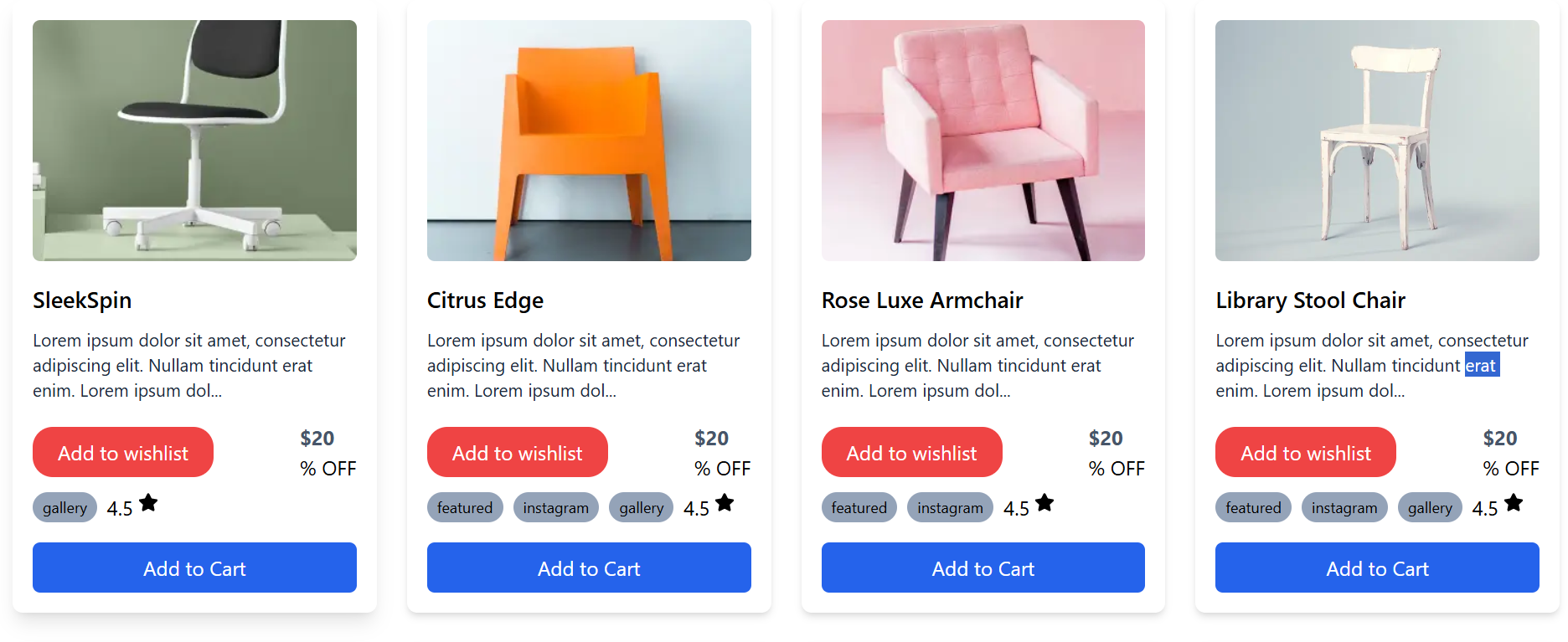
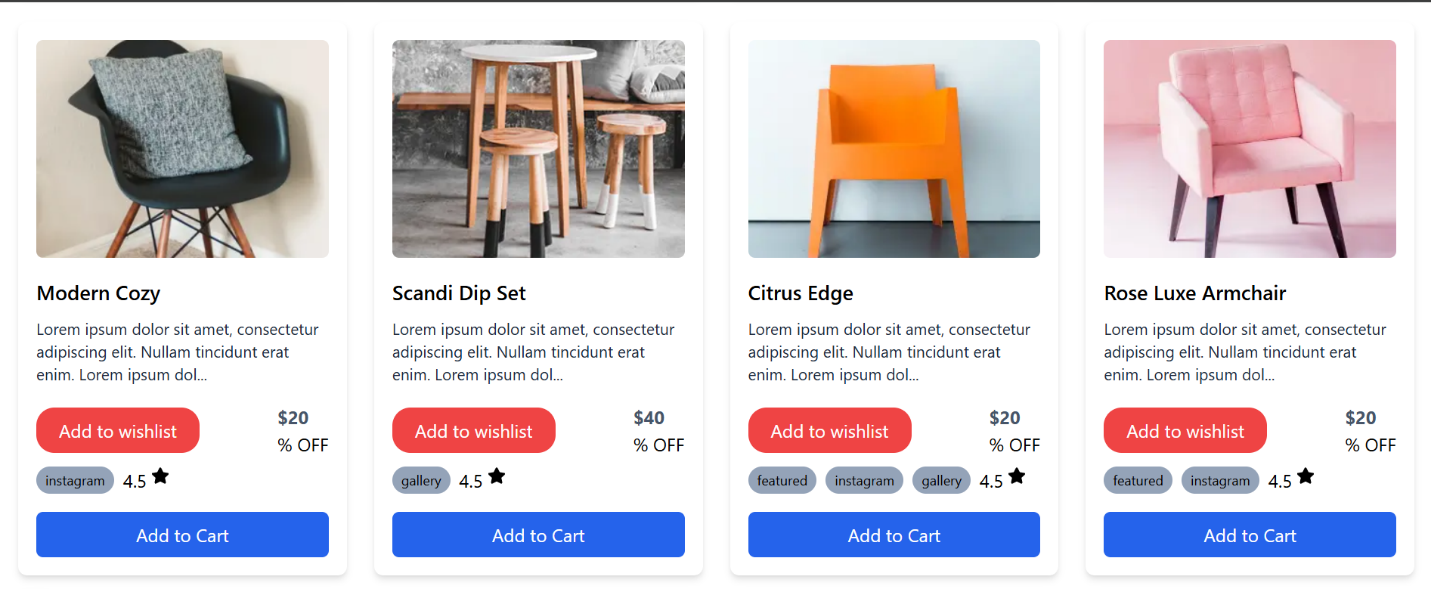
