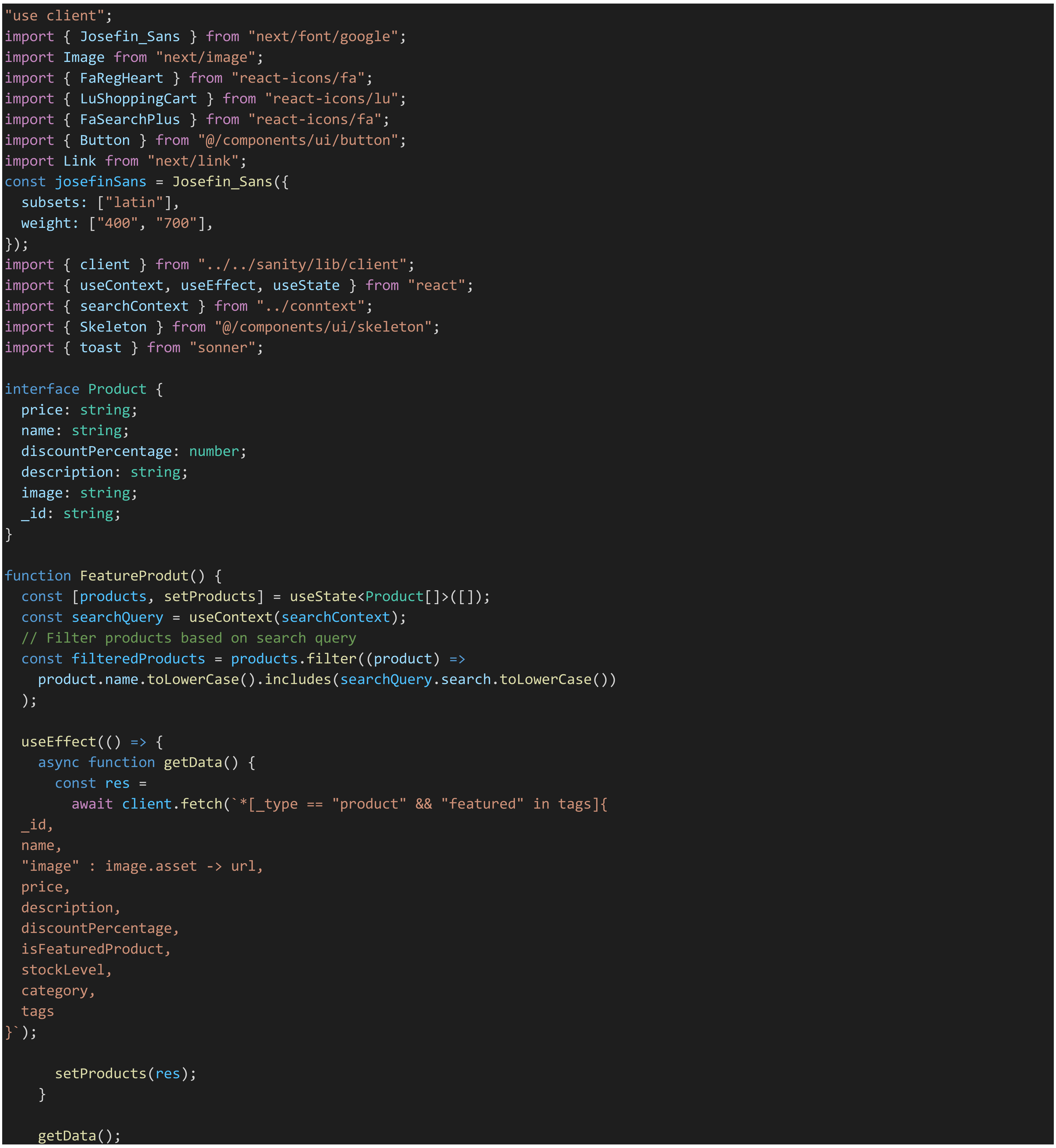
DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR General E-Commerce

Today, I worked on creating multiple components for my website, including **Featured Products**, **Latest Products**, **Trending Products**, **Top Categories**, and other components to manage the frontend UI and logic. I fetched data from **Sanity** based on conditions applied to components or routes and displayed it accordingly.

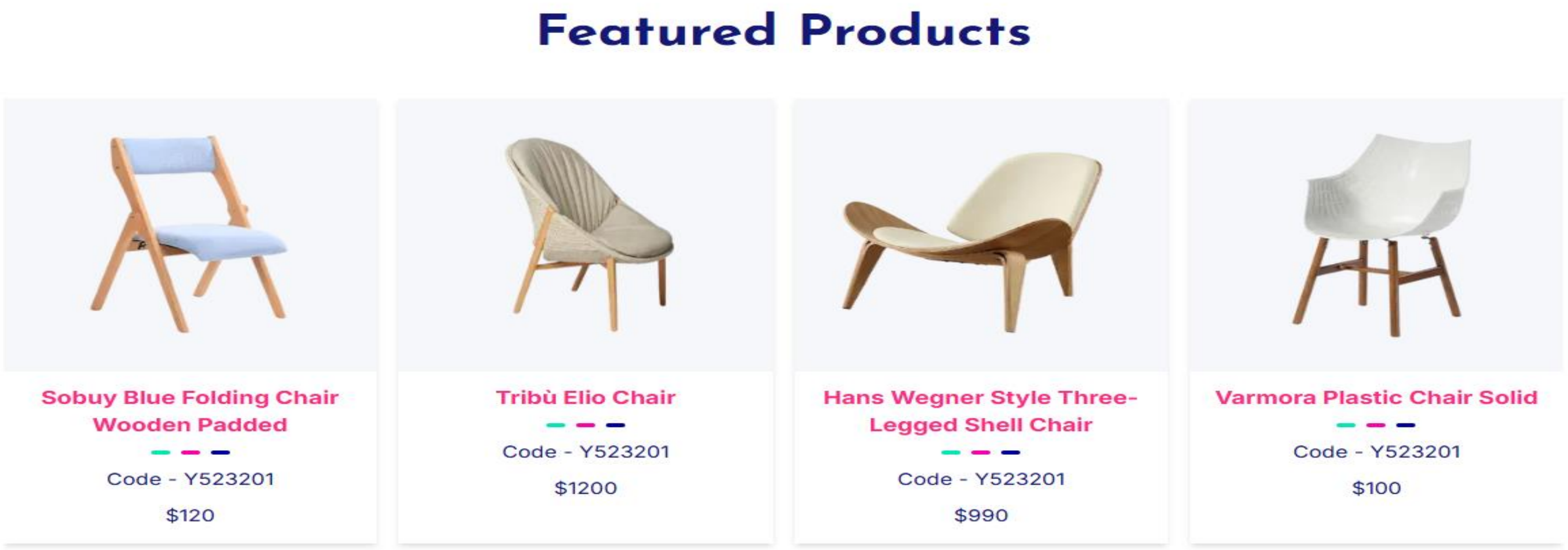
I implemented all the required logic, such as handling the **cart**, **wishlist**, and making the website fully **responsive**. Additionally, I integrated tracking using **ShipEngine** to enhance the user experience.

- ➢ https://ui-ux-hackathon-sable.vercel.app/

✓ Featured Products



```
"use client";
import { Josefin_Sans } from "next/font/google";
import Image from "next/image";
import { FaRegHeart } from "react-icons/fa";
import { LuShoppingCart } from "react-icons/lu";
import { FaSearchPlus } from "react-icons/fa";
import { Button } from "@/components/ui/button";
import Link from "next/link";
const josefinSans = Josefin_Sans({
  subsets: ["latin"],
  weight: ["400", "700"],
});
import { client } from "../../sanity/lib/client";
import { useContext, useEffect, useState } from "react";
import { searchContext } from "../conntext";
import { Skeleton } from "@/components/ui/skeleton";
import { toast } from "sonner";

interface Product {
  price: string;
  name: string;
  discountPercentage: number;
  description: string;
  image: string;
  _id: string;
}

function FeatureProdut() {
  const [products, setProducts] = useState<Product[]>([]);
  const searchQuery = useContext(searchContext);
  // Filter products based on search query
  const filteredProducts = products.filter((product) =>
    product.name.toLowerCase().includes(searchQuery.search.toLowerCase())
  );

  useEffect(() => {
    async function getData() {
      const res =
        await client.fetch(`*[_type == "product" && "featured" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);

      setProducts(res);
    }

    getData();
  }, [filteredProducts]);

  // Custom Button Component for the action
  const ViewCartButton = () => (
    <Link href={"/wishlist"}>
      {" "}
      <button className="bg-[#FB2E86] text-white   w-[90px] py-2 px-4 rounded ">
        WishList
      </button>
    </Link>
  );

  const WishList = (Wishlist: Product) => {
    if (Wishlist) {
      const wishListItems = localStorage.getItem("wishlist");

      const wishlist: Product[] = wishListItems
        ? JSON.parse(wishListItems)
        : [];

      wishlist.push(Wishlist);
      localStorage.setItem("wishlist", JSON.stringify(wishlist));
      toast("⍰ Item successfully added to your Wishlist!", {
        description:
          "You can continue shopping or proceed to view your wishlist by clicking below.",
        action: <ViewCartButton />,
        duration: 4000,
      });
    }
  };

  return (
    <main className="mx-4 lg:mx-0">
      {filteredProducts.length !== 0 ? (
        <>
          <h1
            className={`sm:text-[42px]  text-[27px] mt-16 mb-8 font-bold text-[#151875] ${josefinSans.className} text-
center`}
          >
            Featured Products{" "}
          </h1>
          <section className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-2 lg:grid-cols-4  justify-center gap-4  ">
            {filteredProducts.map((product: any) => {
              return (
                <div
                  key={product.id}
                  className="h-auto  w-full group  shadow-md "
                >
                  <div className="h-[270px]  w-[100%]   lg:mx-0  relative  flex  gap-5 flex-col justify-center items-center
bg-[#F6F7FB]">
                    <Image
                      src={product.image}
                      width={216}
                      height={151}
                      alt="product-image"
                      className="h-[216px] w-[151]"
                    />
                    <div className="group-hover:flex items-center gap-4 absolute hidden top-2 left-2">
                      <Link href={`/${product._id}`}>
                        {" "}
                        <LuShoppingCart className="text-[#00009D]  text-[20px] " />
                      </Link>
                      <FaRegHeart
                        className="text-[#1DB4E7]  "
                        onClick={() => WishList(product)}
                      />
                      <FaSearchPlus className="text-[#1DB4E7]" />
                    </div>
                    <Button className="h-[29px] w-[94px] hidden  hover:bg-[#08D15F]  bg-[#08D15F] p-1 text-[12px] text-white
group-hover:flex justify-center items-center">
                      <Link href={`/${product._id}`}>View Details</Link>
                    </Button>
                  </div>
                  <div className="flex  w-[100%] flex-col justify-center items-center gap-3 mt-3    group-hover:bg-[#2F1AC4]
">
                    <h3 className="text-[#FB2E86] group-hover:text-white text-[18px] font-bold text-center">
                      {product.name}
                    </h3>
                    <div className="flex gap-2">
                      <div className="h-[4px] w-[14px] bg-[#05E6B7] rounded-md"></div>
                      <div className="h-[4px] w-[14px] bg-[#F701A8] rounded-md"></div>
                      <div className="h-[4px] w-[14px] bg-[#00009D] rounded-md"></div>
                    </div>
                    <p className="text-[#151875]  group-hover:text-white">
                      Code - Y523201
                      {/* {`  Code - ${product.code}`} */}
                    </p>

                    <p className="text-[#151875] group-hover:text-white pb-4">
                      ${product.price}
                    </p>
                  </div>
                </div>
              );
            })}
          </section>

          <div className="flex gap-2 my-[80px] justify-center items-center">
            <div className="h-[4px] w-[20px] bg-[#FB2E86] rounded-md"></div>
            <div className="h-[4px] w-[14px] bg-[#FEBAD7] rounded-md"></div>
            <div className="h-[4px] w-[14px] bg-[#FEBAD7] rounded-md"></div>
            <div className="h-[4px] w-[14px] bg-[#FEBAD7] rounded-md"></div>
          </div>
        </>
      ) : (
        <>
          <Skeleton className="h-4 w-[250px]  mt-16 mb-8 bg-gray-400 " />

          <section className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-2 lg:grid-cols-4  justify-center gap-4  ">
            {Array.from({ length: 4 }).map((product: any, index) => {
              return (
                <div key={index} className="h-auto  w-full  shadow-md  pb-3">
                  <div className="h-[270px]  w-[100%]   lg:mx-0  relative  flex  gap-5 flex-col justify-center items-center
bg-gray-300">
                    <Skeleton className="h-[216px] w-[151] bg-gray-400" />
                  </div>
                  <div className="flex  w-[100%] flex-col justify-center items-center gap-3 mt-3    ">
                    <Skeleton className="h-4 w-[250px]  bg-gray-400" />

                    <div className="flex gap-2">
                      <Skeleton className="h-4 w-[250px]  bg-gray-400" />
                    </div>

                    <Skeleton className="h-4 w-[250px]  bg-gray-400" />

                    <Skeleton className="h-4 w-[250px] bg-gray-400" />
                  </div>
                </div>
              );
            })}
          </section>

          <div className="flex gap-2 my-[80px] justify-center items-center">
            <Skeleton className="h-[4px] w-[20px]  rounded-md bg-gray-400 " />
            <Skeleton className="h-[4px] w-[20px]  rounded-md  bg-gray-400" />
            <Skeleton className="h-[4px] w-[20px]  rounded-md bg-gray-400" />
            <Skeleton className="h-[4px] w-[20px]  rounded-md bg-gray-400" />
          </div>
        </>
      )}
    </main>
  );
}

export default FeatureProdut;
```



Featured Products

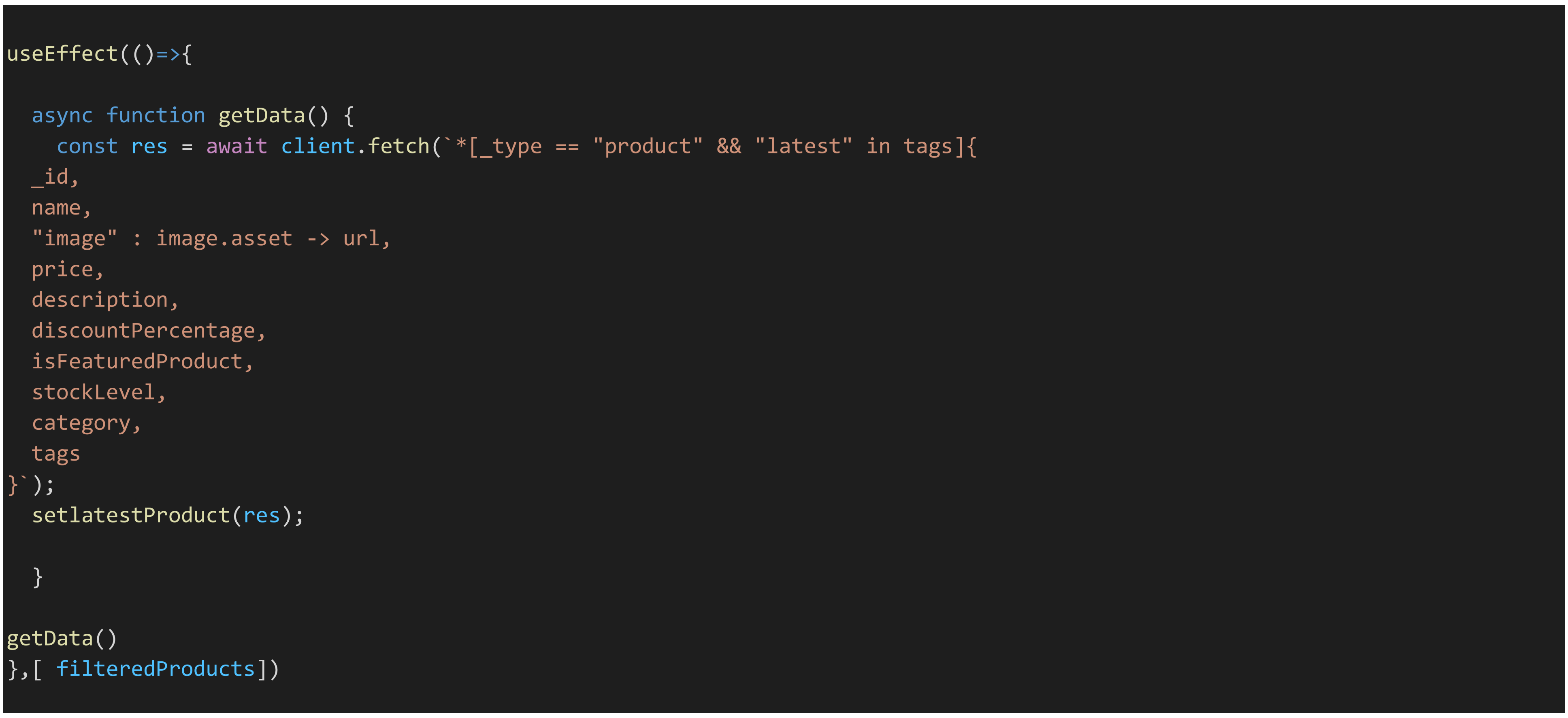
Sobuy Blue Folding Chair Wooden Padded
Code - Y523201
$120

Tribù Elio Chair
Code - Y523201
$1200

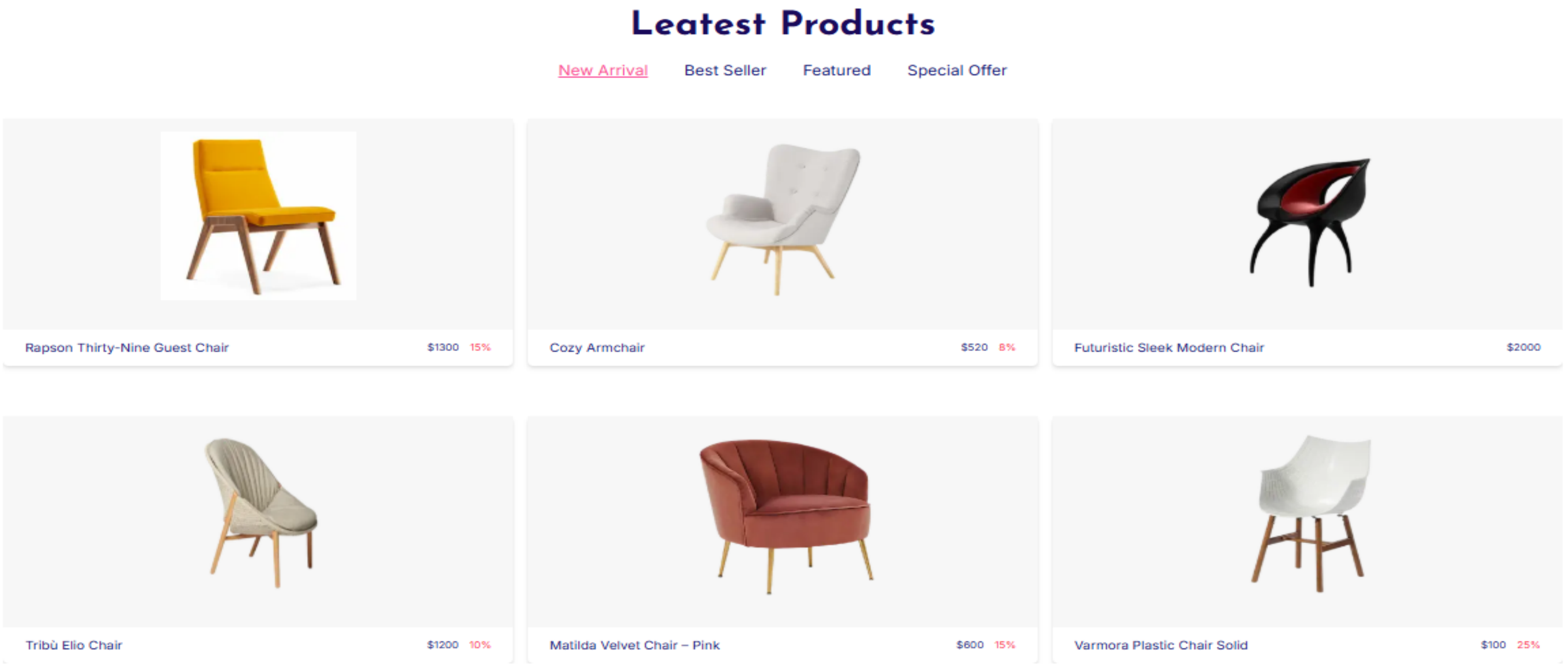
Hans Wegner Style Three-Legged Shell Chair
Code - Y523201
$990

Varmora Plastic Chair Solid
Code - Y523201
$100

- ✓ Latest Products : Just like I fetched "Featured Products" and listed them on the frontend, I have also fetched "Latest Products" , " Trending Product ", and listed them on the frontend.



```
useEffect(()=>{

  async function getData() {
    const res = await client.fetch(`*[_type == "product" && "latest" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);
  setlatestProduct(res);

  }

getData()
},[ filteredProducts])
```



Leatest Products

Rapson Thirty-Nine Guest Chair $1300 15%

Cozy Armchair $520 8%

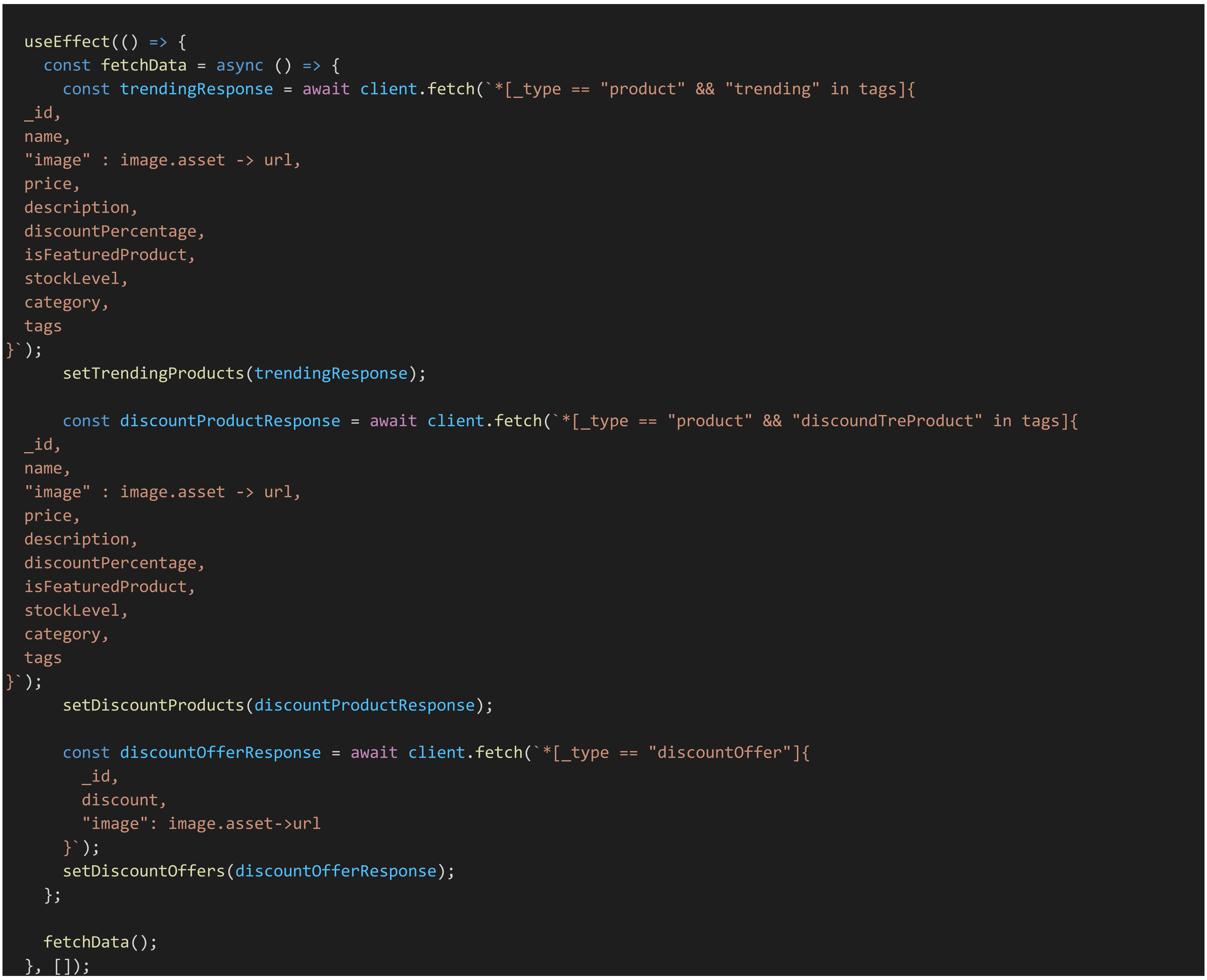
Futuristic Sleek Modern Chair $2000

Tribù Elio Chair $1200 10%

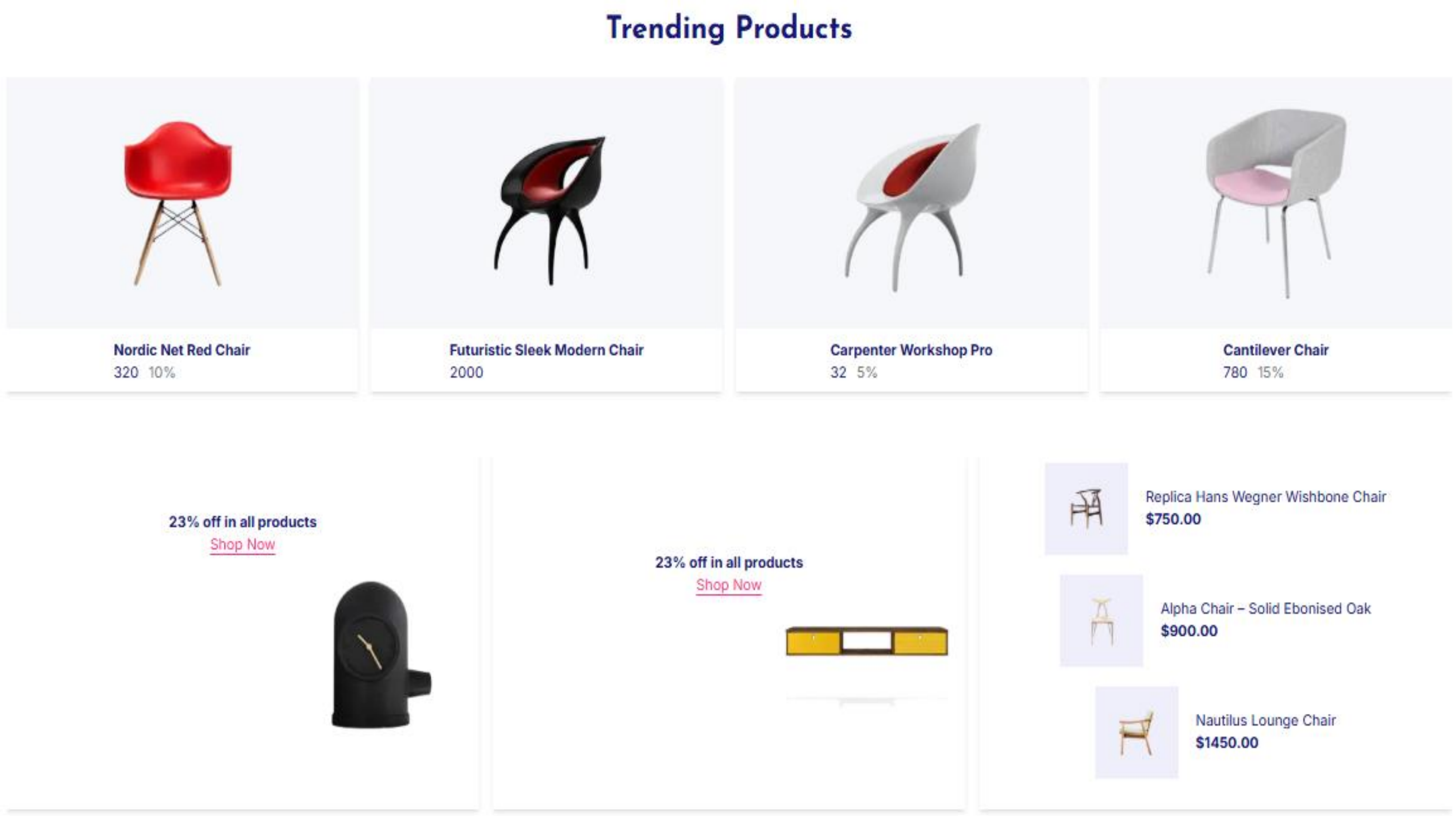
Matilda Velvet Chair – Pink $600 15%

Varmora Plastic Chair Solid $100 25%

✓ Trending Products



```
  useEffect(() => {
    const fetchData = async () => {
      const trendingResponse = await client.fetch(`*[_type == "product" && "trending" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);
      setTrendingProducts(trendingResponse);

      const discountProductResponse = await client.fetch(`*[_type == "product" && "discoundTreProduct" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);
      setDiscountProducts(discountProductResponse);

      const discountOfferResponse = await client.fetch(`*[_type == "discountOffer"]{
        _id,
        discount,
        "image": image.asset->url
      }`);
      setDiscountOffers(discountOfferResponse);
    };

    fetchData();
  }, []);
```



