Day-4 Dynamic Frontend Components - Furnitures-Market:

Overview:

In Day 4, I build dynamic components to display marketplace data fetched from sanity CMS. These components are, modular reuseable and highly responsive. Enhances UI, make the website more user friendly and scaleable.

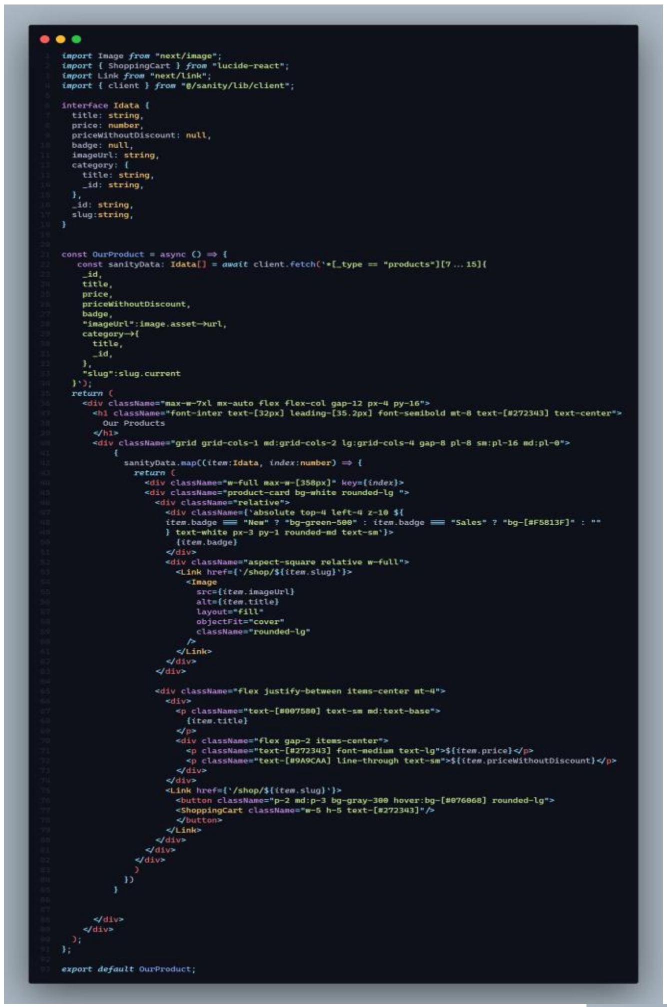
What I have done on Day - 4:

- ❖ Fetching the data from sanity and building dynamic components to display correctly in the product listing page.
- ❖ Implemented sign in functionality and created a middleware for redirecting.
- ❖ Created a skeleton loader for displaying while data is being fetching.
- ❖ Add to cart functionality, users can add the items into cart also can delete items.
- ❖ Created a Checkout page. By clicking checkout button user can fill the form and place the order.
- ❖ Category check: the user can select a category, and items will be obtained based on that category.
- ❖ Created a custom 404 – not found page.
- ❖ Ensuring all the pages are responsive.

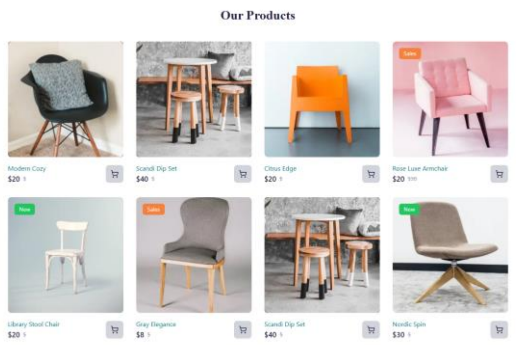
Key components that I have constructed:

