BUILD DYNAMIC FRONTEND COMPONENTS FOR YOU MARKETPLACE

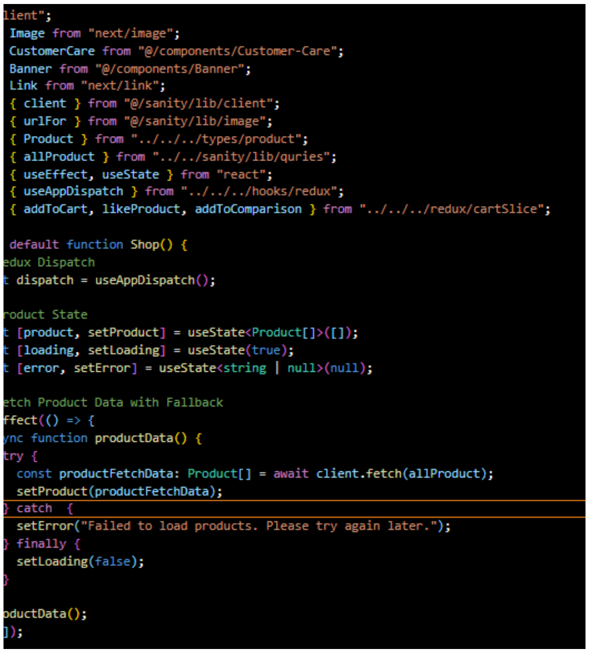
Test cases executed with results:

1. Product Listing Components
2. Product Detail Components
3. Cart Components
4. Wishlist Components
5. Checkout Work Flow Components
6. Product Comparison Components

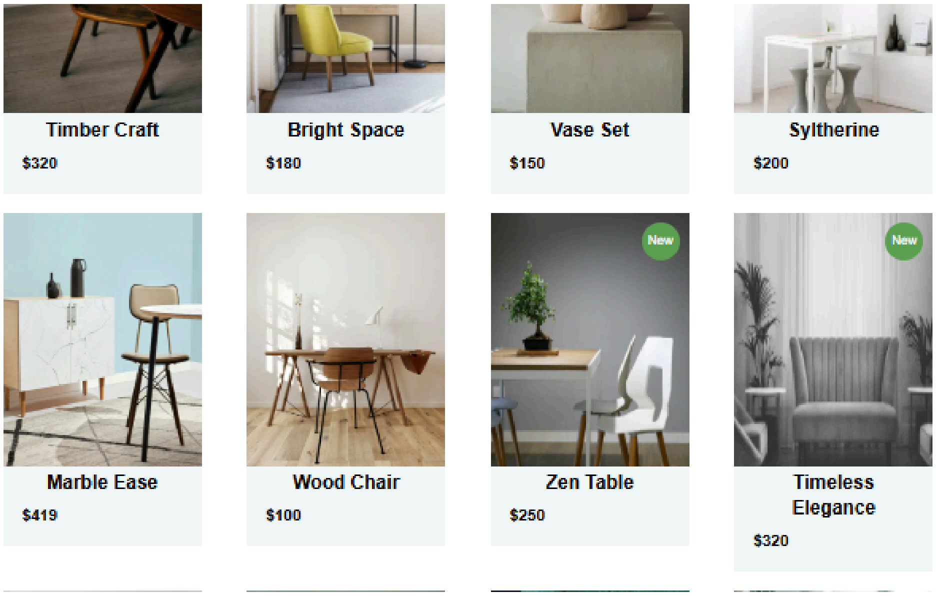
Product Listing Components:

Product listing components are crucial for displaying items on an e-commerce site, as they enhance the shopping experience and encourage conversions. Key elements include high-quality product images that showcase the items, clear and concise product titles, and prominently displayed prices, often accompanied by discounts or original prices. Customer ratings and reviews play a significant role in building trust, and providing potential buyers with insights into the product's quality. Action buttons, such as "Add to Cart" or "View Details," facilitate user interactions, making it easy for customers to engage with the products they are interested in. Together, these components create a visually appealing and functional layout that drives sales.