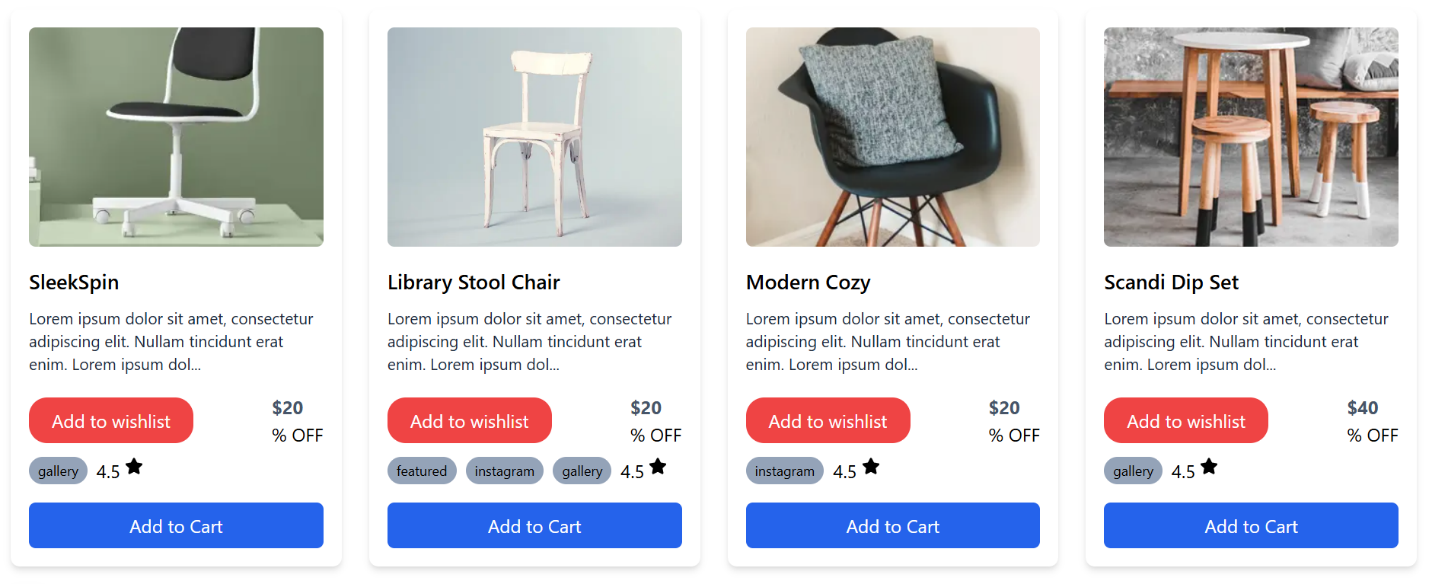
