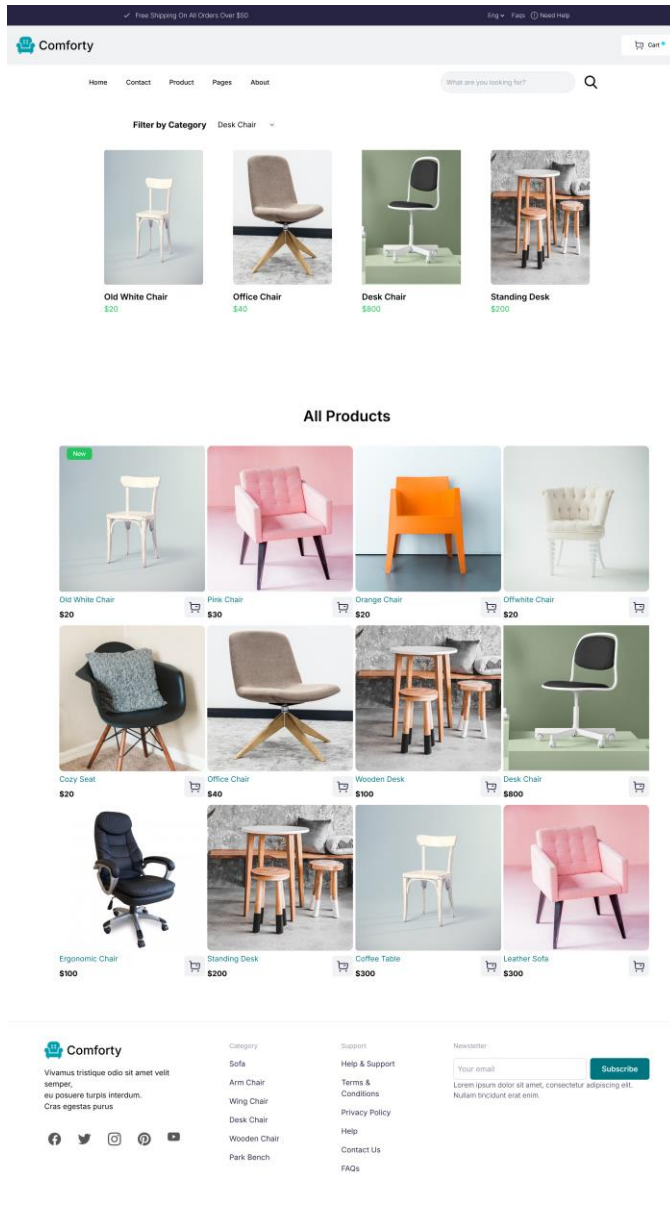


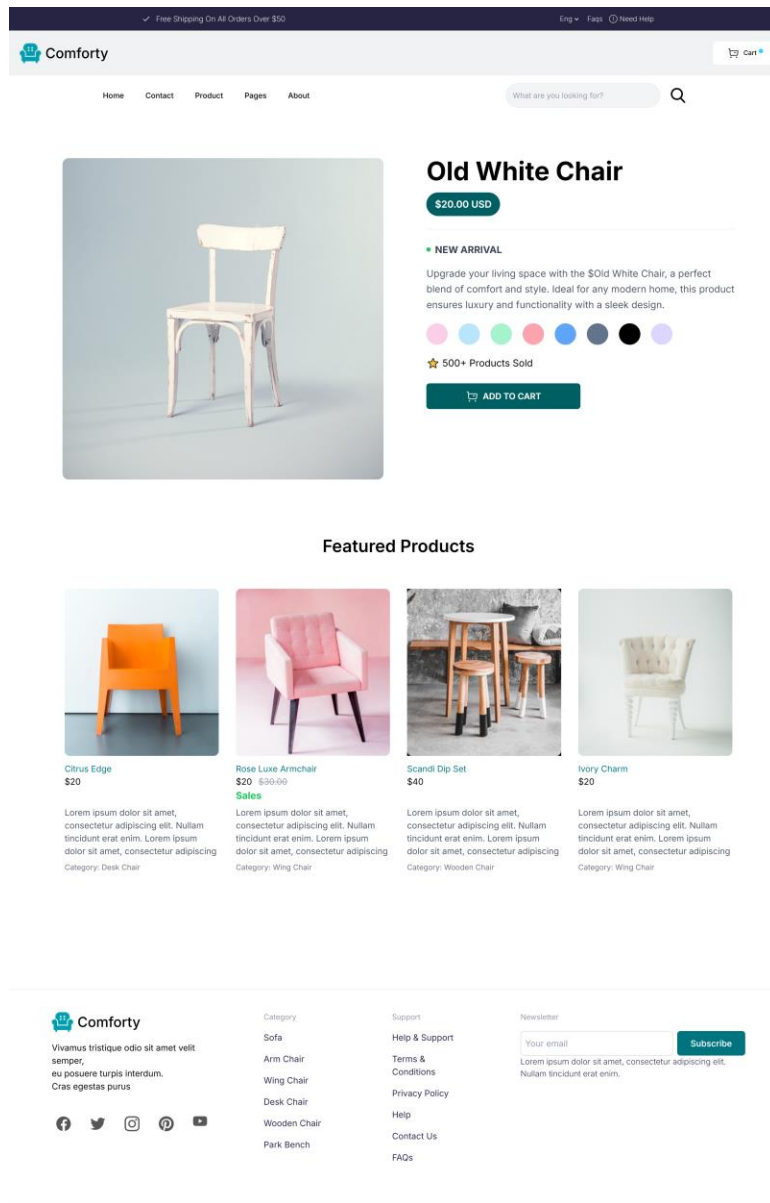
DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