Trending Products

Nordic Net Red Chair
320 10%

Futuristic Sleek Modern Chair
2000

Carpenter Workshop Pro
32 5%

Cantilever Chair
780 15%

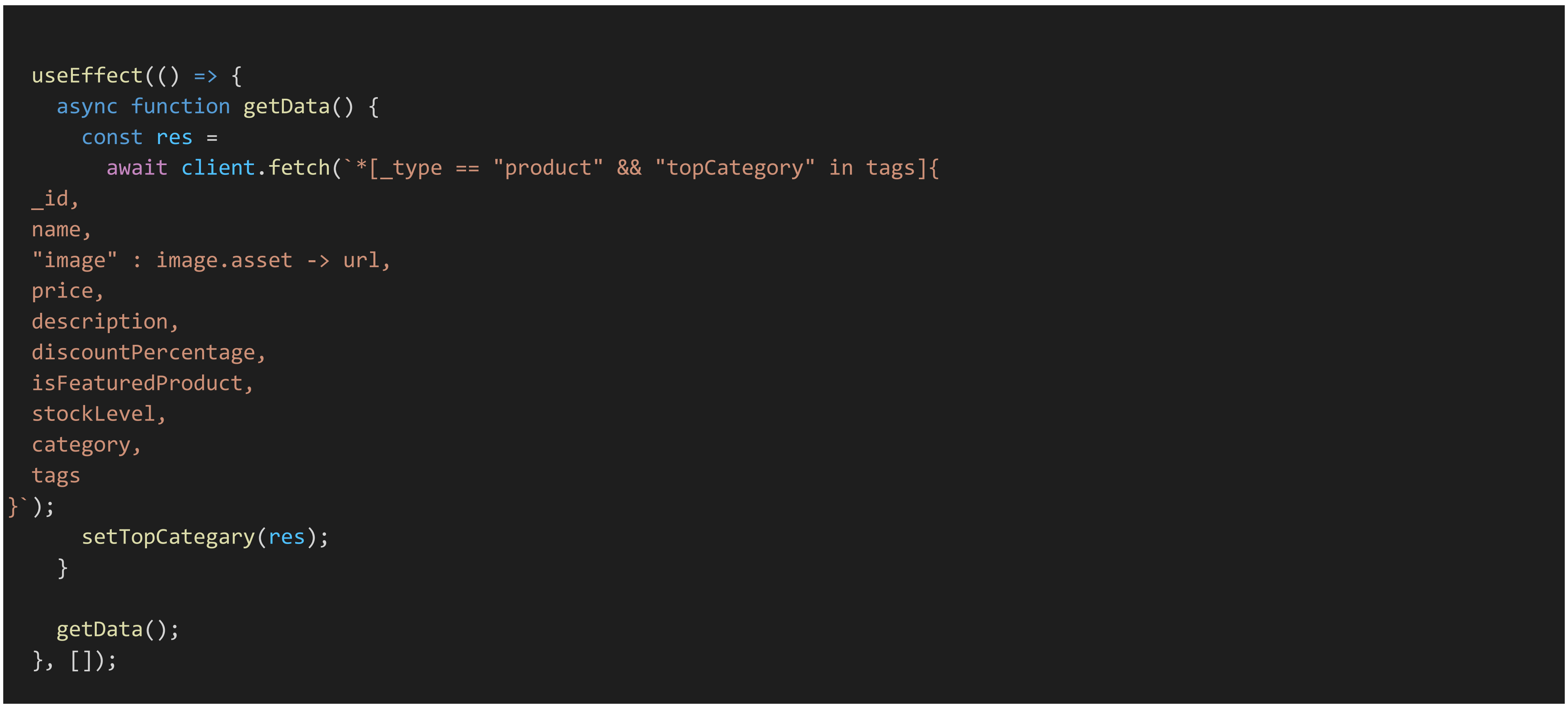
23% off in all products
Shop Now

23% off in all products
Shop Now

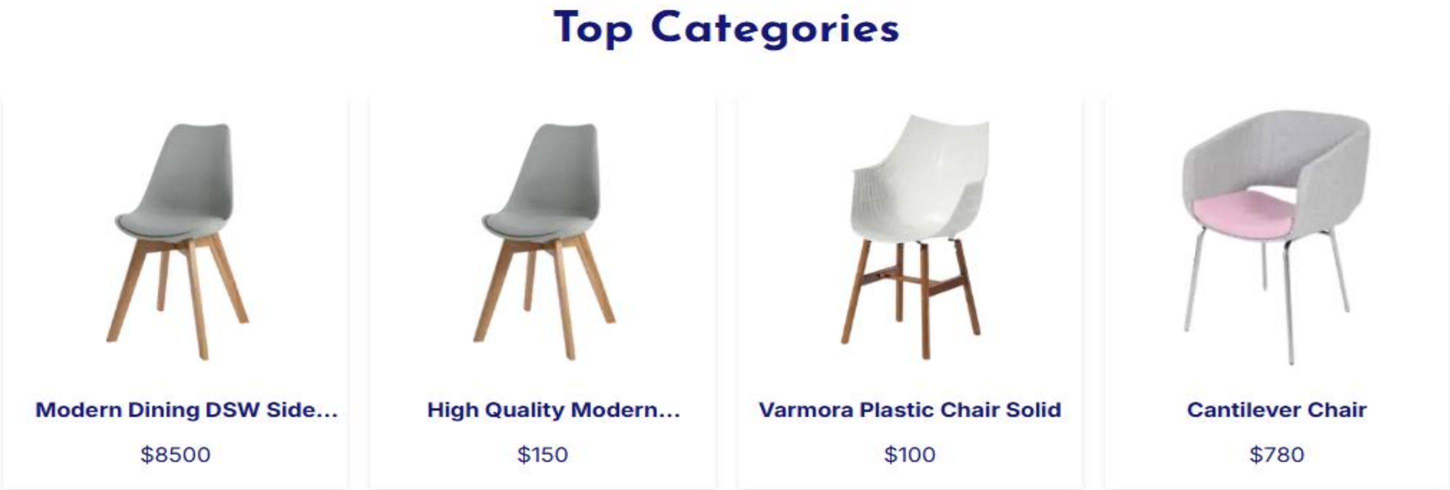
Replica Hans Wegner Wishbone Chair
$750.00

Alpha Chair – Solid Ebonised Oak
$900.00

Nautilus Lounge Chair
$1450.00



```
  useEffect(() => {
    async function getData() {
      const res =
        await client.fetch(`*[_type == "product" && "topCategory" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);
      setTopCategary(res);
    }

    getData();
  }, []);
```



Top Categories

Modern Dining DSW Side...
$8500

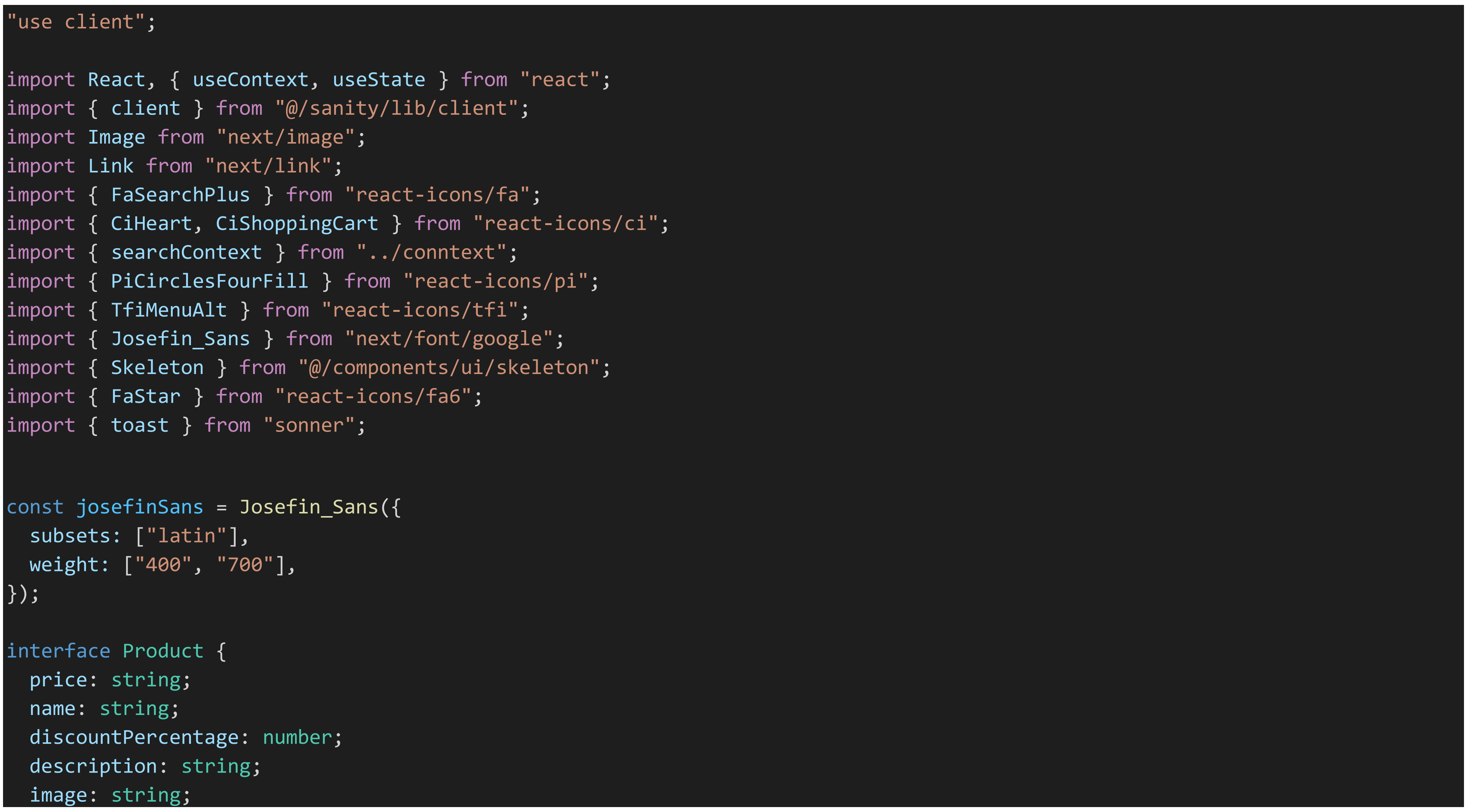
High Quality Modern...
$150

Varmora Plastic Chair Solid
$100

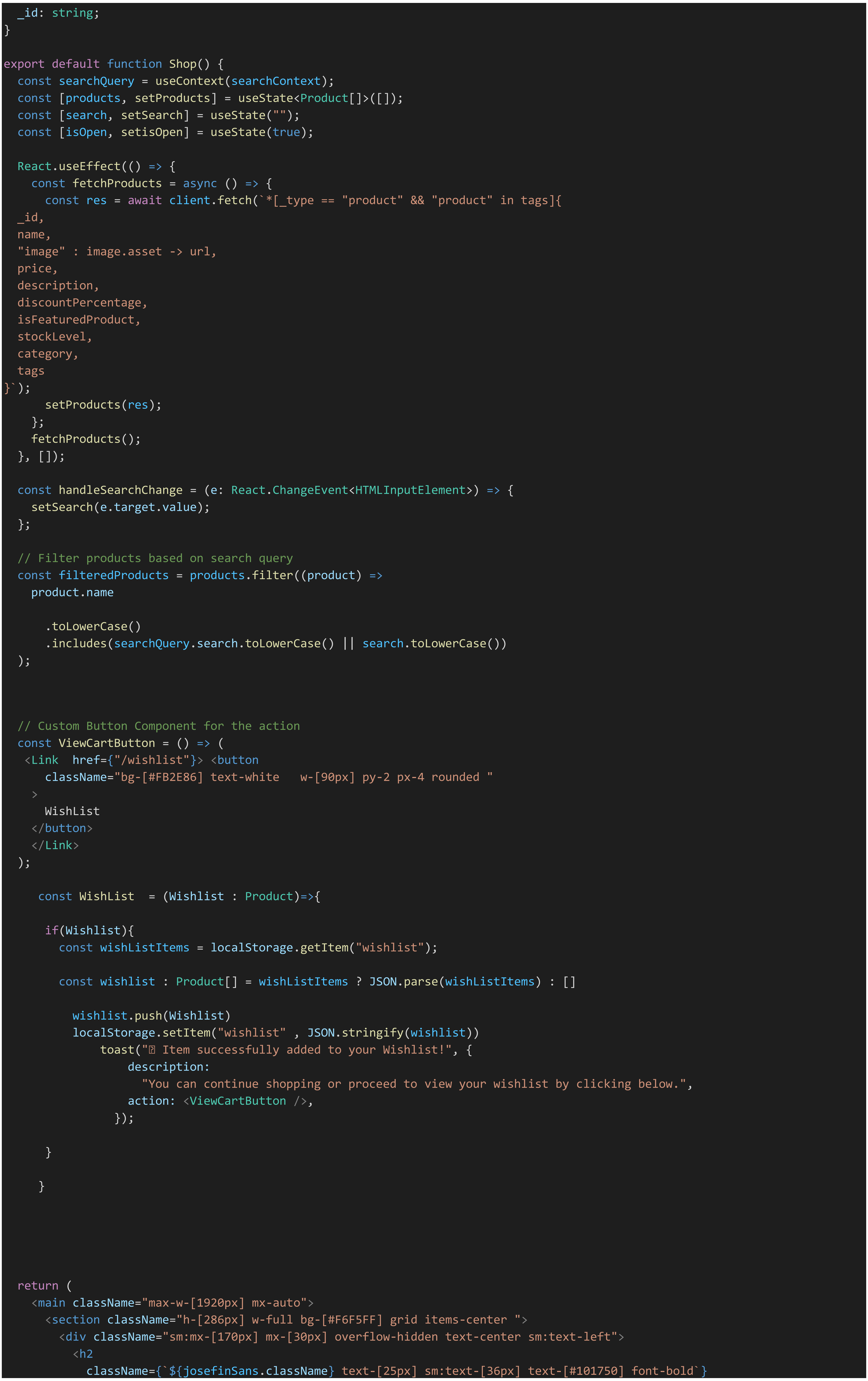
Cantilever Chair
$780

✓ Product Route

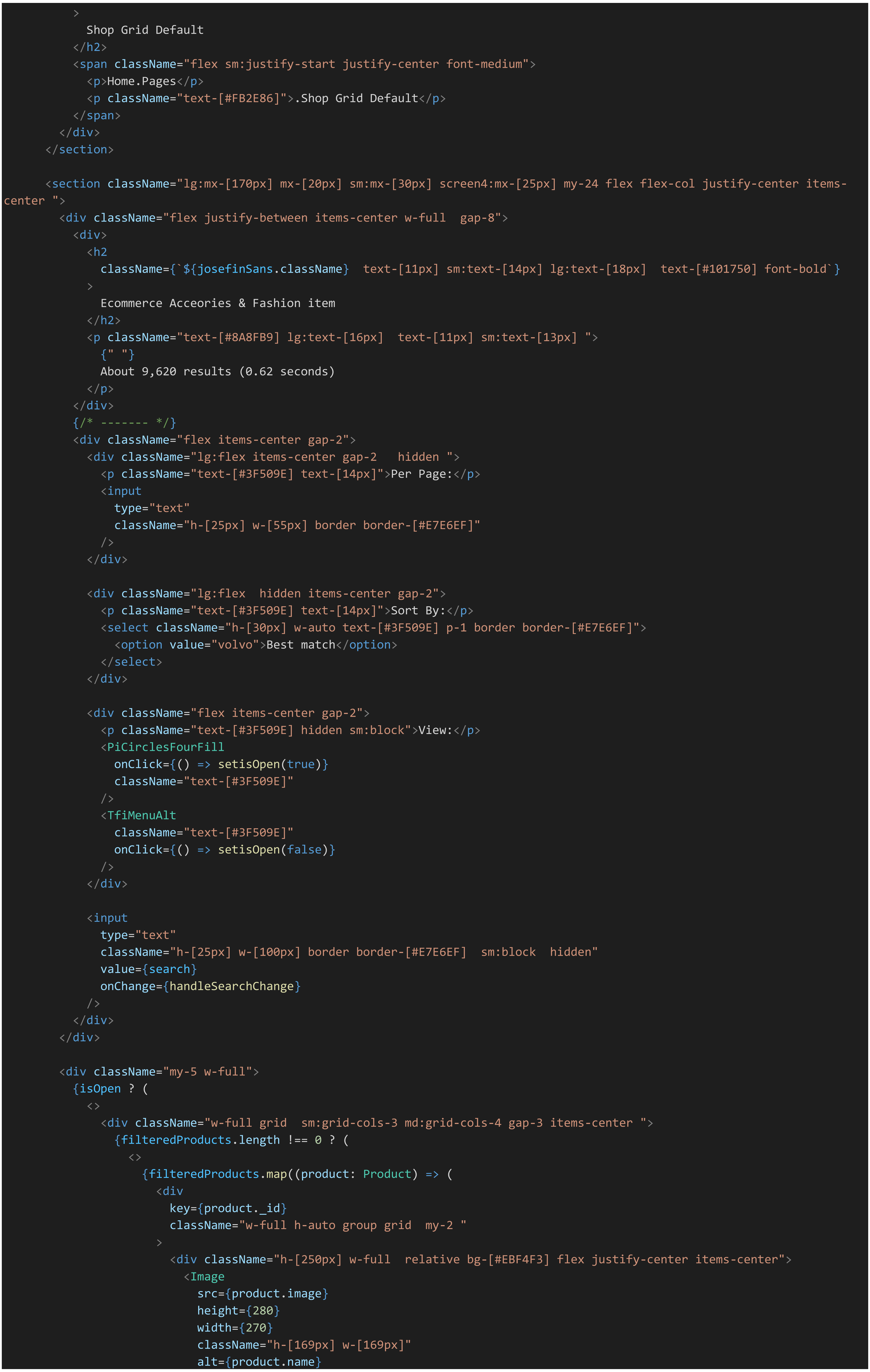
https://ui-ux-hackathon-sable.vercel.app/products



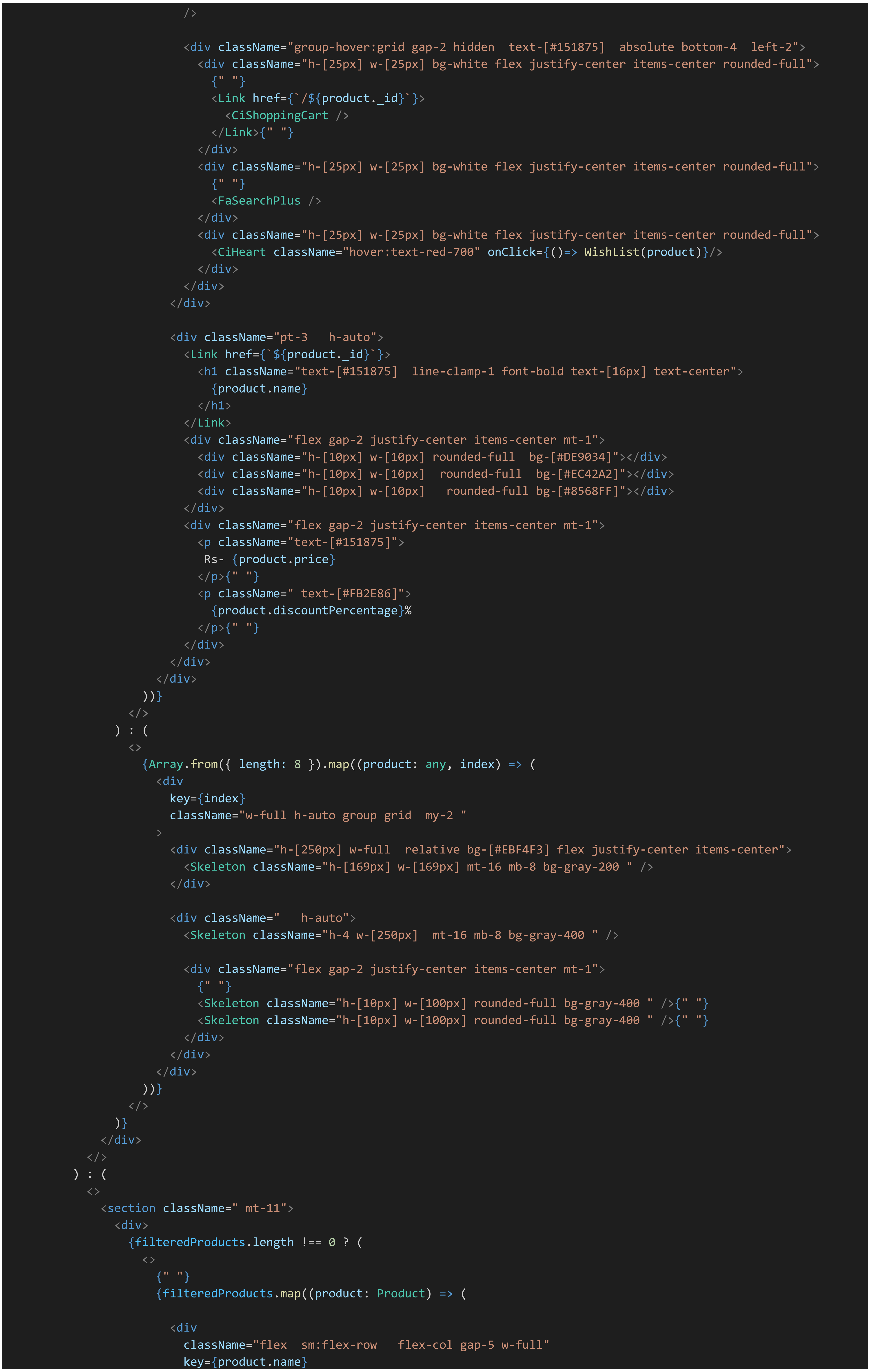
```
"use client";

import React, { useContext, useState } from "react";
import { client } from "@/sanity/lib/client";
import Image from "next/image";
import Link from "next/link";
import { FaSearchPlus } from "react-icons/fa";
import { CiHeart, CiShoppingCart } from "react-icons/ci";
import { searchContext } from "../conntext";
import { PiCirclesFourFill } from "react-icons/pi";
import { TfiMenuAlt } from "react-icons/tfi";
import { Josefin_Sans } from "next/font/google";
import { Skeleton } from "@/components/ui/skeleton";
import { FaStar } from "react-icons/fa6";
import { toast } from "sonner";


const josefinSans = Josefin_Sans({
  subsets: ["latin"],
  weight: ["400", "700"],
});

interface Product {
  price: string;
  name: string;
  discountPercentage: number;
  description: string;
  image: string;
```



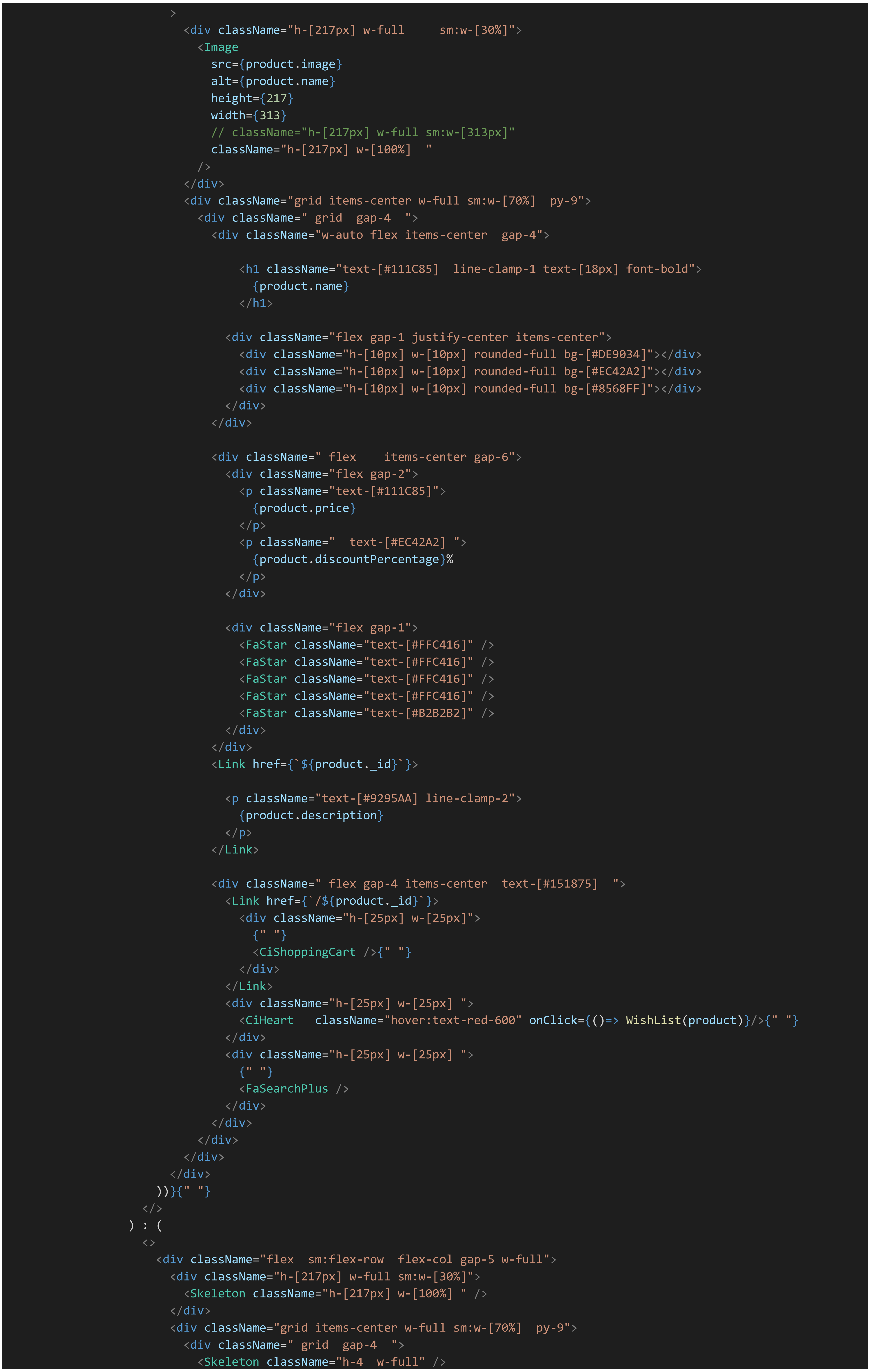
```
  _id: string;
}

export default function Shop() {
  const searchQuery = useContext(searchContext);
  const [products, setProducts] = useState<Product[]>([]);
  const [search, setSearch] = useState("");
  const [isOpen, setisOpen] = useState(true);

  React.useEffect(() => {
    const fetchProducts = async () => {
      const res = await client.fetch(`*[_type == "product" && "product" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);
      setProducts(res);
    };
    fetchProducts();
  }, []);

  const handleSearchChange = (e: React.ChangeEvent<HTMLInputElement>) => {
    setSearch(e.target.value);
  };

  // Filter products based on search query
  const filteredProducts = products.filter((product) =>
    product.name

      .toLowerCase()
      .includes(searchQuery.search.toLowerCase() || search.toLowerCase())
  );



  // Custom Button Component for the action
  const ViewCartButton = () => (
   <Link  href={"/wishlist"}> <button
      className="bg-[#FB2E86] text-white   w-[90px] py-2 px-4 rounded "
    >
      WishList
    </button>
    </Link>
  );

     const WishList  = (Wishlist : Product)=>{

      if(Wishlist){
        const wishListItems = localStorage.getItem("wishlist");

        const wishlist : Product[] = wishListItems ? JSON.parse(wishListItems) : []

          wishlist.push(Wishlist)
          localStorage.setItem("wishlist" , JSON.stringify(wishlist))
              toast("⍰ Item successfully added to your Wishlist!", {
                  description:
                    "You can continue shopping or proceed to view your wishlist by clicking below.",
                  action: <ViewCartButton />,
                });

      }

     }





  return (
    <main className="max-w-[1920px] mx-auto">
      <section className="h-[286px] w-full bg-[#F6F5FF] grid items-center ">
        <div className="sm:mx-[170px] mx-[30px] overflow-hidden text-center sm:text-left">
          <h2
            className={`${josefinSans.className} text-[25px] sm:text-[36px] text-[#101750] font-bold`}
```