1. Product Listing Page:

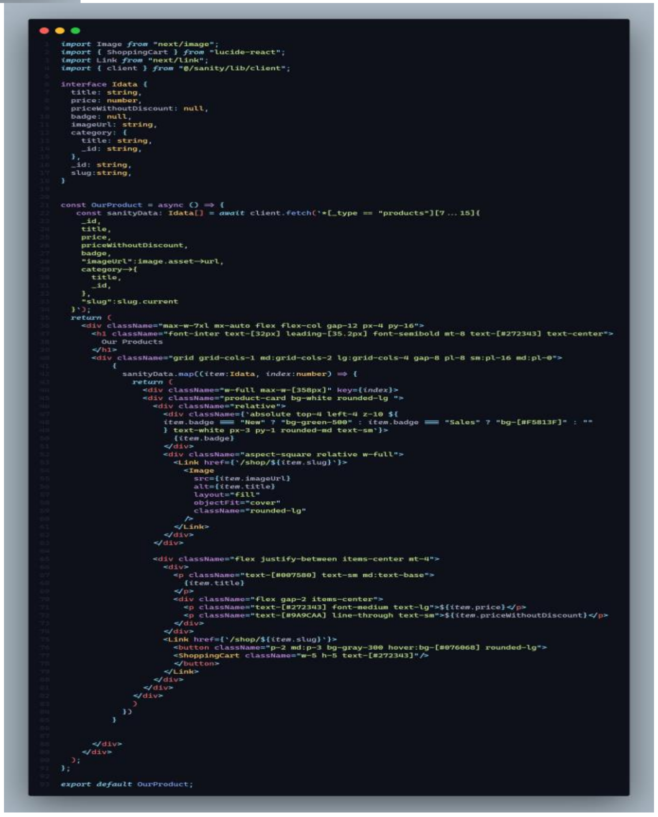
Created a component displayed the data fetching from sanity (Headless CMS). Render dynamically in a grid layout.



```
import Image from "next/image";
import { ShoppingCart } from "lucide-react";
import Link from "next/link";
import { client } from "@/sanity/lib/client";

interface Idata {
  title: string,
  price: number,
  priceWithoutDiscount: null,
  badge: null,
  imageUrl: string,
  category: {
    title: string,
    _id: string,
  },
  _id: string,
  slug:string,
}


const OurProduct = async () => {
   const sanityData: Idata[] = await client.fetch(`*[_type == "products"][7 ... 15]{
    _id,
    title,
    price,
    priceWithoutDiscount,
    badge,
    "imageUrl":image.asset->url,
    category->{
      title,
      _id,
    },
    "slug":slug.current
  }`);
  return (
    <div className="max-w-7xl mx-auto flex flex-col gap-12 px-4 py-16">
      <h1 className="font-inter text-[32px] leading-[35.2px] font-semibold mt-8 text-[#272343] text-center">
        Our Products
      </h1>
      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-8 pl-8 sm:pl-16 md:pl-0">
          {
            sanityData.map((item:Idata, index:number) => {
              return (
                <div className="w-full max-w-[358px]" key={index}>
                <div className="product-card bg-white rounded-lg ">
                  <div className="relative">
                    <div className={`absolute top-4 left-4 z-10 ${
                    item.badge === "New" ? "bg-green-500" : item.badge === "Sales" ? "bg-[#F5813F]" : ""
                    } text-white px-3 py-1 rounded-md text-sm`}>
                      {item.badge}
                    </div>
                    <div className="aspect-square relative w-full">
                      <Link href={`/shop/${item.slug}`}>
                        <Image
                          src={item.imageUrl}
                          alt={item.title}
                          layout="fill"
                          objectFit="cover"
                          className="rounded-lg"
                        />
                      </Link>
                    </div>
                  </div>

                  <div className="flex justify-between items-center mt-4">
                    <div>
                      <p className="text-[#007580] text-sm md:text-base">
                        {item.title}
                      </p>
                      <div className="flex gap-2 items-center">
                        <p className="text-[#272343] font-medium text-lg">${item.price}</p>
                        <p className="text-[#9A9CAA] line-through text-sm">${item.priceWithoutDiscount}</p>
                      </div>
                    </div>
                    <Link href={`/shop/${item.slug}`}>
                      <button className="p-2 md:p-3 bg-gray-300 hover:bg-[#076068] rounded-lg">
                      <ShoppingCart className="w-5 h-5 text-[#272343]"/>
                      </button>
                    </Link>
                  </div>
                </div>
              </div>
              )
            })
          }


      </div>
    </div>
  );
};

export default OurProduct;
```