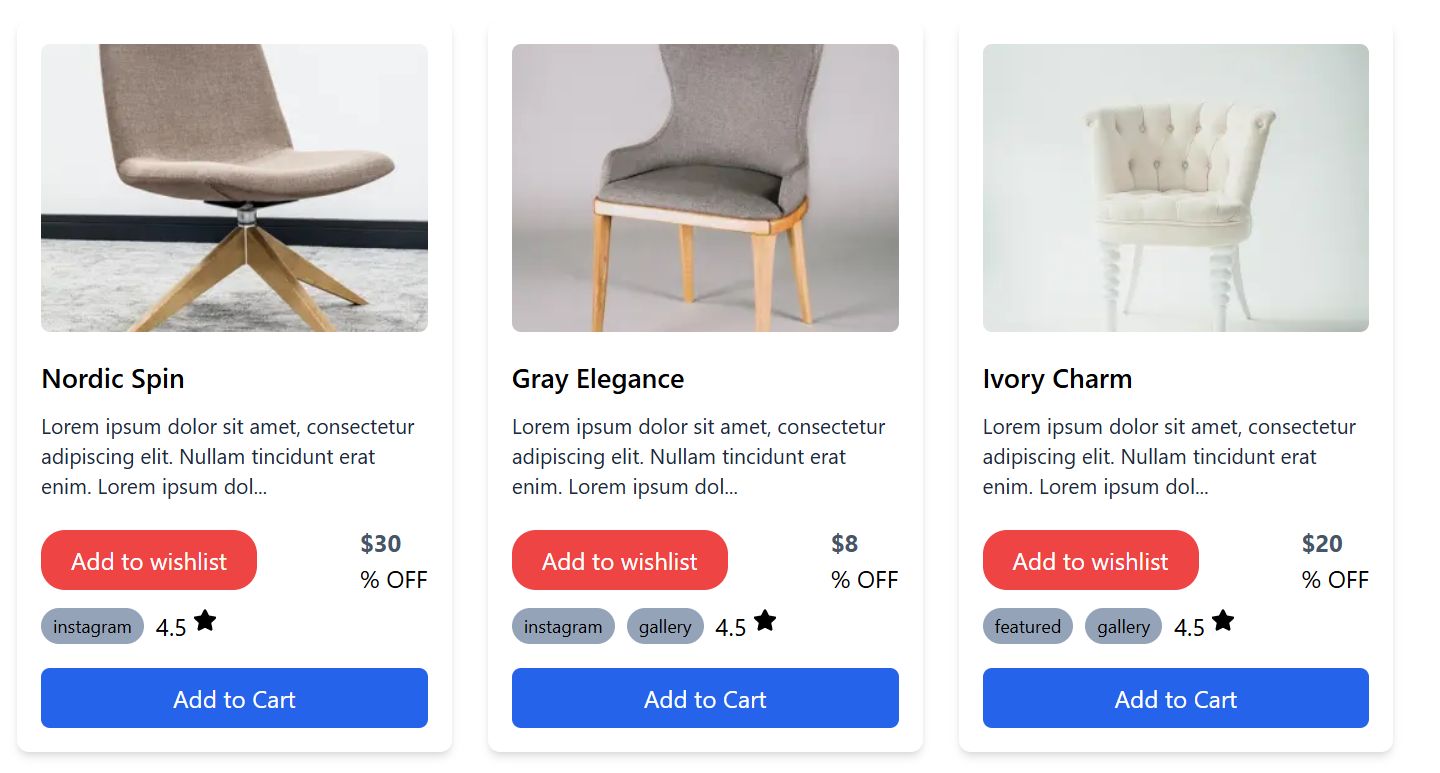
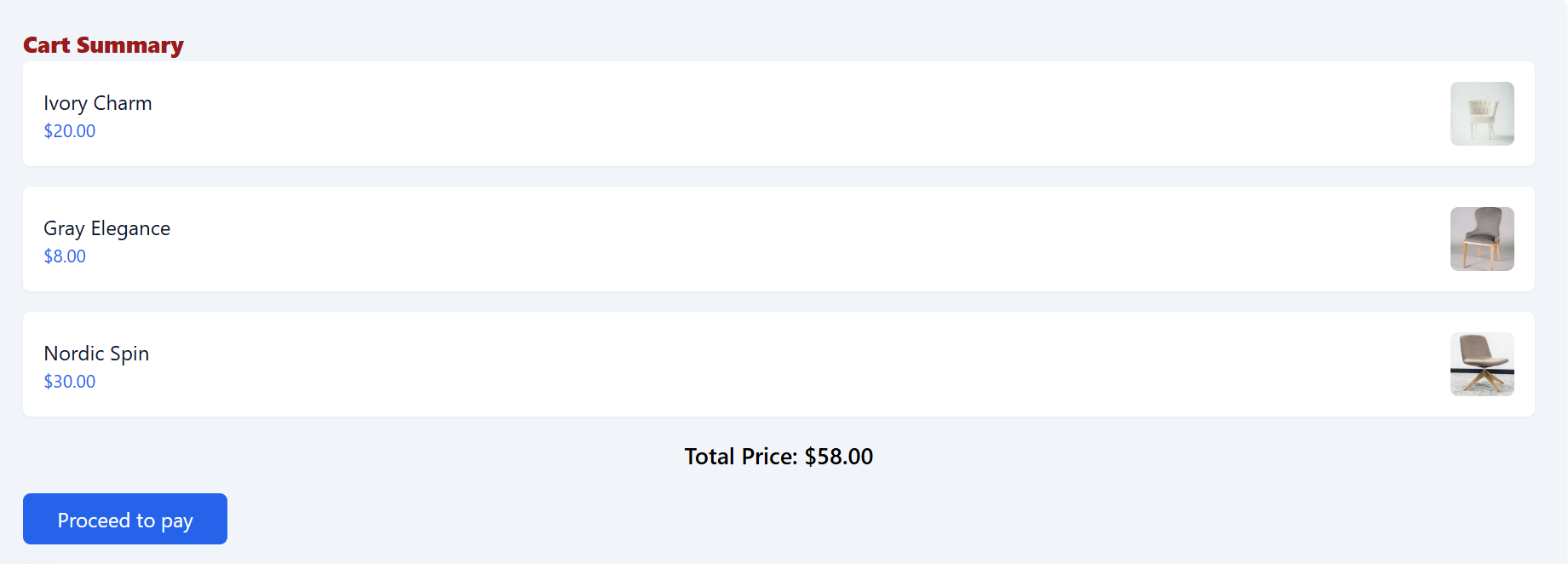
