

Day 4 - Building Dynamic Frontend of Marketplace

Comforty Furniture Website

Dynamic Frontend Overview

Dynamic frontend is essential for creating interactive and engaging web applications. It allows websites to respond to user actions and real-time data changes, making the overall user experience seamless and enjoyable.

Key Features of Dynamic


- **Real-time Interaction:** Enables instant responses to user actions, such as form validations or button state changes.
 - **Live Data Updates:** Automatically updates the user interface based on real-time data changes, ensuring accuracy and relevance.
 - **Modular Design:** Supports reusable components that can be applied across different parts of the application with minimal changes.
 - **Customizability:** Allows for the customization of behavior and appearance through props or dynamic inputs.
-
- **Smooth User Experience:** Creates a seamless and engaging interaction by making the frontend responsive and adaptive to user needs.

Comforty

Cart 2

HomeShopProductPagesAbout

Contact: (808) 555-0111



SleekSpin

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

\$20

Add to Cart

File Edit Selection View Go Run ...

hackathon_3

EXPLORER

HACKATHON_3

app

product

[id]

page.tsx

products.ts

categories.ts

index.ts

migrate.mjs

page.tsx .../product

page.tsx .../product/[id]

```
25
26
27 const SingleProduct = () => {
28   const [data, setData] = useState<Product | null>(null);
29   const [isModalOpen, setIsModalOpen] = useState(false);
30   const [successMessage, setSuccessMessage] = useState("");
31   const { id } = useParams(); // Ensure the id is a string from the route params
32
33   useEffect(() => {
34     if (id) {
35       // Sanity Query to fetch product by _id
36       const query = `[_type == "products" && _id == $id]{
37         _id,
38         title,
39         price,
40         description,
41         category,
42         rating,
43         imageUrl: image.asset->url
44       }`;
45
46       // Log the query to ensure it's being executed correctly
47       console.log("Executing query for product ID:", id);
48
49       // Fetch product data from Sanity
50       client
51         .fetch(query, { id })
52         .then((res) => {
53           console.log("Response from Sanity:", res);
54         });
55     }
56   });
57 }
```

1.