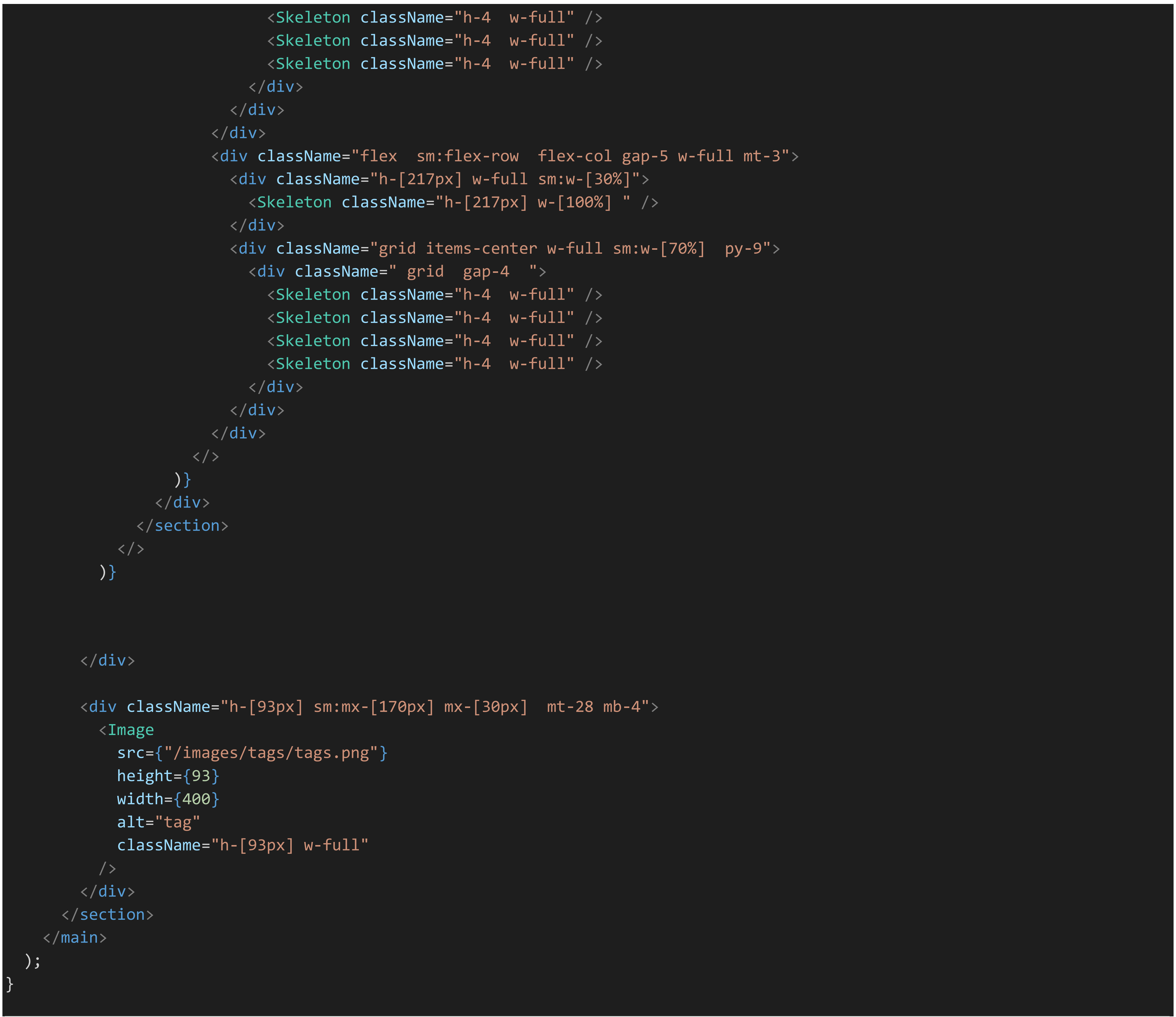
```
          >
            Shop Grid Default
          </h2>
          <span className="flex sm:justify-start justify-center font-medium">
            <p>Home.Pages</p>
            <p className="text-[#FB2E86]">.Shop Grid Default</p>
          </span>
        </div>
      </section>

      <section className="lg:mx-[170px] mx-[20px] sm:mx-[30px] screen4:mx-[25px] my-24 flex flex-col justify-center items-
center ">
        <div className="flex justify-between items-center w-full  gap-8">
          <div>
            <h2
              className={`${josefinSans.className}  text-[11px] sm:text-[14px] lg:text-[18px]  text-[#101750] font-bold`}
            >
              Ecommerce Acceories & Fashion item
            </h2>
            <p className="text-[#8A8FB9] lg:text-[16px]  text-[11px] sm:text-[13px] ">
              {" "}
              About 9,620 results (0.62 seconds)
            </p>
          </div>
          {/* ------- */}
          <div className="flex items-center gap-2">
            <div className="lg:flex items-center gap-2   hidden ">
              <p className="text-[#3F509E] text-[14px]">Per Page:</p>
              <input
                type="text"
                className="h-[25px] w-[55px] border border-[#E7E6EF]"
              />
            </div>

            <div className="lg:flex  hidden items-center gap-2">
              <p className="text-[#3F509E] text-[14px]">Sort By:</p>
              <select className="h-[30px] w-auto text-[#3F509E] p-1 border border-[#E7E6EF]">
                <option value="volvo">Best match</option>
              </select>
            </div>

            <div className="flex items-center gap-2">
              <p className="text-[#3F509E] hidden sm:block">View:</p>
              <PiCirclesFourFill
                onClick={() => setisOpen(true)}
                className="text-[#3F509E]"
              />
              <TfiMenuAlt
                className="text-[#3F509E]"
                onClick={() => setisOpen(false)}
              />
            </div>

            <input
              type="text"
              className="h-[25px] w-[100px] border border-[#E7E6EF]  sm:block  hidden"
              value={search}
              onChange={handleSearchChange}
            />
          </div>
        </div>

        <div className="my-5 w-full">
          {isOpen ? (
            <>
              <div className="w-full grid  sm:grid-cols-3 md:grid-cols-4 gap-3 items-center ">
                {filteredProducts.length !== 0 ? (
                  <>
                    {filteredProducts.map((product: Product) => (
                      <div
                        key={product._id}
                        className="w-full h-auto group grid  my-2 "
                      >
                        <div className="h-[250px] w-full  relative bg-[#EBF4F3] flex justify-center items-center">
                          <Image
                            src={product.image}
                            height={280}
                            width={270}
                            className="h-[169px] w-[169px]"
                            alt={product.name}
```



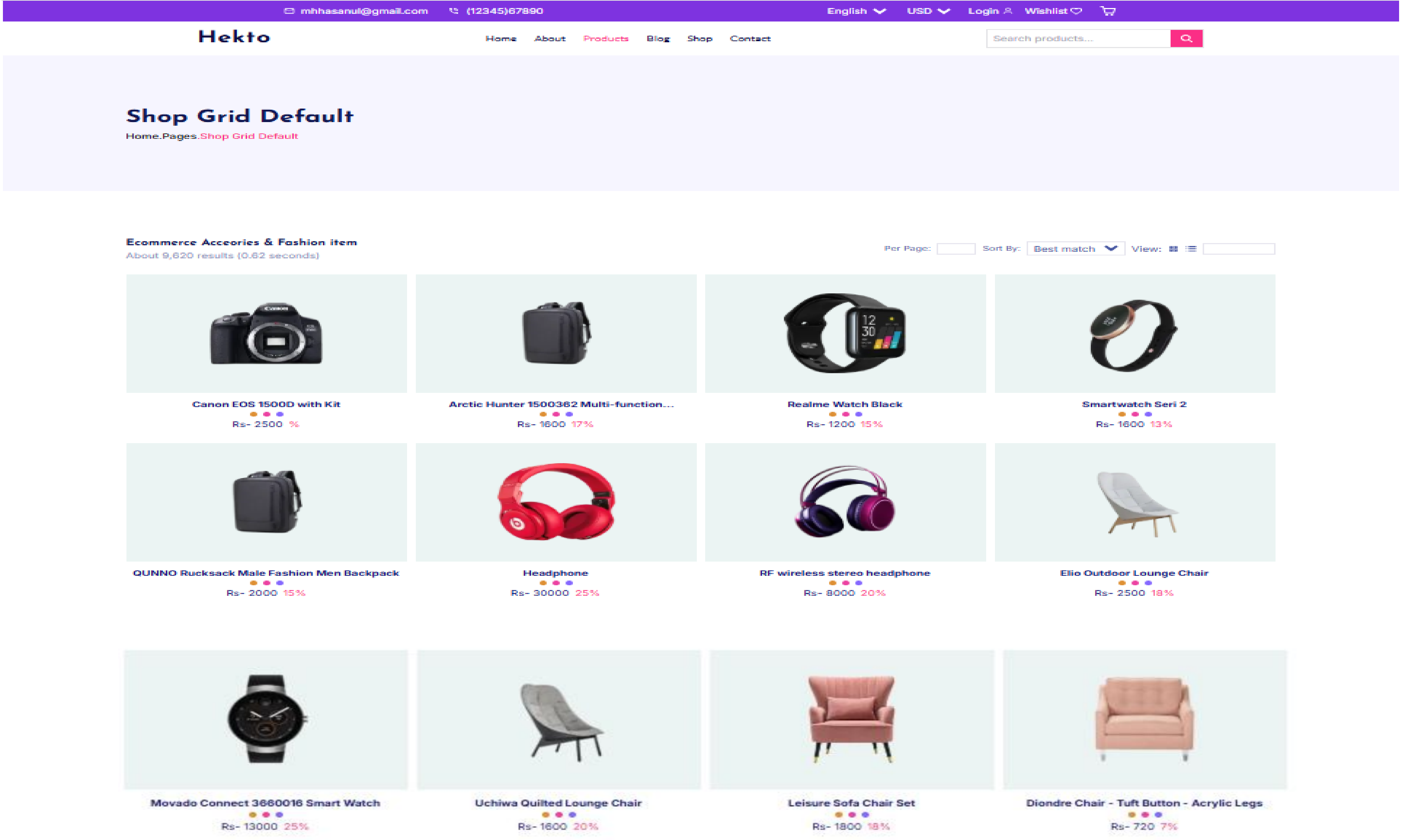
```
                />

                <div className="group-hover:grid gap-2 hidden  text-[#151875]  absolute bottom-4  left-2">
                  <div className="h-[25px] w-[25px] bg-white flex justify-center items-center rounded-full">
                    {" "}
                    <Link href={`/${product._id}`}>
                      <CiShoppingCart />
                    </Link>{" "}
                  </div>
                  <div className="h-[25px] w-[25px] bg-white flex justify-center items-center rounded-full">
                    {" "}
                    <FaSearchPlus />
                  </div>
                  <div className="h-[25px] w-[25px] bg-white flex justify-center items-center rounded-full">
                    <CiHeart className="hover:text-red-700" onClick={()=> WishList(product)}/>
                  </div>
                </div>
              </div>

              <div className="pt-3   h-auto">
                <Link href={`${product._id}`}>
                  <h1 className="text-[#151875]  line-clamp-1 font-bold text-[16px] text-center">
                    {product.name}
                  </h1>
                </Link>
                <div className="flex gap-2 justify-center items-center mt-1">
                  <div className="h-[10px] w-[10px] rounded-full  bg-[#DE9034]"></div>
                  <div className="h-[10px] w-[10px]  rounded-full  bg-[#EC42A2]"></div>
                  <div className="h-[10px] w-[10px]   rounded-full bg-[#8568FF]"></div>
                </div>
                <div className="flex gap-2 justify-center items-center mt-1">
                  <p className="text-[#151875]">
                   Rs- {product.price}
                  </p>{" "}
                  <p className=" text-[#FB2E86]">
                    {product.discountPercentage}%
                  </p>{" "}
                </div>
              </div>
            </div>
          ))}
        </>
      ) : (
        <>
          {Array.from({ length: 8 }).map((product: any, index) => (
            <div
              key={index}
              className="w-full h-auto group grid  my-2 "
            >
              <div className="h-[250px] w-full  relative bg-[#EBF4F3] flex justify-center items-center">
                <Skeleton className="h-[169px] w-[169px] mt-16 mb-8 bg-gray-200 " />
              </div>

              <div className="   h-auto">
                <Skeleton className="h-4 w-[250px]  mt-16 mb-8 bg-gray-400 " />

                <div className="flex gap-2 justify-center items-center mt-1">
                  {" "}
                  <Skeleton className="h-[10px] w-[100px] rounded-full bg-gray-400 " />{" "}
                  <Skeleton className="h-[10px] w-[100px] rounded-full bg-gray-400 " />{" "}
                </div>
              </div>
            </div>
          ))}
        </>
      )}
    </div>
  </>
) : (
  <>
    <section className=" mt-11">
      <div>
        {filteredProducts.length !== 0 ? (
          <>
            {" "}
            {filteredProducts.map((product: Product) => (

              <div
                className="flex  sm:flex-row   flex-col gap-5 w-full"
                key={product.name}
```



```
      >
        <div className="h-[217px] w-full     sm:w-[30%]">
          <Image
            src={product.image}
            alt={product.name}
            height={217}
            width={313}
            // className="h-[217px] w-full sm:w-[313px]"
            className="h-[217px] w-[100%]  "
          />
        </div>
        <div className="grid items-center w-full sm:w-[70%]  py-9">
          <div className=" grid  gap-4  ">
            <div className="w-auto flex items-center  gap-4">

                <h1 className="text-[#111C85]  line-clamp-1 text-[18px] font-bold">
                  {product.name}
                </h1>

              <div className="flex gap-1 justify-center items-center">
                <div className="h-[10px] w-[10px] rounded-full bg-[#DE9034]"></div>
                <div className="h-[10px] w-[10px] rounded-full bg-[#EC42A2]"></div>
                <div className="h-[10px] w-[10px] rounded-full bg-[#8568FF]"></div>
              </div>
            </div>

            <div className=" flex    items-center gap-6">
              <div className="flex gap-2">
                <p className="text-[#111C85]">
                  {product.price}
                </p>
                <p className="  text-[#EC42A2] ">
                  {product.discountPercentage}%
                </p>
              </div>

              <div className="flex gap-1">
                <FaStar className="text-[#FFC416]" />
                <FaStar className="text-[#FFC416]" />
                <FaStar className="text-[#FFC416]" />
                <FaStar className="text-[#FFC416]" />
                <FaStar className="text-[#B2B2B2]" />
              </div>
            </div>
            <Link href={`${product._id}`}>

              <p className="text-[#9295AA] line-clamp-2">
                {product.description}
              </p>
            </Link>

            <div className=" flex gap-4 items-center  text-[#151875]  ">
              <Link href={`/${product._id}`}>
                <div className="h-[25px] w-[25px]">
                  {" "}
                  <CiShoppingCart />{" "}
                </div>
              </Link>
              <div className="h-[25px] w-[25px] ">
                <CiHeart   className="hover:text-red-600" onClick={()=> WishList(product)}/>{" "}
              </div>
              <div className="h-[25px] w-[25px] ">
                {" "}
                <FaSearchPlus />
              </div>
            </div>
          </div>
        </div>
      </div>
    ))}{" "}
  </>
) : (
  <>
    <div className="flex  sm:flex-row  flex-col gap-5 w-full">
      <div className="h-[217px] w-full sm:w-[30%]">
        <Skeleton className="h-[217px] w-[100%] " />
      </div>
      <div className="grid items-center w-full sm:w-[70%]  py-9">
        <div className=" grid  gap-4  ">
          <Skeleton className="h-4  w-full" />
```



```
                            <Skeleton className="h-4  w-full" />
                            <Skeleton className="h-4  w-full" />
                            <Skeleton className="h-4  w-full" />
                          </div>
                        </div>
                      </div>
                      <div className="flex  sm:flex-row  flex-col gap-5 w-full mt-3">
                        <div className="h-[217px] w-full sm:w-[30%]">
                          <Skeleton className="h-[217px] w-[100%] " />
                        </div>
                        <div className="grid items-center w-full sm:w-[70%]  py-9">
                          <div className=" grid  gap-4  ">
                            <Skeleton className="h-4  w-full" />
                            <Skeleton className="h-4  w-full" />
                            <Skeleton className="h-4  w-full" />
                            <Skeleton className="h-4  w-full" />
                          </div>
                        </div>
                      </div>
                    </>
                  )}
                </div>
              </section>
            </>
          )}



        </div>

        <div className="h-[93px] sm:mx-[170px] mx-[30px]  mt-28 mb-4">
          <Image
            src={"/images/tags/tags.png"}
            height={93}
            width={400}
            alt="tag"
            className="h-[93px] w-full"
          />
        </div>
      </section>
    </main>
  );
}
```



Shop Grid Default

Home.Pages.Shop Grid Default

Ecommerce Acceories & Fashion item

About 9,620 results (0.62 seconds)

Canon EOS 1500D with Kit
Rs- 2500 %

Arctic Hunter 1500362 Multi-function...
Rs- 1600 17%

Realme Watch Black
Rs- 1200 15%

Smartwatch Seri 2
Rs- 1600 13%

QUNNO Rucksack Male Fashion Men Backpack
Rs- 2000 15%

Headphone
Rs- 30000 25%

RF wireless stereo headphone
Rs- 8000 20%

Elio Outdoor Lounge Chair
Rs- 2500 18%

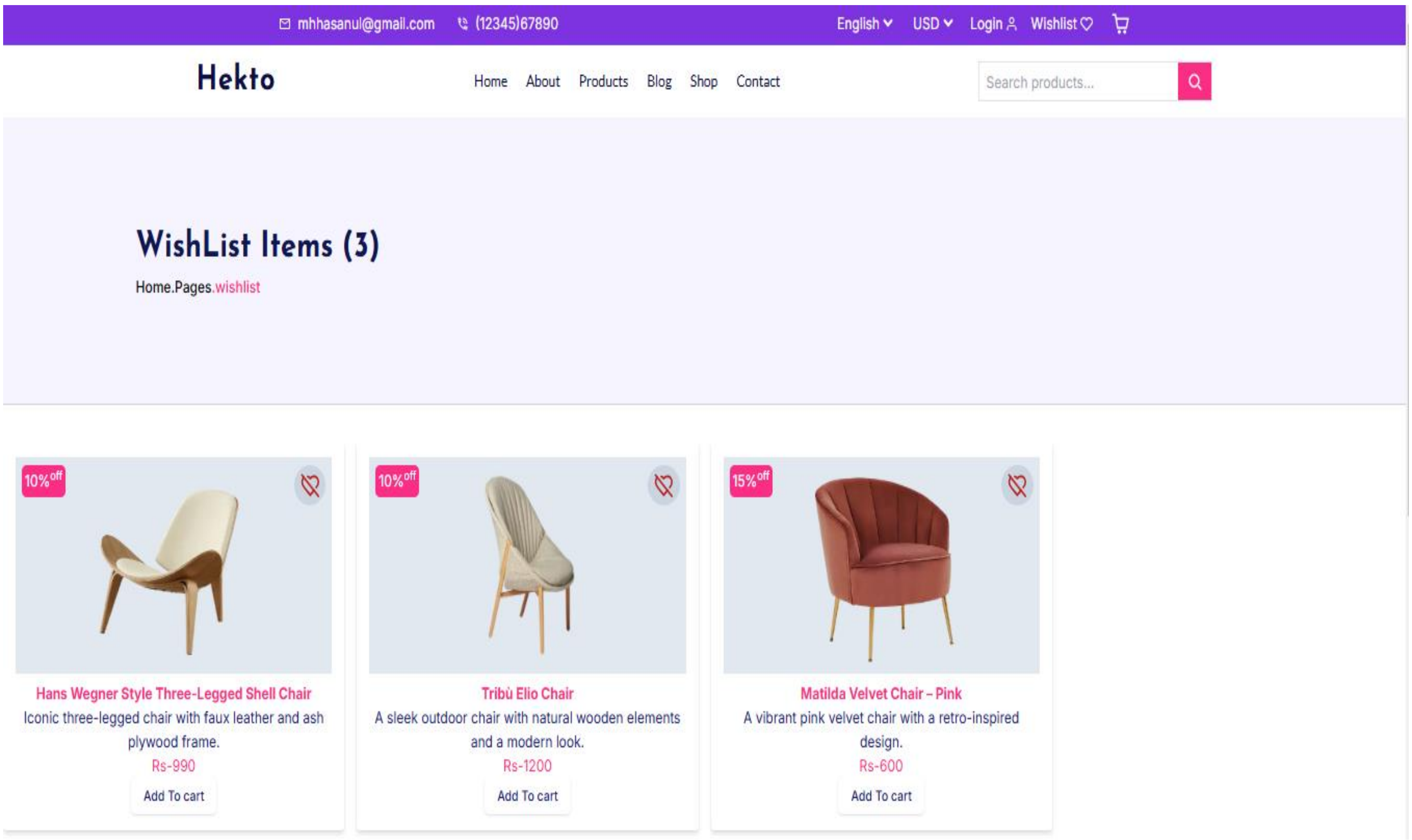
Movado Connect 3660016 Smart Watch
Rs- 13000 25%

Uchiwa Quilted Lounge Chair
Rs- 1600 20%

Leisure Sofa Chair Set
Rs- 1800 18%

Diondre Chair - Tuft Button - Acrylic Legs
Rs- 720 7%

- ✓ Whish List Page add all logic



WishList Items (3)

Home.Pages.wishlist

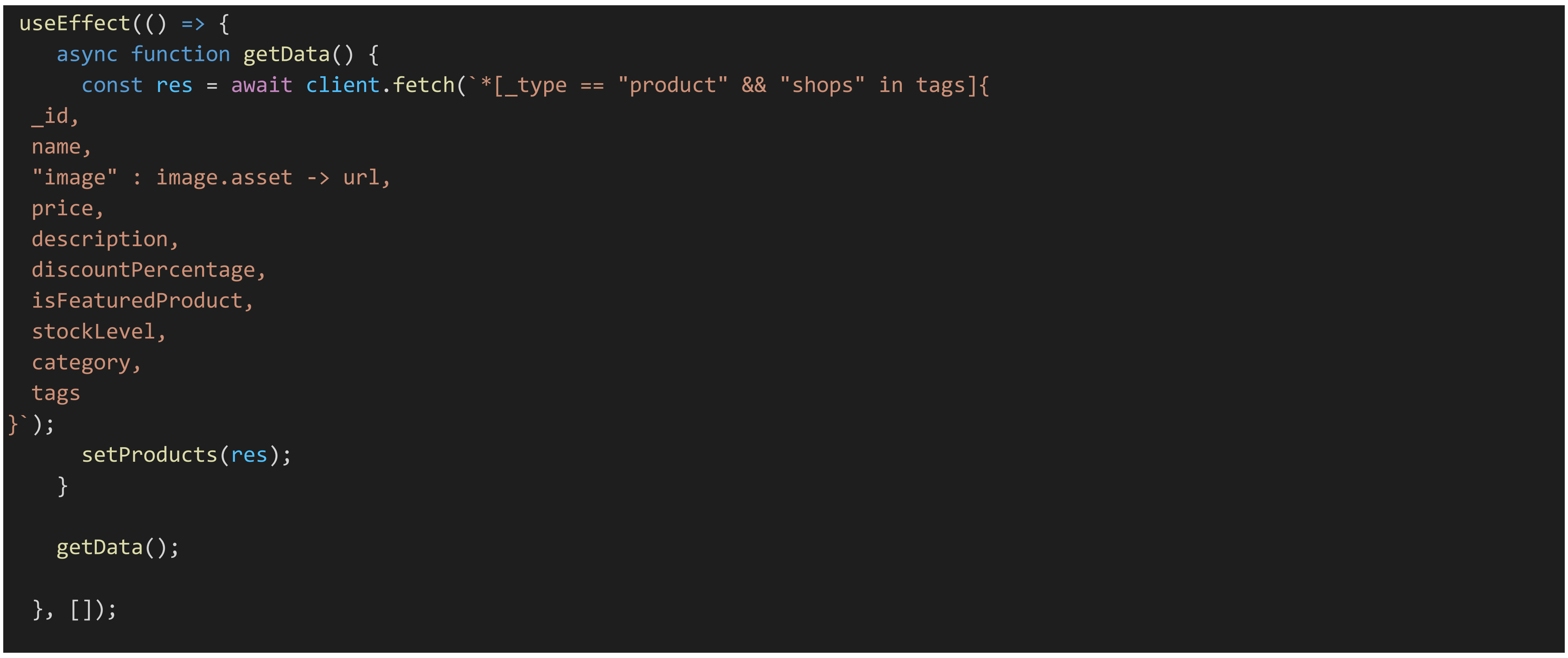
Hans Wegner Style Three-Legged Shell Chair
Iconic three-legged chair with faux leather and ash plywood frame.
Rs-990

Tribù Elio Chair
A sleek outdoor chair with natural wooden elements and a modern look.
Rs-1200

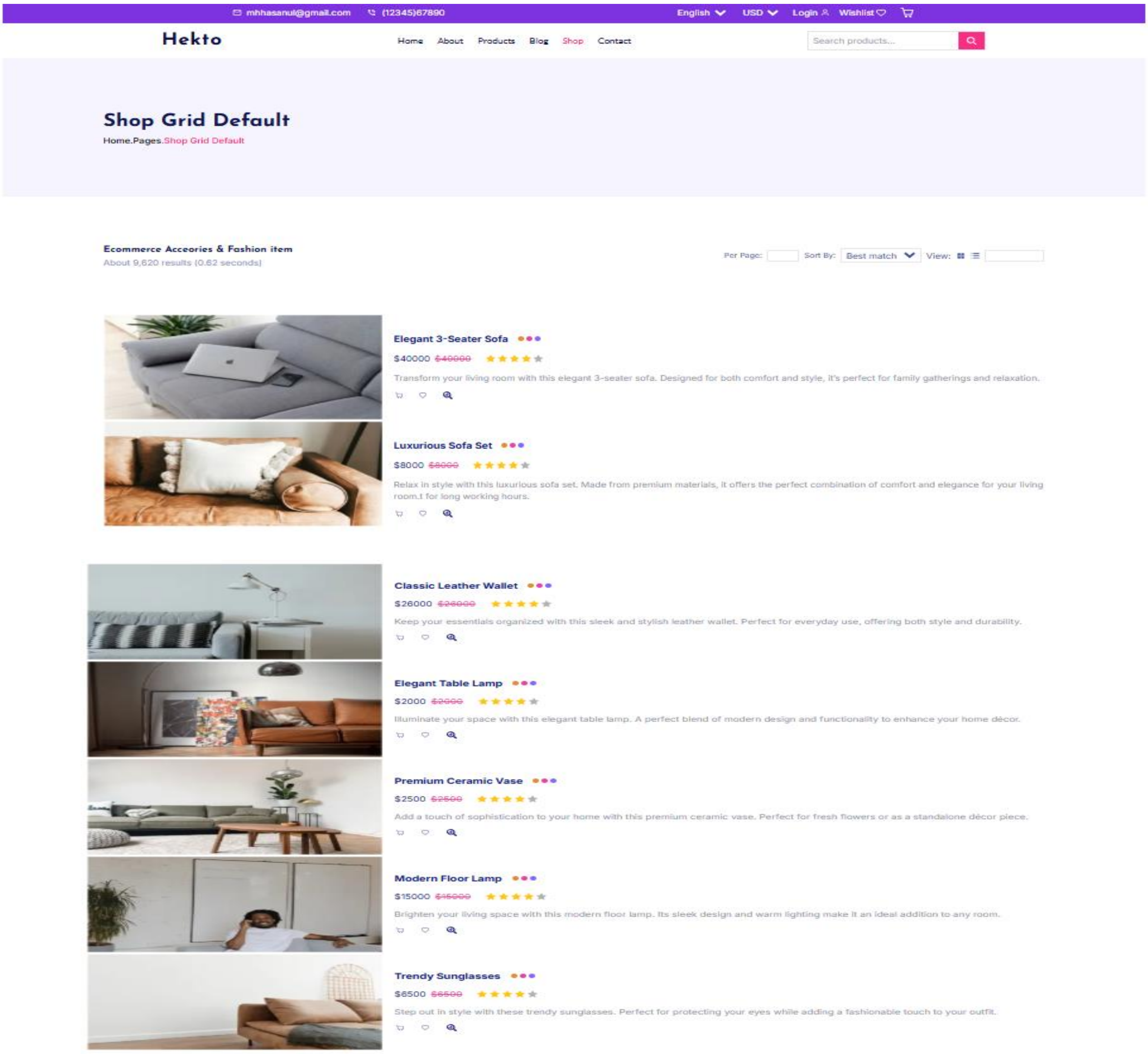
Matilda Velvet Chair – Pink
A vibrant pink velvet chair with a retro-inspired design.
Rs-600

✓ Shop Route

https://ui-ux-hackathon-sable.vercel.app/shop



```
useEffect(() => {
   async function getData() {
     const res = await client.fetch(`*[_type == "product" && "shops" in tags]{
 _id,
 name,
 "image" : image.asset -> url,
 price,
 description,
 discountPercentage,
 isFeaturedProduct,
 stockLevel,
 category,
 tags
}`);
     setProducts(res);
   }

   getData();

 }, []);
```



Shop Grid Default

Home.Pages.Shop Grid Default

Ecommerce Acceories & Fashion item

About 9,620 results (0.62 seconds)

Per Page: Sort By: Best match View:

Elegant 3-Seater Sofa

$40000 ~~$40000~~ ★★★★★

Transform your living room with this elegant 3-seater sofa. Designed for both comfort and style, it's perfect for family gatherings and relaxation.