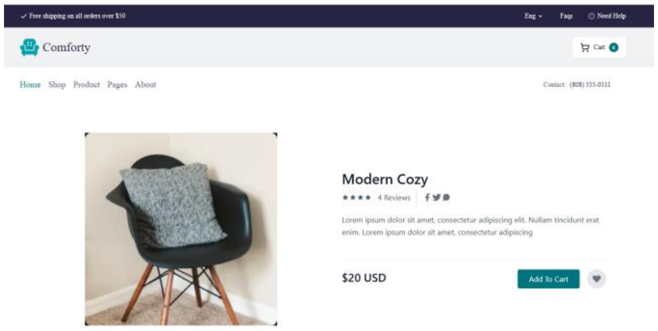
EXPLORE NEW AND POPULAR STYLES



```
import Image from "next/image";
import { ShoppingCart } from "lucide-react";
import Link from "next/link";
import { client } from "@/sanity/lib/client";

interface Idata {
  title: string,
  price: number,
  priceWithoutDiscount: null,
  badge: null,
  imageUrl: string,
  category: {
    title: string,
    _id: string,
  },
  _id: string,
  slug:string,
}


const OurProduct = async () => {
   const sanityData: Idata[] = await client.fetch(`*[_type == "products"][7 ... 15]{
    _id,
    title,
    price,
    priceWithoutDiscount,
    badge,
    "imageUrl":image.asset->url,
    category->{
      title,
      _id,
    },
    "slug":slug.current
  }`);
  return (
    <div className="max-w-7xl mx-auto flex flex-col gap-12 px-4 py-16">
      <h1 className="font-inter text-[32px] leading-[35.2px] font-semibold mt-8 text-[#272343] text-center">
        Our Products
      </h1>
      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-8 pl-8 sm:pl-16 md:pl-0">
          {
            sanityData.map((item:Idata, index:number) => {
              return (
                <div className="w-full max-w-[358px]" key={index}>
                <div className="product-card bg-white rounded-lg ">
                  <div className="relative">
                    <div className={`absolute top-4 left-4 z-10 ${
                    item.badge === "New" ? "bg-green-500" : item.badge === "Sales" ? "bg-[#F5813F]" : ""
                    } text-white px-3 py-1 rounded-md text-sm`}>
                      {item.badge}
                    </div>
                    <div className="aspect-square relative w-full">
                      <Link href={`/shop/${item.slug}`}>
                        <Image
                          src={item.imageUrl}
                          alt={item.title}
                          layout="fill"
                          objectFit="cover"
                          className="rounded-lg"
                        />
                      </Link>
                    </div>
                  </div>

                  <div className="flex justify-between items-center mt-4">
                    <div>
                      <p className="text-[#007580] text-sm md:text-base">
                        {item.title}
                      </p>
                      <div className="flex gap-2 items-center">
                        <p className="text-[#272343] font-medium text-lg">${item.price}</p>
                        <p className="text-[#9A9CAA] line-through text-sm">${item.priceWithoutDiscount}</p>
                      </div>
                    </div>
                    <Link href={`/shop/${item.slug}`}>
                      <button className="p-2 md:p-3 bg-gray-300 hover:bg-[#076068] rounded-lg">
                      <ShoppingCart className="w-5 h-5 text-[#272343]"/>
                      </button>
                    </Link>
                  </div>
                </div>
              </div>
              )
            })
          }


      </div>
    </div>
  );
};

export default OurProduct;
```

2. Product Detailed Component:

Created a dynamic product detailed component, fetching the data from sanity based on slug. Here user can see the detail of the product and can add the items into cart.



✓ Free shipping on all orders over $50 Eng Faqs Need Help

Comforty Cart 0

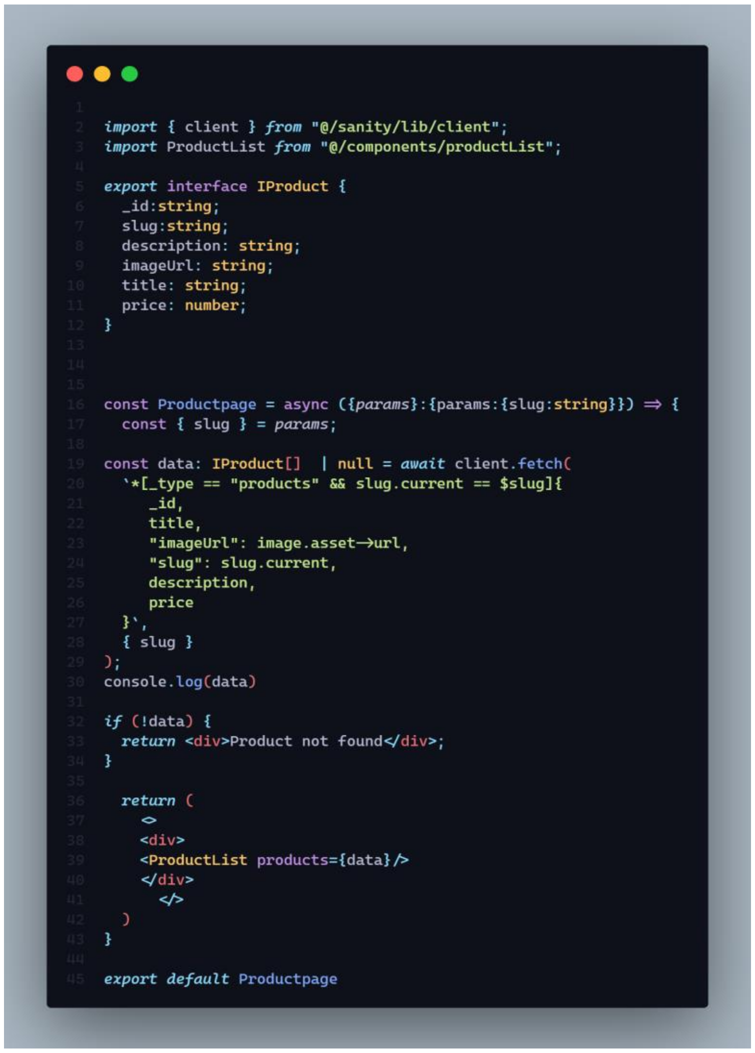
Home Shop Product Pages About Contact: (808) 555-0111

Modern Cozy

★★★★ 4 Reviews

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tincidunt erat enim. Lorem ipsum dolor sit amet, consectetur adipiscing

