Marketplace Technical Foundation - Comforty

Frontend

Next.js: Used for building a modern, fast, and server-rendered frontend, ensuring smooth user experiences and SEO optimization.

Tailwind CSS: Provides a utility-first CSS framework for rapid UI development, making it easy to create responsive designs.

TypeScript: Adds type safety to the code, enhancing maintainability and reducing errors.

ShadCN: Enhances component styling and ensures UI consistency across the application.

Lucide React: Provides customizable, lightweight icons to improve the visual appeal and usability of the interface.

Frontend Key Points

User-Friendly Interface for Browsing Products

Simple navigation and easy-to-use interface for an engaging shopping experience.

Responsive Design for Mobile and Desktop Users

Ensures the site works seamlessly across all device sizes and screen resolutions.

Essential Pages

Home Page:

Showcases featured products and categories for easy exploration.

About Page:

Describes the marketplace's mission and unique value proposition.

Contact Page:

Includes a form for inquiries (Name, Email, Message) and displays contact info (email, phone, business hours).

Products Page:

Lists all available products in an organized grid format with filtering options.

FAQs Page:

Answers common questions related to products, orders, rentals, and returns.

Cart Page:

Displays selected products, quantities, total price, and options to update or proceed to checkout.

Single Product Page:

Provides detailed product information, including features, specifications, and images.

Features

3D Product Images:

Allows users to interactively rotate and view products from all angles.

Product Dimensions:

Displays detailed size information (Length x Width x Height).

Product Description:

Offers comprehensive details about the product’s materials, design, and features.

Product Reviews:

Displays user feedback with options to filter by star rating.

Product Gallery:

Multiple images showing the product from different perspectives and real-life settings.

Related Product Suggestions:

Recommends similar or complementary products to increase cross-selling opportunities.

Backend

Sanity CMS: Acts as the central database for managing and storing product data, customer information, and order records with real-time updates.

Styled Components: Allows for styling components with CSS-in-JS, making the styling more modular and dynamic.

Portable Text: Used in Sanity CMS to manage rich text content (e.g., product descriptions, blog content).

API Endpoints: Manages the communication between the frontend and backend services, including handling requests for product data, order placement, and user authentication.

Sanity Schema

Products Schema

const productSchema = { product_id: "string", product_name: "string", product_price: "number", rental_price: "number", product_description: "string", product_images: ["string"], stock: "number", product_dimensions: { length: "number", width: "number", height: "number" }, product_category: "string" };

Customers Schema

const customerSchema = { customer_id: "string", customer_name: "string", customer_email: "string", customer_phone: "string", customer_address: { street: "string", city: "string", state: "string", postal_code: "string" } };

Orders Schema

const orderSchema = { order_id: "string", customer_id: "string", products: ["string"], total_price: "number", status: "string", order_date: "Date" };

Rentals Schema

const rentalSchema = { rental_id: "string", customer_id: "string", products: ["string"], start_date: "Date", end_date: "Date", rental_price: "number", status: "string" };

Shipping Schema

const shippingSchema = { shipping_id: "string", order: "string", address: { street: "string", city: "string", state: "string", postal_code: "string" }, tracking_number: "string", status: "string", estimated_delivery: "Date" };

Payments Schema

const paymentSchema = { invoice_id: "string", order_id: "string", rental_id: "string", customer_id: "string", total: "number", payment_date: "Date", payment_method: "string" };

Third-Party APIs

Shippo: Integrates shipping APIs to handle real-time shipment tracking and logistics, providing users with accurate shipping status updates.

Payment Gateways:

Third-Party Payment Gateways: Includes services like **Stripe** and **PayPal** for securely processing payments through credit cards and other methods.

OR

Digital Wallets: Integrates **Google Pay**, **Apple Pay**, and **Amazon Pay** to allow users to complete transactions using their digital wallet accounts.