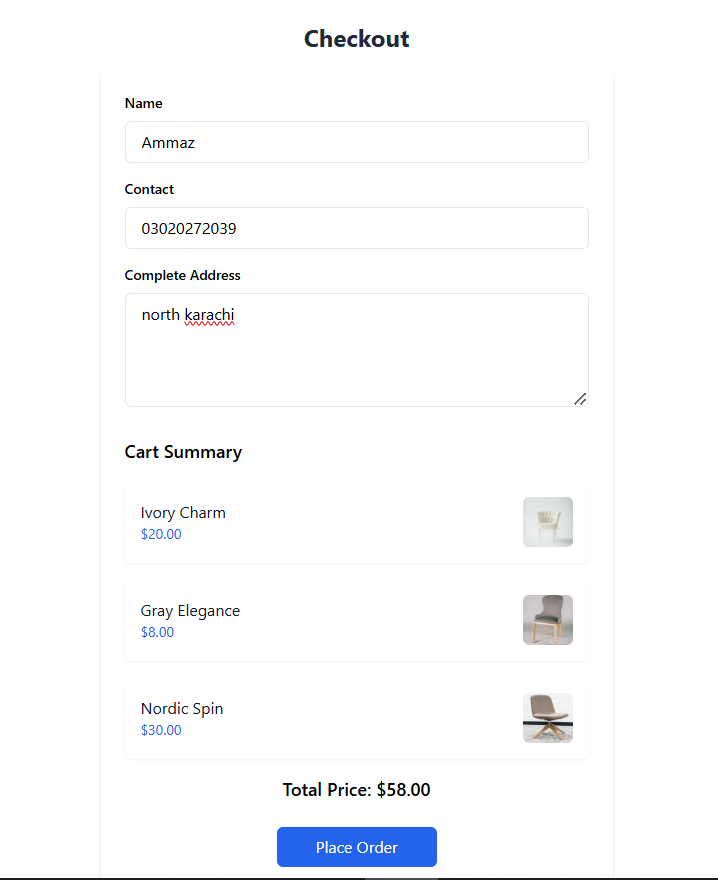
1. **Fetch Api Data:**