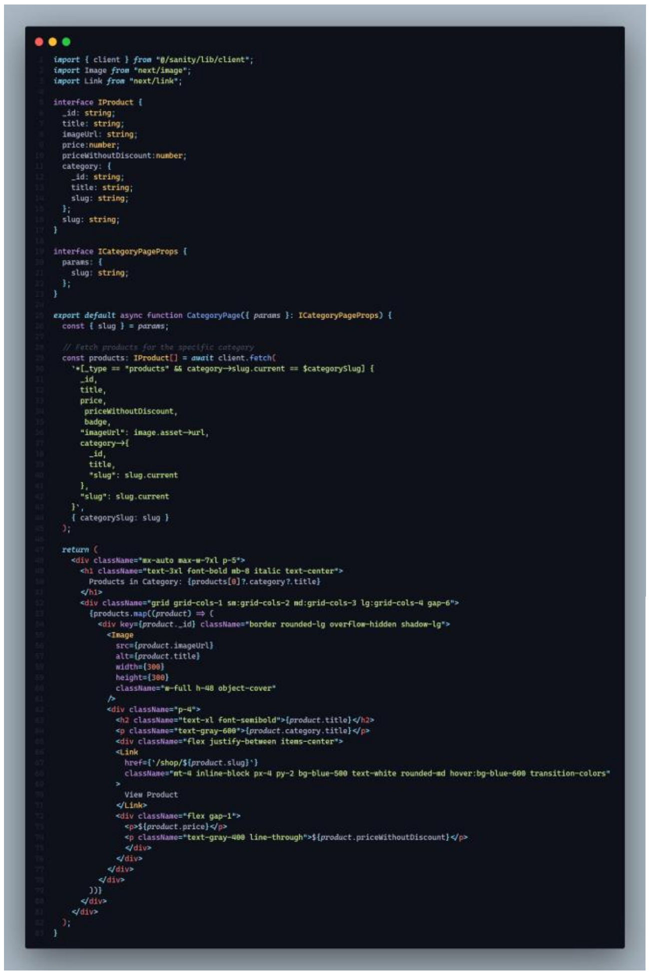
$20 USD Add To Cart



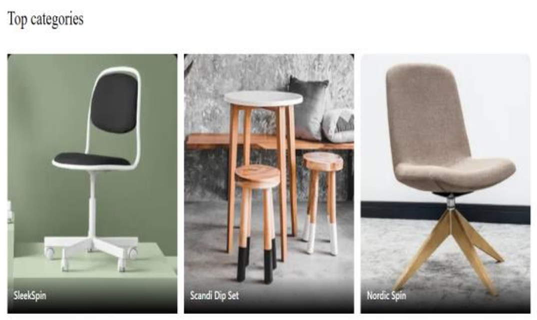
```
import { client } from "@/sanity/lib/client";
import ProductList from "@/components/productList";

export interface IProduct {
  _id:string;
  slug:string;
  description: string;
  imageUrl: string;
  title: string;
  price: number;
}



const Productpage = async ({params}:{params:{slug:string}}) => {
  const { slug } = params;

const data: IProduct[]  | null = await client.fetch(
  `*[_type == "products" && slug.current == $slug]{
     _id,
     title,
     "imageUrl": image.asset->url,
     "slug": slug.current,
     description,
     price
  }`,
  { slug }
);
console.log(data)

if (!data) {
  return <div>Product not found</div>;
}

  return (
    <>
    <div>
    <ProductList products={data}/>
    </div>
      </>
  )
}

export default Productpage
```

3. Category check:

Also implement a category page and where user can select the items according to their category, and can see the all items in the category.