Luxurious Sofa Set

$8000 ~~$8000~~ ★★★★★

Relax in style with this luxurious sofa set. Made from premium materials, it offers the perfect combination of comfort and elegance for your living room.t for long working hours.

Classic Leather Wallet

$26000 ~~$26000~~ ★★★★★

Keep your essentials organized with this sleek and stylish leather wallet. Perfect for everyday use, offering both style and durability.

Elegant Table Lamp

$2000 ~~$2000~~ ★★★★★

Illuminate your space with this elegant table lamp. A perfect blend of modern design and functionality to enhance your home décor.

Premium Ceramic Vase

$2500 ~~$2500~~ ★★★★★

Add a touch of sophistication to your home with this premium ceramic vase. Perfect for fresh flowers or as a standalone décor piece.

Modern Floor Lamp

$15000 ~~$15000~~ ★★★★★

Brighten your living space with this modern floor lamp. Its sleek design and warm lighting make it an ideal addition to any room.

Trendy Sunglasses

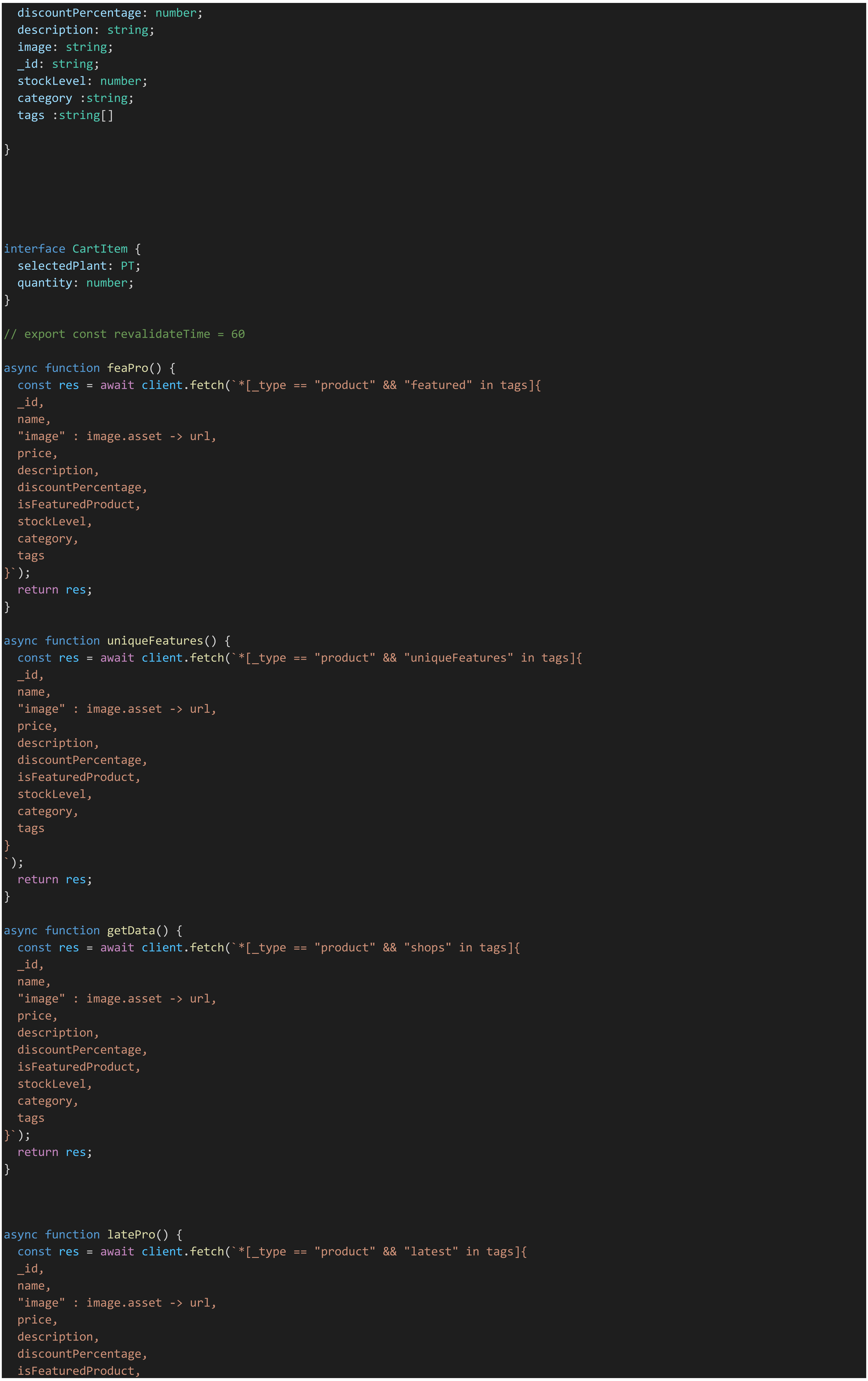
$6500 ~~$6500~~ ★★★★★

Step out in style with these trendy sunglasses. Perfect for protecting your eyes while adding a fashionable touch to your outfit.

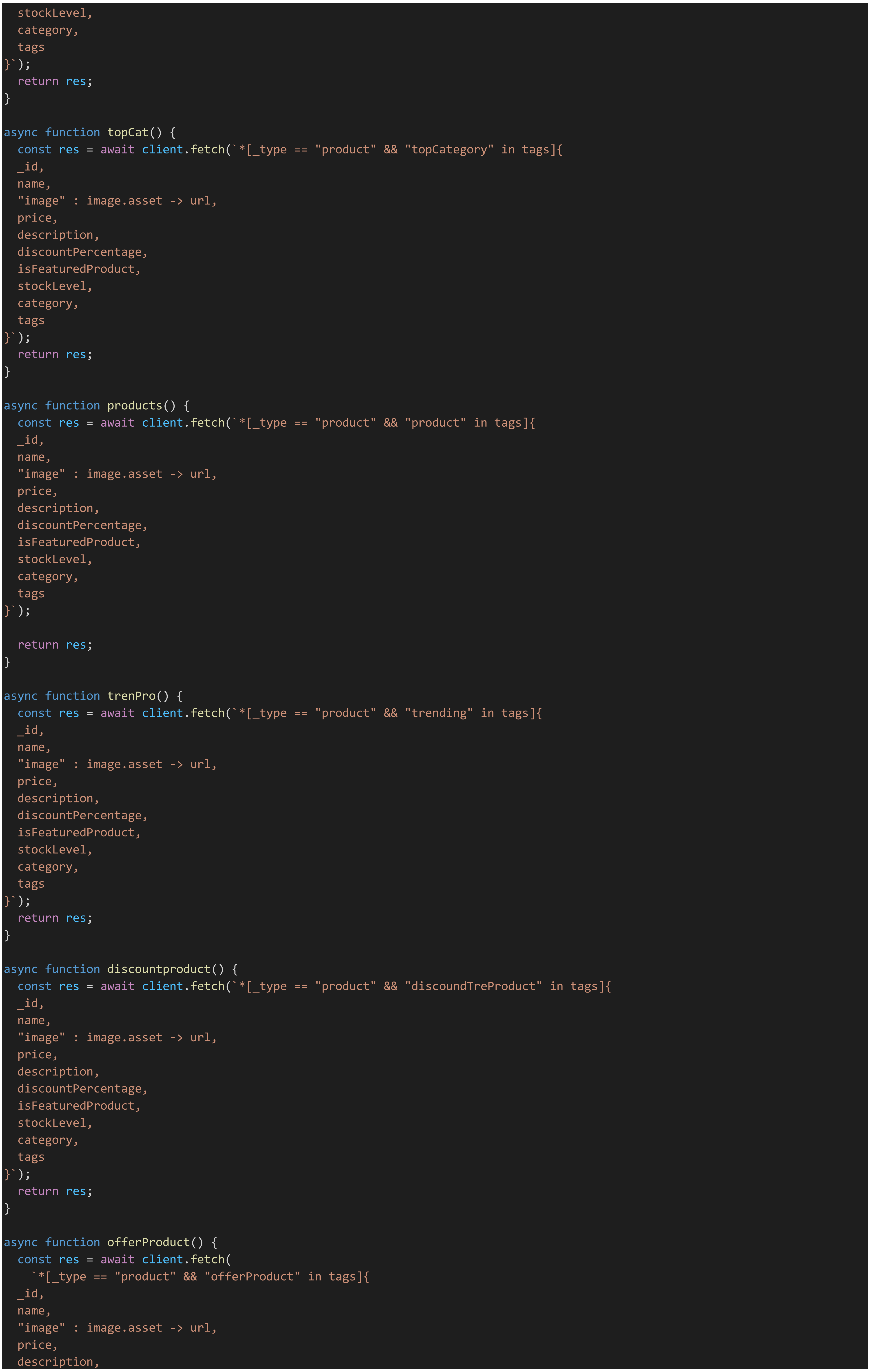
- ✓ Dynamic Route for display product detail page

https://ui-ux-hackathon-sable.vercel.app/jGHNK2T7WRTGSZ6CroBXgK

```
import { Josefin_Sans } from "next/font/google";
import Image from "next/image";
import { FaInstagramSquare, FaStar } from "react-icons/fa";
import { FaFacebook } from "react-icons/fa6";
import { AiFillTwitterCircle } from "react-icons/ai";
import { FaArrowRight } from "react-icons/fa6";
import { client } from "@/sanity/lib/client";
import AddtocardButton from "../components/addtocardButton";
import Wishlistbutton from "../components/wishlistbutton";


const josefinSans = Josefin_Sans({
  subsets: ["latin"],
  weight: ["400", "700"],
});

interface Params {
  id: string;
}

export const revalidate = 3;



interface PT {
  price: string;
  name: string;
```

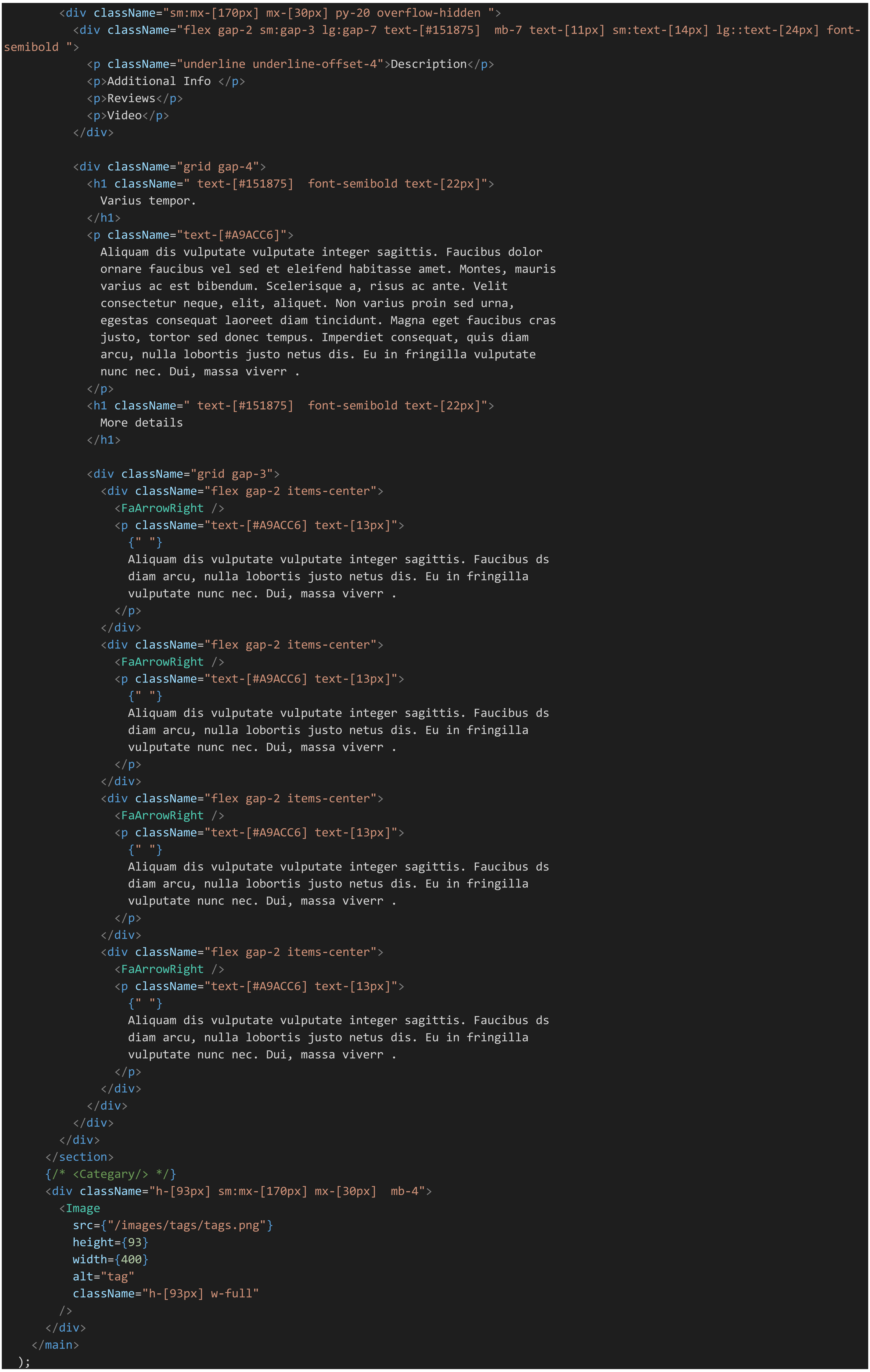


```
  discountPercentage: number;
  description: string;
  image: string;
  _id: string;
  stockLevel: number;
  category :string;
  tags :string[]

}





interface CartItem {
  selectedPlant: PT;
  quantity: number;
}

// export const revalidateTime = 60

async function feaPro() {
  const res = await client.fetch(`*[_type == "product" && "featured" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);
  return res;
}

async function uniqueFeatures() {
  const res = await client.fetch(`*[_type == "product" && "uniqueFeatures" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}
`);
  return res;
}

async function getData() {
  const res = await client.fetch(`*[_type == "product" && "shops" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);
  return res;
}



async function latePro() {
  const res = await client.fetch(`*[_type == "product" && "latest" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
```



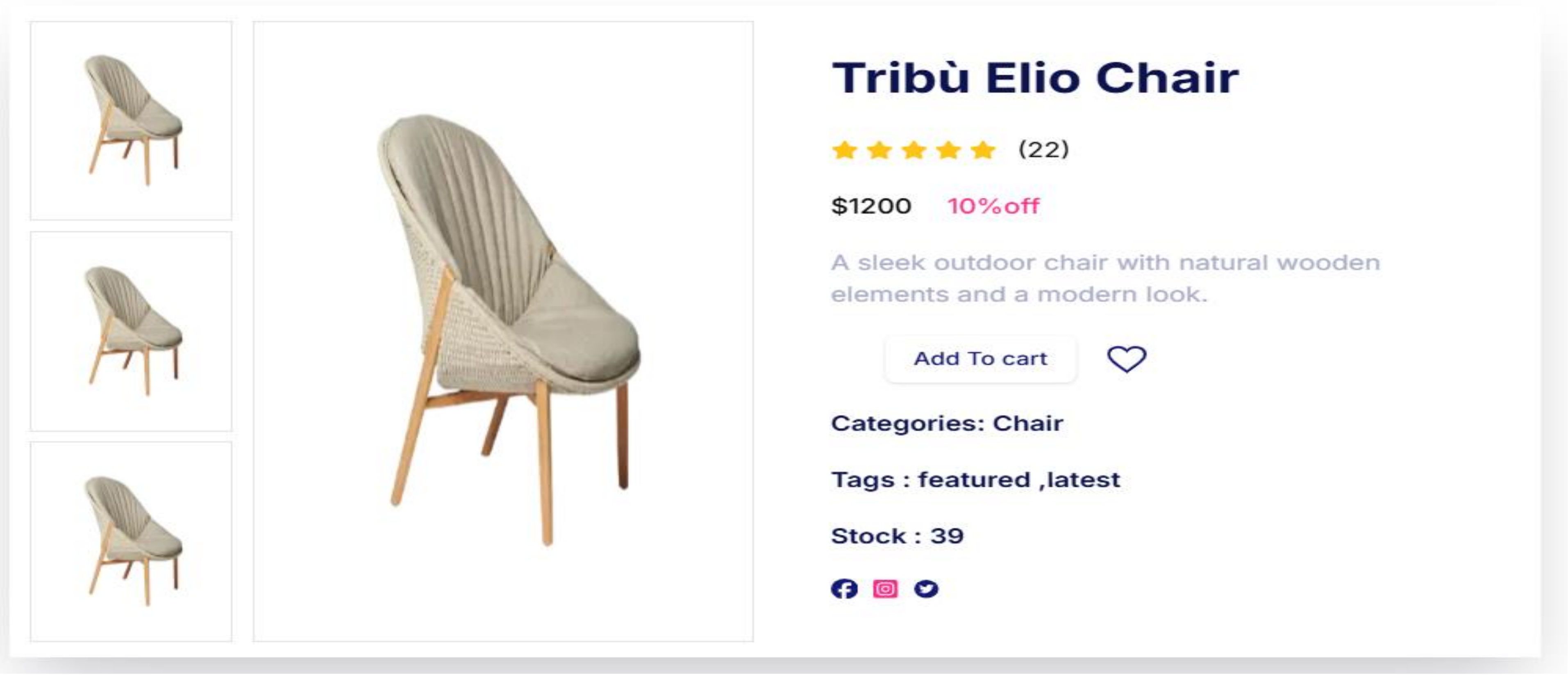
```
  stockLevel,
  category,
  tags
}`);
  return res;
}

async function topCat() {
  const res = await client.fetch(`*[_type == "product" && "topCategory" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);
  return res;
}

async function products() {
  const res = await client.fetch(`*[_type == "product" && "product" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);

  return res;
}

async function trenPro() {
  const res = await client.fetch(`*[_type == "product" && "trending" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);
  return res;
}

async function discountproduct() {
  const res = await client.fetch(`*[_type == "product" && "discoundTreProduct" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}`);
  return res;
}

async function offerProduct() {
  const res = await client.fetch(
    `*[_type == "product" && "offerProduct" in tags]{
  _id,
  name,
  "image" : image.asset -> url,
  price,
  description,
  discountPercentage,
  isFeaturedProduct,
  stockLevel,
  category,
  tags
}
`,
  );

  return res;
}

async function DettailPage({ params }: { params: Params }) {
  const { id } = params;
  let featureProducts = await feaPro();
  let latestProduct = await latePro();
  let topCategary = await topCat();
  let Product = await products();
  let trendingproduct = await trenPro();
  let discountPro = await discountproduct();
  let offerPro = await offerProduct();
  let shop = await getData();
  let uniqueFeature = await uniqueFeatures();

  const allProduct: PT[] = featureProducts.concat(
    latestProduct,
    offerPro,
    uniqueFeature,
    shop,
    discountPro,
    topCategary,
    Product,
    trendingproduct
  );

  const product = allProduct.find((item: any) => String(item._id) === id);
  console.log(product);
  if (!product) {
    return (
      <div className="flex items-center justify-center h-screen text-xl font-semibold">
        product Not Found
      </div>
    );
  }




  return (
    <main className="max-w-[1920px] mx-auto">
      <section className="h-[286px]  w-full  bg-[#F6F5FF]  grid items-center">
        <div className="sm:mx-[170px] mx-[30px] overflow-hidden text-center sm:text-left">
          <h2
            className={`${josefinSans.className} text-[25px]  sm:text-[36px] text-[#101750] font-bold`}
          >
            Product Details
          </h2>
          <span className="flex sm:justify-start justify-center font-medium">
            <p>Home.Pages</p>
            <p className="text-[#FB2E86]">.Product Details</p>
          </span>
        </div>
      </section>
      <section className=" lg:mx-[170px]  mx[30px] ">
        <div className="  shadow-[#c3c3c5] shadow-2xl h-auto  w-full my-24 sm:my-36 flex  sm:flex-row  flex-col">
          <div className="sm:w-[50%] w-full p-3 flex gap-3">
            <div className="w-[151px]  md:grid gap-2 hidden  ">
              <div className="w-full    h-[155px] bg-inherit hover:bg-[#C4C4C4]  border border-[#e5e2e2] flex justify-center
items-center">
                <Image
                  src={product.image}
                  height={200}
                  width={200}
                  alt={product.name}
                  className="h-[130px] w-[130px] p-2"
                />
              </div>
              {/* [#C4C4C4] */}
              <div className="w-full h-[155px] bg-inherit hover:bg-[#C4C4C4]   border border-[#e5e2e2] flex justify-center
items-center">
                <Image
```

```
         src={product.image}
         height={155}
         width={151}
         alt={product.name}
         className="h-[130px] w-[130px] p-2"
        />
       </div>
       <div className="w-full h-[155px] bg-inherit hover:bg-[#C4C4C4]  border border-[#e5e2e2] flex justify-center items-center">
        <Image
         src={product.image}
         height={155}
         width={151}
         alt={product.name}
         className="h-[130px] w-[130px] p-2"
        />
       </div>
      </div>
      <div className="sm:w-[375px] w-full   bg-inherit hover:bg-[#C4C4C4]  border border-[#e5e2e2]  flex justify-center items-center">
       <Image
        src={product.image}
        height={487}
        width={375}
        alt={product.name}
        className=" h-[400px] w-full sm:w-[300px] p-3"
       />
      </div>
     </div>
     <div className="sm:w-[50%]  w-full p-8 ">
      <section className="grid gap-5">
       <h1 className="text-[#0D134E] text-[25px] sm:text-[32px] font-bold">
        {" "}
        {product.name}
       </h1>
       <span className="flex  gap-3 items-center">
        {" "}
        <span className="flex  gap-1 text-[#FFC416]">
         <FaStar /> <FaStar /> <FaStar /> <FaStar /> <FaStar />
        </span>{" "}
        (22)
       </span>
       <span className=" flex items-center gap-5 font-medium">
        <p>${product.price}</p>{" "}
        <p className="  text-[#FB2E86]">
         {product.discountPercentage}%off
        </p>{" "}
       </span>
       {/* <h2 className="text-[#0D134E] font-semibold">Color</h2> */}
       <p className="text-[#A9ACC6]">
        {product.description}
        {/* Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris
        tellus porttitor purus, et volutpat sit. */}
       </p>
       <span className="flex gap-4 items-center  ml-8 text-[#151875]">
        <AddtocardButton product={product} />
        <Wishlistbutton Product={product}/>
        {/* <IoMdHeartEmpty className="text-[28px] hover:text-red-600"  onClick={()=> WishList(product)} /> */}
       </span>
       <h1 className="text-[#0D134E] font-semibold">Categories: {product.category}</h1>
       <h1 className="text-[#0D134E] font-semibold">{`Tags :  ${product.tags.map((item : string)=> `${item}  `)} `}</h1>

       <h1 className="text-[#0D134E] font-semibold">
        Stock : {product.stockLevel}
       </h1>

       <span className="flex items-center  gap-4">
        <span className="flex items-center gap-2">
         {" "}
         <FaFacebook className="text-[#151875] " />
         <FaInstagramSquare className="text-[#FB2E86]" />
         <AiFillTwitterCircle className="text-[#151875]" />{" "}
        </span>{" "}
       </span>
      </section>
     </div>
    </div>
   </section>

   <section className="h-auto  w-full  bg-[#F6F5FF] mb-24 sm:mb-36 ">
```



```
        <div className="sm:mx-[170px] mx-[30px] py-20 overflow-hidden ">
          <div className="flex gap-2 sm:gap-3 lg:gap-7 text-[#151875]  mb-7 text-[11px] sm:text-[14px] lg::text-[24px] font-
semibold ">
            <p className="underline underline-offset-4">Description</p>
            <p>Additional Info </p>
            <p>Reviews</p>
            <p>Video</p>
          </div>

          <div className="grid gap-4">
            <h1 className=" text-[#151875]  font-semibold text-[22px]">
              Varius tempor.
            </h1>
            <p className="text-[#A9ACC6]">
              Aliquam dis vulputate vulputate integer sagittis. Faucibus dolor
              ornare faucibus vel sed et eleifend habitasse amet. Montes, mauris
              varius ac est bibendum. Scelerisque a, risus ac ante. Velit
              consectetur neque, elit, aliquet. Non varius proin sed urna,
              egestas consequat laoreet diam tincidunt. Magna eget faucibus cras
              justo, tortor sed donec tempus. Imperdiet consequat, quis diam
              arcu, nulla lobortis justo netus dis. Eu in fringilla vulputate
              nunc nec. Dui, massa viverr .
            </p>
            <h1 className=" text-[#151875]  font-semibold text-[22px]">
              More details
            </h1>

            <div className="grid gap-3">
              <div className="flex gap-2 items-center">
                <FaArrowRight />
                <p className="text-[#A9ACC6] text-[13px]">
                  {" "}
                  Aliquam dis vulputate vulputate integer sagittis. Faucibus ds
                  diam arcu, nulla lobortis justo netus dis. Eu in fringilla
                  vulputate nunc nec. Dui, massa viverr .
                </p>
              </div>
              <div className="flex gap-2 items-center">
                <FaArrowRight />
                <p className="text-[#A9ACC6] text-[13px]">
                  {" "}
                  Aliquam dis vulputate vulputate integer sagittis. Faucibus ds
                  diam arcu, nulla lobortis justo netus dis. Eu in fringilla
                  vulputate nunc nec. Dui, massa viverr .
                </p>
              </div>
              <div className="flex gap-2 items-center">
                <FaArrowRight />
                <p className="text-[#A9ACC6] text-[13px]">
                  {" "}
                  Aliquam dis vulputate vulputate integer sagittis. Faucibus ds
                  diam arcu, nulla lobortis justo netus dis. Eu in fringilla
                  vulputate nunc nec. Dui, massa viverr .
                </p>
              </div>
              <div className="flex gap-2 items-center">
                <FaArrowRight />
                <p className="text-[#A9ACC6] text-[13px]">
                  {" "}
                  Aliquam dis vulputate vulputate integer sagittis. Faucibus ds
                  diam arcu, nulla lobortis justo netus dis. Eu in fringilla
                  vulputate nunc nec. Dui, massa viverr .
                </p>
              </div>
            </div>
          </div>
        </div>
      </section>
      {/* <Category/> */}
      <div className="h-[93px] sm:mx-[170px] mx-[30px]  mb-4">
        <Image
          src={"/images/tags/tags.png"}
          height={93}
          width={400}
          alt="tag"
          className="h-[93px] w-full"
        />
      </div>
    </main>
  );
```

```
}

export default DettailPage;
```



Tribù Elio Chair

★★★★★ (22)

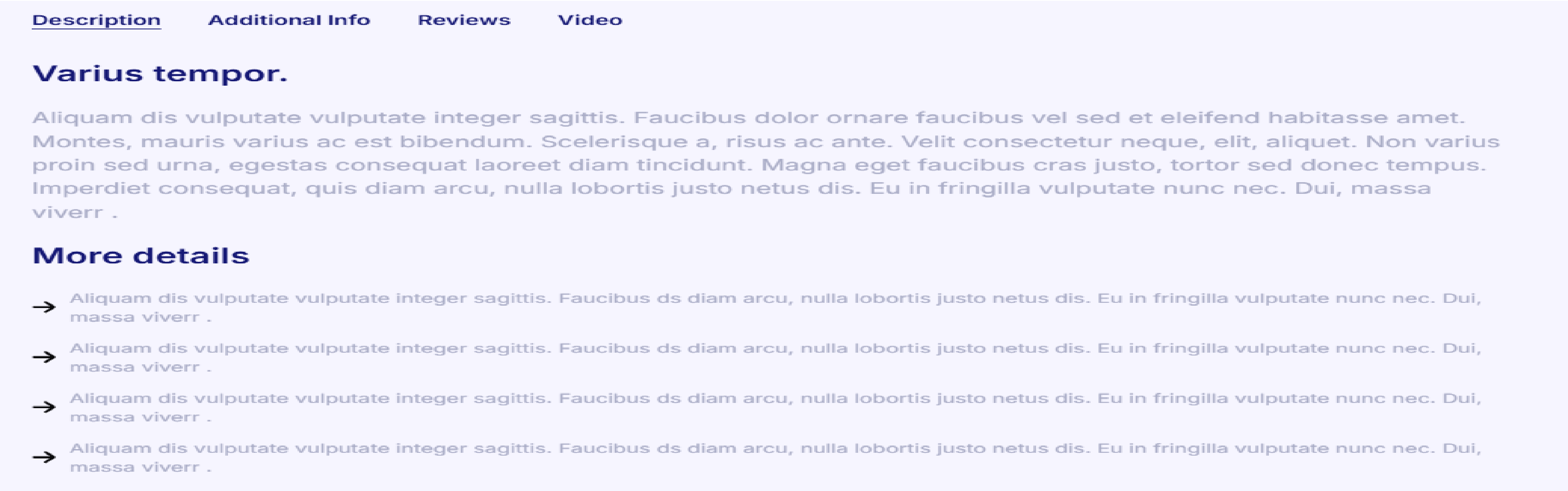
$1200 10%off

A sleek outdoor chair with natural wooden elements and a modern look.