**1.**

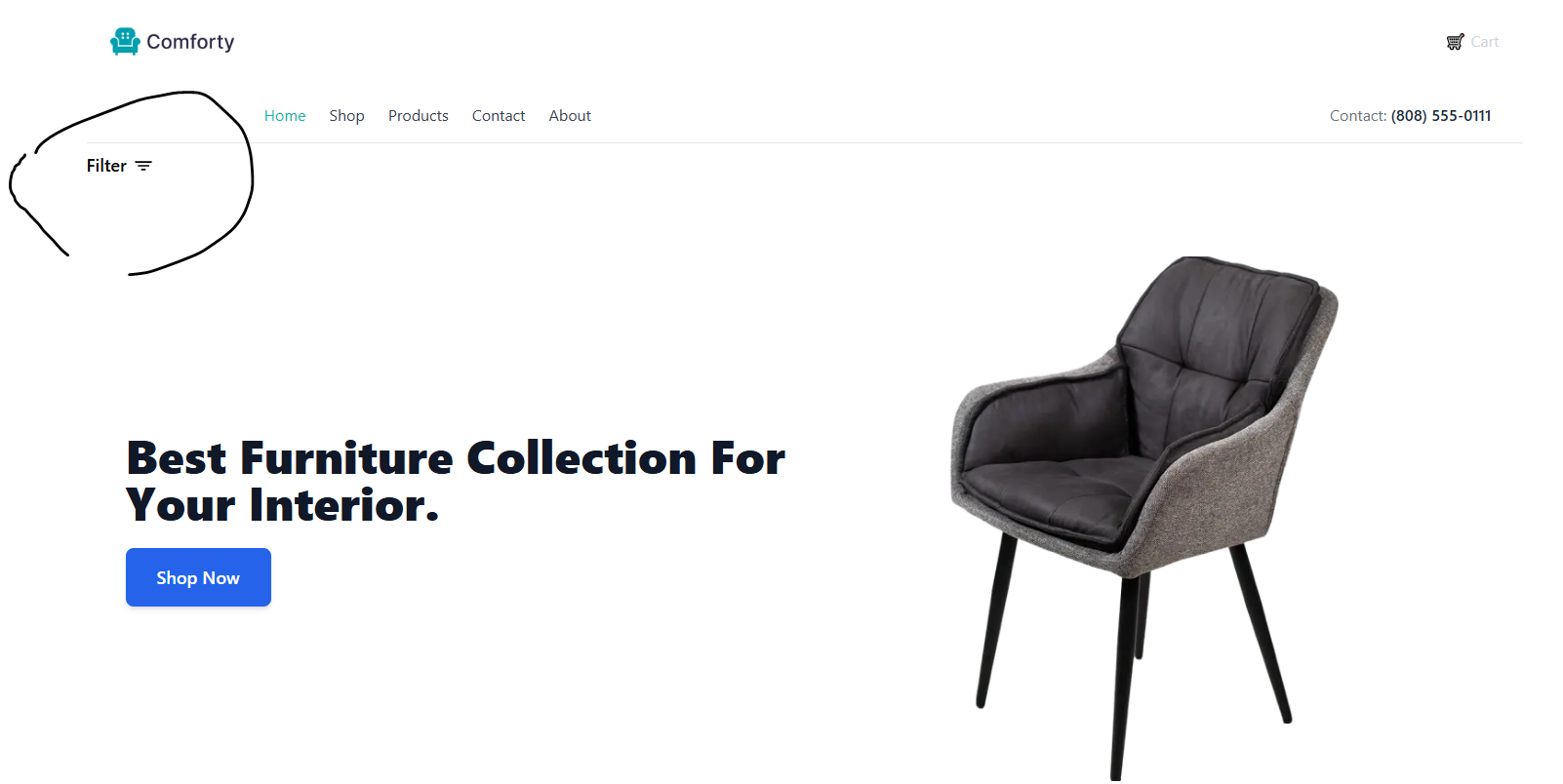
**Add to cart:**

****

**Checkout:**

****

**Filter Products:**

****

**Fetching product data in front end:**

"use client";

import { useState, useEffect } from "react";

import Image from "next/image";

import { useRouter } from "next/navigation";

import { FaStar } from "react-icons/fa";

interface Product {

  \_id: string;

  title: string;

  wishlist: string;

  price: number;

  description: string;

  discountPercentage: number;

  imageUrl: string;

  tags: string[];

}

const ProductCards: React.FC = () => {

  const [products, setProducts] = useState<Product[]>([]);

  const [cart, setCart] = useState<Product[]>([]);

  const [totalPrice, setTotalPrice] = useState<number>(0);

  const router = useRouter();

  // Fetch products from the external API

  const fetchProducts = async () => {

    try {

      const response = await fetch("https://giaic-hackathon-template-08.vercel.app/api/products");

      const data = await response.json();

      setProducts(data);

    } catch (error) {

      console.log("Error Fetching Products:", error);

    }

  };

  function truncateDescription(description: string | undefined | null) {

    if (!description) return '';

    return description.length > 100

      ? description.substring(0, 100) + "..."

      : description;

  }

  const addtoCart = (product: Product) => {

    setCart((prevCart) => [...prevCart, product]);

    setTotalPrice((prevTotalPrice) => prevTotalPrice + product.price);

    alert(`${product.title} has been added to your cart!`);

  };

  useEffect(() => {

    fetchProducts();

  }, []);

  const proceedToPay = () => {

    const cartData = JSON.stringify(cart);

    router.push(`/checkout?cart=${encodeURIComponent(cartData)}`);

  };

  return (

    <div className="p-4">

      <h2 className="text-center text-slate-800 mt-4 mb-4">

        Products from API's Data

      </h2>

      <div className="grid grid-col-1 sm:grid-cols-2 md:grid-col-3 lg:grid-cols-4 gap-6">

        {products.map((product) => (

          <div

            key={product.\_id}

            className="bg-white shadow-md rounded-lg p-4 hover:shadow-lg transition-shadow duration-300"

          >

            <Image

              src={product.imageUrl}

              alt={product.title}

              width={300}

              height={300}

              className="w-full h-48 object-cover rounded-md"