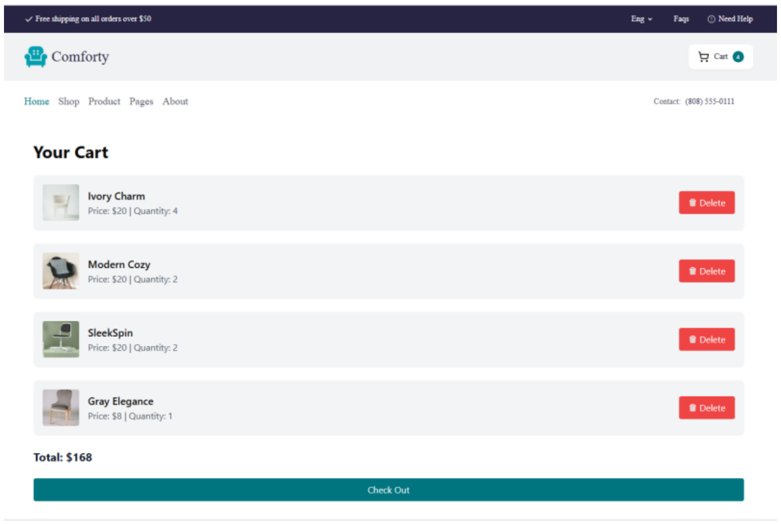


```tsx
import { client } from "@/sanity/lib/client";
import Image from "next/image";
import Link from "next/link";

interface IProduct {
  _id: string;
  title: string;
  imageUrl: string;
  price:number;
  priceWithoutDiscount:number;
  category: {
    _id: string;
    title: string;
    slug: string;
  };
  slug: string;
}

interface ICategoryPageProps {
  params: {
    slug: string;
  };
}

export default async function CategoryPage({ params }: ICategoryPageProps) {
  const { slug } = params;

  // Fetch products for the specific category
  const products: IProduct[] = await client.fetch(
    `*[_type == "products" && category->slug.current == $categorySlug] {
      _id,
      title,
      price,
       priceWithoutDiscount,
       badge,
      "imageUrl": image.asset->url,
      category->{
        _id,
        title,
        "slug": slug.current
      },
      "slug": slug.current
    }`,
    { categorySlug: slug }
  );

  return (
    <div className="mx-auto max-w-7xl p-5">
      <h1 className="text-3xl font-bold mb-8 italic text-center">
        Products in Category: {products[0]?.category?.title}
      </h1>
      <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
        {products.map((product) => (
          <div key={product._id} className="border rounded-lg overflow-hidden shadow-lg">
            <Image
              src={product.imageUrl}
              alt={product.title}
              width={300}
              height={300}
              className="w-full h-48 object-cover"
            />
            <div className="p-4">
              <h2 className="text-xl font-semibold">{product.title}</h2>
              <p className="text-gray-600">{product.category.title}</p>
              <div className="flex justify-between items-center">
              <Link
                href={`/shop/${product.slug}`}
                className="mt-4 inline-block px-4 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600 transition-colors"
              >
                View Product
              </Link>
              <div className="flex gap-1">
                <p>${product.price}</p>
                <p className="text-gray-400 line-through">${product.priceWithoutDiscount}</p>
                </div>
              </div>
            </div>
          </div>
        ))}
      </div>
    </div>
  );
}
```



4. Add To Cart:

Implemented cart page, where user can add items into cart and see the can see the quantity and individual and total prices of cart items.