Output



```
lient";
 Image from "next/image";
 CustomerCare from "@/components/Customer-Care";
 Banner from "@/components/Banner";
 Link from "next/link";
 { client } from "@/sanity/lib/client";
 { urlFor } from "@/sanity/lib/image";
 { Product } from "../../../types/product";
 { allProduct } from "../../sanity/lib/quries";
 { useEffect, useState } from "react";
 { useAppDispatch } from "../../../hooks/redux";
 { addToCart, likeProduct, addToComparison } from "../../../redux/cartSlice";

 default function Shop() {
edux Dispatch
t dispatch = useAppDispatch();

roduct State
t [product, setProduct] = useState<Product[]>([]);
t [loading, setLoading] = useState(true);
t [error, setError] = useState<string | null>(null);

etch Product Data with Fallback
ffect(() => {
ync function productData() {
try {
  const productFetchData: Product[] = await client.fetch(allProduct);
  setProduct(productFetchData);
} catch  {
  setError("Failed to load products. Please try again later.");
} finally {
  setLoading(false);
}

oductData();
]);
```



Timber Craft
$320

Bright Space
$180

Vase Set
$150

Syltherine
$200

Marble Ease
$419

Wood Chair
$100

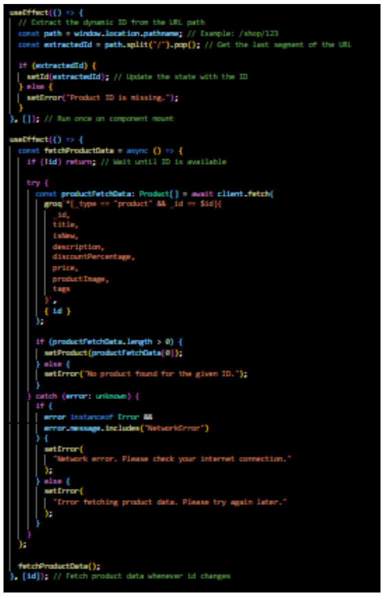
Zen Table
$250

Timeless Elegance
$320

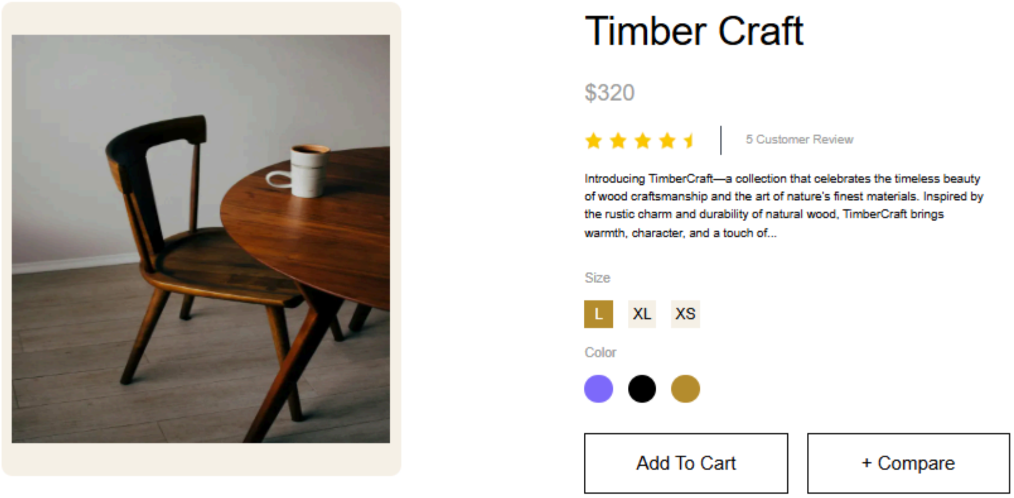
Product Detail Components:

Product detail components are essential for providing customers with comprehensive information about a specific item on an e-commerce site. These components typically include a high-resolution product image that allows users to see the item from various angles, accompanied by a detailed product description that outlines features, specifications, and benefits. Pricing information is prominently displayed, often highlighting any discounts or promotions. Additionally, customer reviews and ratings offer valuable insights into the product's quality and performance, helping to build trust. Key action buttons, such as "Add to Cart" and "Wishlist," enable seamless user interaction, ensuring that customers can easily make purchasing decisions. Together, these components create an engaging and informative product page that enhances the overall shopping experience.