Categories: Chair

Tags : featured ,latest

Stock : 39



Varius tempor.

Aliquam dis vulputate vulputate integer sagittis. Faucibus dolor ornare faucibus vel sed et eleifend habitasse amet. Montes, mauris varius ac est bibendum. Scelerisque a, risus ac ante. Velit consectetur neque, elit, aliquet. Non varius proin sed urna, egestas consequat laoreet diam tincidunt. Magna eget faucibus cras justo, tortor sed donec tempus. Imperdiet consequat, quis diam arcu, nulla lobortis justo netus dis. Eu in fringilla vulputate nunc nec. Dui, massa viverr .

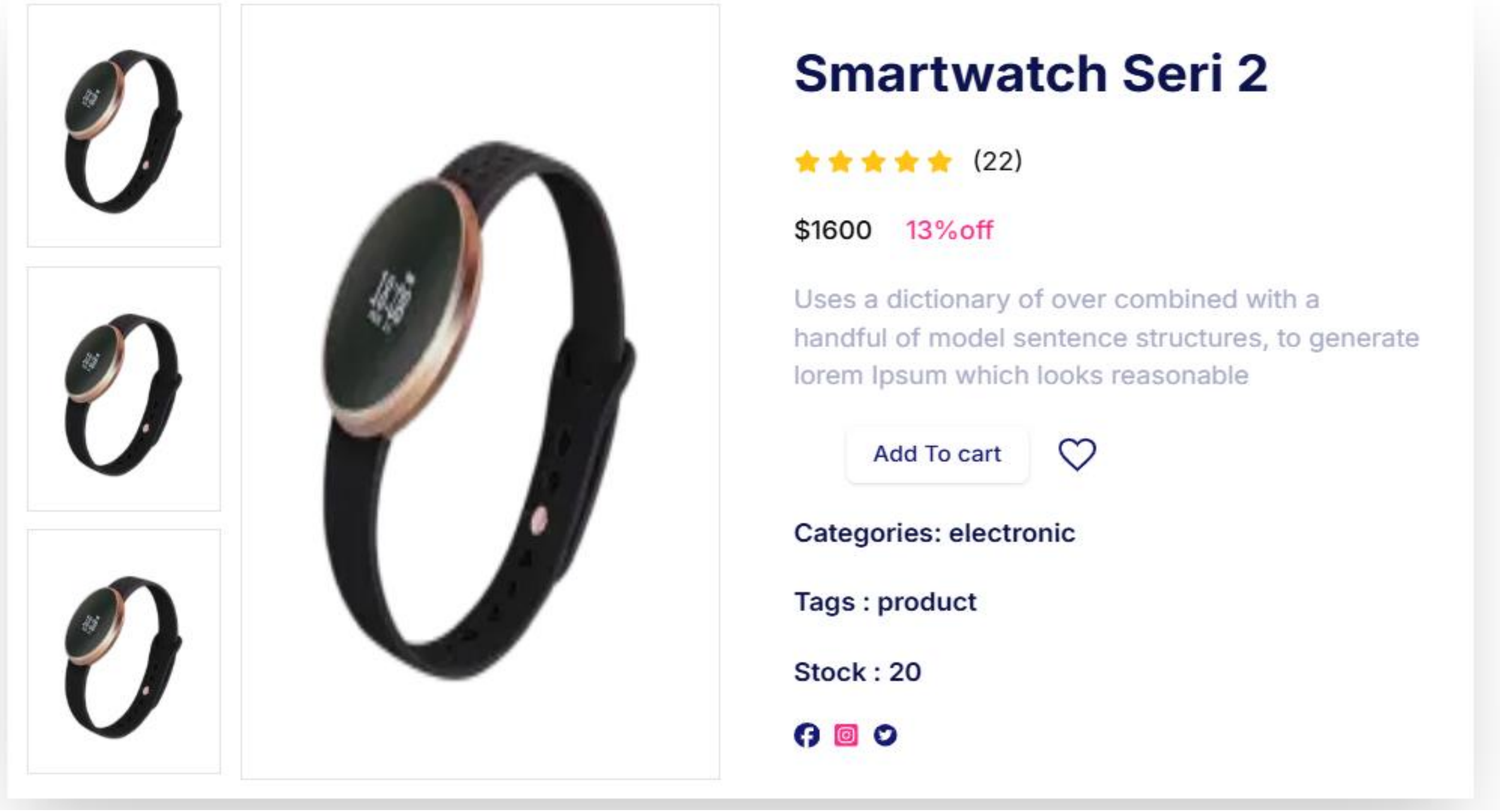
More details

→ Aliquam dis vulputate vulputate integer sagittis. Faucibus ds diam arcu, nulla lobortis justo netus dis. Eu in fringilla vulputate nunc nec. Dui, massa viverr .

→ Aliquam dis vulputate vulputate integer sagittis. Faucibus ds diam arcu, nulla lobortis justo netus dis. Eu in fringilla vulputate nunc nec. Dui, massa viverr .

→ Aliquam dis vulputate vulputate integer sagittis. Faucibus ds diam arcu, nulla lobortis justo netus dis. Eu in fringilla vulputate nunc nec. Dui, massa viverr .

→ Aliquam dis vulputate vulputate integer sagittis. Faucibus ds diam arcu, nulla lobortis justo netus dis. Eu in fringilla vulputate nunc nec. Dui, massa viverr .



Smartwatch Seri 2

★★★★★ (22)

$1600 13%off

Uses a dictionary of over combined with a handful of model sentence structures, to generate lorem Ipsum which looks reasonable

Categories: electronic

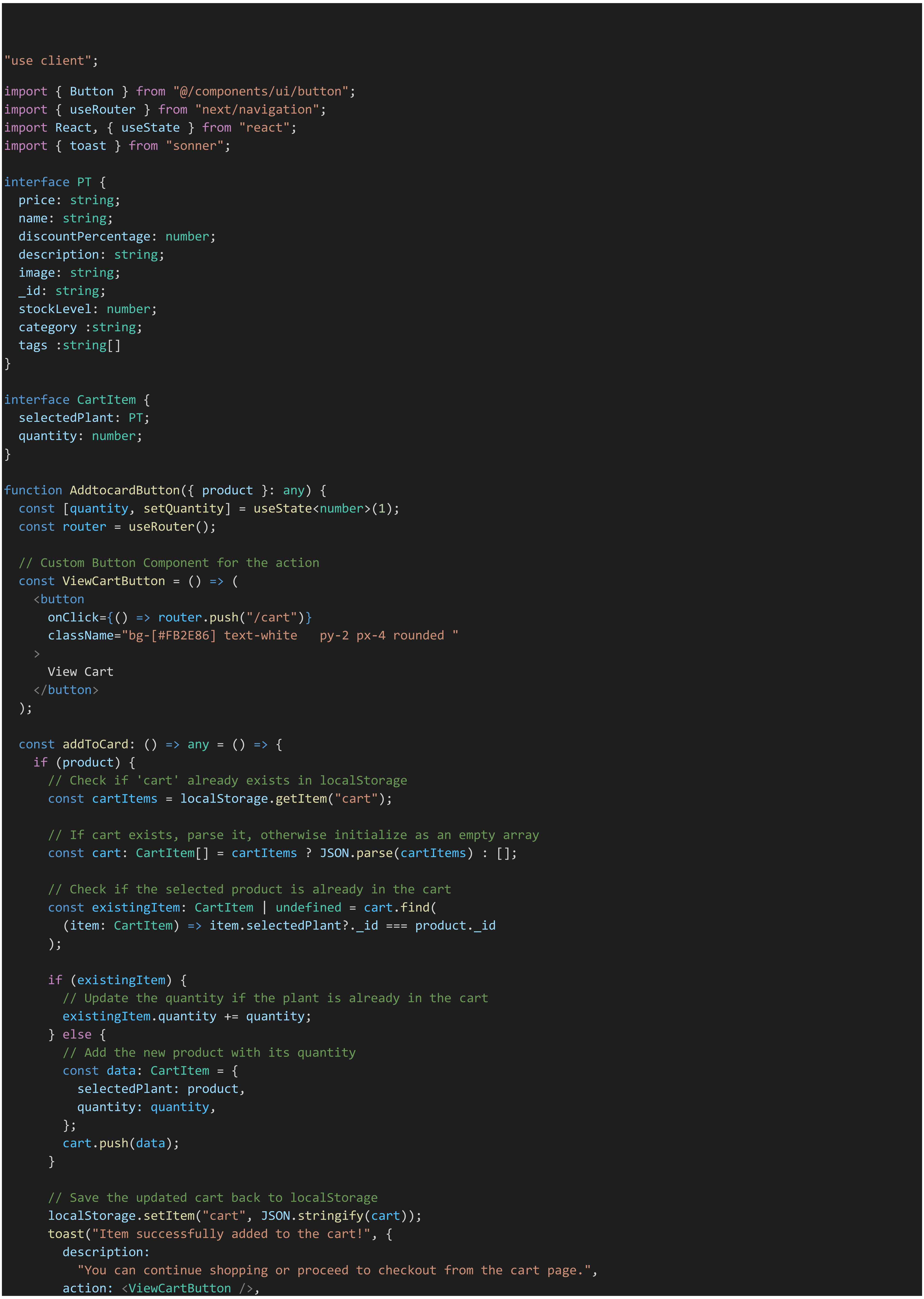
Tags : product

Stock : 20

- ✓ Cart Route

  https://ui-ux-hackathon-sable.vercel.app/cart

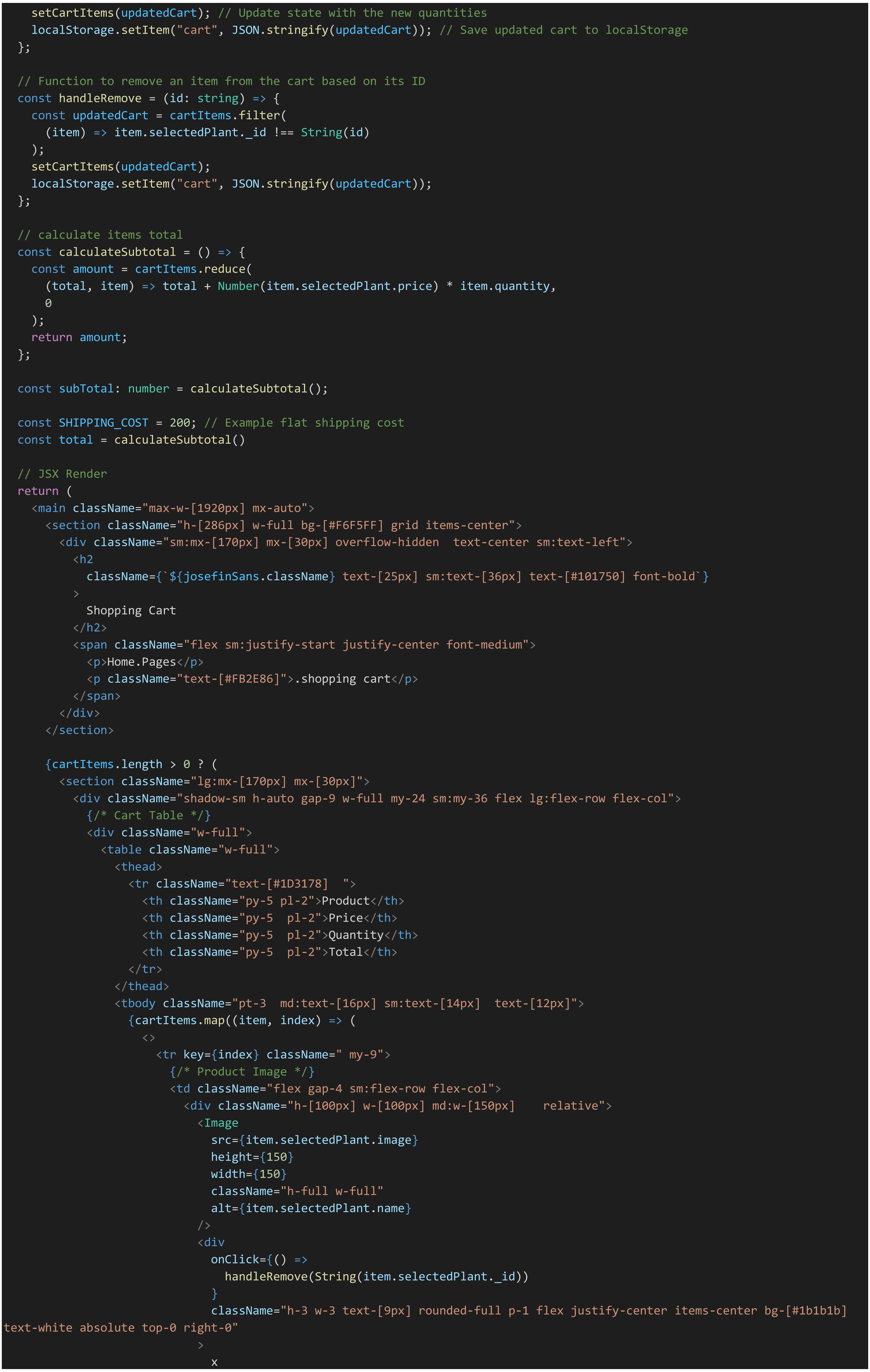
I Use localStorage for display cart items see here logic of send product in localStorage addToCartComponent



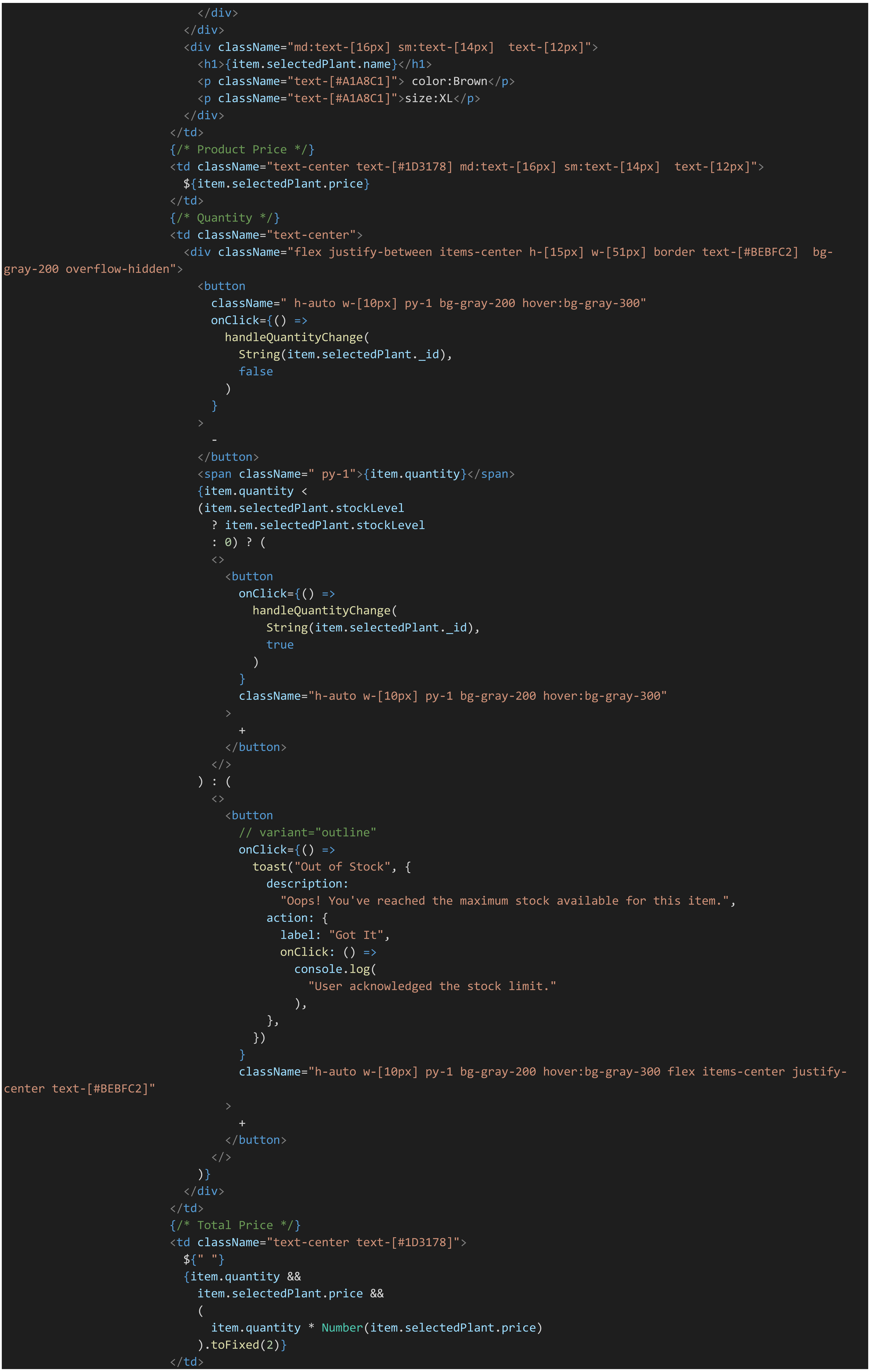
```
"use client";

import { Button } from "@/components/ui/button";
import { useRouter } from "next/navigation";
import React, { useState } from "react";
import { toast } from "sonner";

interface PT {
  price: string;
  name: string;
  discountPercentage: number;
  description: string;
  image: string;
  _id: string;
  stockLevel: number;
  category :string;
  tags :string[]
}

interface CartItem {
  selectedPlant: PT;
  quantity: number;
}

function AddtocardButton({ product }: any) {
  const [quantity, setQuantity] = useState<number>(1);
  const router = useRouter();

  // Custom Button Component for the action
  const ViewCartButton = () => (
    <button
      onClick={() => router.push("/cart")}
      className="bg-[#FB2E86] text-white   py-2 px-4 rounded "
    >
      View Cart
    </button>
  );

  const addToCard: () => any = () => {
    if (product) {
      // Check if 'cart' already exists in localStorage
      const cartItems = localStorage.getItem("cart");

      // If cart exists, parse it, otherwise initialize as an empty array
      const cart: CartItem[] = cartItems ? JSON.parse(cartItems) : [];

      // Check if the selected product is already in the cart
      const existingItem: CartItem | undefined = cart.find(
        (item: CartItem) => item.selectedPlant?._id === product._id
      );

      if (existingItem) {
        // Update the quantity if the plant is already in the cart
        existingItem.quantity += quantity;
      } else {
        // Add the new product with its quantity
        const data: CartItem = {
          selectedPlant: product,
          quantity: quantity,
        };
        cart.push(data);
      }

      // Save the updated cart back to localStorage
      localStorage.setItem("cart", JSON.stringify(cart));
      toast("Item successfully added to the cart!", {
        description:
          "You can continue shopping or proceed to checkout from the cart page.",
        action: <ViewCartButton />,
      });
    }
  };

  return (
    <Button
      onClick={() => addToCard()}
      className="bg-inherit  hover:text-white hover:bg-[#FB2E86] text-[#151875]"
    >
      Add To cart
    </Button>
  );
}

export default AddtocardButton;
```

in this cart page I get localStorage data and display see code here

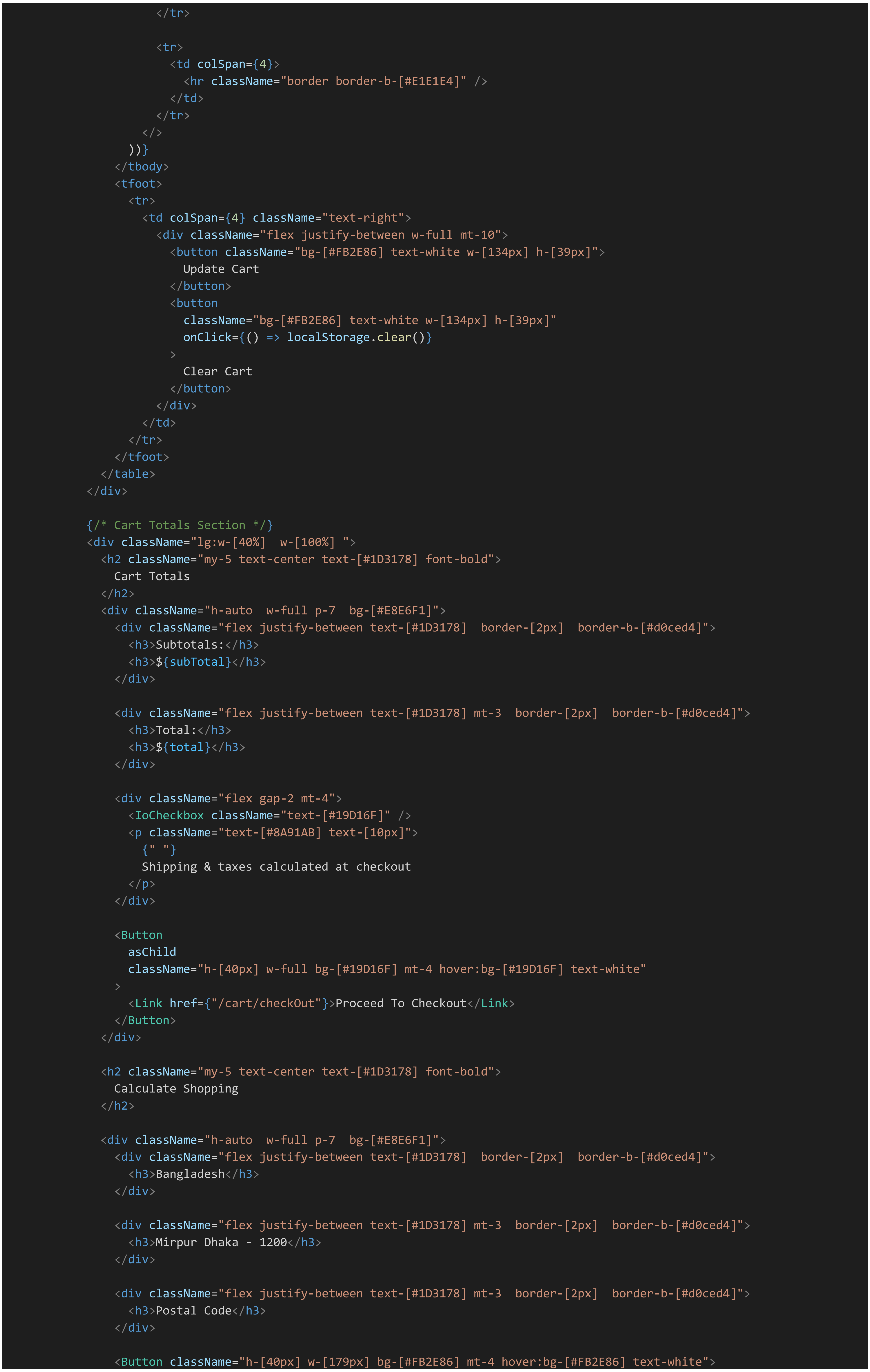
```
"use client";


import React, { useEffect, useState } from "react";
import { Josefin_Sans } from "next/font/google";
import Image from "next/image";
import { IoCheckbox } from "react-icons/io5";
import { Button } from "@/components/ui/button";
import Link from "next/link";

import { toast } from "sonner";

const josefinSans = Josefin_Sans({
  subsets: ["latin"],
  weight: ["400", "700"],
});


interface PT {
  price: string;
  name: string;
  discountPercentage: number;
  description: string;
  image: string;
  _id: string;
  stockLevel: number;
  category :string;
  tags :string[]
}


interface CartItem {
  selectedPlant: PT;
  quantity: number;
}

function Cart() {
  // State to store the cart items
  const [cartItems, setCartItems] = useState<CartItem[]>([]);

  // Get data from localStorage and save it in the state
  useEffect(() => {
    const dataFromLocalStorage = localStorage.getItem("cart");
    if (dataFromLocalStorage) {
      setCartItems(JSON.parse(dataFromLocalStorage));
    } else {
      setCartItems([]);
    }
  }, []);

  // Function to increase or decrease the quantity of a specific plant
  const handleQuantityChange = (id: string, increment: boolean) => {
    const updatedCart = cartItems.map((item) =>
      item.selectedPlant._id === String(id)
        ? {
            ...item,
            quantity: increment
              ? Math.min(item.quantity + 1, item.selectedPlant?.stockLevel ?? 0)
              : Math.max(1, item.quantity - 1),
          }
        : item
    );
```



