HACKATHON 3

Day 4

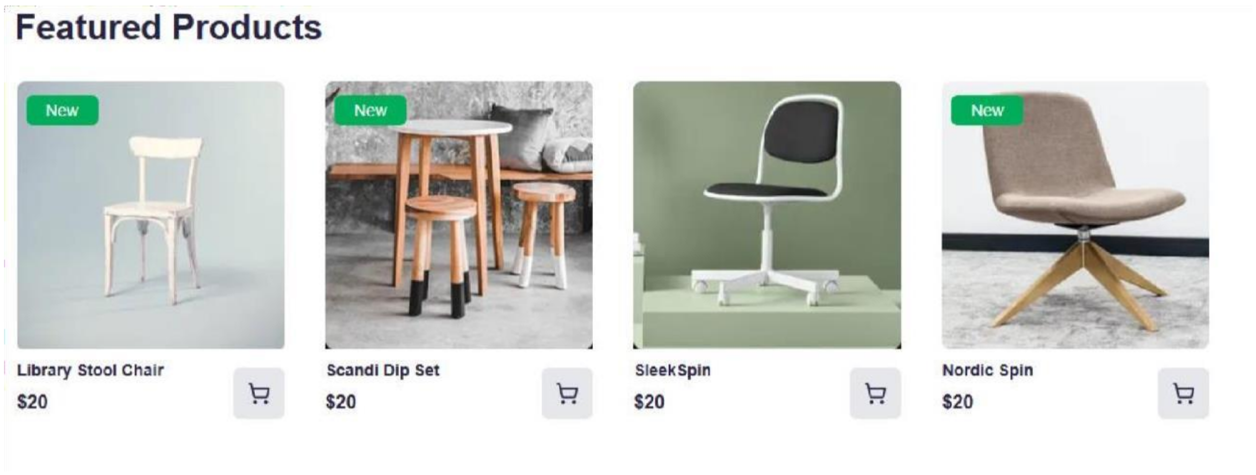
Building Dynamic Front-end Component

Introduction:

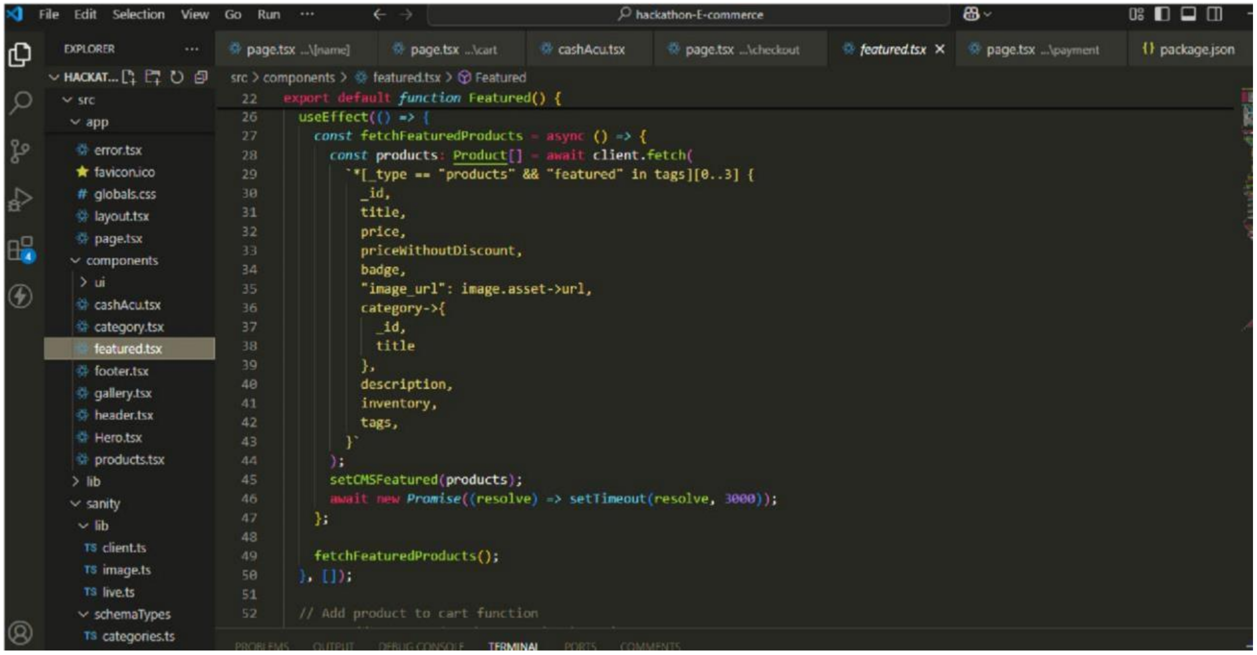
 The purpose of e-commerce is to provide a convenient platform for people to buy and sell products and services online. It allows businesses to reach customers directly, offering them convenience and a wide variety of options. (e.g., Product selling, Online Store, etc.)

Product Listing Component

Render product data dynamically in a grid layout cards displaying product details.