Output:



```
useEffect(() => {
  // Extract the dynamic ID from the URL path
  const path = window.location.pathname; // Example: /shop/123
  const extractedId = path.split("/").pop(); // Get the last segment of the URL

  if (extractedId) {
    setId(extractedId); // Update the state with the ID
  } else {
    setError("Product ID is missing.");
  }
}, []); // Run once on component mount

useEffect(() => {
  const fetchProductData = async () => {
    if (!id) return; // Wait until ID is available

    try {
      const productFetchData: Product[] = await client.fetch(
        groq`*[_type == "product" && _id == $id]{
          _id,
          title,
          isNew,
          description,
          discountPercentage,
          price,
          productImage,
          tags
        }`,
        { id }
      );

      if (productFetchData.length > 0) {
        setProduct(productFetchData[0]);
      } else {
        setError("No product found for the given ID.");
      }
    } catch (error: unknown) {
      if (
        error instanceof Error &&
        error.message.includes("NetworkError")
      ) {
        setError(
          "Network error. Please check your internet connection."
        );
      } else {
        setError(
          "Error fetching product data. Please try again later."
        );
      }
    }
  };

  fetchProductData();
}, [id]); // Fetch product data whenever id changes
```



Timber Craft

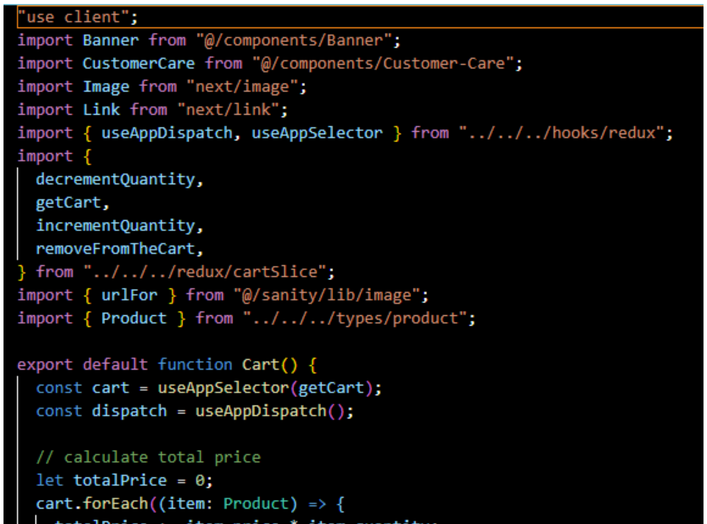
$320

Introducing TimberCraft—a collection that celebrates the timeless beauty of wood craftsmanship and the art of nature's finest materials. Inspired by the rustic charm and durability of natural wood, TimberCraft brings warmth, character, and a touch of...

Cart Components:

Cart components are vital for facilitating the online shopping experience, providing users with a clear overview of their selected items before purchase. These components typically include a list of products added to the cart, complete with images, titles, quantities, and individual prices, allowing customers to review their selections. Users can easily adjust quantities or remove items as needed, ensuring they have control over their purchases. The cart also displays the total cost, including any applicable taxes and shipping fees, helping customers understand the final price. Additionally, action buttons such as "Proceed to Checkout" and "Continue Shopping" guide users through the purchasing process, creating a seamless transition from browsing to buying. Together, these elements enhance usability and encourage customers to complete their transactions.