            />

            <div className="mt-4">

              <h2 className="text-lg font-semibold">{product.title}</h2>

              <p className="text-slate-800 mt-2 text-sm">

                {truncateDescription(product.description)}

              </p>

              <div className="flex justify-between items-center mt-4">

                <div>

                  <button className="w-36 py-2 bg-red-500 hover:bg-red-400 rounded-2xl text-white"

      >

                    Add to wishlist

                  </button>

                </div>

                <div>

                  <p className="text-slate-600 font-bold">${product.price}</p>

                  {product.discountPercentage}% OFF

                </div>

              </div>

              <div className="mt-2 flex flex-wrap gap-2">

                {product.tags.map((tag, index) => (

                  <span

                    key={index}

                    className="text-xs bg-slate-400 text-black rounded-full px-2 py-1"

                  >

                    {tag}

                  </span>

                ))}

                <div className="flex gap-1">4.5 <FaStar /></div>

              </div>

              <button

                className="mt-4 w-full bg-blue-600 text-white py-2 rounded-md hover:bg-blue-700"

                onClick={() => addtoCart(product)}

              >

                Add to Cart

              </button>

            </div>

          </div>

        ))}

      </div>

      <div className="mt-8 bg-slate-100 p-6 rounded-lg shadow-md">

        <h2 className="text-lg font-black text-red-800">Cart Summary</h2>

        {cart.length > 0 ? (

          <div>

            <ul className="space-y-4">

              {cart.map((item, index) => (

                <li

                  key={index}

                  className="flex justify-between items-center bg-white shadow-sm p-4 rounded-md"

                >

                  <div>

                    <p className="font-md text-slate-900">{item.title}</p>

                    <p className="text-sm text-blue-600">

                      ${item.price.toFixed(2)}

                    </p>

                  </div>

                  <Image

                    src={item.imageUrl}

                    alt={item.title}

                    width={50}

                    height={50}

                    className="rounded-md"

                  />

                </li>

              ))}

            </ul>

            <div className="mt-4 text-lg font-semibold text-center">

              <p>Total Price: ${totalPrice.toFixed(2)}</p>

            </div>

          </div>

        ) : (

          <p className="text-black text-center">

            Your Cart is Empty. Please Add Products.

          </p>

        )}

        <button className="mt-4 w-40 bg-blue-600 text-white py-2 rounded-md hover:bg-blue-700"

        onClick={proceedToPay}>

        Proceed to pay

      </button>

      </div>

    </div>

  );

};

export default ProductCards;

**Check out:**

"use client";

import { useState, useEffect } from "react";

import { useRouter } from "next/navigation";

import Image from "next/image";

import { useSearchParams } from "next/navigation";

interface CartProduct {

  \_id: string;

  title: string;

  price: number;

  imageUrl: string;

}

const Checkout = () => {

  const [name, setName] = useState("");

  const [contact, setContact] = useState("");

  const [address, setAddress] = useState("");

  const [cart, setCart] = useState<CartProduct[]>([]);

  const [totalPrice, setTotalPrice] = useState<number>(0);

  const router = useRouter();

  const searchParams = useSearchParams();

  const cartData = searchParams.get("cart");

  const parsedCart = cartData ? JSON.parse(cartData) : [];

  useEffect(() => {

    const total = parsedCart.reduce((acc: number, item: CartProduct) => acc + item.price, 0);

    setTotalPrice(total);

  }, [parsedCart]);

  const handleSubmit = (e: React.FormEvent) => {

    e.preventDefault();

    console.log({ name, contact, address, cart: parsedCart });

    alert("Order placed successfully!");

    const cartData = JSON.stringify(cart);

    router.push(`/checkout?cart=${encodeURIComponent(cartData)}`);

  };

  return (

    <div className="p-4">

      <h2 className="text-center font-bold text-2xl text-slate-800 mt-4 mb-4">Checkout</h2>

      <form onSubmit={handleSubmit} className="max-w-lg mx-auto bg-white p-6 rounded-lg shadow-md">

        <div className="mb-4">

          <label className="block text-sm font-semibold mb-2">Name</label>

          <input

            type="text"

            value={name}

            onChange={(e) => setName(e.target.value)}

            className="w-full px-4 py-2 border rounded-md"

            required

          />

        </div>

        <div className="mb-4">

          <label className="block text-sm font-semibold mb-2">Contact</label>

          <input

            type="text"

            value={contact}

            onChange={(e) => setContact(e.target.value)}

            className="w-full px-4 py-2 border rounded-md"

            required

          />

        </div>

        <div className="mb-4">

          <label className="block text-sm font-semibold mb-2">Complete Address</label>

          <textarea

            value={address}

            onChange={(e) => setAddress(e.target.value)}

            className="w-full px-4 py-2 border rounded-md"

            rows={4}

            required

          />

        </div>

        <h3 className="text-lg font-semibold mt-6">Cart Summary</h3>

        {parsedCart.length > 0 ? (

          <>

            <ul className="space-y-4 mt-4">

              {parsedCart.map((item: CartProduct, index: number) => (

                <li key={index} className="flex justify-between items-center bg-white shadow-sm p-4 rounded-md">