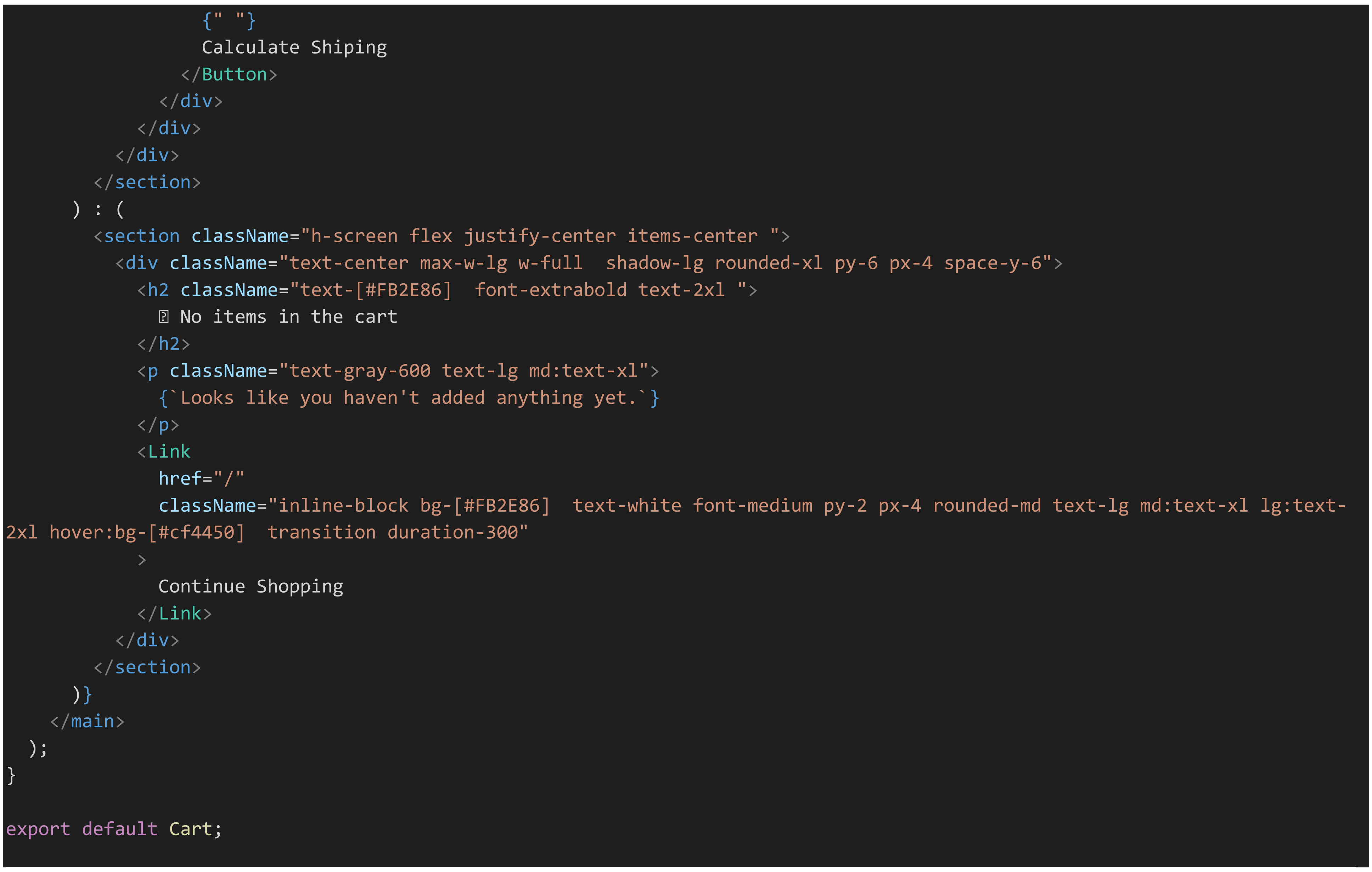
```
    setCartItems(updatedCart); // Update state with the new quantities
    localStorage.setItem("cart", JSON.stringify(updatedCart)); // Save updated cart to localStorage
  };

  // Function to remove an item from the cart based on its ID
  const handleRemove = (id: string) => {
    const updatedCart = cartItems.filter(
      (item) => item.selectedPlant._id !== String(id)
    );
    setCartItems(updatedCart);
    localStorage.setItem("cart", JSON.stringify(updatedCart));
  };

  // calculate items total
  const calculateSubtotal = () => {
    const amount = cartItems.reduce(
      (total, item) => total + Number(item.selectedPlant.price) * item.quantity,
      0
    );
    return amount;
  };

  const subTotal: number = calculateSubtotal();

  const SHIPPING_COST = 200; // Example flat shipping cost
  const total = calculateSubtotal()

  // JSX Render
  return (
    <main className="max-w-[1920px] mx-auto">
      <section className="h-[286px] w-full bg-[#F6F5FF] grid items-center">
        <div className="sm:mx-[170px] mx-[30px] overflow-hidden  text-center sm:text-left">
          <h2
            className={`${josefinSans.className} text-[25px] sm:text-[36px] text-[#101750] font-bold`}
          >
            Shopping Cart
          </h2>
          <span className="flex sm:justify-start justify-center font-medium">
            <p>Home.Pages</p>
            <p className="text-[#FB2E86]">.shopping cart</p>
          </span>
        </div>
      </section>

      {cartItems.length > 0 ? (
        <section className="lg:mx-[170px] mx-[30px]">
          <div className="shadow-sm h-auto gap-9 w-full my-24 sm:my-36 flex lg:flex-row flex-col">
            {/* Cart Table */}
            <div className="w-full">
              <table className="w-full">
                <thead>
                  <tr className="text-[#1D3178]  ">
                    <th className="py-5 pl-2">Product</th>
                    <th className="py-5  pl-2">Price</th>
                    <th className="py-5  pl-2">Quantity</th>
                    <th className="py-5  pl-2">Total</th>
                  </tr>
                </thead>
                <tbody className="pt-3  md:text-[16px] sm:text-[14px]  text-[12px]">
                  {cartItems.map((item, index) => (
                    <>
                      <tr key={index} className=" my-9">
                        {/* Product Image */}
                        <td className="flex gap-4 sm:flex-row flex-col">
                          <div className="h-[100px] w-[100px] md:w-[150px]    relative">
                            <Image
                              src={item.selectedPlant.image}
                              height={150}
                              width={150}
                              className="h-full w-full"
                              alt={item.selectedPlant.name}
                            />
                            <div
                              onClick={() =>
                                handleRemove(String(item.selectedPlant._id))
                              }
                              className="h-3 w-3 text-[9px] rounded-full p-1 flex justify-center items-center bg-[#1b1b1b]
text-white absolute top-0 right-0"
                            >
                              x
```



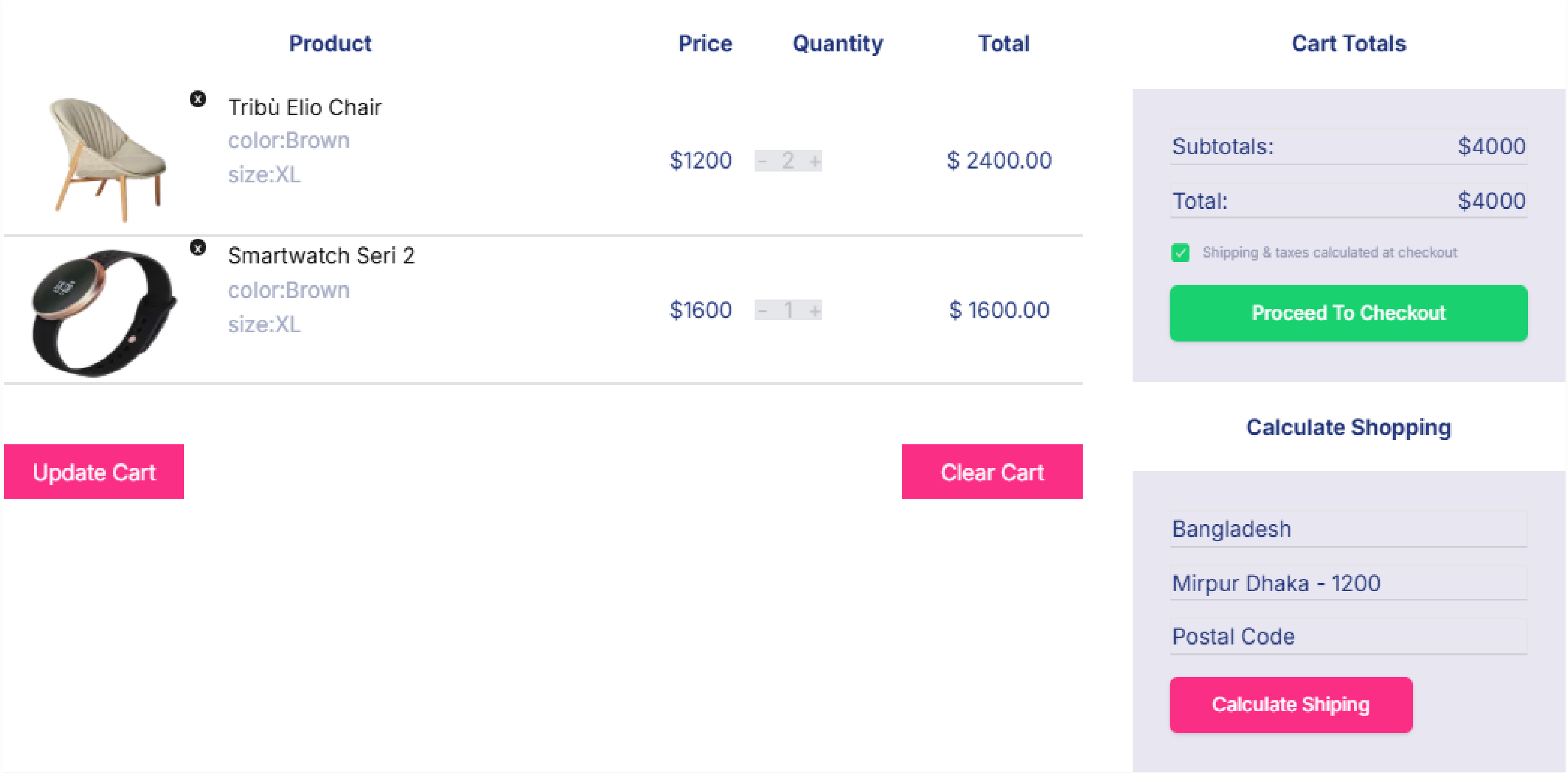
```
                            </div>
                          </div>
                          <div className="md:text-[16px] sm:text-[14px]  text-[12px]">
                            <h1>{item.selectedPlant.name}</h1>
                            <p className="text-[#A1A8C1]"> color:Brown</p>
                            <p className="text-[#A1A8C1]">size:XL</p>
                          </div>
                        </td>
                        {/* Product Price */}
                        <td className="text-center text-[#1D3178] md:text-[16px] sm:text-[14px]  text-[12px]">
                          ${item.selectedPlant.price}
                        </td>
                        {/* Quantity */}
                        <td className="text-center">
                          <div className="flex justify-between items-center h-[15px] w-[51px] border text-[#BEBFC2]  bg-gray-200 overflow-hidden">
                            <button
                              className=" h-auto w-[10px] py-1 bg-gray-200 hover:bg-gray-300"
                              onClick={() =>
                                handleQuantityChange(
                                  String(item.selectedPlant._id),
                                  false
                                )
                              }
                            >
                              -
                            </button>
                            <span className=" py-1">{item.quantity}</span>
                            {item.quantity <
                            (item.selectedPlant.stockLevel
                              ? item.selectedPlant.stockLevel
                              : 0) ? (
                              <>
                                <button
                                  onClick={() =>
                                    handleQuantityChange(
                                      String(item.selectedPlant._id),
                                      true
                                    )
                                  }
                                  className="h-auto w-[10px] py-1 bg-gray-200 hover:bg-gray-300"
                                >
                                  +
                                </button>
                              </>
                            ) : (
                              <>
                                <button
                                  // variant="outline"
                                  onClick={() =>
                                    toast("Out of Stock", {
                                      description:
                                        "Oops! You've reached the maximum stock available for this item.",
                                      action: {
                                        label: "Got It",
                                        onClick: () =>
                                          console.log(
                                            "User acknowledged the stock limit."
                                          ),
                                      },
                                    })
                                  }
                                  className="h-auto w-[10px] py-1 bg-gray-200 hover:bg-gray-300 flex items-center justify-center text-[#BEBFC2]"

                                >
                                  +
                                </button>
                              </>
                            )}
                          </div>
                        </td>
                        {/* Total Price */}
                        <td className="text-center text-[#1D3178]">
                          ${" "}
                          {item.quantity &&
                            item.selectedPlant.price &&
                            (
                              item.quantity * Number(item.selectedPlant.price)
                            ).toFixed(2)}
                        </td>
```



```
                      </tr>

                      <tr>
                        <td colSpan={4}>
                          <hr className="border border-b-[#E1E1E4]" />
                        </td>
                      </tr>
                    </>
                  ))}
                </tbody>
                <tfoot>
                  <tr>
                    <td colSpan={4} className="text-right">
                      <div className="flex justify-between w-full mt-10">
                        <button className="bg-[#FB2E86] text-white w-[134px] h-[39px]">
                          Update Cart
                        </button>
                        <button
                          className="bg-[#FB2E86] text-white w-[134px] h-[39px]"
                          onClick={() => localStorage.clear()}
                        >
                          Clear Cart
                        </button>
                      </div>
                    </td>
                  </tr>
                </tfoot>
              </table>
            </div>

            {/* Cart Totals Section */}
            <div className="lg:w-[40%]  w-[100%] ">
              <h2 className="my-5 text-center text-[#1D3178] font-bold">
                Cart Totals
              </h2>
              <div className="h-auto  w-full p-7  bg-[#E8E6F1]">
                <div className="flex justify-between text-[#1D3178]  border-[2px]  border-b-[#d0ced4]">
                  <h3>Subtotals:</h3>
                  <h3>${subTotal}</h3>
                </div>

                <div className="flex justify-between text-[#1D3178] mt-3  border-[2px]  border-b-[#d0ced4]">
                  <h3>Total:</h3>
                  <h3>${total}</h3>
                </div>

                <div className="flex gap-2 mt-4">
                  <IoCheckbox className="text-[#19D16F]" />
                  <p className="text-[#8A91AB] text-[10px]">
                    {" "}
                    Shipping & taxes calculated at checkout
                  </p>
                </div>

                <Button
                  asChild
                  className="h-[40px] w-full bg-[#19D16F] mt-4 hover:bg-[#19D16F] text-white"
                >
                  <Link href={"/cart/checkOut"}>Proceed To Checkout</Link>
                </Button>
              </div>

              <h2 className="my-5 text-center text-[#1D3178] font-bold">
                Calculate Shopping
              </h2>

              <div className="h-auto  w-full p-7  bg-[#E8E6F1]">
                <div className="flex justify-between text-[#1D3178]  border-[2px]  border-b-[#d0ced4]">
                  <h3>Bangladesh</h3>
                </div>

                <div className="flex justify-between text-[#1D3178] mt-3  border-[2px]  border-b-[#d0ced4]">
                  <h3>Mirpur Dhaka - 1200</h3>
                </div>

                <div className="flex justify-between text-[#1D3178] mt-3  border-[2px]  border-b-[#d0ced4]">
                  <h3>Postal Code</h3>
                </div>

                <Button className="h-[40px] w-[179px] bg-[#FB2E86] mt-4 hover:bg-[#FB2E86] text-white">
```



```
                  {" "}
                  Calculate Shiping
                </Button>
              </div>
            </div>
          </div>
        </section>
      ) : (
        <section className="h-screen flex justify-center items-center ">
          <div className="text-center max-w-lg w-full  shadow-lg rounded-xl py-6 px-4 space-y-6">
            <h2 className="text-[#FB2E86]  font-extrabold text-2xl ">
              □ No items in the cart
            </h2>
            <p className="text-gray-600 text-lg md:text-xl">
              {`Looks like you haven't added anything yet.`}
            </p>
            <Link
              href="/"
              className="inline-block bg-[#FB2E86]  text-white font-medium py-2 px-4 rounded-md text-lg md:text-xl lg:text-
2xl hover:bg-[#cf4450]  transition duration-300"
            >
              Continue Shopping
            </Link>
          </div>
        </section>
      )}
    </main>
  );
}

export default Cart;
```



Product		Price	Quantity	Total
	Tribù Elio Chair color:Brown size:XL	$1200	- 2 +	$ 2400.00
	Smartwatch Seri 2 color:Brown size:XL	$1600	- 1 +	$ 1600.00

Update Cart

Clear Cart

Cart Totals

Subtotals:	$4000
Total:	$4000

☑ Shipping & taxes calculated at checkout

Proceed To Checkout

Calculate Shopping

Bangladesh

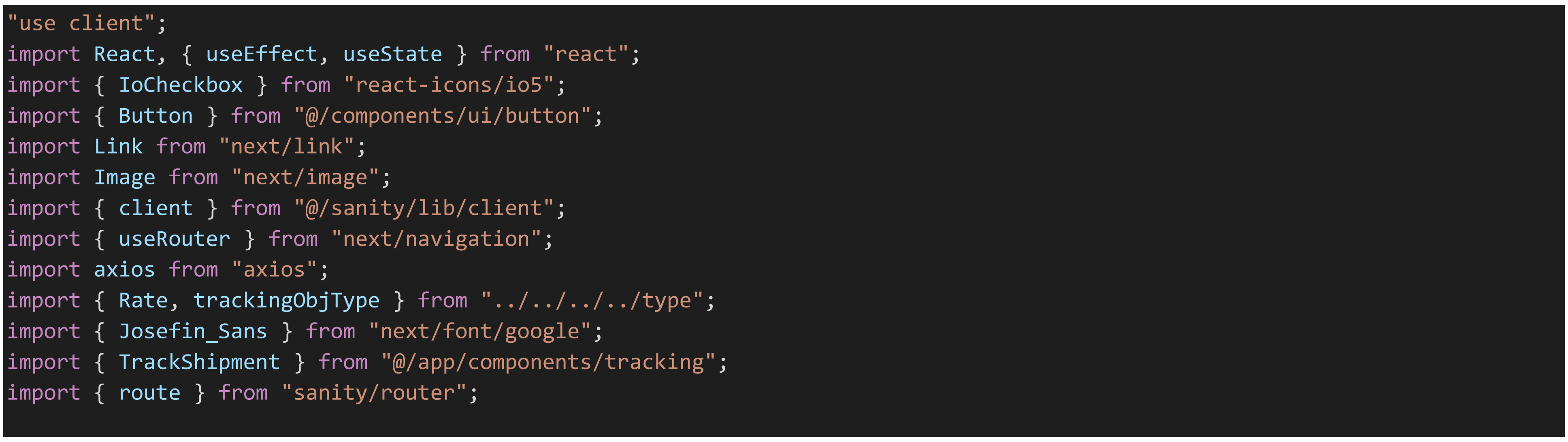
Mirpur Dhaka - 1200

Postal Code

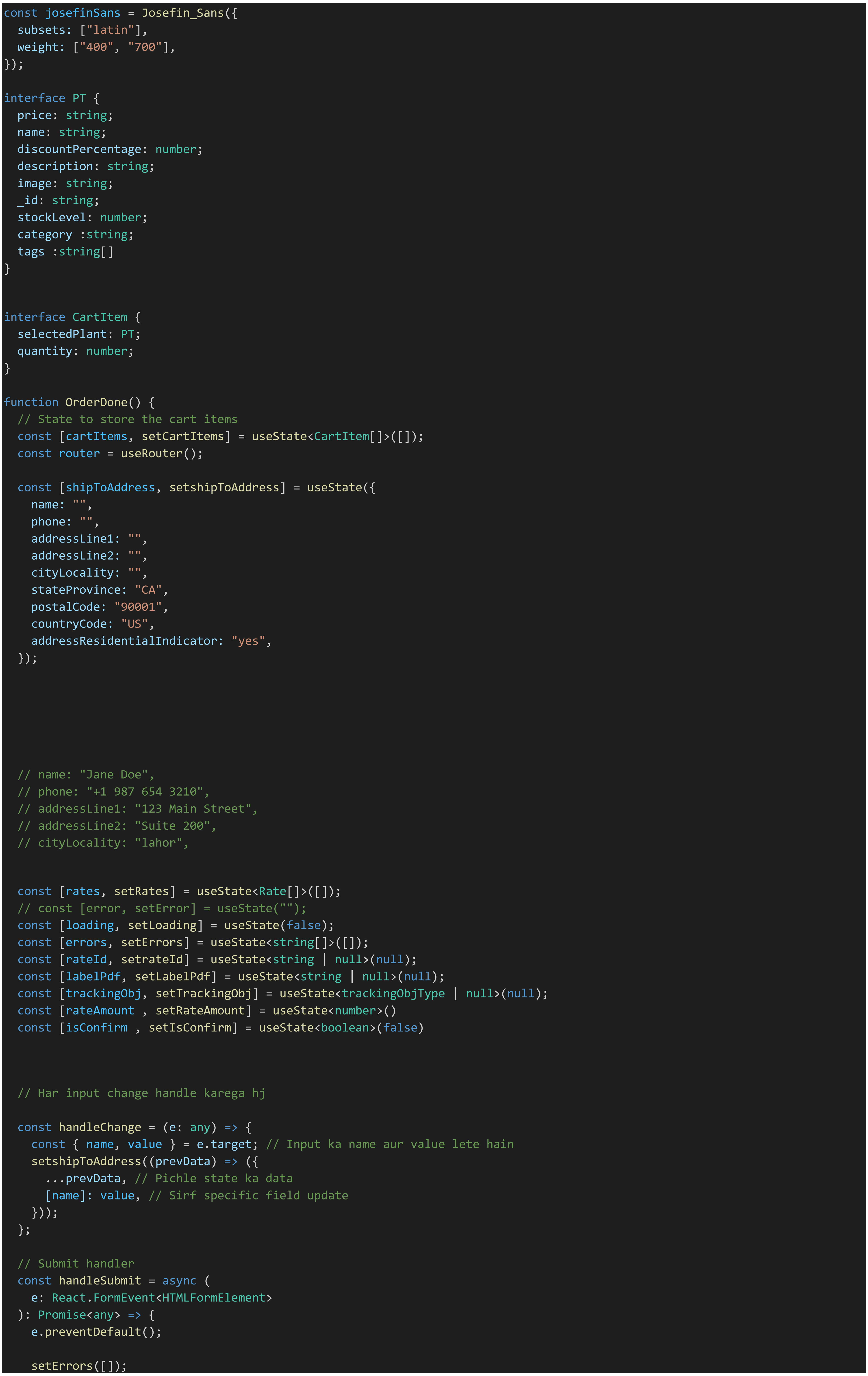
Calculate Shiping

✓ CheckOut Page here I applay all shipment logic and price and checkOut

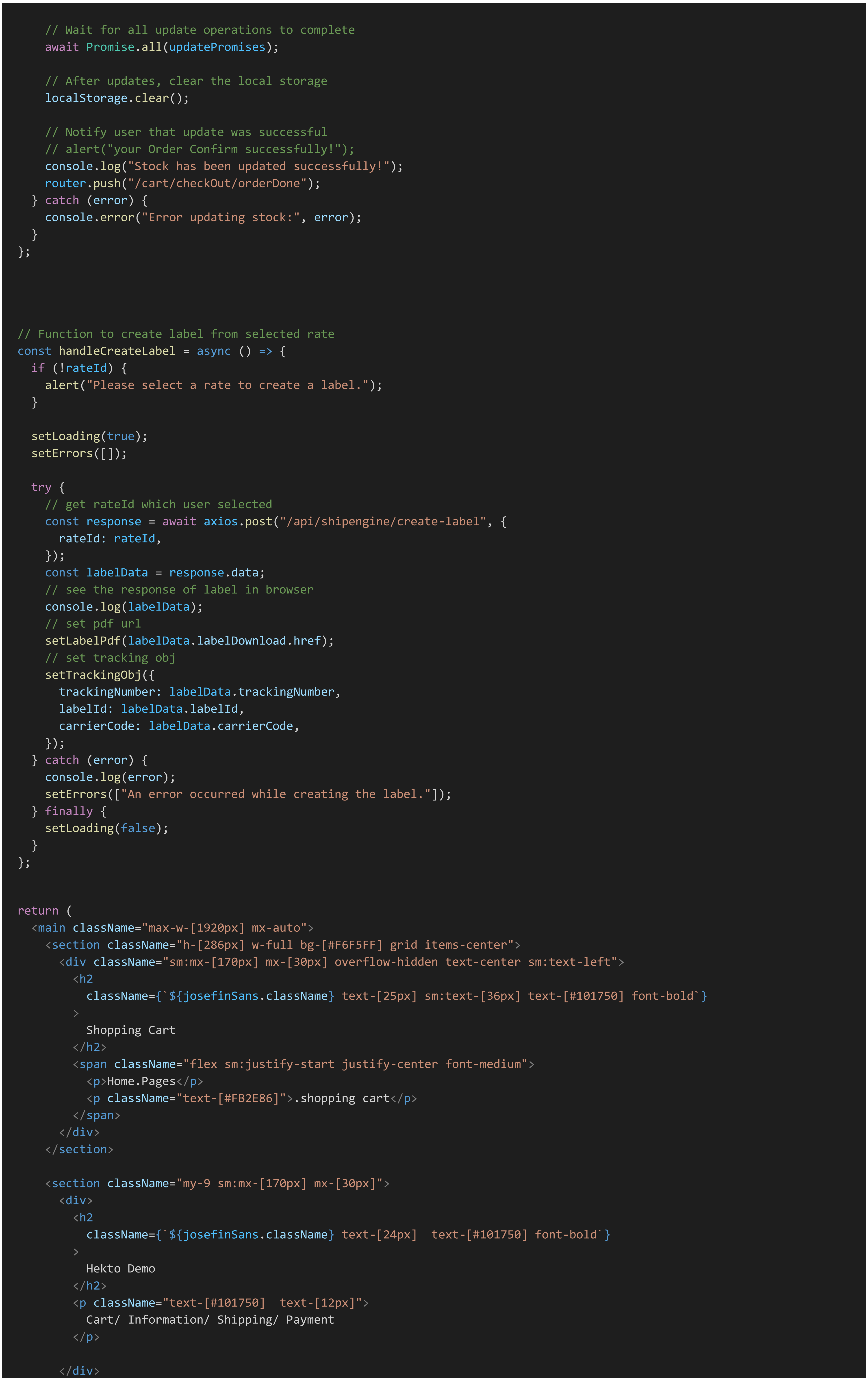
https://ui-ux-hackathon-sable.vercel.app/cart/checkOut



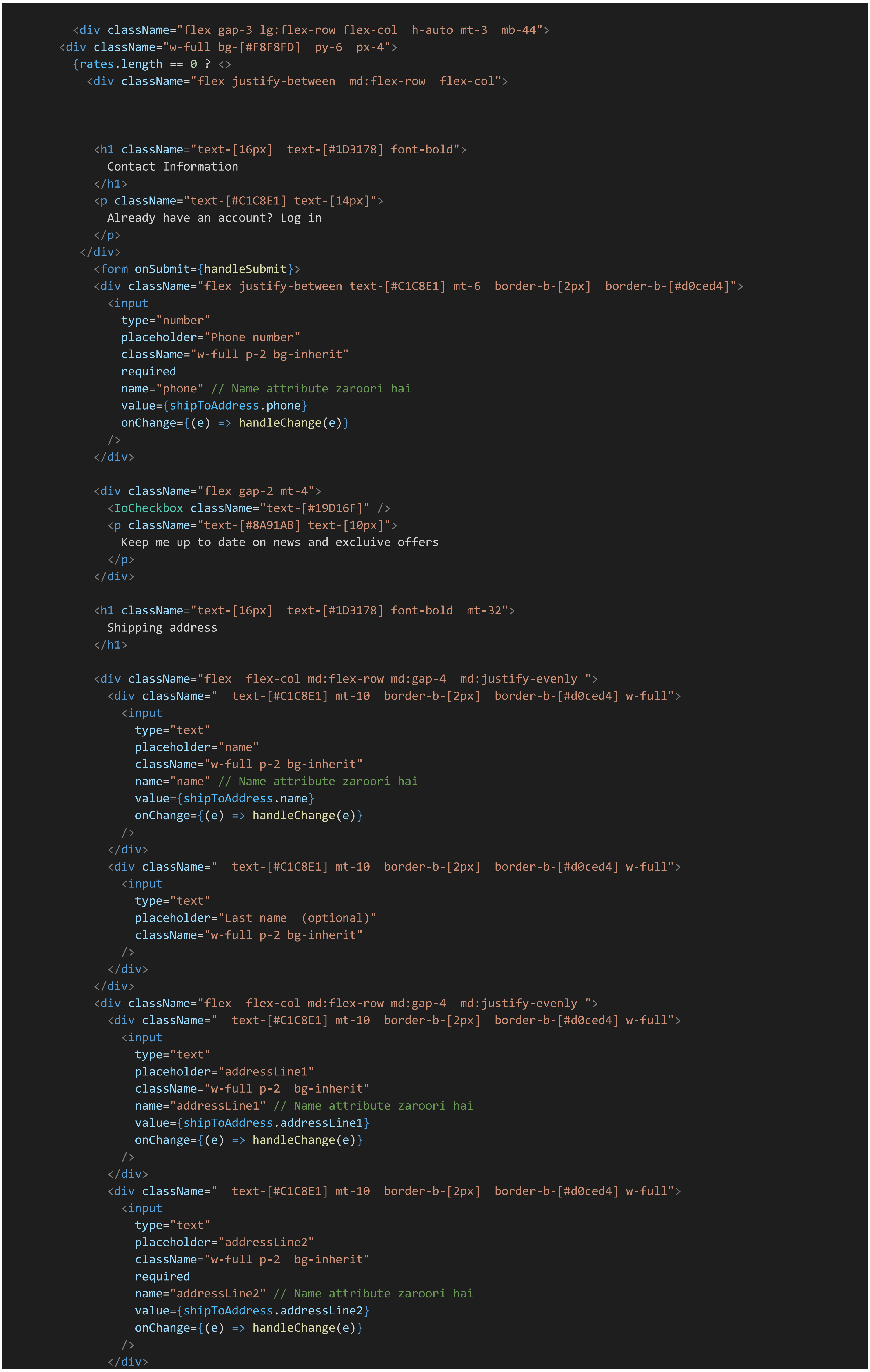
```
"use client";
import React, { useEffect, useState } from "react";
import { IoCheckbox } from "react-icons/io5";
import { Button } from "@/components/ui/button";
import Link from "next/link";
import Image from "next/image";
import { client } from "@/sanity/lib/client";
import { useRouter } from "next/navigation";
import axios from "axios";
import { Rate, trackingObjType } from "../../../../type";
import { Josefin_Sans } from "next/font/google";
import { TrackShipment } from "@/app/components/tracking";
import { route } from "sanity/router";
```