Output

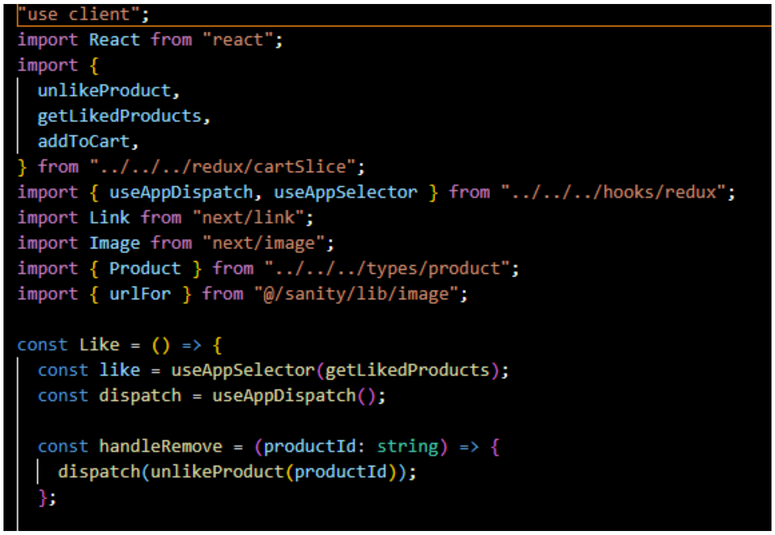


```
"use client";
import Banner from "@/components/Banner";
import CustomerCare from "@/components/Customer-Care";
import Image from "next/image";
import Link from "next/link";
import { useAppDispatch, useAppSelector } from "../../../hooks/redux";
import {
  decrementQuantity,
  getCart,
  incrementQuantity,
  removeFromTheCart,
} from "../../../redux/cartSlice";
import { urlFor } from "@/sanity/lib/image";
import { Product } from "../../../types/product";

export default function Cart() {
  const cart = useAppSelector(getCart);
  const dispatch = useAppDispatch();

  // calculate total price
  let totalPrice = 0;
  cart.forEach((item: Product) => {
```

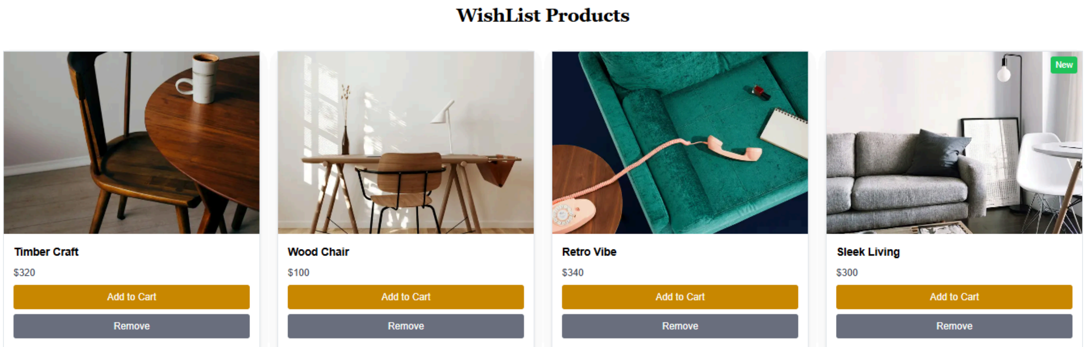
Wishlist Components:

These components typically include a visually appealing list of products, complete with images, titles, and prices, enabling customers to easily browse their saved items. Users can add or remove products from their wishlist with a simple click, providing flexibility as their preferences change. Additionally, wishlist components often feature options to share the wishlist with friends or family, making it easy to communicate gift ideas. Some platforms may even notify users about price drops or restocks for items in their wishlist, enhancing engagement and encouraging future purchases. Overall, these components create a personalized shopping experience that fosters customer loyalty and convenience.

Output



```
"use client";
import React from "react";
import {
  unlikeProduct,
  getLikedProducts,
  addToCart,
} from "../../../redux/cartSlice";
import { useAppDispatch, useAppSelector } from "../../../hooks/redux";
import Link from "next/link";
import Image from "next/image";
import { Product } from "../../../types/product";
import { urlFor } from "@/sanity/lib/image";

const Like = () => {
  const like = useAppSelector(getLikedProducts);
  const dispatch = useAppDispatch();

  const handleRemove = (productId: string) => {
    dispatch(unlikeProduct(productId));
  };
```