Your Cart

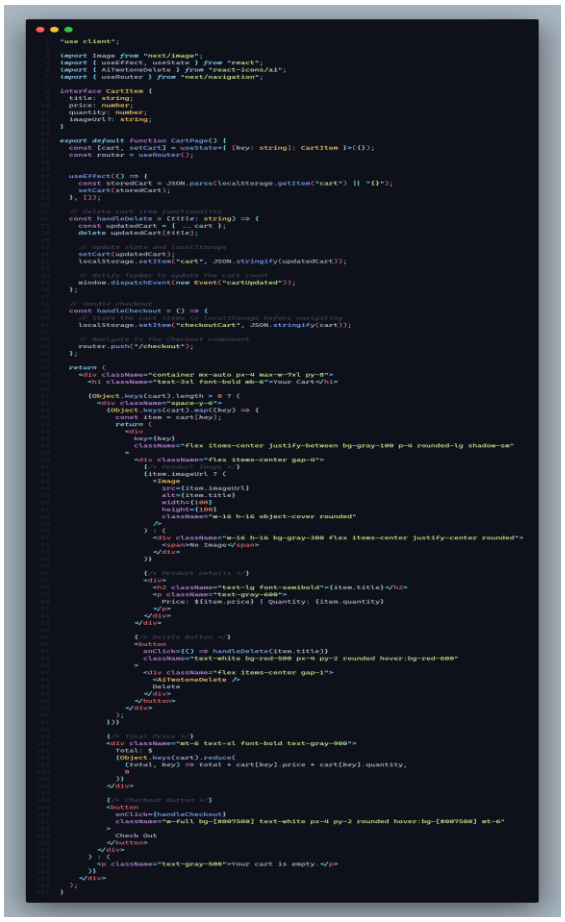
Ivory Charm
Price: $20 | Quantity: 4

Modern Cozy
Price: $20 | Quantity: 2

SleekSpin
Price: $20 | Quantity: 2

Gray Elegance
Price: $8 | Quantity: 1

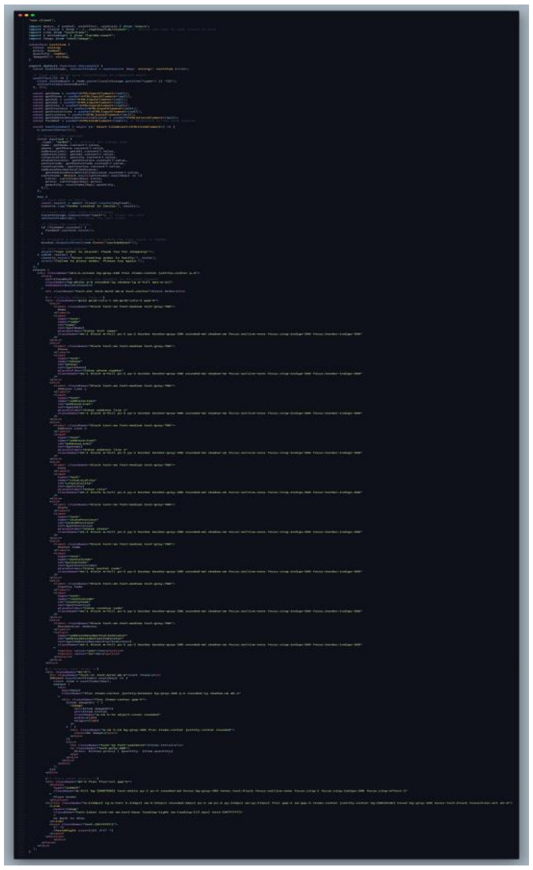
Total: $168



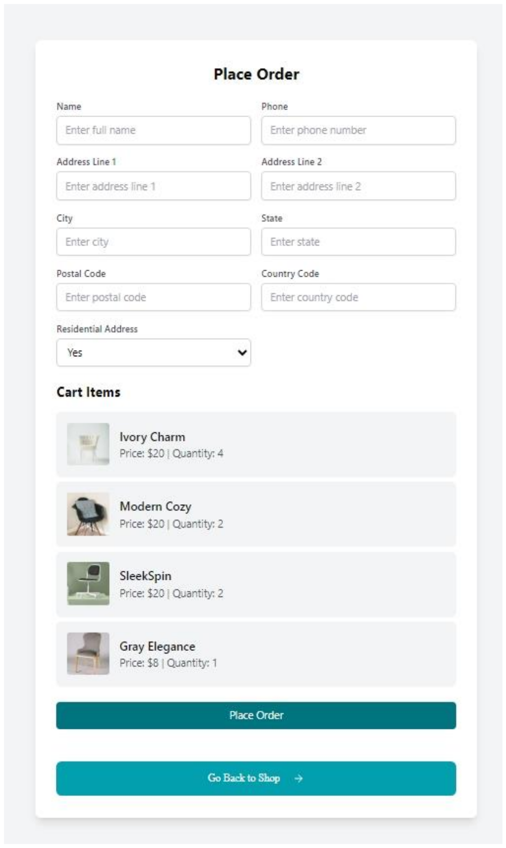
```
"use client";

import Image from "next/image";
import { useEffect, useState } from "react";
import { AiTwotoneDelete } from "react-icons/ai";
import { useRouter } from "next/navigation";

interface CartItem {
  title: string;
  price: number;
  quantity: number;
  imageUrl?: string;
}

export default function CartPage() {
  const [cart, setCart] = useState<{ [key: string]: CartItem }>({});
  const router = useRouter();


  useEffect(() => {
    const storedCart = JSON.parse(localStorage.getItem("cart") || "{}");
    setCart(storedCart);
  }, []);

  // Delete cart item functionality
  const handleDelete = (title: string) => {
    const updatedCart = { ...cart };
    delete updatedCart[title];

    // Update state and localStorage
    setCart(updatedCart);
    localStorage.setItem("cart", JSON.stringify(updatedCart));

    // Notify TopBar to update the cart count
    window.dispatchEvent(new Event("cartUpdated"));
  };

  // Handle checkout
  const handleCheckout = () => {
    // Store the cart items in localStorage before navigating
    localStorage.setItem("checkoutCart", JSON.stringify(cart));

    // Navigate to the Checkout component
    router.push("/checkout");
  };

  return (
    <div className="container mx-auto px-4 max-w-7xl py-8">
      <h1 className="text-3xl font-bold mb-6">Your Cart</h1>

      {Object.keys(cart).length > 0 ? (
        <div className="space-y-6">
          {Object.keys(cart).map((key) => {
            const item = cart[key];
            return (
              <div
                key={key}
                className="flex items-center justify-between bg-gray-100 p-4 rounded-lg shadow-sm"
              >
                <div className="flex items-center gap-4">
                  {/* Product Image */}
                  {item.imageUrl ? (
                    <Image
                      src={item.imageUrl}
                      alt={item.title}
                      width={100}
                      height={100}
                      className="w-16 h-16 object-cover rounded"
                    />
                  ) : (
                    <div className="w-16 h-16 bg-gray-300 flex items-center justify-center rounded">
                      <span>No Image</span>
                    </div>
                  )}

                  {/* Product Details */}
                  <div>
                    <h2 className="text-lg font-semibold">{item.title}</h2>
                    <p className="text-gray-600">
                      Price: ${item.price} | Quantity: {item.quantity}
                    </p>
                  </div>
                </div>

                {/* Delete Button */}
                <button
                  onClick={() => handleDelete(item.title)}
                  className="text-white bg-red-500 px-4 py-2 rounded hover:bg-red-600"
                >
                  <div className="flex items-center gap-1">
                    <AiTwotoneDelete />
                    Delete
                  </div>
                </button>
              </div>
            );
          })}

          {/* Total Price */}
          <div className="mt-6 text-xl font-bold text-gray-900">
            Total: $
            {Object.keys(cart).reduce(
              (total, key) => total + cart[key].price * cart[key].quantity,
              0
            )}
          </div>

          {/* Checkout Button */}
          <button
            onClick={handleCheckout}
            className="w-full bg-[#007580] text-white px-4 py-2 rounded hover:bg-[#007580] mt-6"
          >
            Check Out
          </button>
        </div>
      ) : (
        <p className="text-gray-500">Your cart is empty.</p>
      )}
    </div>
  );
}
```