                  <div>

                    <p className="font-md text-slate-900">{item.title}</p>

                    <p className="text-sm text-blue-600">${item.price.toFixed(2)}</p>

                  </div>

                  <Image

                    src={item.imageUrl}

                    alt={item.title}

                    width={50}

                    height={50}

                    className="rounded-md"

                  />

                </li>

              ))}

            </ul>

            <div className="mt-4 text-lg font-semibold text-center">

              <p>Total Price: ${totalPrice.toFixed(2)}</p>

            </div>

          </>

        ) : (

          <p className="text-center text-gray-500">No products in cart</p>

        )}

        <div className="mt-6 flex justify-center">

          <button

            type="submit"

            className="w-40 bg-blue-600 text-white py-2 rounded-md hover:bg-blue-700"

          >

            Place Order

          </button>

        </div>

      </form>

    </div>

  );

};

export default Checkout;

**Cart:**

import React from "react";

import Image from "next/image";

import { CiHeart } from "react-icons/ci";

import { RiDeleteBin6Line } from "react-icons/ri";

const Cart = () => {

  const cartItems = [

    {

      id: 1,

      name: "Library Stool Chair",

      description: "Ashen Slate/Cobalt Bliss",

      size: "L",

      quantity: 1,

      price: 99,

      imgSrc: "/02.jpg",

    },

    {

      id: 2,

      name: "Library Stool Chair",

      description: "Ashen Slate/Cobalt Bliss",

      size: "L",

      quantity: 1,

      price: 99,

      imgSrc: "/02.jpg",

    },

  ];

  const subtotal = cartItems.reduce((acc, item) => acc + item.price, 0);

  return (

    <div className="max-w-[1321px] mx-auto px-4 py-8">

      <div className="grid grid-cols-1 lg:grid-cols-3 gap-8">

        <div className="lg:col-span-2">

          <h2 className="text-[22px] font-medium pl-3 mb-6">Bag</h2>

          {cartItems.map((item) => (

            <div

              key={item.id}

              className="flex flex-col sm:flex-row items-center sm:justify-between bg-white p-4 rounded-lg shadow-md mb-4"

            >

              <div className="flex flex-col sm:flex-row items-center space-y-4 sm:space-y-0 sm:space-x-4">

                <div className="w-24 h-24 bg-gray-200 rounded overflow-hidden">

                  <Image

                    src={item.imgSrc}

                    alt={`Image of ${item.name}`}

                    width={150}

                    height={150}

                  />

                </div>

                <div className="text-center sm:text-left">

                  <h3 className="text-[16px] font-normal text-[#272343] mb-3">

                    {item.name}

                  </h3>

                  <p className="text-sm text-gray-500 mb-1">{item.description}</p>

                  <div className="flex justify-center sm:justify-start space-x-12">

                    <p className="text-[15px] font-normal text-[#757575]">

                      Size: {item.size}

                    </p>

                    <p className="text-[15px] font-normal text-[#757575]">

                      Quantity: {item.quantity}

                    </p>

                  </div>

                  <div className="flex justify-center sm:justify-start gap-3 mt-3 text-gray-500 hover:text-black cursor-pointer">

                    <CiHeart size={20} />

                    <RiDeleteBin6Line size={20} />

                  </div>

                </div>

              </div>

              <div className="flex justify-center sm:justify-end gap-3 mt-4 sm:mt-0">

                <p className="text-[16px] font-normal text-[#111111]">MRP:</p>

                <p className="text-[16px] font-normal text-[#111111]">

                  ${item.price}

                </p>

              </div>

            </div>

          ))}

        </div>

        <div>

          <h2 className="text-2xl font-bold mb-6">Summary</h2>

          <div className="bg-white p-6 rounded-lg shadow-md">

            <div className="flex justify-between mb-4">

              <p className="text-lg">Subtotal</p>

              <p className="text-lg font-semibold">${subtotal.toFixed(2)}</p>

            </div>

            <div className="flex justify-between mb-4">

              <p className="text-lg">Estimated Delivery & Handling</p>

              <p className="text-lg font-semibold text-green-500">Free</p>

            </div>

            <hr className="mb-4" />

            <div className="flex justify-between mb-6">

              <p className="text-xl font-bold">Total</p>

              <p className="text-xl font-bold">${subtotal.toFixed(2)}</p>

            </div>

            <button className="w-full h-[60px] rounded-[30px] text-white bg-[#029FAE] hover:bg-[#028693]">