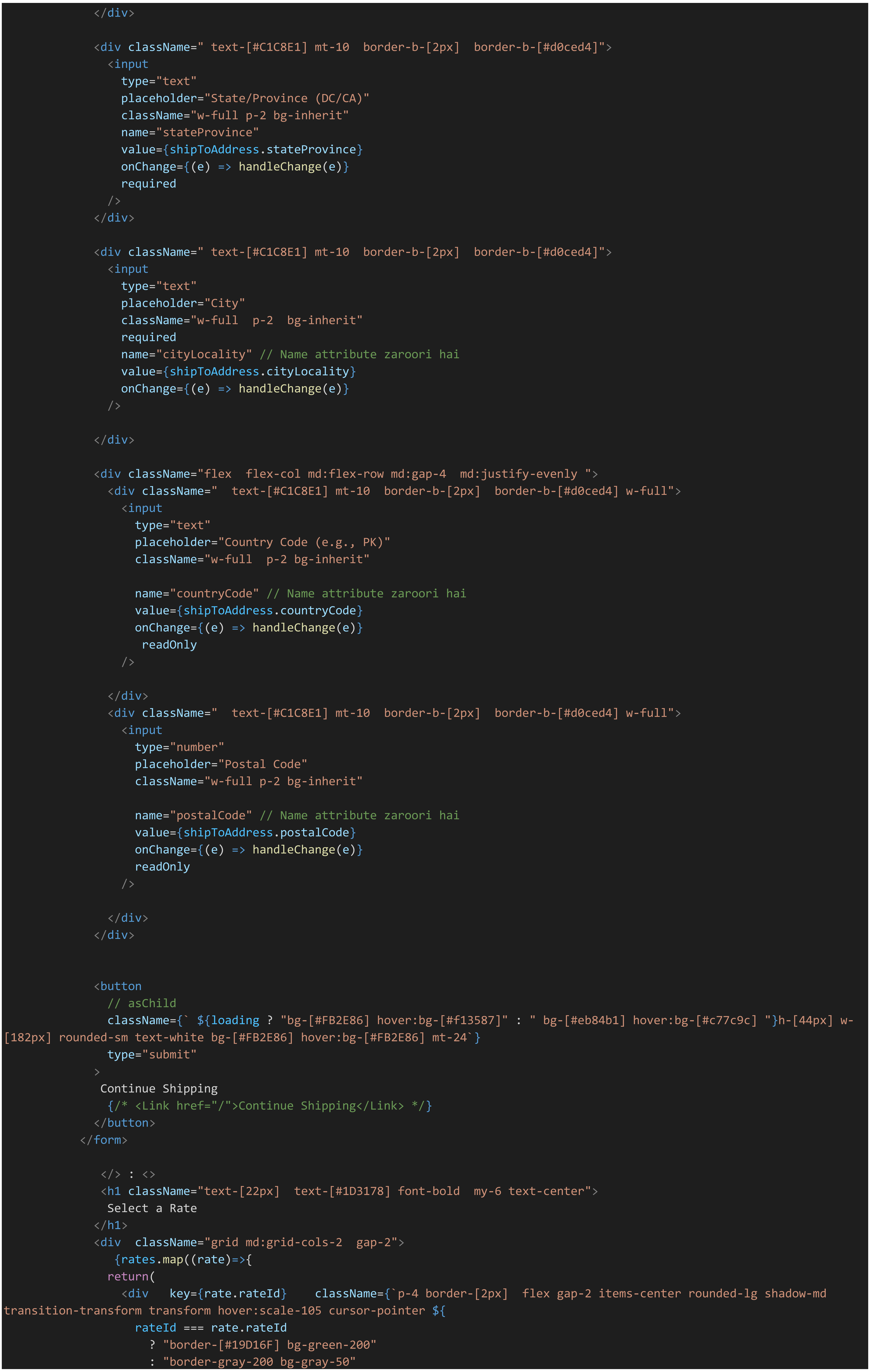
```
const josefinSans = Josefin_Sans({
  subsets: ["latin"],
  weight: ["400", "700"],
});

interface PT {
  price: string;
  name: string;
  discountPercentage: number;
  description: string;
  image: string;
  _id: string;
  stockLevel: number;
  category :string;
  tags :string[]
}


interface CartItem {
  selectedPlant: PT;
  quantity: number;
}

function OrderDone() {
  // State to store the cart items
  const [cartItems, setCartItems] = useState<CartItem[]>([]);
  const router = useRouter();

  const [shipToAddress, setshipToAddress] = useState({
    name: "",
    phone: "",
    addressLine1: "",
    addressLine2: "",
    cityLocality: "",
    stateProvince: "CA",
    postalCode: "90001",
    countryCode: "US",
    addressResidentialIndicator: "yes",
  });






  // name: "Jane Doe",
  // phone: "+1 987 654 3210",
  // addressLine1: "123 Main Street",
  // addressLine2: "Suite 200",
  // cityLocality: "lahor",


  const [rates, setRates] = useState<Rate[]>([]);
  // const [error, setError] = useState("");
  const [loading, setLoading] = useState(false);
  const [errors, setErrors] = useState<string[]>([]);
  const [rateId, setrateId] = useState<string | null>(null);
  const [labelPdf, setLabelPdf] = useState<string | null>(null);
  const [trackingObj, setTrackingObj] = useState<trackingObjType | null>(null);
  const [rateAmount , setRateAmount] = useState<number>()
  const [isConfirm , setIsConfirm] = useState<boolean>(false)



  // Har input change handle karega hj

  const handleChange = (e: any) => {
    const { name, value } = e.target; // Input ka name aur value lete hain
    setshipToAddress((prevData) => ({
      ...prevData, // Pichle state ka data
      [name]: value, // Sirf specific field update
    }));
  };

  // Submit handler
  const handleSubmit = async (
    e: React.FormEvent<HTMLFormElement>
  ): Promise<any> => {
    e.preventDefault();

    setErrors([]);
    setRates([]);

    try {
      const response = await axios.post("/api/shipengine/get-rates", {
        shipToAddress,
        // map the cart products which can be shipped and use only weight and dimensions

        packages:  [
          {
            weight: {
              value: 2,
              unit: "pound"
            },
            dimensions: {
              length: 10,
              width: 5,
              height: 8,
              unit: "inch"
            }
          }

        ]
      });
      // see the response in browser
      console.log(response.data);
      setLoading(true);

      // Update the state with the fetched rates
      setRates(response.data.rateResponse.rates);
      console.log("rates................." , rates)
    } catch (error) {
      console.log(error);
      setErrors(["An error occurred while fetching rates."]);
    } finally {
      setLoading(false);
    }


  };

  // Get data from localStorage and save it in the state
  useEffect(() => {
    const dataFromLocalStorage = localStorage.getItem("cart");
    if (dataFromLocalStorage) {
      setCartItems(JSON.parse(dataFromLocalStorage));
    } else {
      setCartItems([]);
    }
  }, []);

  const calculateSubtotal: () => number = () => {
    return cartItems.reduce(
      (total, item) =>
        total + Number(item.selectedPlant.price) * Number(item.quantity),
      0
    );
  };

  const subTotal: number = calculateSubtotal();

  const SHIPPING_COST : number | undefined = rateAmount ? rateAmount : 0; // Example flat shipping cost
  const total = calculateSubtotal() + SHIPPING_COST;

  const handleUpdate = async () => {
    try {
      const updatePromises = cartItems.map(async (item) => {
        if (
          item.selectedPlant.stockLevel &&
          item.quantity &&
          item.selectedPlant._id
        ) {
          // Ensure stock and quantity are numbers before updating
          await client
            .patch(item.selectedPlant._id)
            .set({
              stockLevel: item.selectedPlant.stockLevel - item.quantity,
            })
            .commit();
        }
      });
```



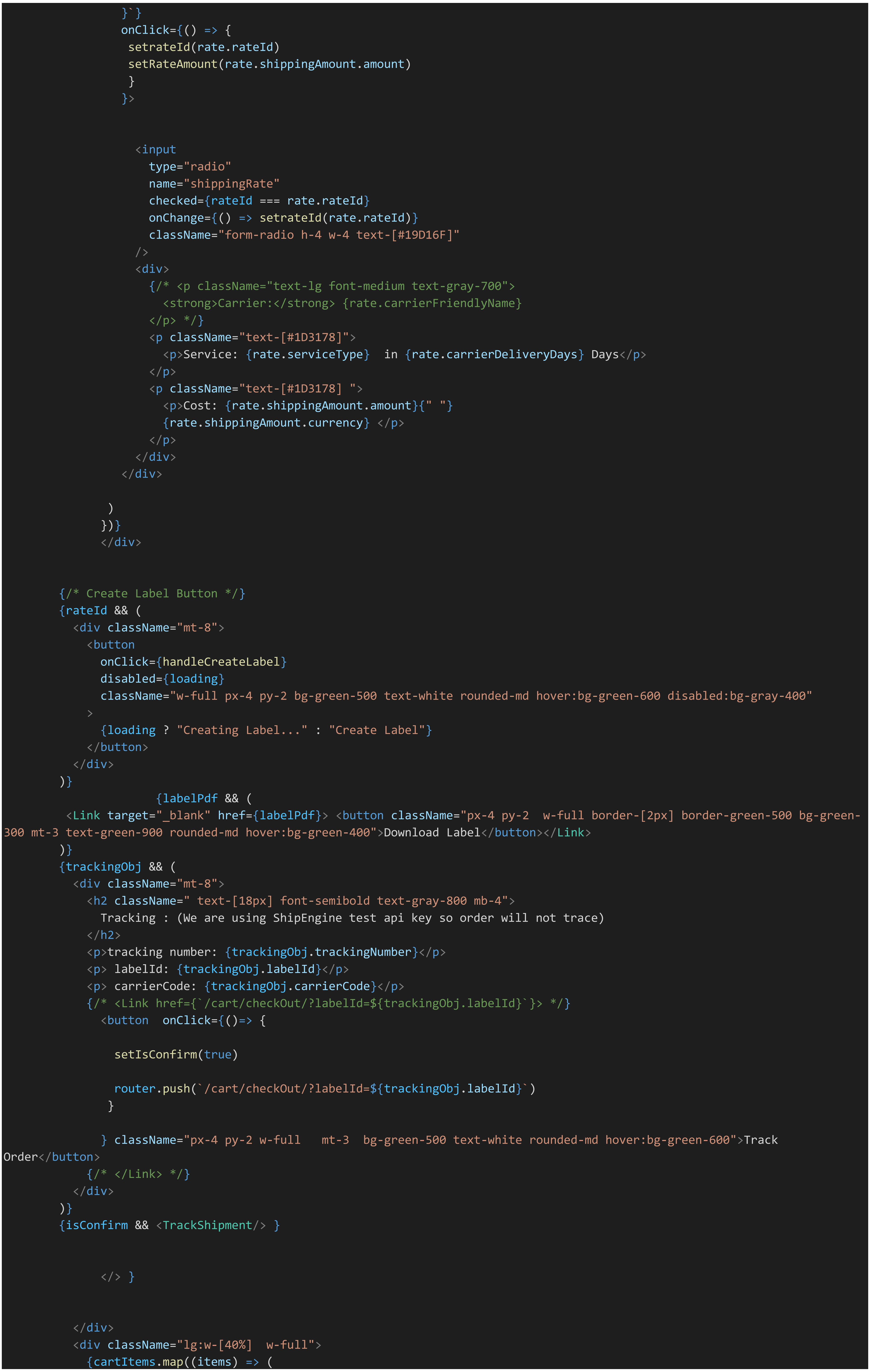
```
    // Wait for all update operations to complete
    await Promise.all(updatePromises);

    // After updates, clear the local storage
    localStorage.clear();

    // Notify user that update was successful
    // alert("your Order Confirm successfully!");
    console.log("Stock has been updated successfully!");
    router.push("/cart/checkOut/orderDone");
  } catch (error) {
    console.error("Error updating stock:", error);
  }
};




// Function to create label from selected rate
const handleCreateLabel = async () => {
  if (!rateId) {
    alert("Please select a rate to create a label.");
  }

  setLoading(true);
  setErrors([]);

  try {
    // get rateId which user selected
    const response = await axios.post("/api/shipengine/create-label", {
      rateId: rateId,
    });
    const labelData = response.data;
    // see the response of label in browser
    console.log(labelData);
    // set pdf url
    setLabelPdf(labelData.labelDownload.href);
    // set tracking obj
    setTrackingObj({
      trackingNumber: labelData.trackingNumber,
      labelId: labelData.labelId,
      carrierCode: labelData.carrierCode,
    });
  } catch (error) {
    console.log(error);
    setErrors(["An error occurred while creating the label."]);
  } finally {
    setLoading(false);
  }
};


return (
  <main className="max-w-[1920px] mx-auto">
    <section className="h-[286px] w-full bg-[#F6F5FF] grid items-center">
      <div className="sm:mx-[170px] mx-[30px] overflow-hidden text-center sm:text-left">
        <h2
          className={`${josefinSans.className} text-[25px] sm:text-[36px] text-[#101750] font-bold`}
        >
          Shopping Cart
        </h2>
        <span className="flex sm:justify-start justify-center font-medium">
          <p>Home.Pages</p>
          <p className="text-[#FB2E86]">.shopping cart</p>
        </span>
      </div>
    </section>

    <section className="my-9 sm:mx-[170px] mx-[30px]">
      <div>
        <h2
          className={`${josefinSans.className} text-[24px]  text-[#101750] font-bold`}
        >
          Hekto Demo
        </h2>
        <p className="text-[#101750]  text-[12px]">
          Cart/ Information/ Shipping/ Payment
        </p>

      </div>
```



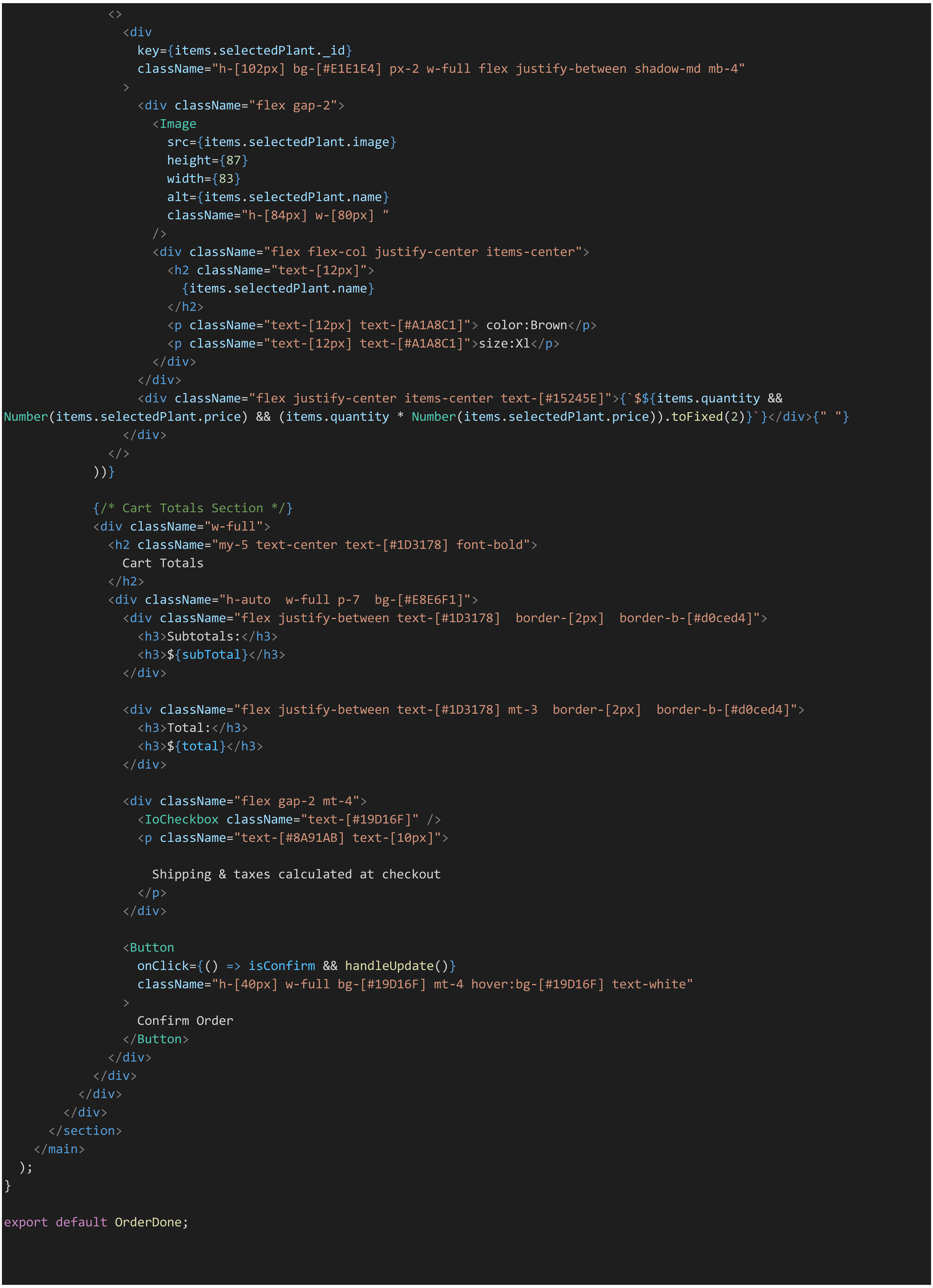
```
  <div className="flex gap-3 lg:flex-row flex-col  h-auto mt-3  mb-44">
<div className="w-full bg-[#F8F8FD]  py-6  px-4">
  {rates.length == 0 ? <>
    <div className="flex justify-between  md:flex-row  flex-col">



     <h1 className="text-[16px]  text-[#1D3178] font-bold">
       Contact Information
     </h1>
     <p className="text-[#C1C8E1] text-[14px]">
       Already have an account? Log in
     </p>
   </div>
     <form onSubmit={handleSubmit}>
     <div className="flex justify-between text-[#C1C8E1] mt-6  border-b-[2px]  border-b-[#d0ced4]">
       <input
         type="number"
         placeholder="Phone number"
         className="w-full p-2 bg-inherit"
         required
         name="phone" // Name attribute zaroori hai
         value={shipToAddress.phone}
         onChange={(e) => handleChange(e)}
       />
     </div>

     <div className="flex gap-2 mt-4">
       <IoCheckbox className="text-[#19D16F]" />
       <p className="text-[#8A91AB] text-[10px]">
         Keep me up to date on news and excluive offers
       </p>
     </div>

     <h1 className="text-[16px]  text-[#1D3178] font-bold  mt-32">
       Shipping address
     </h1>

     <div className="flex  flex-col md:flex-row md:gap-4  md:justify-evenly ">
       <div className="  text-[#C1C8E1] mt-10  border-b-[2px]  border-b-[#d0ced4] w-full">
         <input
           type="text"
           placeholder="name"
           className="w-full p-2 bg-inherit"
           name="name" // Name attribute zaroori hai
           value={shipToAddress.name}
           onChange={(e) => handleChange(e)}
         />
       </div>
       <div className="  text-[#C1C8E1] mt-10  border-b-[2px]  border-b-[#d0ced4] w-full">
         <input
           type="text"
           placeholder="Last name  (optional)"
           className="w-full p-2 bg-inherit"
         />
       </div>
     </div>
     <div className="flex  flex-col md:flex-row md:gap-4  md:justify-evenly ">
       <div className="  text-[#C1C8E1] mt-10  border-b-[2px]  border-b-[#d0ced4] w-full">
         <input
           type="text"
           placeholder="addressLine1"
           className="w-full p-2  bg-inherit"
           name="addressLine1" // Name attribute zaroori hai
           value={shipToAddress.addressLine1}
           onChange={(e) => handleChange(e)}
         />
       </div>
       <div className="  text-[#C1C8E1] mt-10  border-b-[2px]  border-b-[#d0ced4] w-full">
         <input
           type="text"
           placeholder="addressLine2"
           className="w-full p-2  bg-inherit"
           required
           name="addressLine2" // Name attribute zaroori hai
           value={shipToAddress.addressLine2}
           onChange={(e) => handleChange(e)}
         />
       </div>
```



```
            </div>

            <div className=" text-[#C1C8E1] mt-10  border-b-[2px]  border-b-[#d0ced4]">
              <input
                type="text"
                placeholder="State/Province (DC/CA)"
                className="w-full p-2 bg-inherit"
                name="stateProvince"
                value={shipToAddress.stateProvince}
                onChange={(e) => handleChange(e)}
                required
              />
            </div>

            <div className=" text-[#C1C8E1] mt-10  border-b-[2px]  border-b-[#d0ced4]">
              <input
                type="text"
                placeholder="City"
                className="w-full  p-2  bg-inherit"
                required
                name="cityLocality" // Name attribute zaroori hai
                value={shipToAddress.cityLocality}
                onChange={(e) => handleChange(e)}
              />

            </div>

            <div className="flex  flex-col md:flex-row md:gap-4  md:justify-evenly ">
              <div className="  text-[#C1C8E1] mt-10  border-b-[2px]  border-b-[#d0ced4] w-full">
                <input
                  type="text"
                  placeholder="Country Code (e.g., PK)"
                  className="w-full  p-2 bg-inherit"

                  name="countryCode" // Name attribute zaroori hai
                  value={shipToAddress.countryCode}
                  onChange={(e) => handleChange(e)}
                   readOnly
                />

              </div>
              <div className="  text-[#C1C8E1] mt-10  border-b-[2px]  border-b-[#d0ced4] w-full">
                <input
                  type="number"
                  placeholder="Postal Code"
                  className="w-full p-2 bg-inherit"

                  name="postalCode" // Name attribute zaroori hai
                  value={shipToAddress.postalCode}
                  onChange={(e) => handleChange(e)}
                  readOnly
                />

              </div>
            </div>


            <button
              // asChild
              className={` ${loading ? "bg-[#FB2E86] hover:bg-[#f13587]" : " bg-[#eb84b1] hover:bg-[#c77c9c] "}h-[44px] w-
[182px] rounded-sm text-white bg-[#FB2E86] hover:bg-[#FB2E86] mt-24`}
              type="submit"
            >
             Continue Shipping
              {/* <Link href="/">Continue Shipping</Link> */}
            </button>
          </form>

             </> : <>
             <h1 className="text-[22px]  text-[#1D3178] font-bold  my-6 text-center">
              Select a Rate
            </h1>
            <div  className="grid md:grid-cols-2  gap-2">
               {rates.map((rate)=>{
              return(
                <div   key={rate.rateId}    className={`p-4 border-[2px]  flex gap-2 items-center rounded-lg shadow-md
transition-transform transform hover:scale-105 cursor-pointer ${
                  rateId === rate.rateId
                    ? "border-[#19D16F] bg-green-200"
                    : "border-gray-200 bg-gray-50"
```



```
                 }`}
                 onClick={() => {
                  setrateId(rate.rateId)
                  setRateAmount(rate.shippingAmount.amount)
                  }
                 }>


                   <input
                     type="radio"
                     name="shippingRate"
                     checked={rateId === rate.rateId}
                     onChange={() => setrateId(rate.rateId)}
                     className="form-radio h-4 w-4 text-[#19D16F]"
                   />
                   <div>
                     {/* <p className="text-lg font-medium text-gray-700">
                       <strong>Carrier:</strong> {rate.carrierFriendlyName}
                     </p> */}
                     <p className="text-[#1D3178]">
                       <p>Service: {rate.serviceType}  in {rate.carrierDeliveryDays} Days</p>
                     </p>
                     <p className="text-[#1D3178] ">
                       <p>Cost: {rate.shippingAmount.amount}{" "}
                       {rate.shippingAmount.currency} </p>
                     </p>
                   </div>
                 </div>

               )
              })}
              </div>


        {/* Create Label Button */}
        {rateId && (
          <div className="mt-8">
            <button
              onClick={handleCreateLabel}
              disabled={loading}
              className="w-full px-4 py-2 bg-green-500 text-white rounded-md hover:bg-green-600 disabled:bg-gray-400"
            >
              {loading ? "Creating Label..." : "Create Label"}
            </button>
          </div>
        )}
                      {labelPdf && (
         <Link target="_blank" href={labelPdf}> <button className="px-4 py-2  w-full border-[2px] border-green-500 bg-green-300 mt-3 text-green-900 rounded-md hover:bg-green-400">Download Label</button></Link>
        )}
        {trackingObj && (
          <div className="mt-8">
            <h2 className=" text-[18px] font-semibold text-gray-800 mb-4">
              Tracking : (We are using ShipEngine test api key so order will not trace)
            </h2>
            <p>tracking number: {trackingObj.trackingNumber}</p>
            <p> labelId: {trackingObj.labelId}</p>
            <p> carrierCode: {trackingObj.carrierCode}</p>
            {/* <Link href={`/cart/checkOut/?labelId=${trackingObj.labelId}`}> */}
              <button  onClick={()=> {

                setIsConfirm(true)

                router.push(`/cart/checkOut/?labelId=${trackingObj.labelId}`)
               }

              } className="px-4 py-2 w-full   mt-3  bg-green-500 text-white rounded-md hover:bg-green-600">Track Order</button>
            {/* </Link> */}
          </div>
        )}
        {isConfirm && <TrackShipment/> }


              </> }


          </div>
          <div className="lg:w-[40%]  w-full">
            {cartItems.map((items) => (
```



```
              <>
                <div
                  key={items.selectedPlant._id}
                  className="h-[102px] bg-[#E1E1E4] px-2 w-full flex justify-between shadow-md mb-4"
                >
                  <div className="flex gap-2">
                    <Image
                      src={items.selectedPlant.image}
                      height={87}
                      width={83}
                      alt={items.selectedPlant.name}
                      className="h-[84px] w-[80px] "
                    />
                    <div className="flex flex-col justify-center items-center">
                      <h2 className="text-[12px]">
                        {items.selectedPlant.name}
                      </h2>
                      <p className="text-[12px] text-[#A1A8C1]"> color:Brown</p>
                      <p className="text-[12px] text-[#A1A8C1]">size:Xl</p>
                    </div>
                  </div>
                  <div className="flex justify-center items-center text-[#15245E]">{`$${items.quantity &&
Number(items.selectedPlant.price) && (items.quantity * Number(items.selectedPlant.price)).toFixed(2)}`}</div>{" "}
                </div>
              </>
            ))}

            {/* Cart Totals Section */}
            <div className="w-full">
              <h2 className="my-5 text-center text-[#1D3178] font-bold">
                Cart Totals
              </h2>
              <div className="h-auto  w-full p-7  bg-[#E8E6F1]">
                <div className="flex justify-between text-[#1D3178]  border-[2px]  border-b-[#d0ced4]">
                  <h3>Subtotals:</h3>
                  <h3>${subTotal}</h3>
                </div>

                <div className="flex justify-between text-[#1D3178] mt-3  border-[2px]  border-b-[#d0ced4]">
                  <h3>Total:</h3>
                  <h3>${total}</h3>
                </div>

                <div className="flex gap-2 mt-4">
                  <IoCheckbox className="text-[#19D16F]" />
                  <p className="text-[#8A91AB] text-[10px]">

                    Shipping & taxes calculated at checkout
                  </p>
                </div>

                <Button
                  onClick={() => isConfirm && handleUpdate()}
                  className="h-[40px] w-full bg-[#19D16F] mt-4 hover:bg-[#19D16F] text-white"
                >
                  Confirm Order
                </Button>
              </div>
            </div>
          </div>
        </div>
      </section>
    </main>
  );
}

export default OrderDone;
```

- ➢ api route that I hit in prev code for shippinge

  api/shipengine/get-rates/route.ts

```
import { shipEngine } from "@/helper/shipEngine";

import { NextRequest } from "next/server";

export async function POST(req: NextRequest) {
  const { shipToAddress, packages } = await req.json();
  try {


      // Validate required fields
      if (!shipToAddress || !packages) {
        return new Response(
          JSON.stringify({
            error: "Missing required fields: shipeToAddress and packages",
          }),
          { status: 400 }
        );
      }



      const params = {
        shipment: {
          shipTo: shipToAddress,
          shipFrom: {
            name: "John Doe",
            phone: "+1 555 123 4567",
            addressLine1: "742 Evergreen Terrace",
            addressLine2: "Apt 101",
            cityLocality: "Springfield",
            stateProvince: "IL",
            postalCode: "74000",
            countryCode: "US",
            addressResidentialIndicator: "no"as const    // TypeScript will now treat it as a literal


          },
          packages: packages,
        },
        rateOptions: {
          carrierIds: [
            process.env.SHIPENGINE_FIRST_COURIER || "",
            process.env.SHIPENGINE_SECOND_COURIER || "",
            process.env.SHIPENGINE_THIRD_COURIER || "",
            process.env.SHIPENGINE_FOURTH_COURIER || "",
          ].filter(Boolean),
        },
      }
    const shipmentDetails = await shipEngine.getRatesWithShipmentDetails(params);



    // Log details for debugging
    console.log("Ship To Address:", shipToAddress);
    console.log("Packages:", packages);
    console.log("Shipment Details:", shipmentDetails);


    return new Response(JSON.stringify(shipmentDetails), { status: 200 });
  } catch (error : any) {
    return new Response(JSON.stringify({ error: error.message }), { status: 500 });
  }
}
```