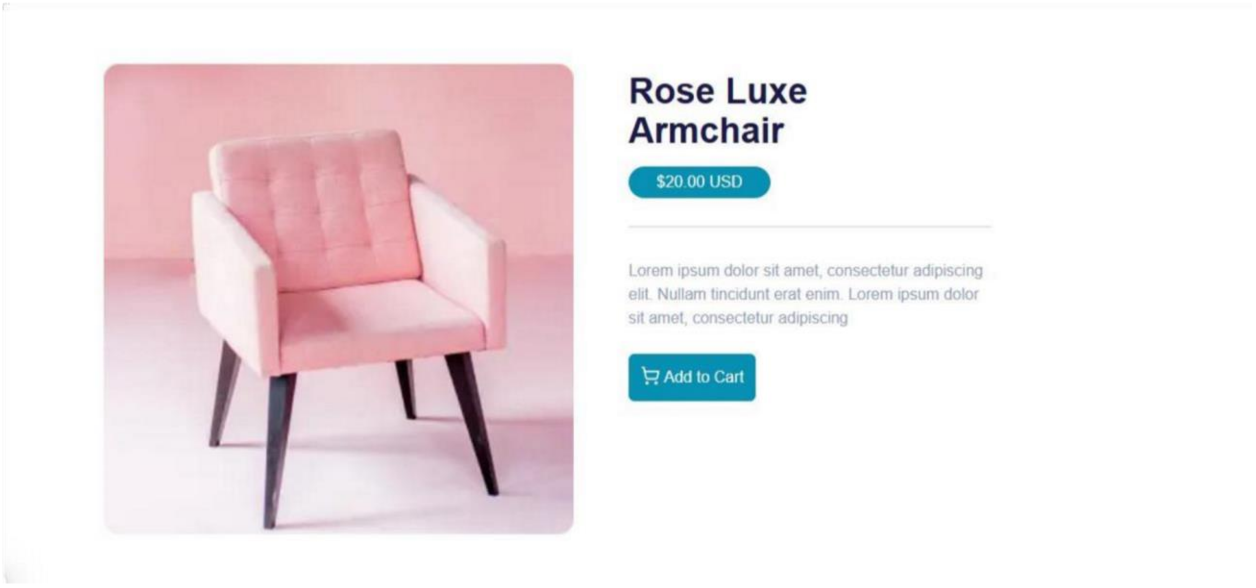
Product Listing Component Code



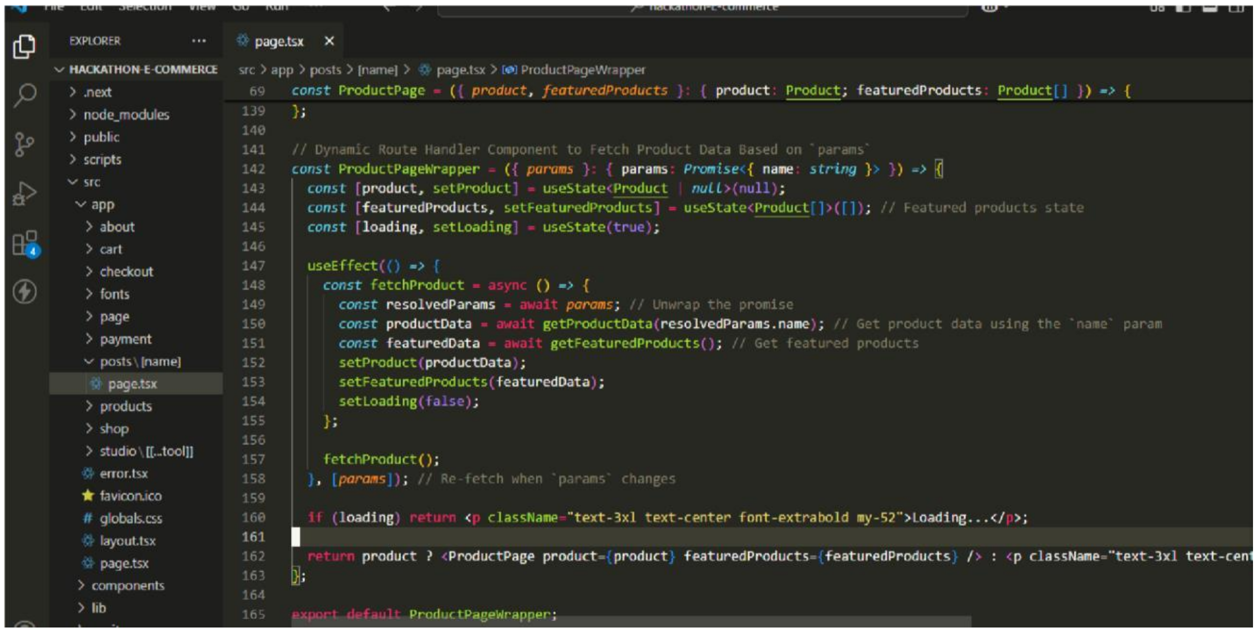
```
export default function Featured() {
  useEffect(() => {
    const fetchFeaturedProducts = async () => {
      const products: Product[] = await client.fetch(
        `*[_type == "products" && "featured" in tags][0..3] {
          _id,
          title,
          price,
          priceWithoutDiscount,
          badge,
          "image_url": image.asset->url,
          category->{
            _id,
            title
          },
          description,
          inventory,
          tags,
        }`
      );
      setCMSFeatured(products);
      await new Promise((resolve) => setTimeout(resolve, 3000));
    };

    fetchFeaturedProducts();
  }, []);

  // Add product to cart function
```

Product Detail

Create individual product detail pages using dynamic routing in Next.js.


Rose Luxe Armchair
$20.00 USD
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tincidunt erat enim. Lorem ipsum dolor sit amet, consectetur adipiscing
Add to Cart

