

E-commerce Application

Table of Contents

- Introduction
- Architecture
- Features
- Components
- User Flows
- API Endpoints
- Data Models
- Deployment
- Conclusion

Abstract:

High-Level Documentation of eCommerce MERN App

This document serves as a high-level overview of the architecture, components, and functionality of the eCommerce MERN app. The app is designed to provide users with an intuitive and secure platform for exploring, selecting, and purchasing products online.

Architecture and Technology Stack:

The eCommerce MERN app follows a three-tier architecture, leveraging the power of MongoDB, Express.js, React, and Node.js (MERN stack). This architecture ensures a robust foundation for creating a responsive and dynamic web application.

Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the

current project description to represent a suitable model for coding.

This document is also

intended to help detect contradictions prior to coding, and can be used as a reference manual

for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:

- o Security

- o Reliability

- o Maintainability

- o Portability

- o Reusability

- o Application compatibility

- o Resource utilization

- o Serviceability

1. Introduction

The eCommerce MERN app is a web-based platform built using the MongoDB,

Express.js, React, and Node.js stack. It enables users to interact with a variety of products, a

dd them to their cart, and complete purchases securely. The app follows modern best practices for security, user experience, and performance.

Architecture

The app follows a three-tier architecture:

Front-End (Client): Built using React, the client-side renders the UI and communicates with the server via API calls.

Back-End (Server): Implemented with Node.js and Express.js, the server handles API requests, business logic, and interacts with the database.

Database (MongoDB): Stores user information, product details, orders, and other relevant data.

Features

User Registration and Login

Product Listings with Filters and Search

Product Detail Pages

Shopping Cart and Checkout Process

Components:

The app is divided into reusable components for modularity and maintainability. Notable components include:

- Navbar
- ProductList
- ProductDetail
- Cart
- Checkout

User Flows

- User Registration and Login
- Browse Products
- View Product Details
- Add Products to Cart
- Proceed to Checkout
- Complete Payment

API Endpoints

- I. GET /api/products: Fetches a list of products.
- II. GET /api/products/:id: Fetches product details by ID.
- III. POST /api/cart/add: Adds a product to the cart.
- IV. POST /api/cart/remove: Removes a product from the cart.
- V. POST /api/orders/checkout: Initiates the checkout process.

Data Models

- User: Stores user information including name, email, password hash, and role.
- Product: Represents a product with details like name, description, price, and image URL.
- Cart: Tracks user's cart items.
- Order: Records order details, products, total amount, and user information.

Deployment

The app can be deployed using platforms like Heroku, AWS, or DigitalOcean. The database can be hosted on MongoDB Atlas for scalability and reliability.

Conclusion

The eCommerce MERN app offers a seamless shopping experience for users, enabling them to browse, select, and purchase products securely. The technology stack ensures optimal performance and maintainability, and the app's architecture supports future enhancements and scalability.

This high-level documentation provides an overview of the eCommerce MERN app's structure, features, and components. Detailed documentation for each component, API, and module should be created to guide developers during implementation and maintenance.