

Day 4 - Building Dynamic Frontend Components for My Marketplace

Prepared by: Ammara Rajput

Objective

Today, my primary focus was to build dynamic frontend components for the marketplace using React and Next.js. The goal was to create reusable components for product listings, category filters, and navigation, ensuring responsiveness and scalability.

Key Outcomes

1. Dynamic Components Creation:

- Built and tested reusable React components to display marketplace data dynamically.
- Implemented dynamic routing in Next.js for product and category pages.

2. State Management:

- Managed application state using hooks like `useState` and `useEffect`.
- Ensured smooth communication between components via props and context.

3. Responsive Design:

- Made the components responsive using Tailwind CSS.
- Tested the UI on different screen sizes to ensure consistency.

Steps I Followed

Step 1: Planning and Component Breakdown

Before starting, I created a clear structure for the components based on the marketplace requirements:

- **Header Component:** Contains the logo, navigation links, and a search bar.
- **Product Listing Component:** Dynamically displays products with details like price and image.
- **Category Filter Component:** Allows users to filter products by category.

I also drafted a folder structure to keep the project organized.

Step 2: Building Components in Next.js

I built and tested the components incrementally:

1. Header Component:

- Designed the navigation bar using Tailwind CSS.
- Added dynamic links for categories fetched via API calls.

2. Product Listing Component:

- Mapped over the API data to dynamically render product cards.
- Each product card includes an image, title, price, and a "View Details" button that links to the product page.

3. Category Filter Component:

- Created buttons and dropdowns for categories.
- Connected the filter functionality to API calls to display relevant products.

Step 3: Dynamic Routing with Next.js

To make the marketplace user-friendly, I implemented dynamic routes:

- **Product Details Page:**

- Used `getStaticPaths` and `getStaticProps` to generate individual product pages.
- Displayed detailed information fetched from the API.

- **Category Pages:**

- Dynamically rendered products based on the selected category.

Step 4: State Management and Interactivity

To make the components interactive, I worked on state management:

- Managed the product list and category state using React hooks (`useState`).
- Used `useEffect` to fetch updated data whenever a category was selected.

Step 5: Styling and Responsiveness

I styled the components using Tailwind CSS:

- Used flexbox and grid utilities for layout.
- Applied responsive classes to ensure the UI looked good on mobile, tablet, and desktop.
- Added hover effects for buttons and product cards to improve user experience.