WishList Products

Timber Craft
$320

Wood Chair
$100

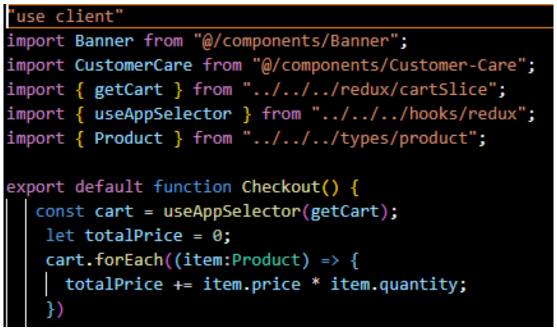
Retro Vibe
$340

Sleek Living
$300

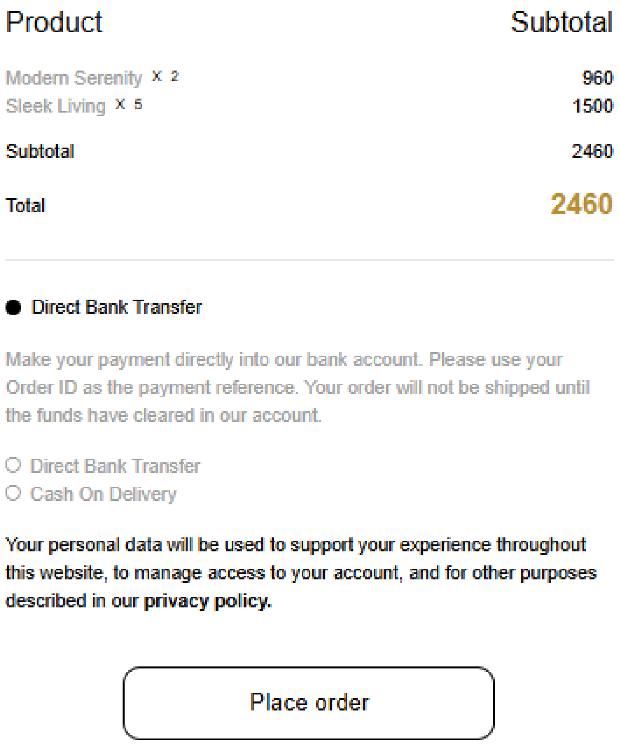
Checkout Work Flow Components:

This workflow typically begins with a summary of the items in the cart, allowing users to review their selections and make any last-minute adjustments. Next, customers are prompted to enter their shipping information, including their address and preferred shipping method. Payment options are presented, offering various methods such as credit cards, digital wallets, and PayPal, to accommodate different user preferences. Additionally, the checkout process often includes a review step, where customers can confirm their order details before finalizing the transaction. Clear navigation and progress indicators throughout the workflow help reduce friction, ensuring that customers feel confident and informed as they complete their purchases. Overall, these components play a crucial role in enhancing user satisfaction and minimizing cart abandonment.

Output



```
"use client"
import Banner from "@/components/Banner";
import CustomerCare from "@/components/Customer-Care";
import { getCart } from "../../../redux/cartSlice";
import { useAppSelector } from "../../../hooks/redux";
import { Product } from "../../../types/product";

export default function Checkout() {
   const cart = useAppSelector(getCart);
    let totalPrice = 0;
    cart.forEach((item:Product) => {
      totalPrice += item.price * item.quantity;
    })
```



Product	Subtotal
Modern Serenity X 2	960
Sleek Living X 5	1500
Subtotal	2460
Total	2460

● Direct Bank Transfer

Make your payment directly into our bank account. Please use your Order ID as the payment reference. Your order will not be shipped until the funds have cleared in our account.

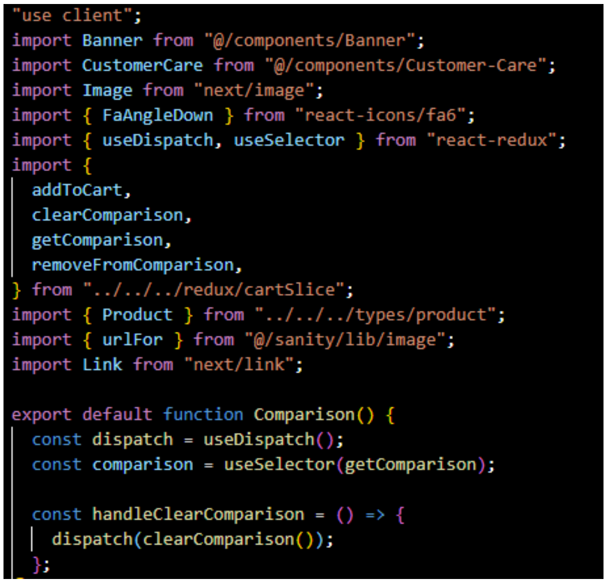
○ Direct Bank Transfer
○ Cash On Delivery