5. Checkout Component:

   In this page user can provide the details and place their order successfully and when their order is placed successfully, user receive a pop up of thank you for shopping and the cart items will be empty.



```
[illegible]
```



Place Order

Name: Enter full name
Phone: Enter phone number
Address Line 1: Enter address line 1
Address Line 2: Enter address line 2
City: Enter city
State: Enter state
Postal Code: Enter postal code
Country Code: Enter country code
Residential Address: Yes

Cart Items

Ivory Charm
Price: $20 | Quantity: 4

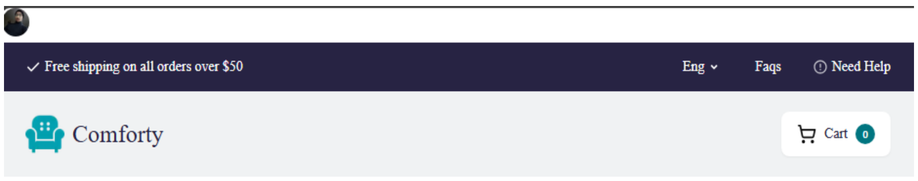
Modern Cozy
Price: $20 | Quantity: 2

SleekSpin
Price: $20 | Quantity: 2

Gray Elegance
Price: $8 | Quantity: 1

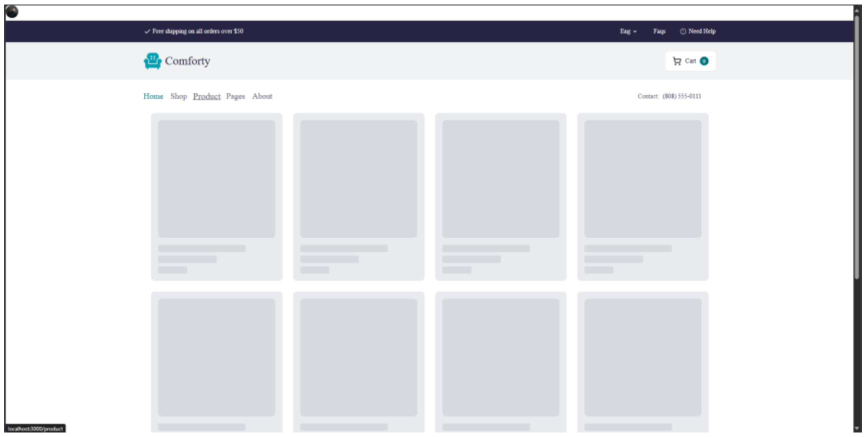
6. Implemented sign in functionality:

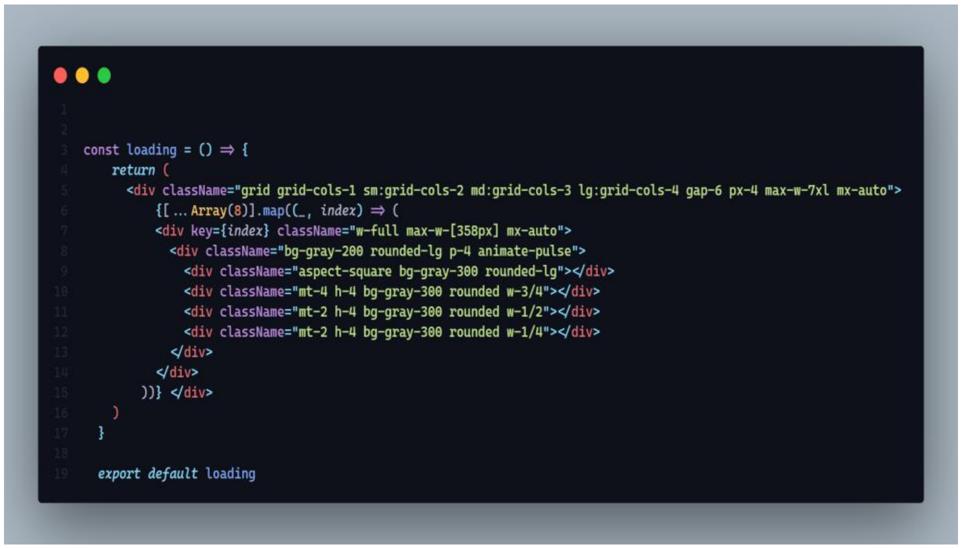
User can sign in into the website from their google or GitHub account. And after sign in, can see the detail of their profile.