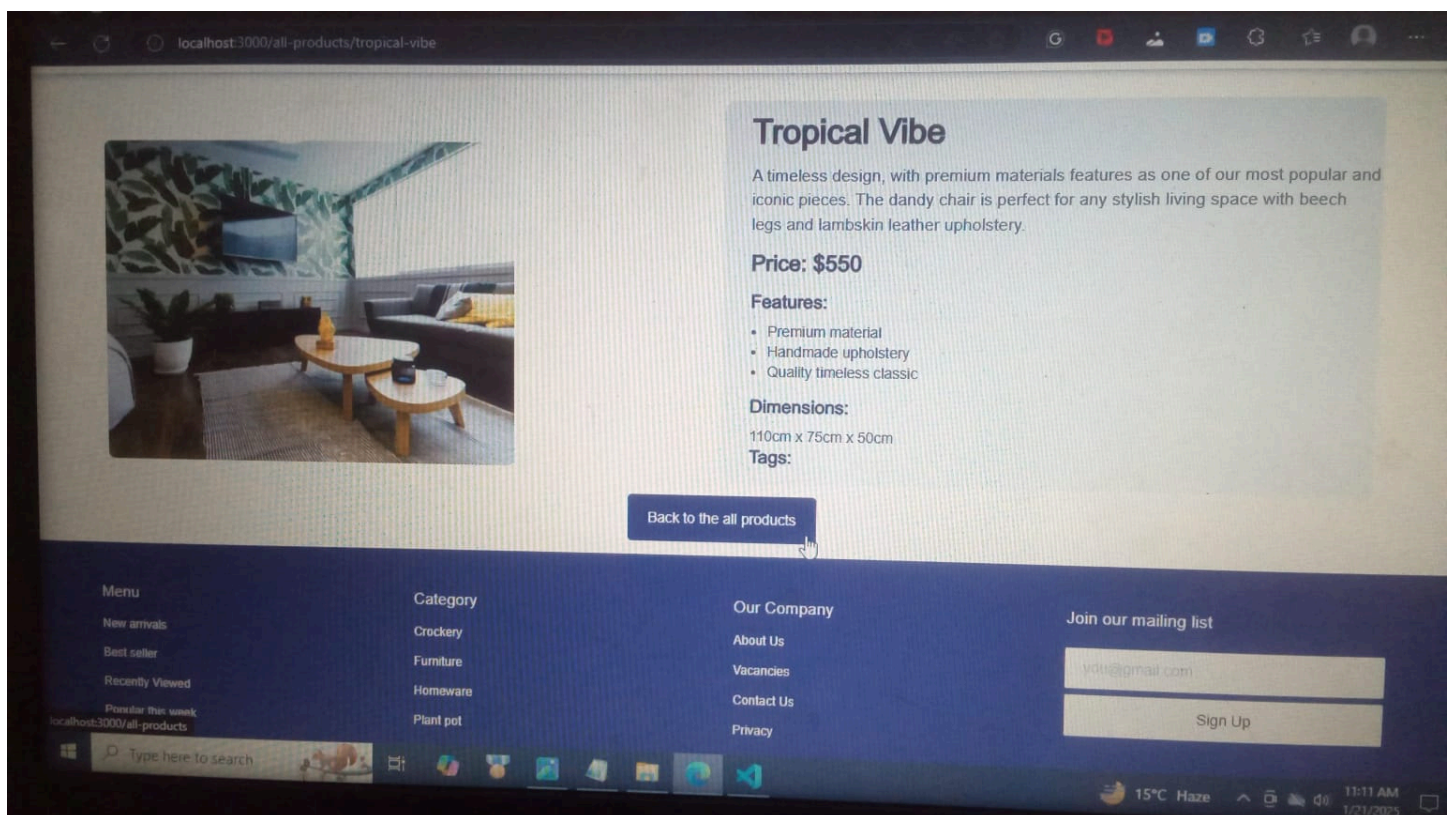
Deliverables and Submission

Here's what I've prepared for submission:

1. Code Files:

```
src > app > Components > AllProductsPreview.tsx > ProductsPreview
5 import Image from 'next/image'; // importing image from next/image
6
7 interface Product {
8   _id: string;
9   name: string;
10  price: number;
11  imageUrl: string;
12  slug: {
13    current: string;
14  };
15 }
16
17 export default function ProductsPreview() {
18   const [products, setProducts] = useState<Product[]>([]);
19   const [loading, setLoading] = useState<boolean>(true);
20   const [error, setError] = useState<string | null>(null);
21
22   useEffect(() => {
23     const fetchProducts = async () => {
24       try {
25         const query = `
26           *[_type == "product"]{0..5}{
27             _id,
28             name,
29             price,
30             "imageUrl": image.asset->url,
31             slug
32           }
33         `;
34         const fetchedProducts: Product[] = await sanityFetch({ query });
35         setProducts(fetchedProducts);
36       } catch (err) {
37         setError('Failed to load products.');
```

1. Screenshots:



Featured Products



The Poplar suede sofa
Price: \$980



Tropical Vibe
Price: \$550



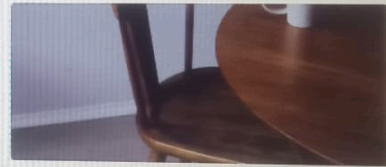
Sleek Living
Price: \$300



Serene Seat
Price: \$350



Nordic Elegance
Price: \$280



TimberCraft
Price: \$320

[View All Products](#)

All Products



The Poplar suede sofa
Price: \$980



Tropical Vibe
Price: \$550



Sleek Living
Price: \$300



Serene Seat
Price: \$350



Nordic Elegance
Price: \$280



TimberCraft
Price: \$320

[Previous](#)

[1](#)

[2](#)

[3](#)

[4](#)

[Next](#)