

Hackathon Day 04

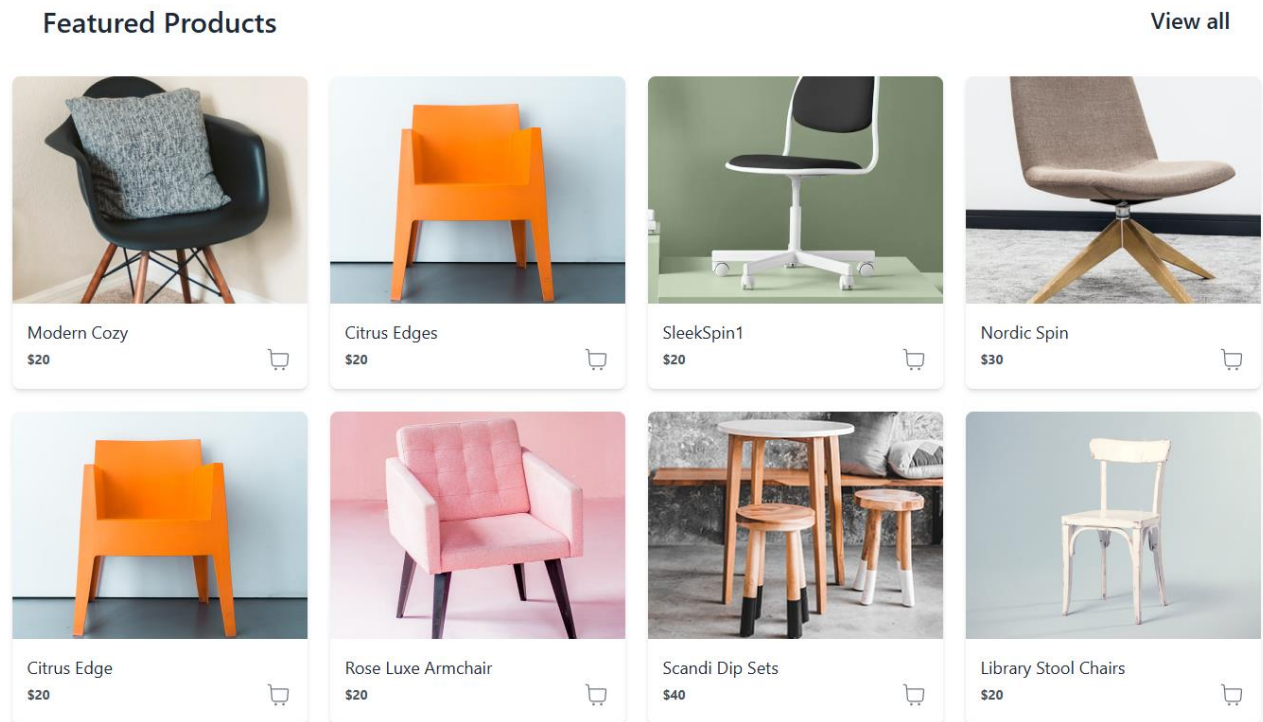
BUILDING DYNAMIC FRONTEND COMPONENTS FOR OUR MARKETPLACE

Marketplace Name:

Comforty Ecommerce

Step 1:

The Product Listing Page With Dynamic Data.



