unit testing components and utility functions.
Integration Testing: Tests the interaction between different parts of the app, including API endpoints and database interactions.

7. Conclusion

This low-level documentation provides detailed insights into the eCommerce MERN app's architecture, modules, components, and interactions.
It serves as a reference guide for developers involved in the app's development, maintenance, and enhancements.
Further documentation and code comments are available within the codebase to assist developers during implementation.

Developers are encouraged to follow best practices, maintain code quality, and ensure the security and performance of the app while making enhancements or modifications.