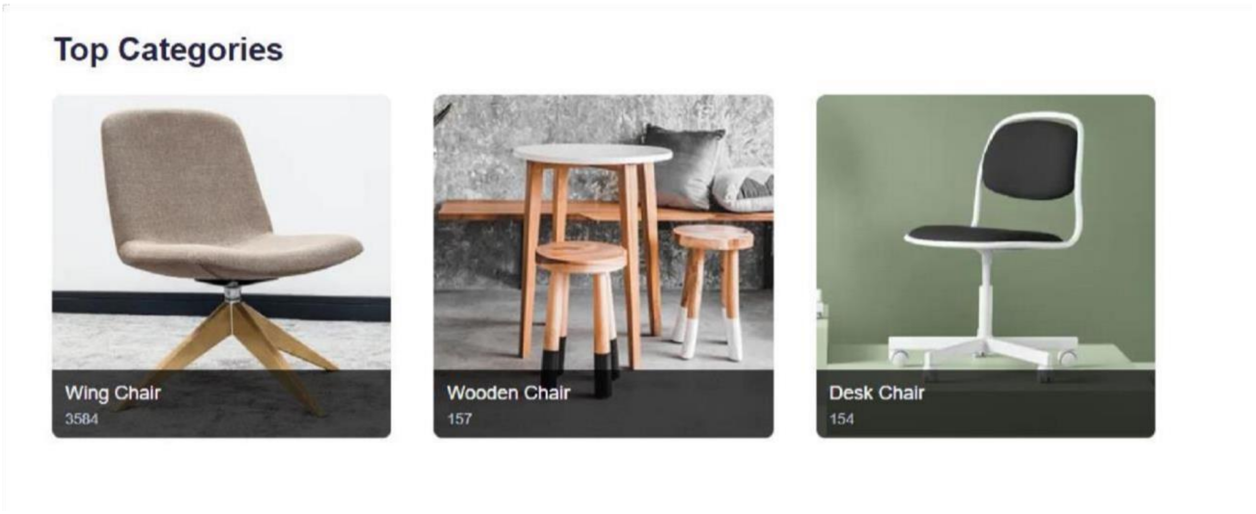
Product Detail Code



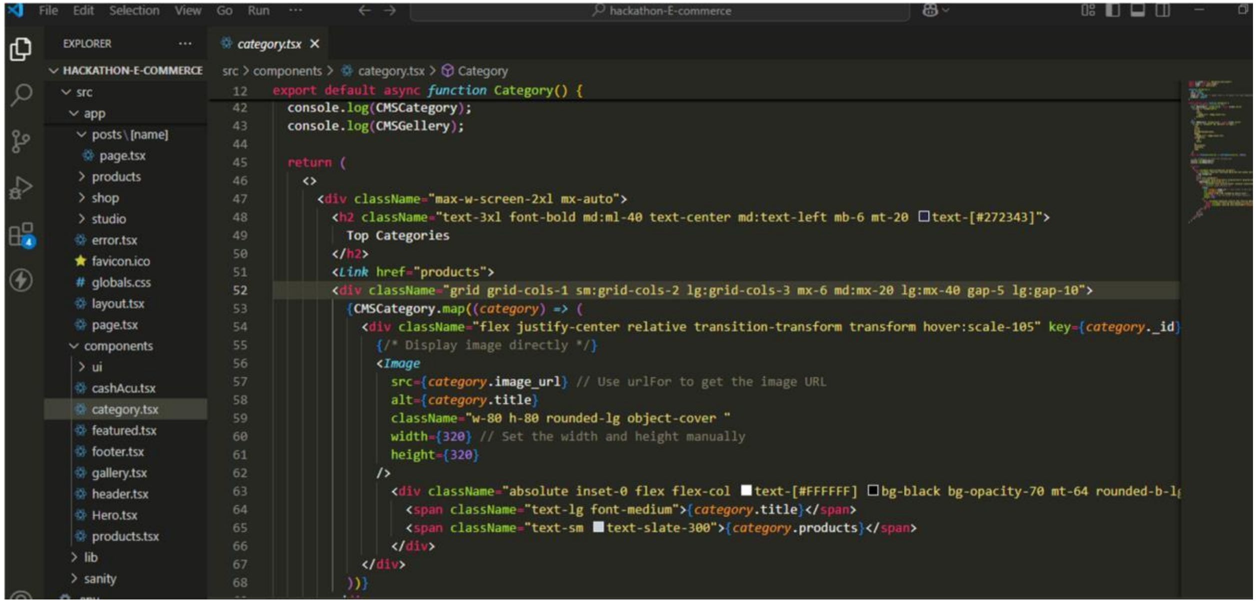
```tsx
const ProductPage = ({ product, featuredProducts }: { product: Product; featuredProducts: Product[] }) => {
};

// Dynamic Route Handler Component to Fetch Product Data Based on `params`
const ProductPageWrapper = ({ params }: { params: Promise<{ name: string }> }) => {
  const [product, setProduct] = useState<Product | null>(null);
  const [featuredProducts, setFeaturedProducts] = useState<Product[]>([]); // Featured products state
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    const fetchProduct = async () => {
      const resolvedParams = await params; // Unwrap the promise
      const productData = await getProductData(resolvedParams.name); // Get product data using the `name` param
      const featuredData = await getFeaturedProducts(); // Get featured products
      setProduct(productData);
      setFeaturedProducts(featuredData);
      setLoading(false);
    };

    fetchProduct();
  }, [params]); // Re-fetch when `params` changes

  if (loading) return <p className="text-3xl text-center font-extrabold my-52">Loading...</p>;

  return product ? <ProductPage product={product} featuredProducts={featuredProducts} /> : <p className="text-3xl text-cent
};

export default ProductPageWrapper;
```

Category Component:

Display categories dynamically fetched from the data source. Enable filtering of products by selected categories.



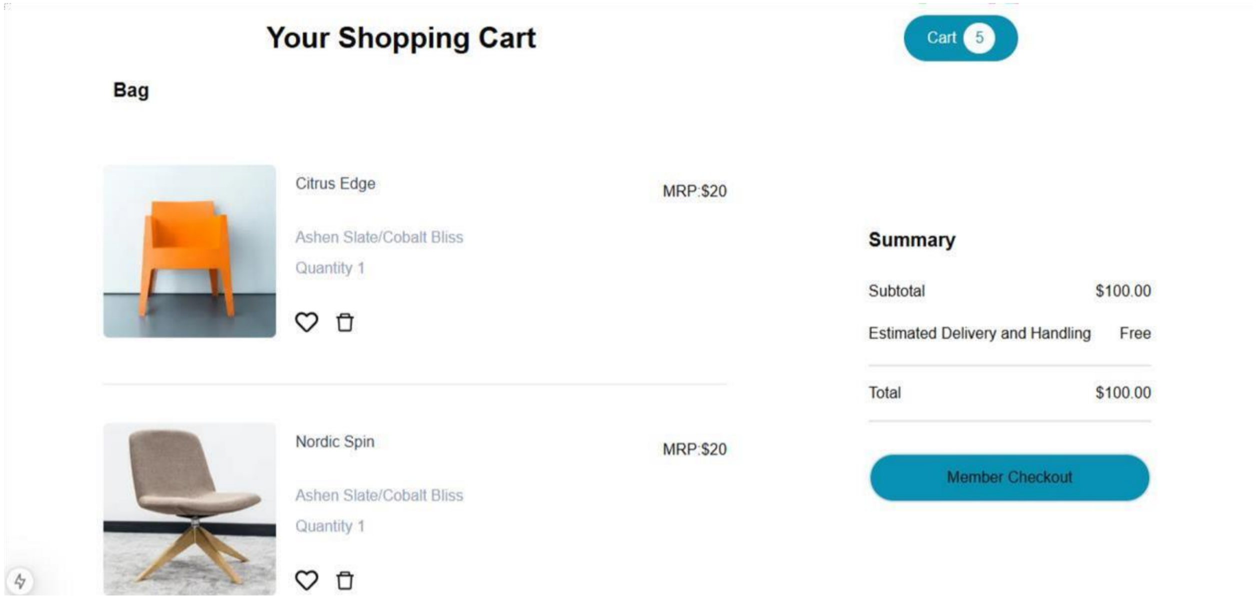
Category Component Code:



```tsx
export default async function Category() {
  console.log(CMSCategory);
  console.log(CMSGellery);

  return (
    <>
      <div className="max-w-screen-2xl mx-auto">
        <h2 className="text-3xl font-bold md:ml-40 text-center md:text-left mb-6 mt-20 text-[#272343]">
          Top Categories
        </h2>
        <Link href="products">
        <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 mx-6 md:mx-20 lg:mx-40 gap-5 lg:gap-10">
          {CMSCategory.map((category) => (
            <div className="flex justify-center relative transition-transform transform hover:scale-105" key={category._id}
              {/* Display image directly */}
              <Image
                src={category.image_url} // Use urlFor to get the image URL
                alt={category.title}
                className="w-80 h-80 rounded-lg object-cover "
                width={320} // Set the width and height manually
                height={320}
              />
                <div className="absolute inset-0 flex flex-col text-[#FFFFFF] bg-black bg-opacity-70 mt-64 rounded-b-l
                  <span className="text-lg font-medium">{category.title}</span>
                  <span className="text-sm text-slate-300">{category.products}</span>
                </div>
            </div>
          ))}
```

Cart Component:

Display added items, quantity, and total price. Use state management for tracking cart items.