Code Snippets Of This Page:

```

3 import { client } from '@sanity/lib/client';
4 import React from 'react';
5 import Image from 'next/image';
6 import { simpleProduct } from '@interface';
7 import { FaHeart, FaRegEye } from 'react-icons/fa';
8 import { CiShoppingCart } from 'react-icons/ci';
9 import Link from 'next/link';
10
11 async function getData() {
12   const query = `
13     *[_type == 'products']{
14       _id,
15       title,
16       price,
17       priceWithoutDiscount,
18       badge,
19       description,
20       inventory,
21       tags,
22       "imageUrl": image.asset->url,
23       "slug":slug.current,
24       category,
25     }
26   `;
27   const data: simpleProduct[] = await client.fetch(query);
28   return data;
29 }
30
31 const ProductCart = async () => {
32   const data = await getData();
33
34   return (
35     <>
36     <section className="relative bg-white py-8 px-4 sm:px-6 mt-16 sm:mt-20 lg:mt-24">
37       <div className="container mx-auto max-w-[1320px]">
38         <div className="text-center mb-8">
39           <h1 className="text-3xl font-semibold text-gray-800">Our Products</h1>
40         </div>
41
42         <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
43           {data.map((product) => (
44             <div
45               key={product._id}
46               className="bg-white shadow-md rounded-lg overflow-hidden"
47             >
48               <img
49                 src={product.imageUrl}
50                 alt={product.title}
51                 className="w-full h-60 object-cover transition-transform duration-300 hover:scale-110"
52               />
53               <div className="p-4">
54                 <Link href={`/product2/${product.slug}`}>
55                   <h2 className="text-lg hover:text-cyan-600 text-gray-800">
56                     {product.title}
57                   </h2>
58                   </Link>
59                   <div className="flex justify-between items-center">
60                     <p className="text-sm text-gray-600 font-bold">${product.price}</p>
61                     <CiShoppingCart className="text-gray-500 text-3xl hover:bg-cyan-500 cursor-pointer" />
62                   </div>
63                 </div>
64             </div>
65           ))}
66         </div>
67       </div>
68     </section>
69   </>
70 )
71

```

Step 2:

Individual Product Detail Pages With Accurate Routing And Data Rendering:



SleekSpin1

 In Stock

\$20.00

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

Tags:

gallery

Quantity:

Add to Card

Code Snippets:

```

const totalPrice = (data.price * quantity).toFixed(2); // Total price calculation

return (
  <>
    <div className="bg-white mt-24">
      <div className="mx-auto max-w-screen-xl px-4 md:px-8">
        <div className="grid gap-8 sm:grid-cols-1 md:grid-cols-2">
          { /* Image Gallery */ }
          <img
            src={imageUrl}
            width={400}
            height={400}
            alt={data.title}
            className="w-[600px] h-[400px] object-cover transition-transform duration-300 hover:scale-110"
          />

          { /* Right Section - Details */ }
          <div>
            <h1 className="text-2xl font-bold">{data.title || 'Product Name'}</h1>
            <div className="flex items-center space-x-2 mt-2">
              <span className={`text-sm flex items-center ${data.inventory > 0 ? 'text-green-600' : 'text-red-600'} font-medium`}>
                <FaCircle className="mr-1" />
                {data.inventory > 0 ? 'In Stock' : 'Out of Stock'}
              </span>
            </div>
            <p className="text-2xl font-bold text-red-500 mt-4">${totalPrice}</p>
            <p className="text-gray-600 mt-2">{data.description || 'No description available.'}</p>

            { /* Tags Section */ }
            <div className="mt-4">
              <h3 className="text-sm font-medium text-gray-700">Tags:</h3>
              <div className="flex flex-wrap space-x-2 gap-2 mt-2">
                {data.tags}
              </div>
            </div>

            { /* Quantity Selector */ }
            <div className="mt-4">
              <label htmlFor="quantity" className="block text-sm font-medium text-gray-700">
                Quantity:
              </label>
              <input
                id="quantity"
                type="number"
                value={quantity}
                min={1}
                max={data.inventory || 1}
                onChange={(e) => setQuantity(Math.min(data.inventory, Math.max(1, +e.target.value)))}
                className="mt-1 block w-16 border border-gray-300 rounded-md p-2"
              />
            </div>

            { /* Buttons */ }
            <div className="mt-6 flex space-x-4">
              <button className="bg-cyan-500 text-white px-6 py-3 rounded-md hover:bg-red-600">
                Add to Card
              </button>
            </div>
          </div>
        </div>
      </div>
    </>
  )