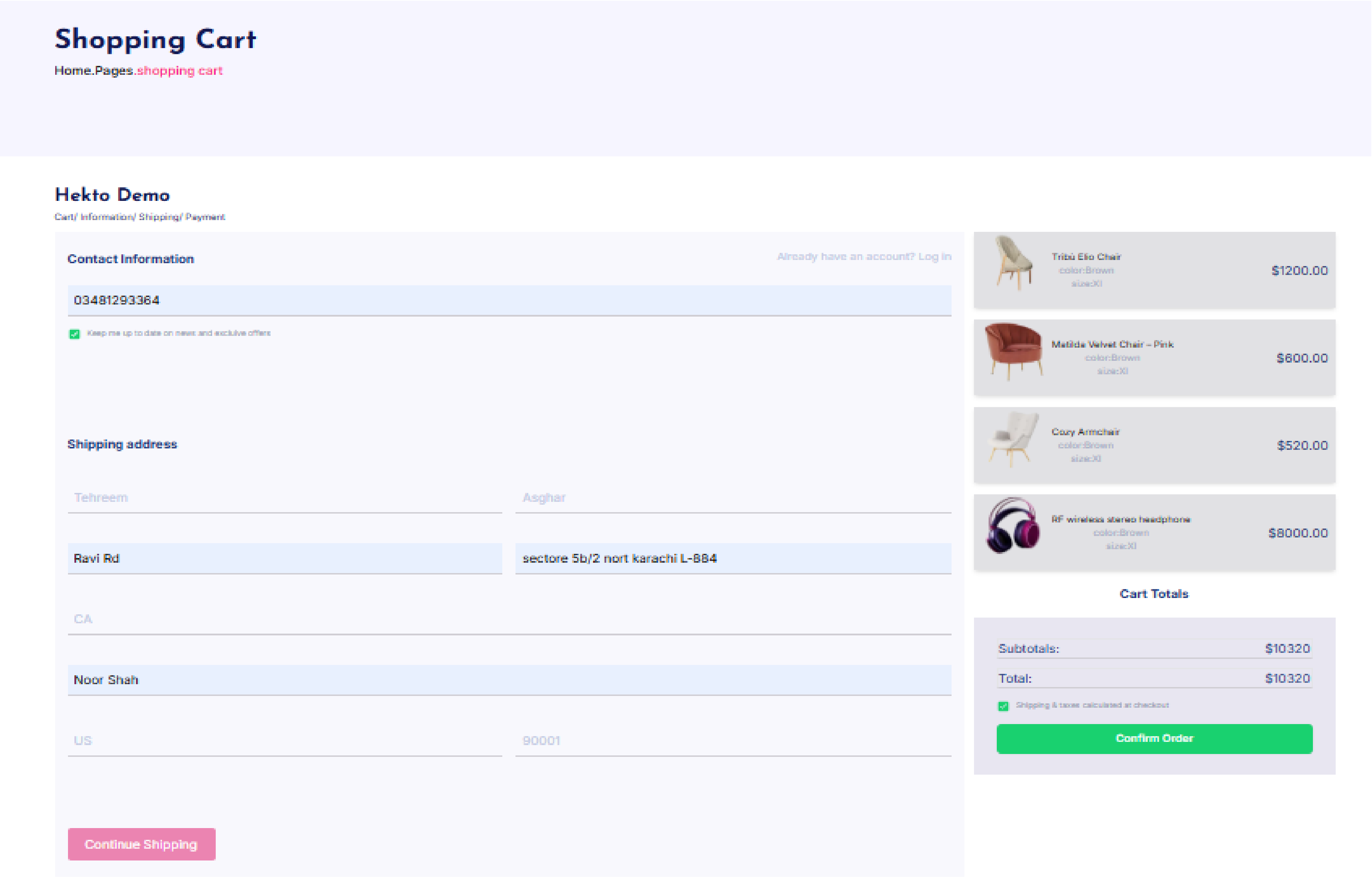
- src/app/api/shipengine/create-label/route.ts

```
import { shipEngine } from "@/helper/shipEngine";
import { NextRequest, NextResponse } from "next/server";

export async function POST(req: NextRequest) {
  try {
    const { rateId } = await req.json();
    const label = await shipEngine.createLabelFromRate({ rateId });
    return NextResponse.json(label, { status: 200 });
  } catch (error) {
    return NextResponse.json(
      { error: "An error occurred while creating the label" },
      { status: 500 }
    );
  }
}
```

- src/app/api/shipengine/tracking/[labelId]/route.ts

```
import { shipEngine } from "../../../../../helper/shipEngine";
import { NextRequest, NextResponse } from "next/server";

export async function GET(
  req: NextRequest,
  { params }: {
  params: Promise<{ labelId: string }>
}) {
  const labelId = (await params).labelId;
  if (!labelId) {
    return new Response(JSON.stringify({ error: "Missing required fields" }), {
      status: 400,
    });
  }

  try {
    // you have two options to track
    // you can track using label id
    const label = await shipEngine.trackUsingLabelId(labelId);
    // or
    // you can track using carrier code and tracking number
    // const label = await shipengine.trackUsingCarrierCodeAndTrackingNumber({
    //   carrierCode: "carrier code", // Replace with the actual carrier code
    //   trackingNumber: "tracking number", // Replace with the actual tracking number
    // });

    console.log(label);

    return NextResponse.json(label, { status: 200 });
  } catch (error) {
    console.log(error);
    return new Response(JSON.stringify({ error: error }), {
      status: 500,
    });
  }
}
```

ShipEngine Rates create see here in frontend fill this form and create rates



Shopping Cart

Home.Pages.shopping cart

Hekto Demo

Cart/ Information/ Shipping/ Payment

Contact Information

Already have an account? Log in

03481293364

☑ Keep me up to date on news and exclusive offers

Shipping address

Tehreem
Asghar
Ravi Rd
sectore 5b/2 nort karachi L-884
CA
Noor Shah
US
90001

Continue Shipping

Tribù Elio Chair color:Brown size:Xl	$1200.00
Matilda Velvet Chair – Pink color:Brown size:Xl	$600.00
Cozy Armchair color:Brown size:Xl	$520.00
RF wireless stereo headphone color:Brown size:Xl	$8000.00

Cart Totals

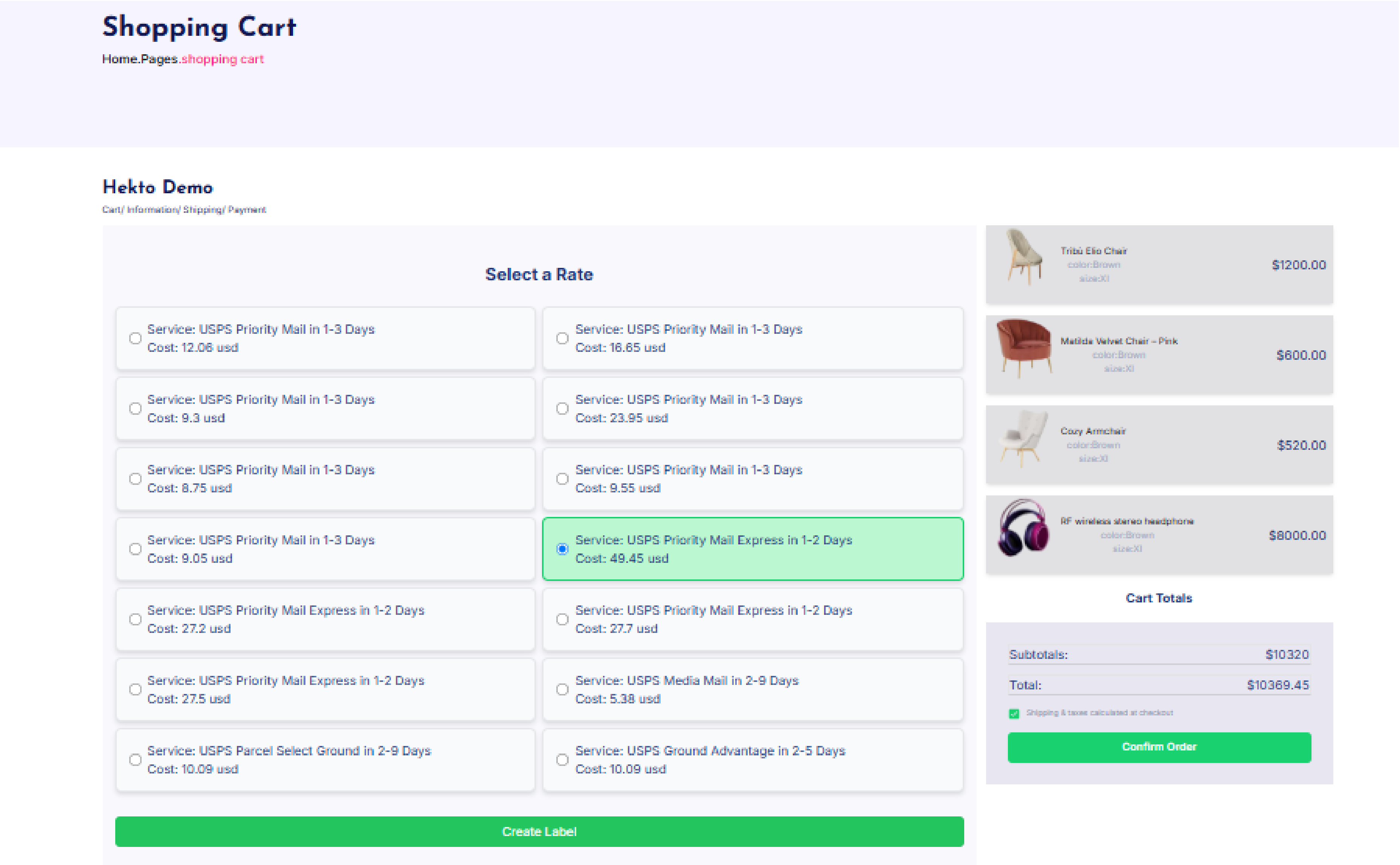
Subtotals: $10320

Total: $10320

☑ Shipping & taxes calculated at checkout

Confirm Order

After click continue Shiping create Rates see here



Shopping Cart

Home.Pages.shopping cart

Hekto Demo

Cart/ Information/ Shipping/ Payment

Select a Rate

○ Service: USPS Priority Mail in 1-3 Days
Cost: 12.06 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 16.65 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 9.3 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 23.95 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 8.75 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 9.55 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 9.05 usd

● Service: USPS Priority Mail Express in 1-2 Days
Cost: 49.45 usd

○ Service: USPS Priority Mail Express in 1-2 Days
Cost: 27.2 usd

○ Service: USPS Priority Mail Express in 1-2 Days
Cost: 27.7 usd

○ Service: USPS Priority Mail Express in 1-2 Days
Cost: 27.5 usd

○ Service: USPS Media Mail in 2-9 Days
Cost: 5.38 usd

○ Service: USPS Parcel Select Ground in 2-9 Days
Cost: 10.09 usd

○ Service: USPS Ground Advantage in 2-5 Days
Cost: 10.09 usd

Create Label

Tribù Elio Chair color:Brown size:Xl	$1200.00
Matilda Velvet Chair – Pink color:Brown size:Xl	$600.00
Cozy Armchair color:Brown size:Xl	$520.00
RF wireless stereo headphone color:Brown size:Xl	$8000.00

Cart Totals

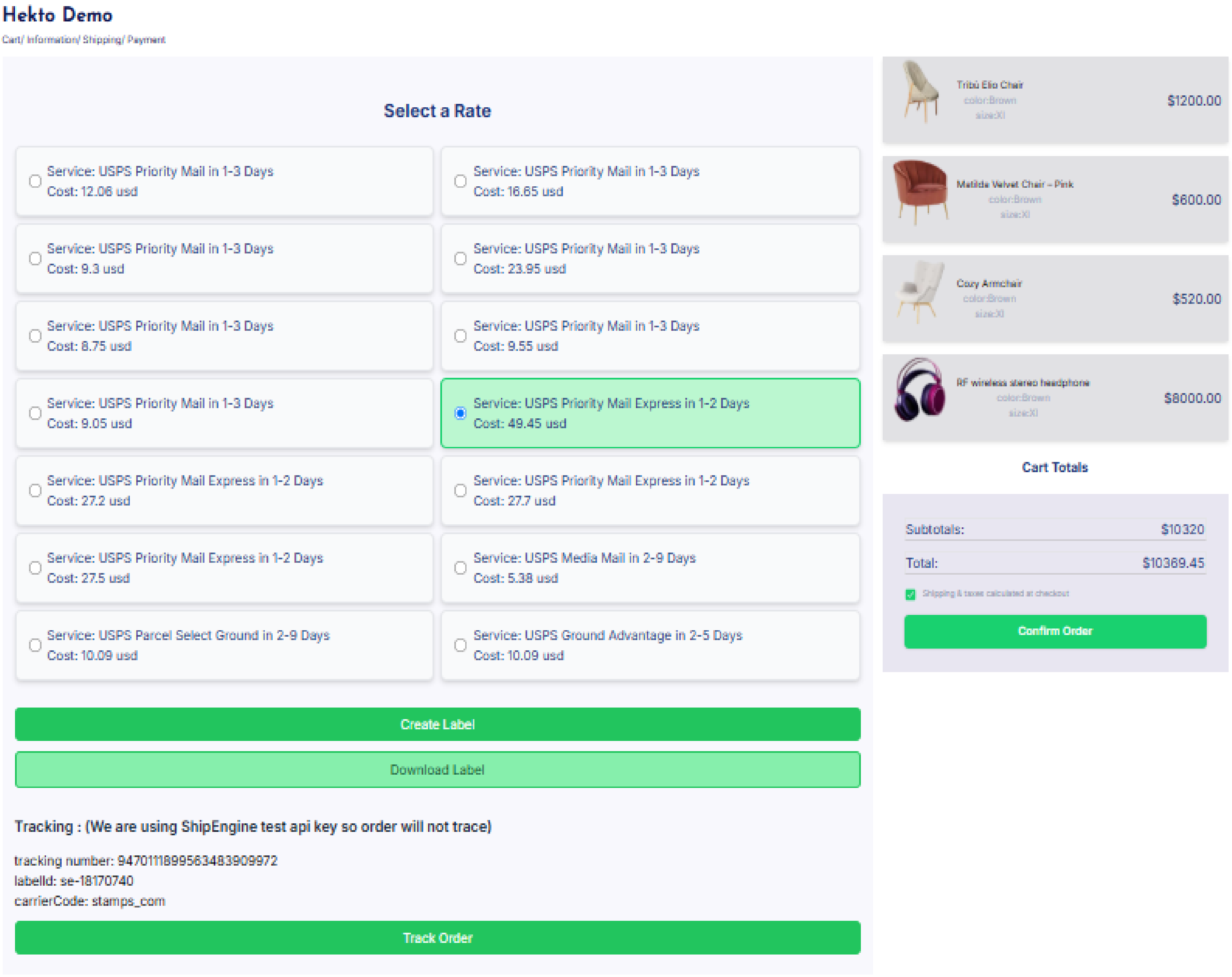
Subtotals: $10320

Total: $10369.45

☑ Shipping & taxes calculated at checkout

Confirm Order

Create Lable



Hekto Demo

Cart/ Information/ Shipping/ Payment

Select a Rate

○ Service: USPS Priority Mail in 1-3 Days
Cost: 12.06 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 16.65 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 9.3 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 23.95 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 8.75 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 9.55 usd

○ Service: USPS Priority Mail in 1-3 Days
Cost: 9.05 usd

● Service: USPS Priority Mail Express in 1-2 Days
Cost: 49.45 usd

○ Service: USPS Priority Mail Express in 1-2 Days
Cost: 27.2 usd

○ Service: USPS Priority Mail Express in 1-2 Days
Cost: 27.7 usd

○ Service: USPS Priority Mail Express in 1-2 Days
Cost: 27.5 usd

○ Service: USPS Media Mail in 2-9 Days
Cost: 5.38 usd

○ Service: USPS Parcel Select Ground in 2-9 Days
Cost: 10.09 usd

○ Service: USPS Ground Advantage in 2-5 Days
Cost: 10.09 usd

Tracking : (We are using ShipEngine test api key so order will not trace)

tracking number: 9470111899563483909972
labelId: se-18170740
carrierCode: stamps_com

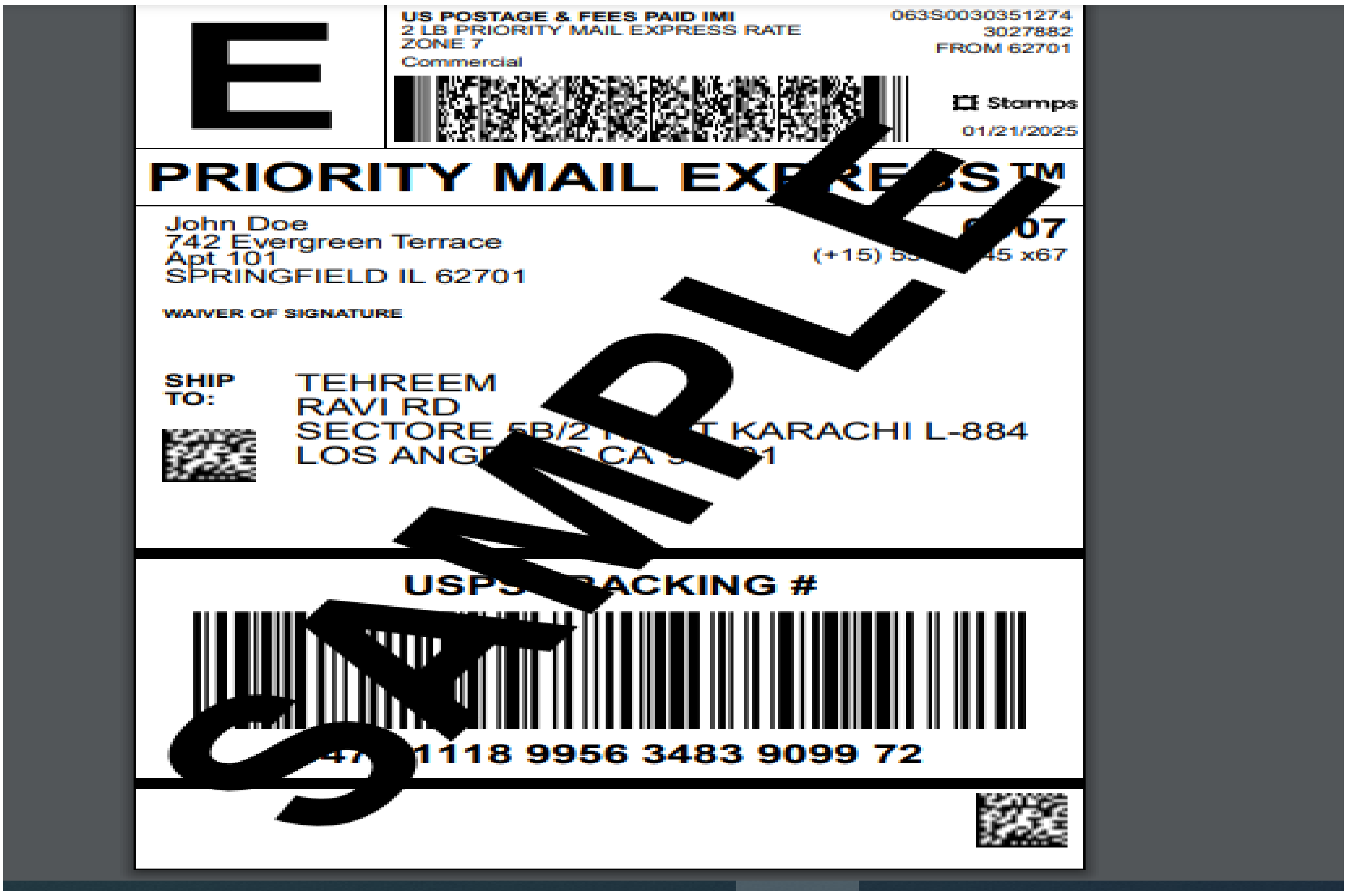
Tribù Elio Chair color:Brown size:Xl	$1200.00
Matilda Velvet Chair – Pink color:Brown size:Xl	$600.00
Cozy Armchair color:Brown size:Xl	$520.00
RF wireless stereo headphone color:Brown size:Xl	$8000.00

Cart Totals

Subtotals:	$10320
Total:	$10369.45

☑ Shipping & taxes calculated at checkout

Download Lable pdf



E

US POSTAGE & FEES PAID IMI
2 LB PRIORITY MAIL EXPRESS RATE
ZONE 7
Commercial

063S0030351274
3027882
FROM 62701

Stamps
01/21/2025

PRIORITY MAIL EXPRESS™

John Doe
742 Evergreen Terrace
Apt 101
SPRINGFIELD IL 62701

07
(+15) 55 45 x67

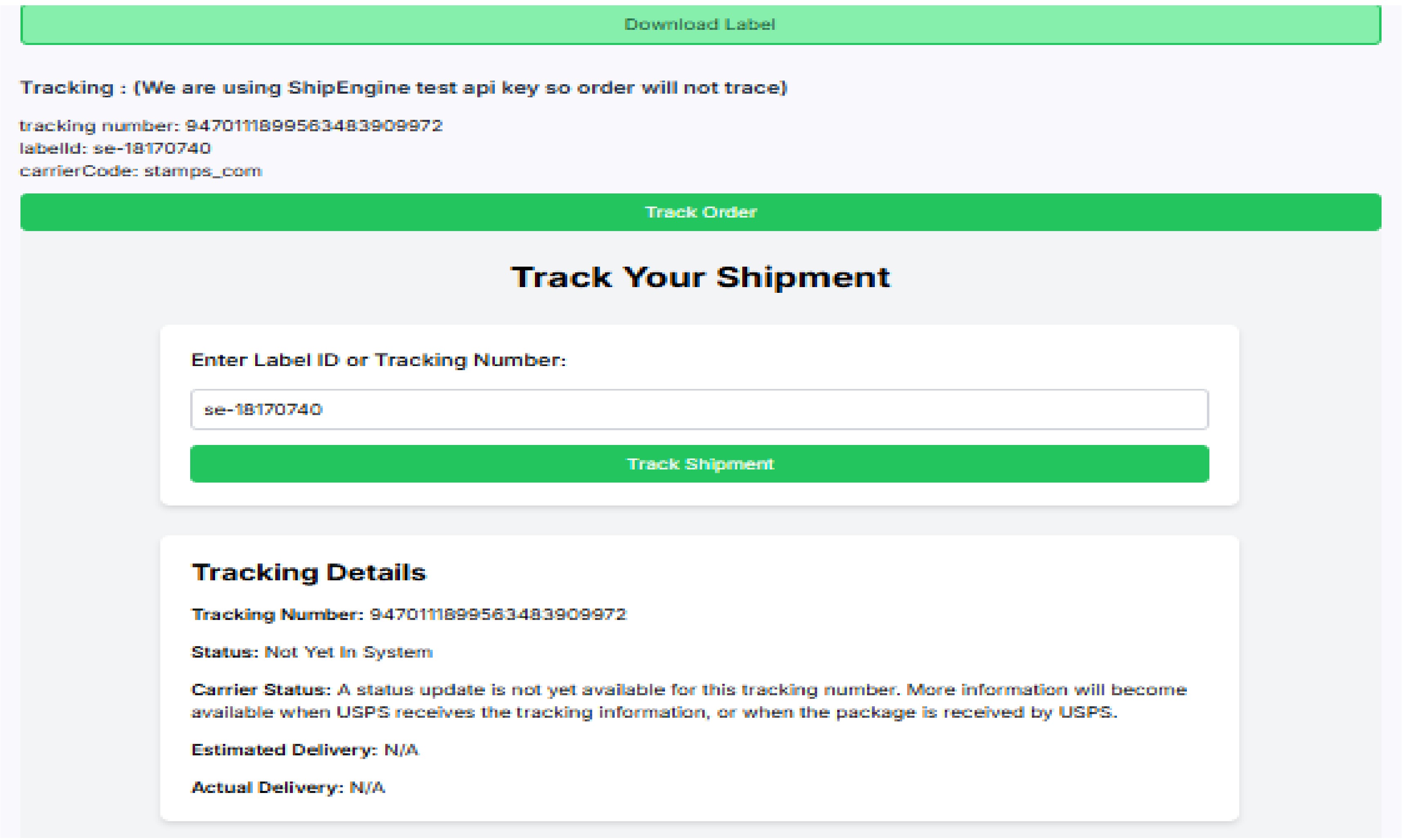
WAIVER OF SIGNATURE

SHIP TO:
TEHREEM
RAVI RD
SECTORE 5B/2 T KARACHI L-884
LOS ANG CA 9 1

USPS ACKING #

47 1118 9956 3483 9099 72

Track Your Shipment



Tracking : (We are using ShipEngine test api key so order will not trace)

tracking number: 9470111899563483909972
labelId: se-18170740
carrierCode: stamps_com

Track Your Shipment

Enter Label ID or Tracking Number:

se-18170740

Tracking Details

Tracking Number: 9470111899563483909972

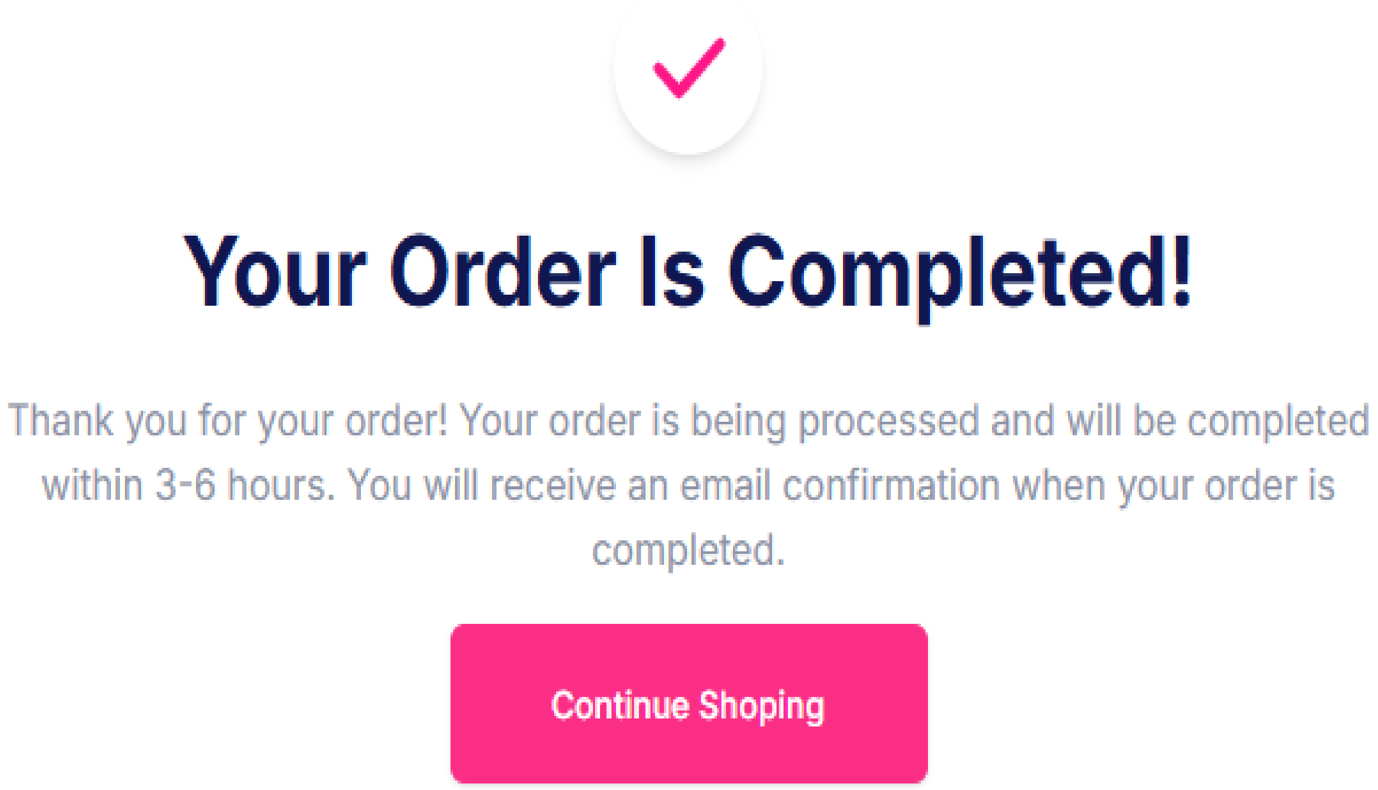
Status: Not Yet In System

Carrier Status: A status update is not yet available for this tracking number. More information will become available when USPS receives the tracking information, or when the package is received by USPS.

Estimated Delivery: N/A

Actual Delivery: N/A

After Shipping click button Confirm Order



Your Order Is Completed!

Thank you for your order! Your order is being processed and will be completed within 3-6 hours. You will receive an email confirmation when your order is completed.