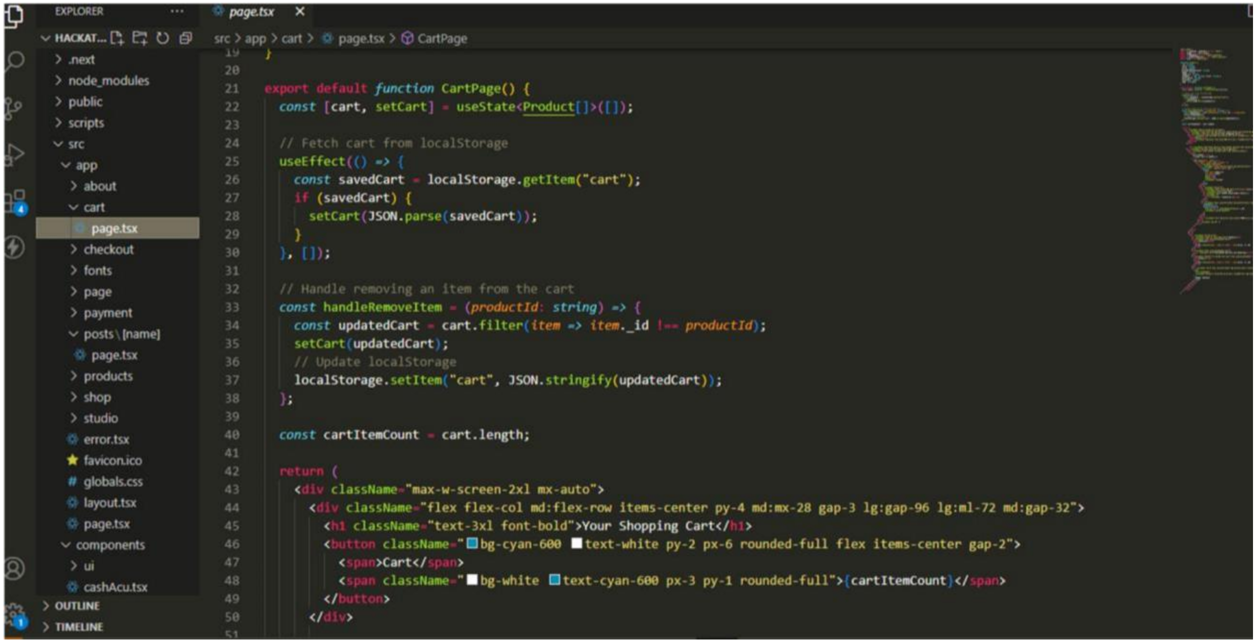


Citrus Edge MRP:$20
Ashen Slate/Cobalt Bliss
Quantity 1

Nordic Spin MRP:$20
Ashen Slate/Cobalt Bliss
Quantity 1

Summary
Subtotal $100.00
Estimated Delivery and Handling Free
Total $100.00

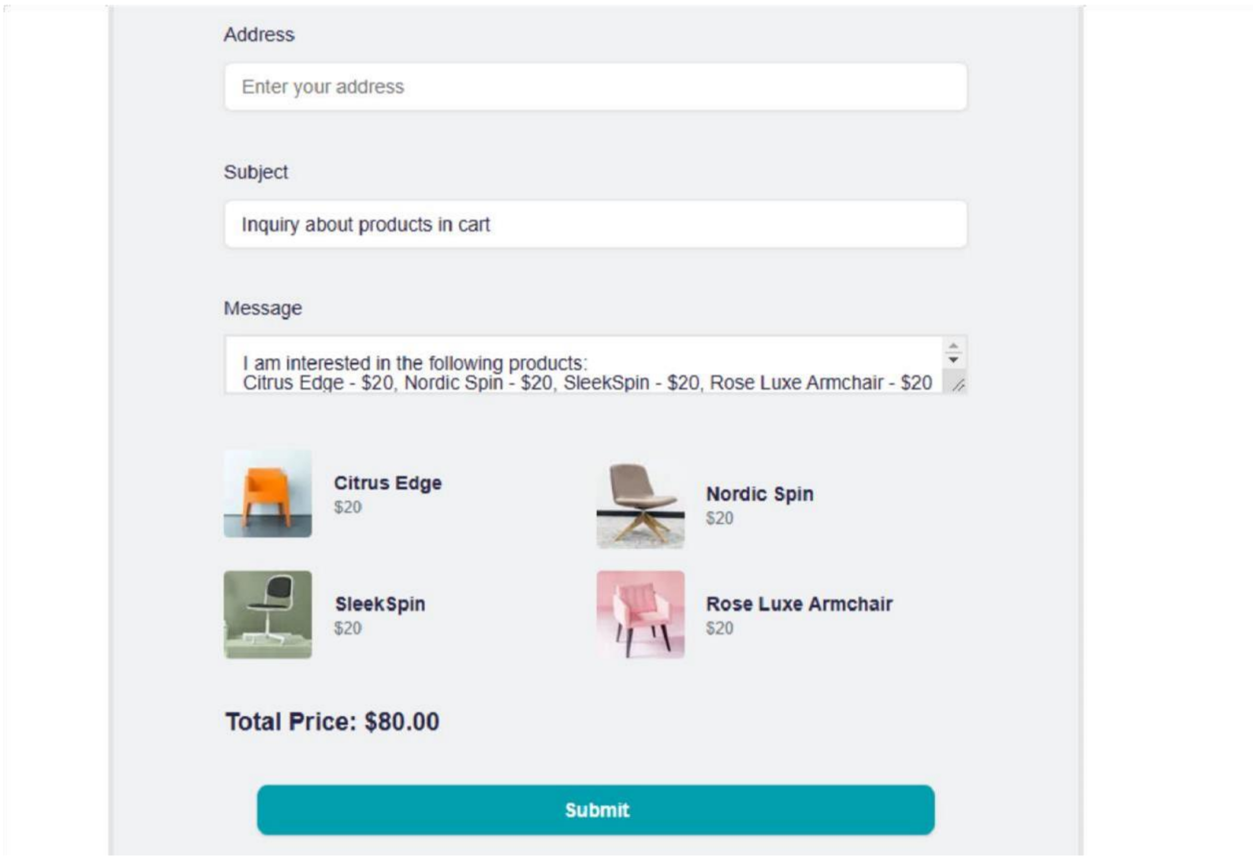
Cart Component Code:



```tsx
}

export default function CartPage() {
  const [cart, setCart] = useState<Product[]>([]);

  // Fetch cart from localStorage
  useEffect(() => {
    const savedCart = localStorage.getItem("cart");
    if (savedCart) {
      setCart(JSON.parse(savedCart));
    }
  }, []);

  // Handle removing an item from the cart
  const handleRemoveItem = (productId: string) => {
    const updatedCart = cart.filter(item => item._id !== productId);
    setCart(updatedCart);
    // Update localStorage
    localStorage.setItem("cart", JSON.stringify(updatedCart));
  };

  const cartItemCount = cart.length;

  return (
    <div className="max-w-screen-2xl mx-auto">
      <div className="flex flex-col md:flex-row items-center py-4 md:mx-28 gap-3 lg:gap-96 lg:ml-72 md:gap-32">
        <h1 className="text-3xl font-bold">Your Shopping Cart</h1>
        <button className="bg-cyan-600 text-white py-2 px-6 rounded-full flex items-center gap-2">
          <span>Cart</span>
          <span className="bg-white text-cyan-600 px-3 py-1 rounded-full">{cartItemCount}</span>
        </button>
      </div>
```

Checkout Flow Component:

Create a multi-step form for checkout, including fields for Billing and shipping address Payment details (mock implementation).



Address

Enter your address

Subject

Inquiry about products in cart

Message

I am interested in the following products:
Citrus Edge - $20, Nordic Spin - $20, SleekSpin - $20, Rose Luxe Armchair - $20