```

```

'use client';

import { fullProduct } from '@interface';
import { client } from '@sanity/lib/client';
import React, { useEffect, useState } from 'react';
import { FaCircle } from 'react-icons/fa6';
import { use } from 'react';
import ProductCart from '@components/ProductCart';

async function getData(slug: string) {
  const query = `
    *[_type == "products" && slug.current == "${slug}"][0]{
      _id,
      title,
      price,
      priceWithoutDiscount,
      badge,
      description,
      inventory,
      tags,
      "imageUrl": image.asset->url,
      "slug": slug.current,
      category,
      rating
    }
  `;
  const data = await client.fetch(query);
  return data;
}

export default function ProductPage({ params }: { params: Promise<{ slug: string }> }) {
  const { slug } = use(params); // Unwrap params

  const [data, setData] = useState<fullProduct | null>(null);
  const [quantity, setQuantity] = useState(1); // Default quantity

  useEffect(() => {
    const fetchData = async () => {
      const productData = await getData(slug);
      setData(productData);
    };

    fetchData();
  }, [slug]);

  if (!data) {
    return <div className="text-center py-16">Product not found.</div>;
  }
}

```

Step 3:

Category Page:

Top Categories



Wooden Chair



Code Snippets:

```
// components > @ CategorySec > ...
57 // export default CategorySec
58 import { client } from '@sanity/lib/client';
59 import React from 'react';
60
61 // Fetching Data
62 async function getData() {
63   const query =
64     `[_type == 'categories']{
65       title,
66       imageUrl: image.asset->url,
67     }
68   `;
69   const data = await client.fetch(query);
70   return data;
71 }
72
73 const CategorySec = async () => {
74   const data: { title: string; imageUrl: string }[] = await getData();
75
76   return (
77     <section className="text-gray-400 body-font">
78       <div className="container mx-auto max-w-[1320px] px-4 sm:px-6 lg:px-8 sm:mb-[400px] lg:mb-0 md:mb-0">
79         {/* Section Heading */}
80         <div className="flex flex-col text-left w-full mb-12">
81           <h1 className="sm:text-3xl text-2xl font-medium title-font text-gray-900">
82             Top Categories
83           </h1>
84         </div>
85
86         {/* Grid of Cards */}
87         <div className="flex flex-wrap justify-center mb-4">
88           {data.map((category, index) => (
89             <div key={index} className="p-4 w-full sm:w-1/2 lg:w-1/3">
90               <div className="relative group bg-gray-100 overflow-hidden shadow-lg rounded-lg">
91                 {/* Image */}
92                 <img
93                   alt={category.title}
94                   className="w-full h-[424px] object-cover transform transition-transform duration-500 group-hover:scale-110"
95                   src={category.imageUrl}
96                 />
97
98                 {/* Hidden Content */}
99                 <div className="absolute inset-0 bg-opacity-75 opacity-0 group-hover:opacity-100 transition-opacity duration-500 flex items-center justify-center">
100                   <button className="w-full absolute bottom-0 bg-gray-500 opacity-80 text-white px-4 py-2 rounded-sm hover:opacity-100 transition-opacity duration-300 mt-2 text-center text-2xl">
101                     {category.title}
102                   </button>
103                 </div>
104               </div>
105             </div>
106           ))}
107         </div>
108       </div>
109     </section>
110   );
111 };
112
113 export default CategorySec;
114
```

Step 4:

Scripts Or Logic For API Integration And Dynamic Routing:

Handle API Integration:

- Fetch data from APIs dynamically for products, categories, or other marketplace entities.
- Use state management to store global states.
- Implement error handling and loading states.

Dynamic Routing:

- Utilize a routing library like React Router or Next.js's built-in routing for SPA or SSR.
- Implement route parameters (/product/:id) for dynamic pages like product details.
- Use route guards if necessary to handle user permissions.

Prepared by

Anoosha Naz

Monday 2 to 5

(00064668)