              Member Checkout

            </button>

          </div>

        </div>

      </div>

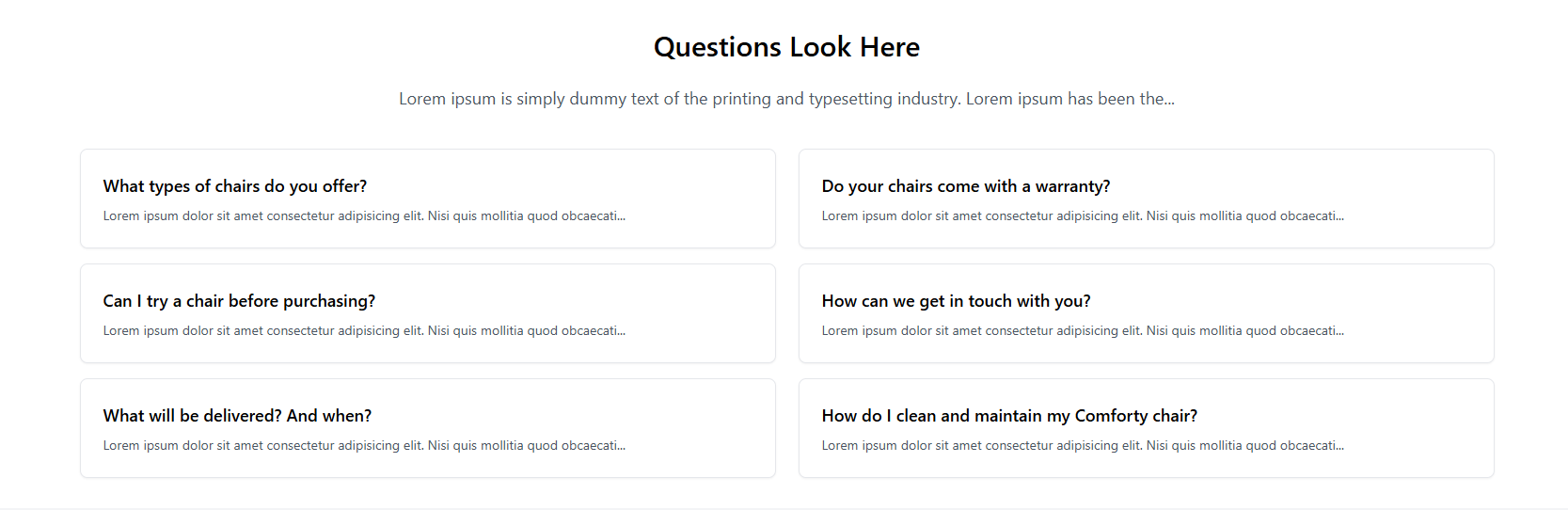
    </div>

  );

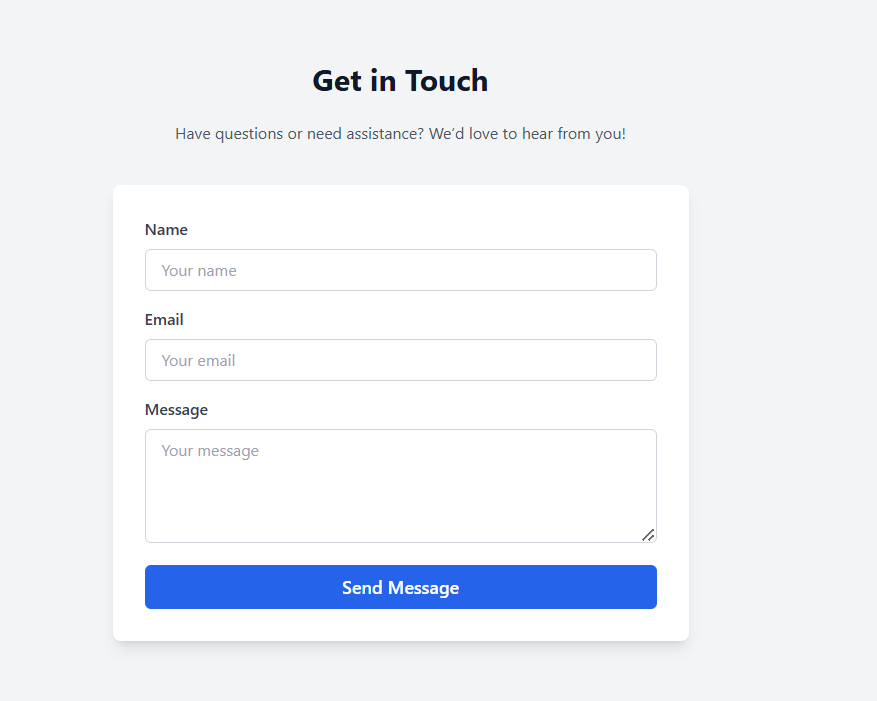
};

export default Cart;

**FAQ’S:**



**Query form :**



**Categories:**