Citrus Edge
$20

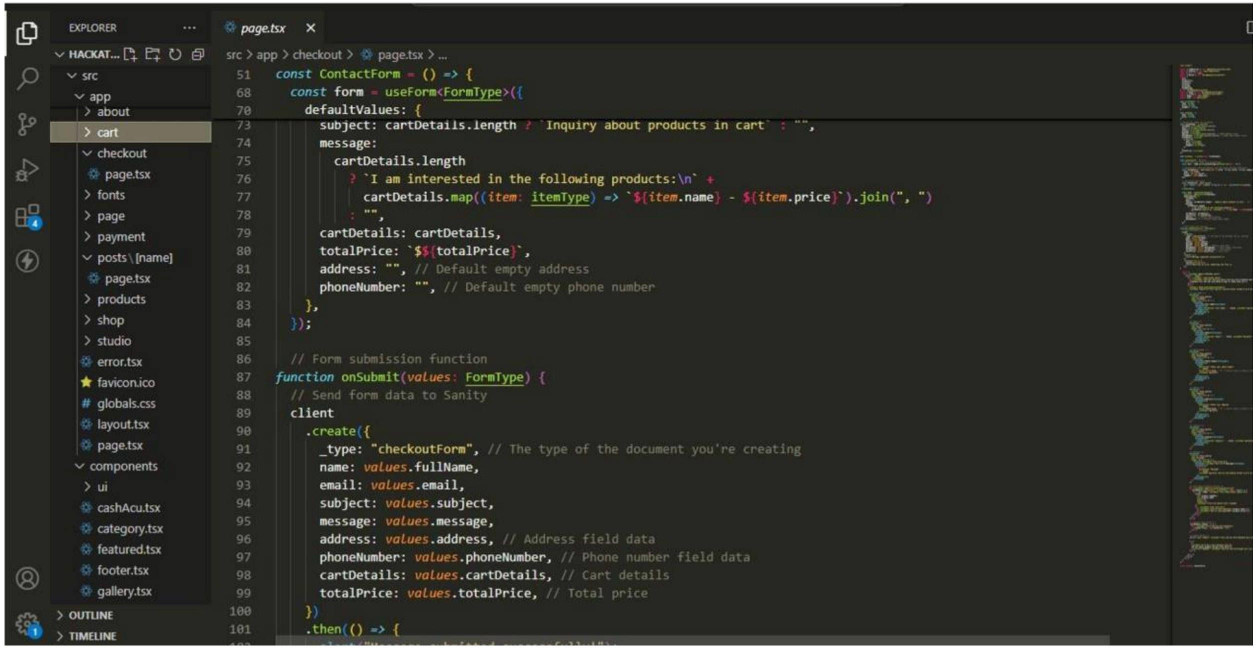
Nordic Spin
$20

SleekSpin
$20

Rose Luxe Armchair
$20

Total Price: $80.00

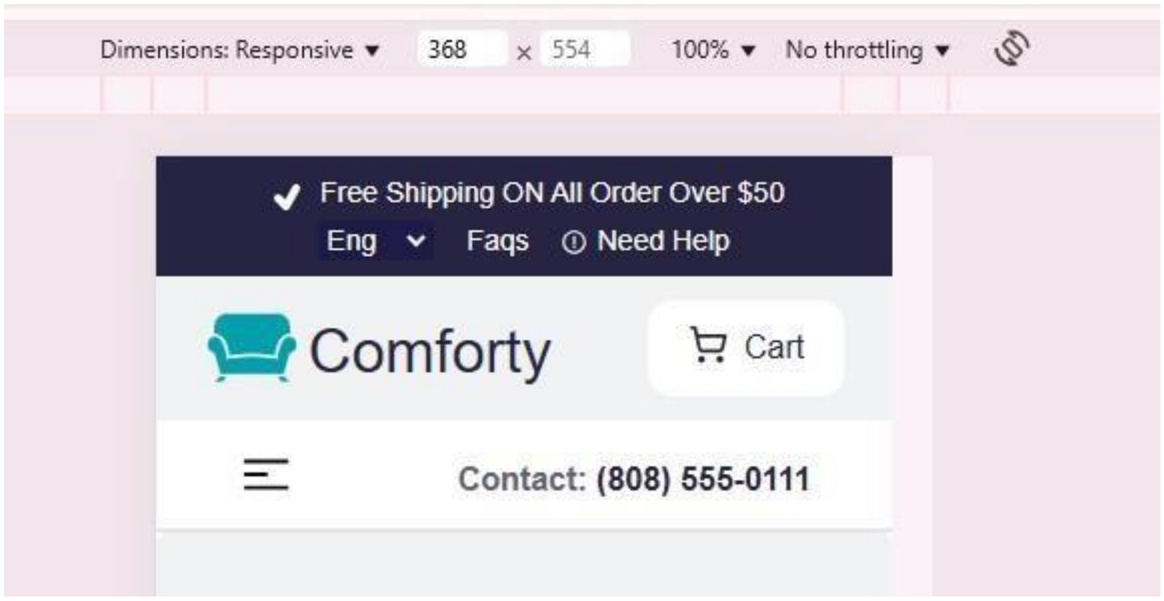
Checkout Flow Component Code:



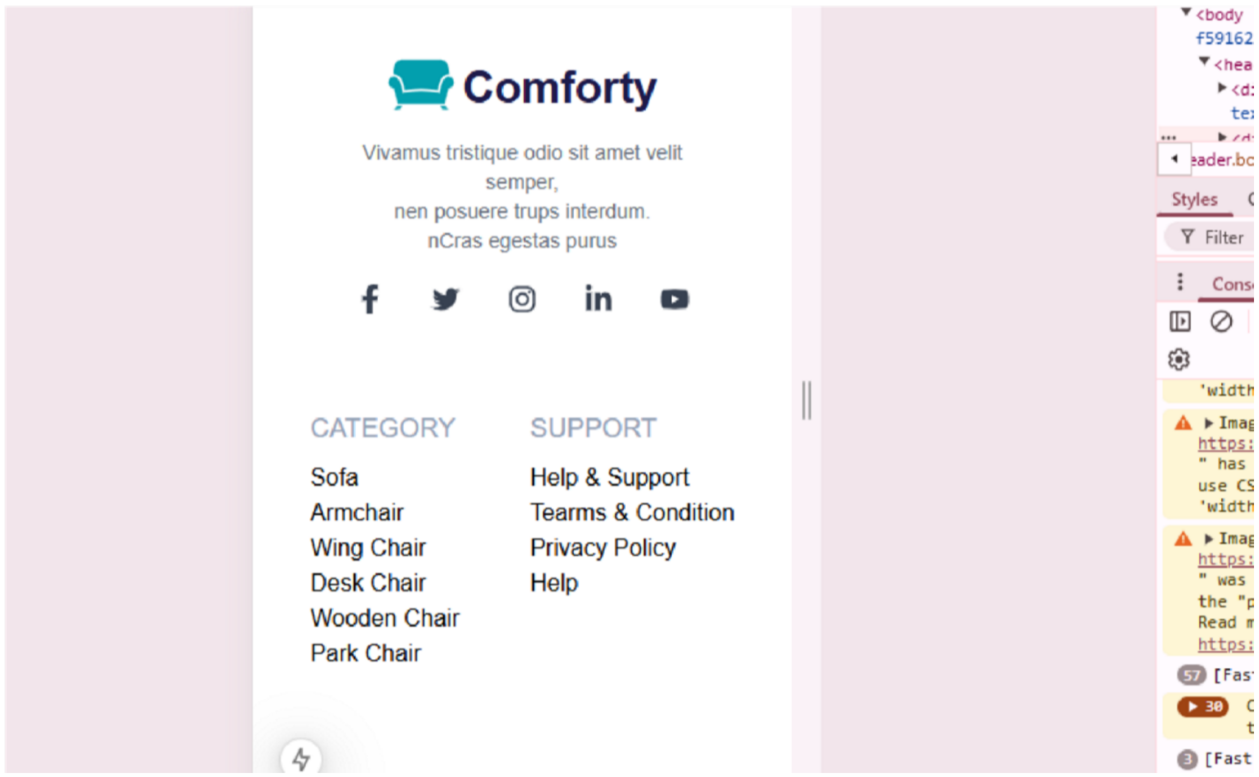
```tsx
const ContactForm = () => {
  const form = useForm<FormType>({
    defaultValues: {
      subject: cartDetails.length ? `Inquiry about products in cart` : "",
      message:
        cartDetails.length
          ? `I am interested in the following products:\n` +
            cartDetails.map((item: itemType) => `${item.name} - ${item.price}`).join(", ")
          : "",
      cartDetails: cartDetails,
      totalPrice: `$${totalPrice}`,
      address: "", // Default empty address
      phoneNumber: "", // Default empty phone number
    },
  });

  // Form submission function
function onSubmit(values: FormType) {
  // Send form data to Sanity
  client
    .create({
      _type: "checkoutForm", // The type of the document you're creating
      name: values.fullName,
      email: values.email,
      subject: values.subject,
      message: values.message,
      address: values.address, // Address field data
      phoneNumber: values.phoneNumber, // Phone number field data
      cartDetails: values.cartDetails, // Cart details
      totalPrice: values.totalPrice, // Total price
    })
    .then(() => {
```

Footer and Header Components:

Build consistent navigation and branding elements. responsiveness and accessibility.