7. Skeleton Loader:

Created a skeleton loader in loading.tsx file. So, while fetching when component mount, at that time skeleton loader will appear on the UI for good user experience.

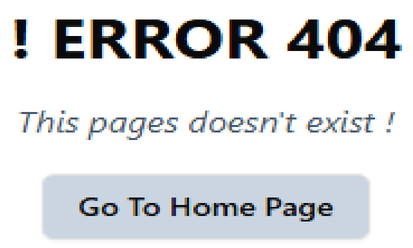




```
const loading = () ⇒ {
   return (
     <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6 px-4 max-w-7xl mx-auto">
         {[ ... Array(8)].map((_, index) ⇒ (
         <div key={index} className="w-full max-w-[358px] mx-auto">
           <div className="bg-gray-200 rounded-lg p-4 animate-pulse">
             <div className="aspect-square bg-gray-300 rounded-lg"></div>
             <div className="mt-4 h-4 bg-gray-300 rounded w-3/4"></div>
             <div className="mt-2 h-4 bg-gray-300 rounded w-1/2"></div>
             <div className="mt-2 h-4 bg-gray-300 rounded w-1/4"></div>
           </div>
         </div>
       ))} </div>
   )
 }

 export default loading
```

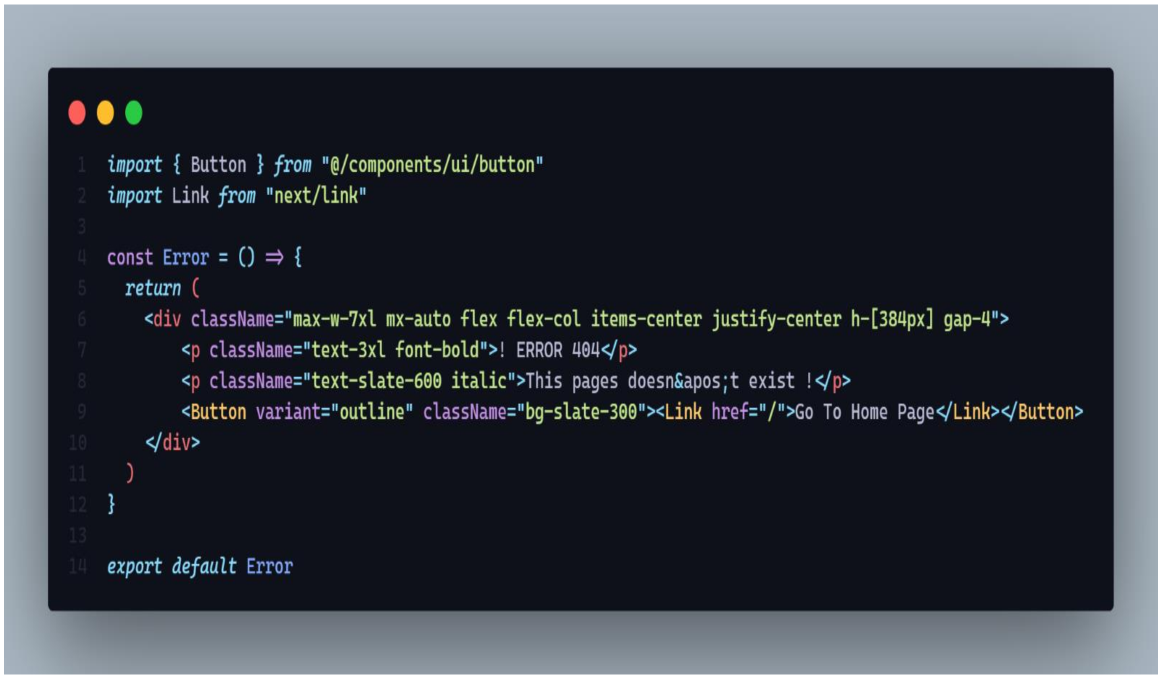
8. Custom 404 – not found page:

   Also create a custom 404 page in not-found.tsx for seamless user experience. Providing a button to go to home page if route is incorrect.



! ERROR 404

This pages doesn't exist !



```
import { Button } from "@/components/ui/button"
import Link from "next/link"

const Error = () ⇒ {
  return (
    <div className="max-w-7xl mx-auto flex flex-col items-center justify-center h-[384px] gap-4">
        <p className="text-3xl font-bold">! ERROR 404</p>
        <p className="text-slate-600 italic">This pages doesn&apos;t exist !</p>
        <Button variant="outline" className="bg-slate-300"><Link href="/">Go To Home Page</Link></Button>
    </div>
  )
}

export default Error
```

And while doing all the things I also make sure that all the pages should be responsive and correctly structured.

THANK YOU!