Marketplace Technical Foundation - Furniture

Prepared By: Ayesha (Rising Star)

Overview

This document details the system architecture for a modern e-commerce platform focused on affordable and Pinterest-inspired furniture. The platform leverages Next.js 14 with TypeScript and integrates Sanity CMS for dynamic content management. It is designed to provide a seamless shopping experience for customers while equipping admins with efficient management tools.

High-Level System Architecture

Frontend Structure:

- Framework: Next.js 14 with TypeScript for server-side rendering (SSR), dynamic routing, and enhanced performance.

Pages:

- General Pages: Home, About, Products, Product Details (Dynamic), Cart, Admin Panel.

- User Pages: Login, Sign Up, User Portal (Order and Shipment Details).

- Admin Pages: Analytics, Dashboard, Orders, Stock Management, User Management.

Reusable Components:

- UI Components: CardComponent.tsx, HeroSection.tsx, PopularProduct.tsx.

- Product Components: ProductComponent.tsx, ProductCardDetails.tsx.

- Order Components: CheckoutModal.tsx, PaymentForm.tsx, DisplayShipmentDetails.tsx.

- Cart Components: UserCartComponent.tsx, CartItem.tsx.

- Authentication Components: UserLogin.tsx, UserSignup.tsx.

Marketplace Technical Foundation - Furniture

CMS (Sanity)

Sanity Studio manages structured data for:

- Products: Names, prices, images, categories, and inventory levels.

- Users: Authentication data and order history.

- Orders: Purchased items, quantities, and shipping details.

- Shipments: Tracks shipment statuses via the Shippo API.

- Analytics: Tracks sales performance, revenue, and product popularity.

Key Features:

- Data Schemas: Define structures for Products, Orders, Users, Inventory, and Analytics.

- GROQ Queries: Enable real-time data fetching from Sanity.

APIs and Integrations

Mock APIs:

- /api/products/productData: Dynamically fetches product data.

- /api/shipments/:carrier/:trackingNumber: Fetches shipment tracking details from Shippo.

Payment Gateway (Stripe):

- Handles secure payment processing with Stripe Elements.

- Supports dummy transactions for testing.

Shipment Tracking (Shippo API):

- Provides real-time tracking information.

- Integrated into the user?s order history page.

Marketplace Technical Foundation - Furniture

Workflow Overview

User Workflow:

- Browse Products: Explore dynamically fetched furniture categories from Sanity CMS.

- Add to Cart: Users can add items to their cart without logging in. Cart data is stored locally.

- Checkout: Users log in or sign up during checkout if not already authenticated.

- Payment and Shipment: Payments are processed securely via Stripe. Shipment tracking is generated through the Shippo API.

- Order Management: Users can track orders via the User Portal.

Admin Workflow:

- Admin Login: Secured via AdminLogin.tsx with role-based access.

- Analytics: View sales data, revenue, and product performance using Analytics.tsx.

- Inventory Management: Manage stock and product details via Stock.tsx.

- Order Management: Oversee user orders and details with Orders.tsx.

- Navigation: Seamlessly access admin features using SideBar.tsx.

Collaboration Notes

Challenges:

- Designing efficient schemas for Products, Users, and Orders in Sanity.

- Debugging the integration of APIs with the frontend.

- Mapping workflows like cart management, payment, and shipment tracking.

Learning Outcomes:

- Improved skills in Next.js, TypeScript, and CMS integration.

- Hands-on experience in designing scalable APIs.

- Enhanced understanding of e-commerce workflows.

Feedback Adaptations:

- Improved the cart interface based on user feedback.

- Enhanced analytics visuals for better insights.

Technologies and Tools

- Frontend: Next.js 14 with TypeScript.

- CMS: Sanity CMS for content and data management.

- Payment Gateway: Stripe for payment processing.

- Shipment Tracking: Shippo API for live updates.

- Deployment: Vercel for reliable hosting.

Conclusion

This system architecture ensures scalability, efficiency, and a user-friendly experience. By combining reusable components, dynamic rendering, and robust CMS support, the platform is optimized for both users and admins. Continuous feedback integration will further enhance the platform's functionality and usability.