Footer

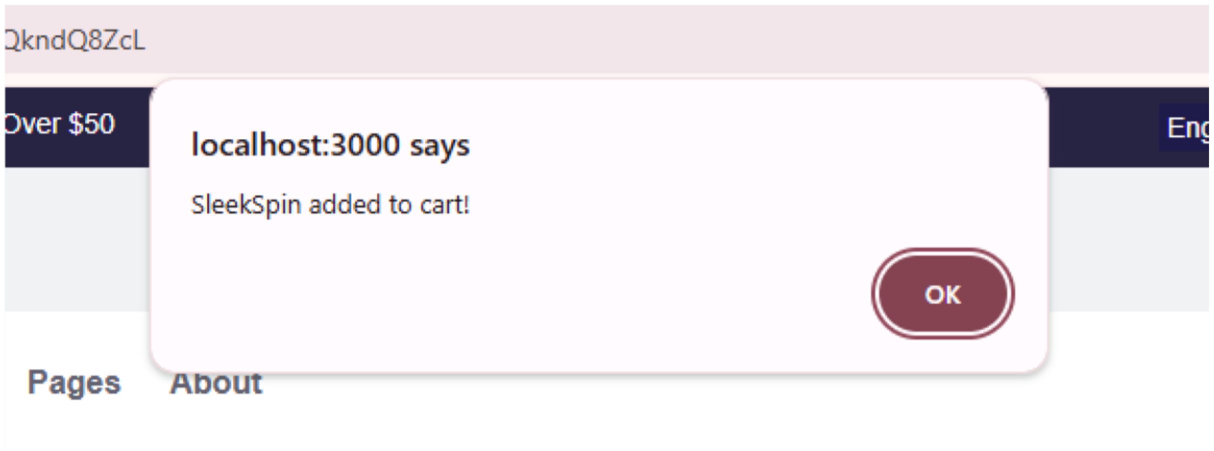

Comforty

Vivamus tristique odio sit amet velit semper,
nen posuere trups interdum.
nCras egestas purus

CATEGORY	SUPPORT
Sofa	Help & Support
Armchair	Teams & Condition
Wing Chair	Privacy Policy
Desk Chair	Help
Wooden Chair	
Park Chair	

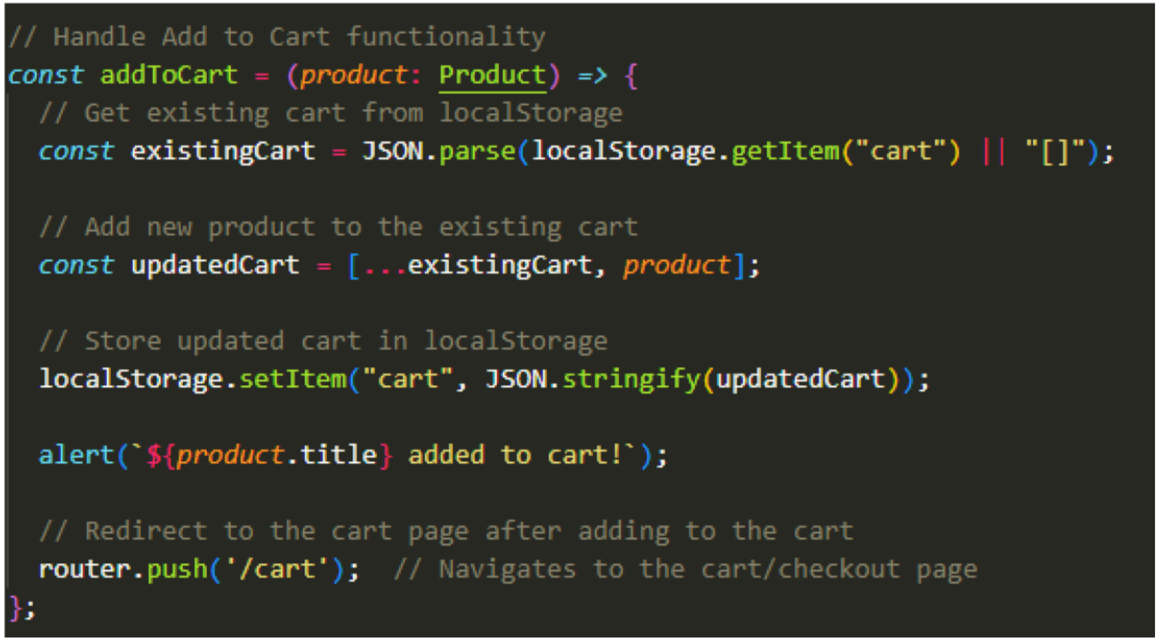
Notifications Component:

Show real-time alerts for actions like adding to cart, errors, or successful purchases. Use toast notifications or modal windows


localhost:3000 says

SleekSpin added to cart!

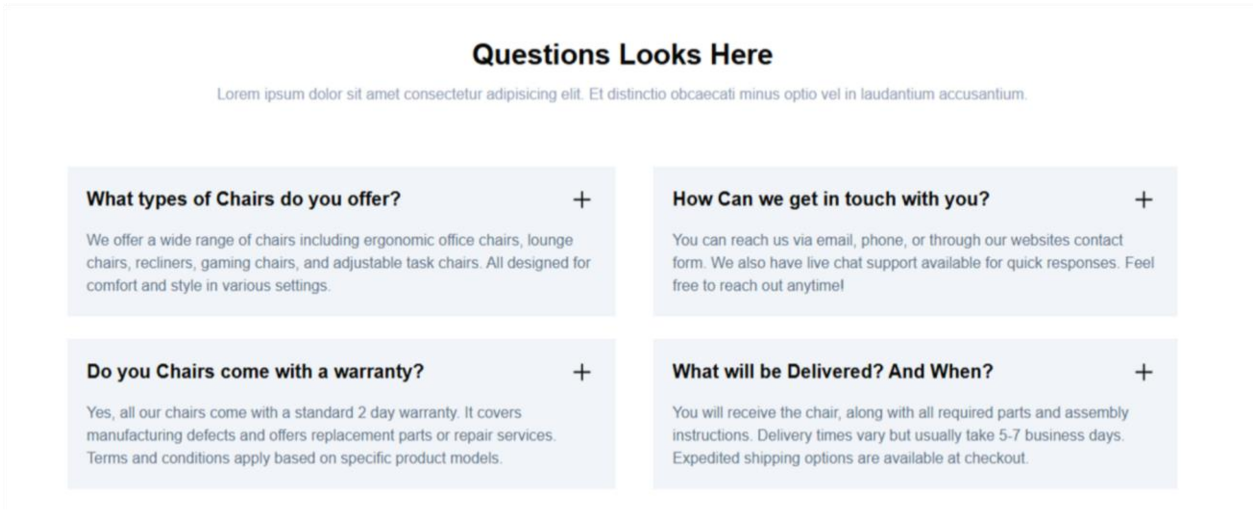
Notifications Component Code:



```
// Handle Add to Cart functionality
const addToCart = (product: Product) => {
  // Get existing cart from localStorage
  const existingCart = JSON.parse(localStorage.getItem("cart") || "[]");

  // Add new product to the existing cart
  const updatedCart = [...existingCart, product];

  // Store updated cart in localStorage
  localStorage.setItem("cart", JSON.stringify(updatedCart));

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

  // Redirect to the cart page after adding to the cart
  router.push('/cart');  // Navigates to the cart/checkout page
};
```

FAQ and Help Center Component:

Include a searchable FAQ section. Add contact forms or chatbot integrations for support.