Your personal data will be used to support your experience throughout this website, to manage access to your account, and for other purposes described in our **privacy policy.**

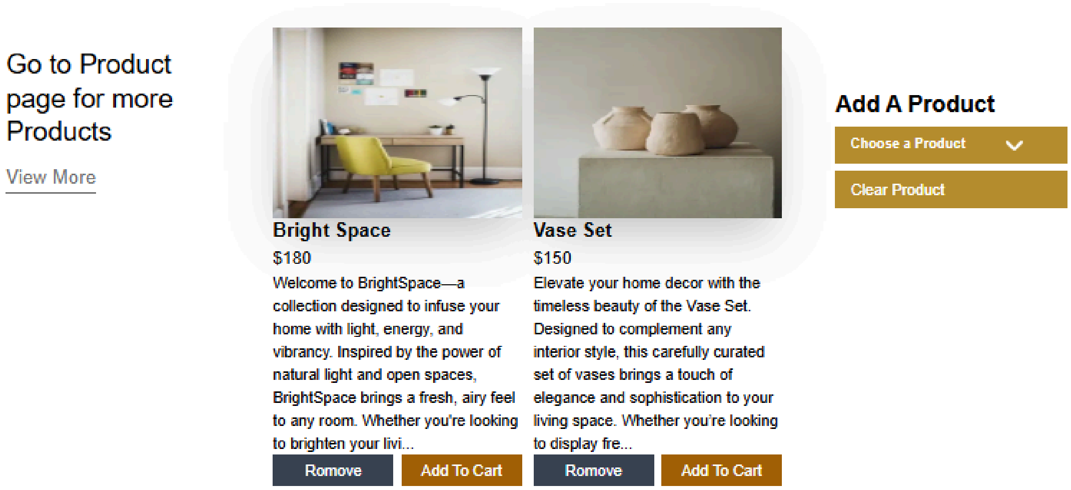
Product Comparison Components:

These components typically feature a grid or table format that displays key product attributes, such as price, specifications, features, and customer ratings, allowing users to easily identify the differences and similarities between products. Users can select which items to compare, ensuring that the information presented is relevant to their decision-making process. This functionality not only enhances the shopping experience by empowering customers with the information they need but also helps reduce decision fatigue by simplifying the evaluation of options. Overall, product comparison components facilitate informed purchasing decisions and can lead to higher customer satisfaction.

Output



```
"use client";
import Banner from "@/components/Banner";
import CustomerCare from "@/components/Customer-Care";
import Image from "next/image";
import { FaAngleDown } from "react-icons/fa6";
import { useDispatch, useSelector } from "react-redux";
import {
  addToCart,
  clearComparison,
  getComparison,
  removeFromComparison,
} from "../../../redux/cartSlice";
import { Product } from "../../../types/product";
import { urlFor } from "@/sanity/lib/image";
import Link from "next/link";

export default function Comparison() {
  const dispatch = useDispatch();
  const comparison = useSelector(getComparison);

  const handleClearComparison = () => {
    dispatch(clearComparison());
  };
```



Go to Product page for more Products

View More

Bright Space

$180

Welcome to BrightSpace—a collection designed to infuse your home with light, energy, and vibrancy. Inspired by the power of natural light and open spaces, BrightSpace brings a fresh, airy feel to any room. Whether you're looking to brighten your livi...

Vase Set

$150

Elevate your home decor with the timeless beauty of the Vase Set. Designed to complement any interior style, this carefully curated set of vases brings a touch of elegance and sophistication to your living space. Whether you're looking to display fre...

Add A Product