Questions Looks Here

Lorem ipsum dolor sit amet consectetur adipisicing elit. Et distinctio obcaecati minus optio vel in laudantium accusantium.

What types of Chairs do you offer? +

We offer a wide range of chairs including ergonomic office chairs, lounge chairs, recliners, gaming chairs, and adjustable task chairs. All designed for comfort and style in various settings.

How Can we get in touch with you? +

You can reach us via email, phone, or through our websites contact form. We also have live chat support available for quick responses. Feel free to reach out anytime!

Do you Chairs come with a warranty? +

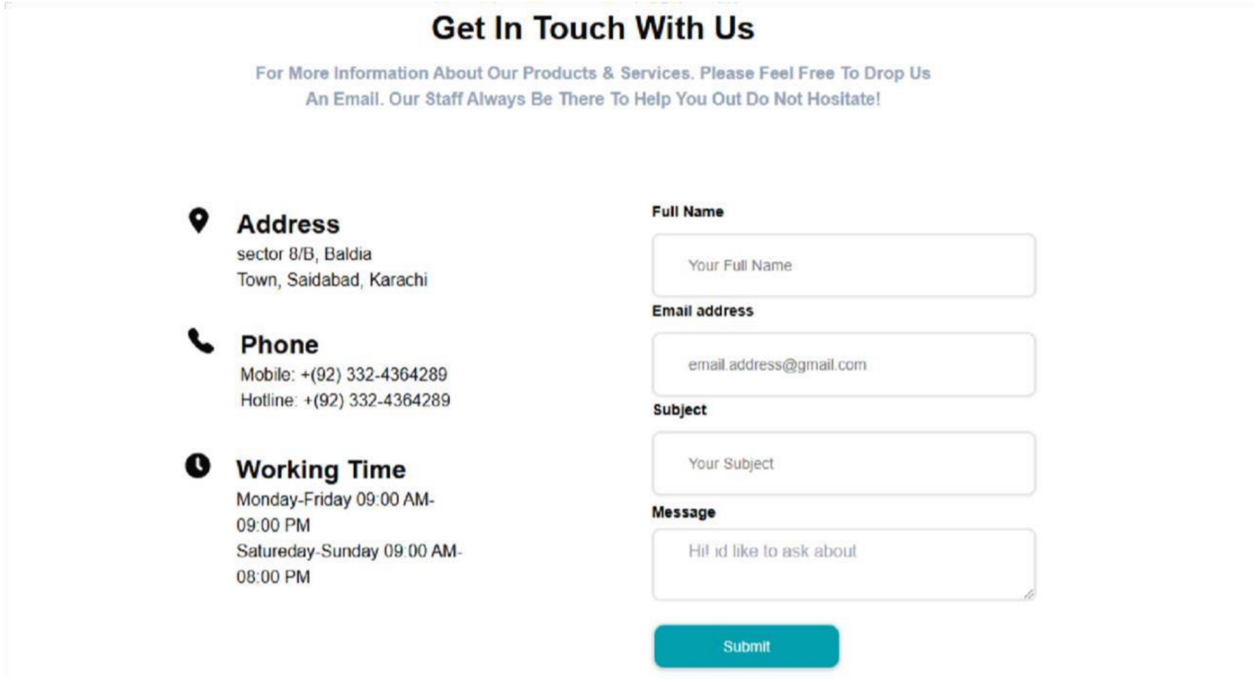
Yes, all our chairs come with a standard 2 day warranty. It covers manufacturing defects and offers replacement parts or repair services. Terms and conditions apply based on specific product models.

What will be Delivered? And When? +

You will receive the chair, along with all required parts and assembly instructions. Delivery times vary but usually take 5-7 business days. Expedited shipping options are available at checkout.

Help Center:

It form is work after submit data push on sanity.


Get In Touch With Us

For More Information About Our Products & Services. Please Feel Free To Drop Us An Email. Our Staff Always Be There To Help You Out Do Not Hositate!

Address
sector 8/B, Baldia Town, Saidabad, Karachi

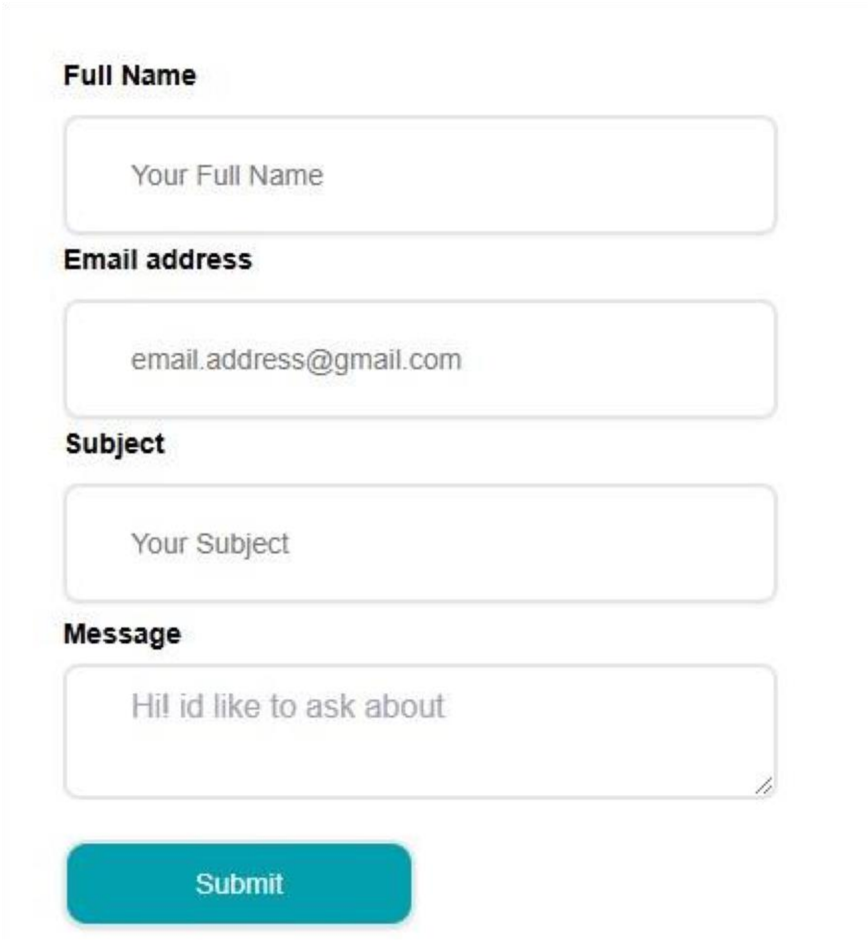
Phone
Mobile: +(92) 332-4364289
Hotline: +(92) 332-4364289

Working Time
Monday-Friday 09:00 AM-09:00 PM
Satureday-Sunday 09:00 AM-08:00 PM

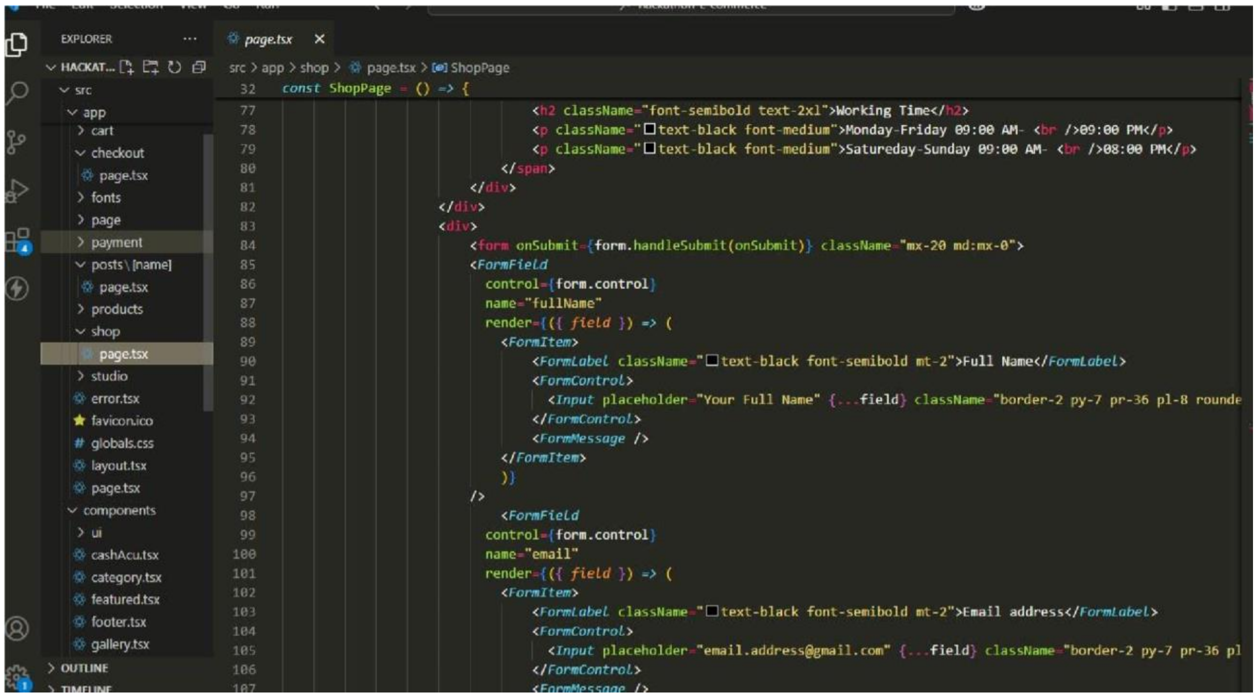
Full Name
Your Full Name
Email address
email.address@gmail.com
Subject
Your Subject
Message
Hi! id like to ask about
Submit

Customer Feedback Component:

Create a form for users to submit feedback about the marketplace or specific products. Display aggregated feedback for admins to review.


Full Name
Your Full Name
Email address
email.address@gmail.com
Subject
Your Subject
Message
Hi! id like to ask about
Submit

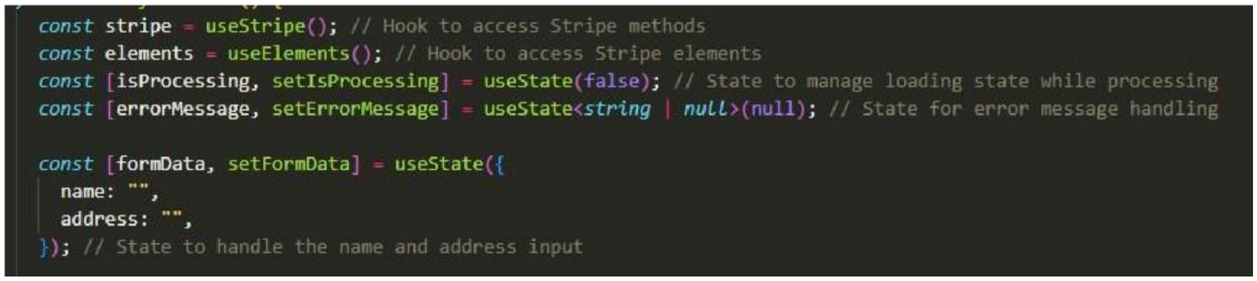
Feedback Code:



```
const ShopPage = () => {
                <h2 className="font-semibold text-2xl">Working Time</h2>
                <p className="text-black font-medium">Monday-Friday 09:00 AM- <br />09:00 PM</p>
                <p className="text-black font-medium">Satureday-Sunday 09:00 AM- <br />08:00 PM</p>
              </span>
            </div>
          </div>
          <div>
            <form onSubmit={form.handleSubmit(onSubmit)} className="mx-20 md:mx-0">
            <FormField
             control={form.control}
             name="fullName"
             render={({ field }) => (
               <FormItem>
                 <FormLabel className="text-black font-semibold mt-2">Full Name</FormLabel>
                 <FormControl>
                   <Input placeholder="Your Full Name" {...field} className="border-2 py-7 pr-36 pl-8 rounde
                 </FormControl>
                 <FormMessage />
               </FormItem>
               )}
            />
               <FormField
             control={form.control}
             name="email"
             render={({ field }) => (
               <FormItem>
                 <FormLabel className="text-black font-semibold mt-2">Email address</FormLabel>
                 <FormControl>
                   <Input placeholder="email.address@gmail.com" {...field} className="border-2 py-7 pr-36 pl
                 </FormControl>
                 <FormMessage />
```

State Management:

Use React state or context to manage data across components. Example: Use useState for local component state and useContext for global state.



```
const stripe = useStripe(); // Hook to access Stripe methods
const elements = useElements(); // Hook to access Stripe elements
const [isProcessing, setIsProcessing] = useState(false); // State to manage loading state while processing
const [errorMessage, setErrorMessage] = useState<string | null>(null); // State for error message handling

const [formData, setFormData] = useState({
  name: "",
  address: "",
}); // State to handle the name and address input
```

Step Taken to Build and Integrate Component:

- **Front-end Development:**
  - Create a website Design (e.g., homepage, product page, card, etc.)
  - Using the framework Next.js Front-end Technology. o Build a responsive design Using Tailwind CSS.

Back-end Development:

- We will Learn take after Hackathon Authentication and Autherization. Now I use backend sanity CMS technology.

Payment Gateway Integration:

- Payment Gateway (e.g., jaazcash, eeasypaisa) integrate user can easily Payment.

Component Integration:

- Card system, and Checkout Process Integrate.

Conclusive:

In conclusion, e-commerce offers a user-friendly experience by providing easy access to a wide range of products and services from the comfort of home. It streamlines transactions, making shopping convenient, fast, and efficient.