The product listing page with dynamic data:



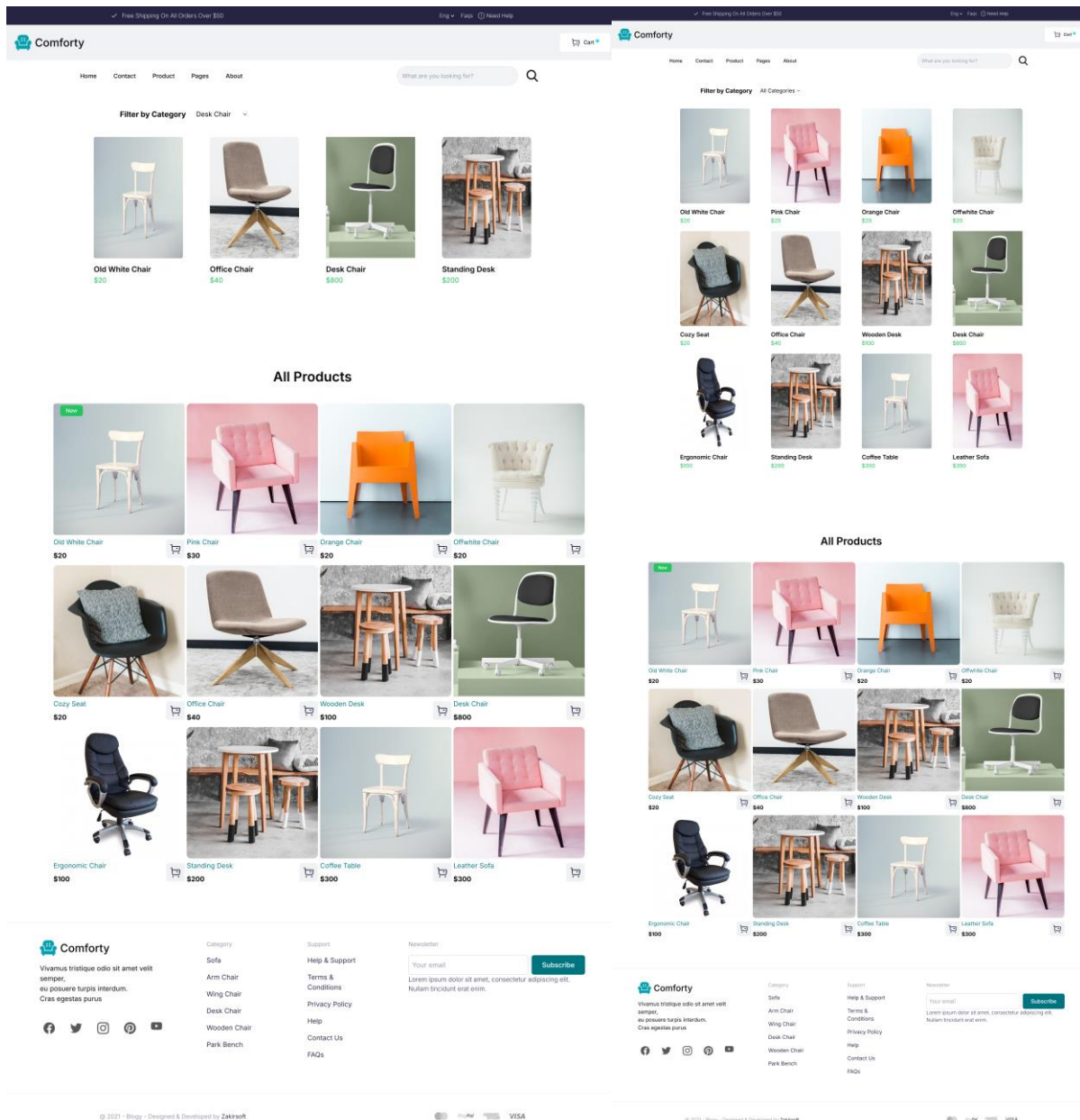
Individual product detail pages with accurate routing and data rendering:

DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

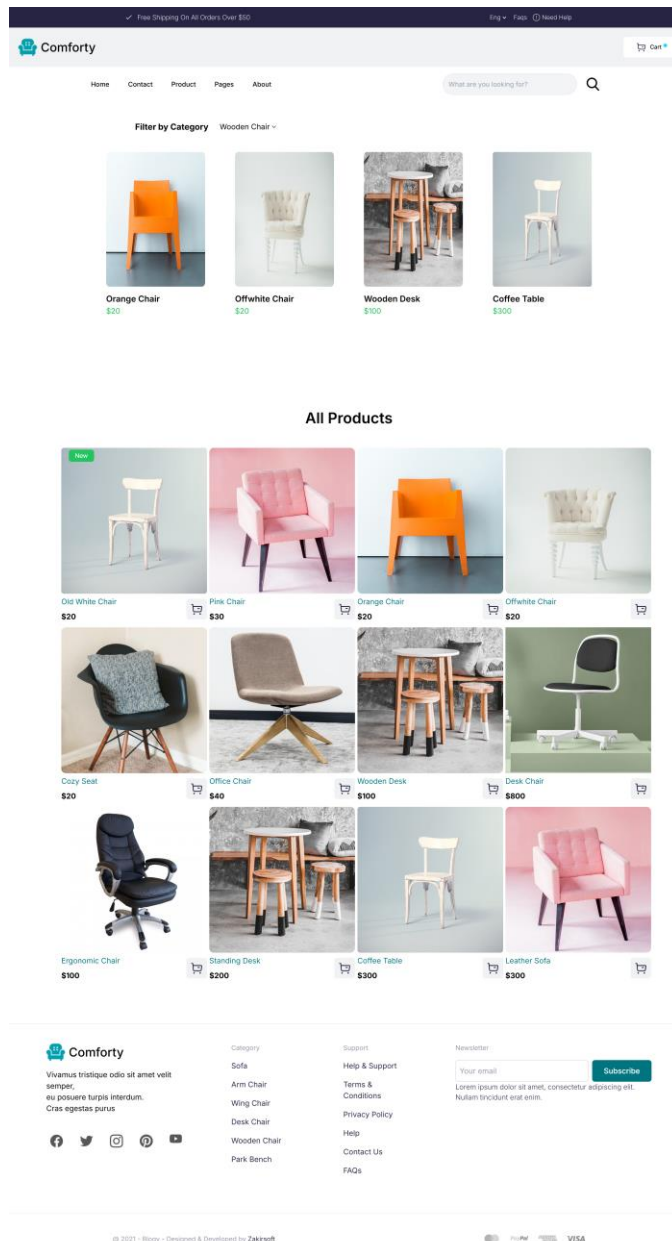


Working category filters:

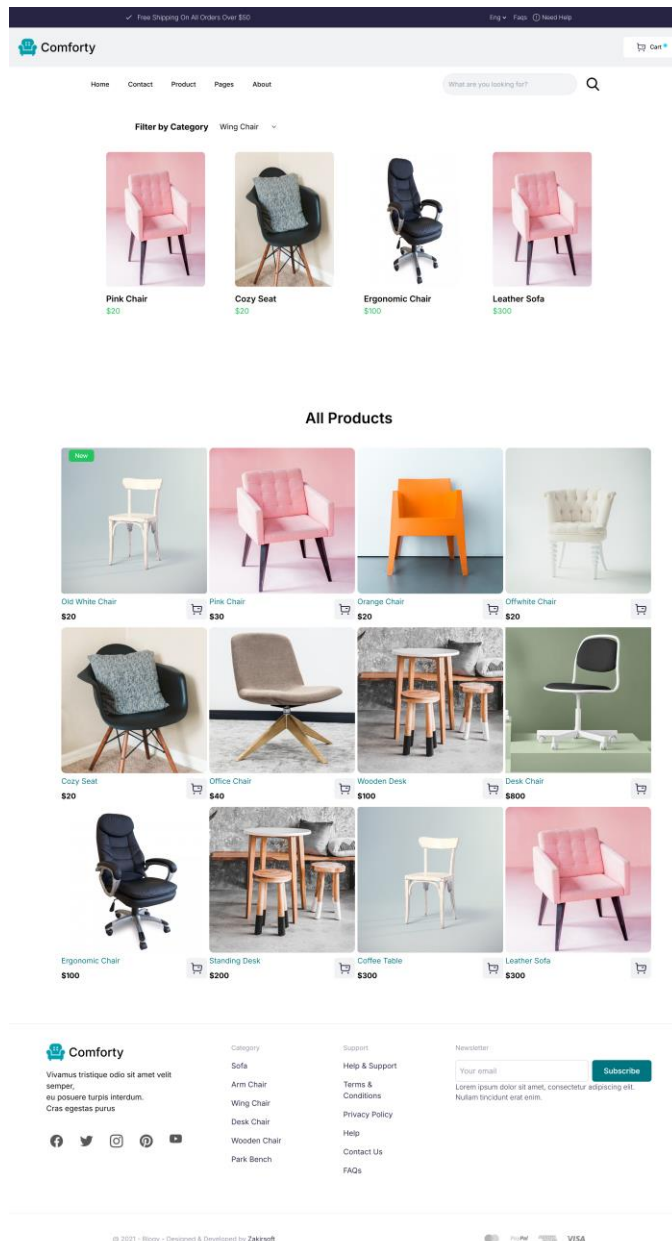
DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE



DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

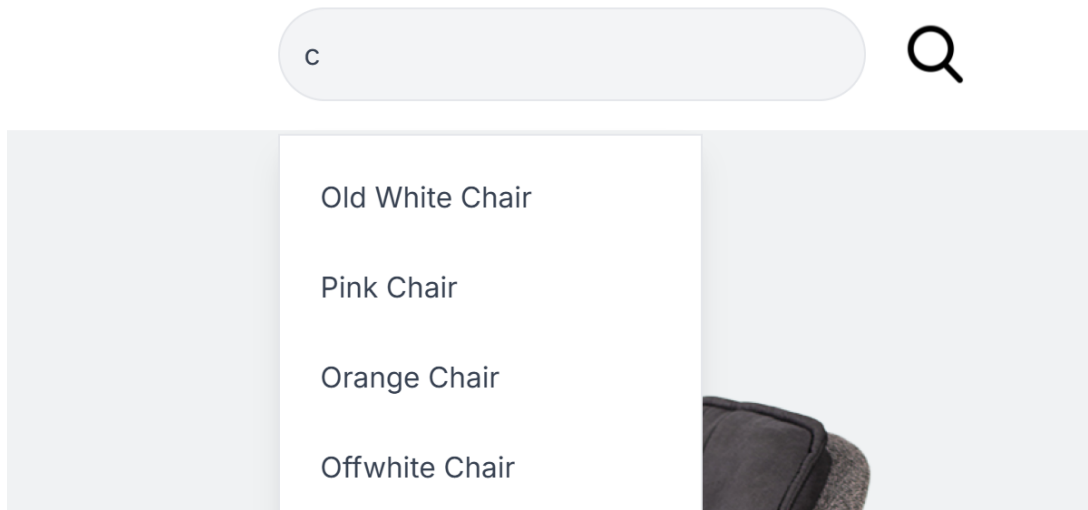
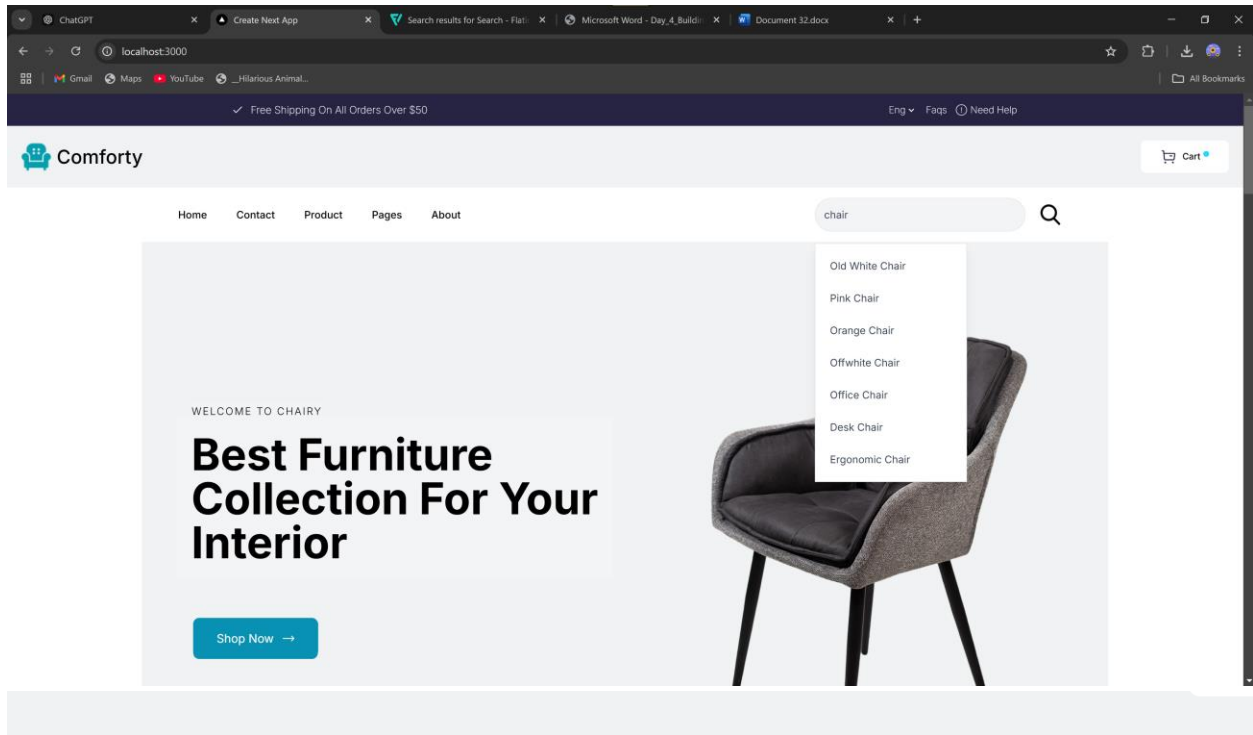


DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

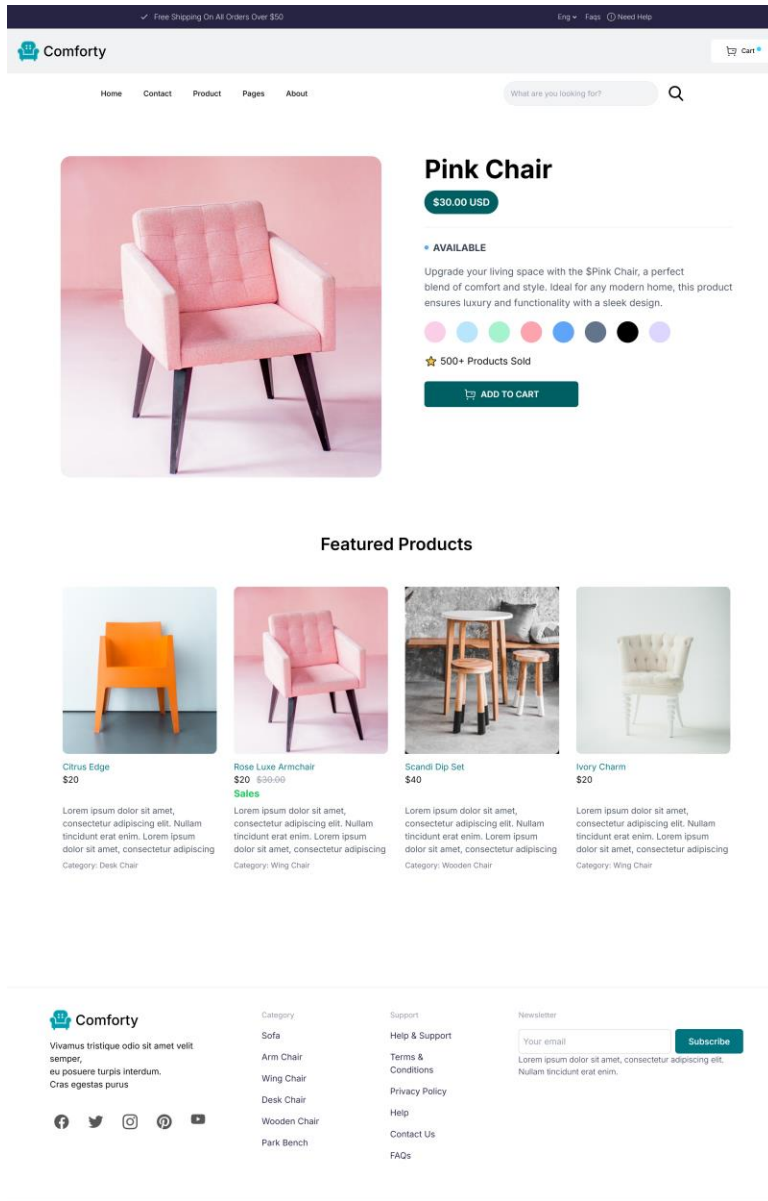


Working search bar:

DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE



DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE



CODE DELIVERABLES

PRODUCT CARD

DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

```
src > app > products > page.tsx > Products > useEffect() callback > fetchProducts
17
18 export default function Products() {
19   const [products, setProducts] = useState<Product[]>([]);
20   const [cart, setCart] = useState<Product[]>([]);
21
22   useEffect(() => {
23     async function fetchProducts() {
24       try {
25         const response = await fetch("/api/products");
26         if (response.ok) {
27           const data: Product[] = await response.json();
28           setProducts(data);
29         } else {
30           console.error("Failed to fetch: ${response.status}");
31         }
32       } catch (error) {}
33       console.error("Failed to fetch products:", error);
34     }
35     fetchProducts();
36   }, []);
37
38   const addToCart = (product: Product) => {
39     const updatedCart = [...cart, product];
40     setCart(updatedCart);
41     localStorage.setItem("cart", JSON.stringify(updatedCart));
42   };
43
44   return (
45     <main>
46       <Header />
47       <CategoryComponent />
48       <div>
49         <h2 className="text-[32px] font-semibold mt-[173px] text-center">
50           All Products
51         </h2>
52
53         <div className="grid sm:grid-cols-1 md:grid-cols-2 lg:grid-cols-4 justify-center items-center gap-x-5 mt-10 ml-28 mr-24">
54           {products.map((product) => (
55             <Link key={product.id} href={`/products/${product.id}`} passHref>
56               <div className="card group w-80 h-96 relative cursor-pointer">
57                 <div className="img relative">
58                   <img
59                     className="w-[312px] h-[312px]"
60                     src={product.imageUrl}
61                     alt={product.name}
62                   />
63                 <div className="flex justify-between -mt-[295px] items-center">
64                   {product.isNew && (
65                     <button className="absolute w-[54px] h-[26px] text-[13px] rounded-md ml-4 bg-green-500 text-white">
66                       New
67                     </button>
68                   )}
69                 </div>
70               </div>
71             )}
72         </div>
73       </div>
74     </main>
75   );
76 }
```


DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

```
header.tsx M  page.tsx ..app M  page.tsx ..products M X  routes.ts M  search.tsx U  page.tsx ..[id] U
src > app > products >  page.tsx >  Products >  useEffect() callback >  fetchProducts
18  export default function Products() {
55  (products.map((product) => {
59      <img
60        className="w-[312px] h-[312px]"
61        src={product.imageUrl}
62        alt={product.name}
63      />
64      <div className="flex justify-between -mt-[295px] items-center">
65        {product.isNew && (
66          <button className="absolute w-[54px] h-[26px] text-[13px] rounded-md ml-4 bg-green-500 text-white">
67            New
68          </button>
69        )}
70      </div>
71      <div className="flex justify-between items-center">
72        <div className="mt-[300px]">
73          <h4 className="text-[16px] font-normal text-[#007580]">
74            {product.name}
75          </h4>
76          <h5 className="mt-2 font-bold">
77            ${product.salePrice || product.price}
78          </h5>
79        </div>
80        <div
81          className="w-[44px] h-[44px] mt-[300px] mr-2 rounded-md relative group overflow-hidden flex flex-col items-center bg-[#f0f3] hover:bg-[#007580] transition-colors duration-300 cursor-p
82          onClick={() => addToCart(product)}
83        >
84          
89          
94        </div>
95      </div>
96    </div>
97  </div>
98  </Link>
99  )}
100 </div>
101 </div>
102 <Footer />
103 </main>
104 </>
105 }
106
```

PRODUCT LIST

DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

```
header.tsx M  page.tsx ...app M  page.tsx ...products M  route.ts M  search.tsx U  page.tsx ...[id] U X
src > app > products > [id] > page.tsx > Product
7
8  type Product = {
9    id: number;
10   name: string;
11   price: number;
12   imageUrl: string;
13   isNew: boolean;
14   salePrice?: number;
15 };
16
17 export default function ProductDetail() {
18   const [product, setProduct] = useState<Product | null>(null);
19   const { id } = useParams();
20
21   useEffect(() => {
22     if (id) {
23       const fetchProduct = async () => {
24         try {
25           const response = await fetch(`/api/products`);
26           if (response.ok) {
27             const products: Product[] = await response.json();
28             const foundProduct = products.find((p) => p.id.toString() === id);
29             setProduct(foundProduct || null);
30           } else {
31             console.error(`Error: ${response.status}`);
32           }
33         } catch (error) {
34           console.error("Error fetching product:", error);
35         }
36       };
37       fetchProduct();
38     }
39   }, [id]);
40
41   if (!product) {
42     return (
43       <main>
44         <Header />
45         <div className="flex justify-center items-center min-h-screen">
46           <p className="text-xl">Loading product details...</p>
47         </div>
48         <Footer />
49       </main>
50     );
51   }
52
53   return (
54     <main>
55       <Header />
56
57       <div className="flex flex-col md:flex-row justify-center md:gap-20 mt-20 px-4 md:px-0">
58         <div>
59           <h2>Product Details</h2>
60           <p>Product ID: {product.id}</p>
61           <p>Name: {product.name}</p>
62           <p>Price: {product.price}</p>
63           <p>Image URL: {product.imageUrl}</p>
64           <p>Is New: {product.isNew}</p>
65           <p>Sale Price: {product.salePrice}</p>
66         </div>
67         <div>
68           <h3>Product Description</h3>
69           <p>This is a detailed description of the product, showing its features and benefits. It is designed to provide users with all the information they need to make a purchase decision.</p>
70         </div>
71       </div>
72     </main>
73   );
74 }
```

DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

```
header.tsx M  page.tsx ...app M  page.tsx ...products M  route.ts M  search.tsx U  page.tsx ...[id] U X
src > app > products > [id] > page.tsx > Product
17 export default function ProductDetail() {
18   /* Product Detail Section */
58   <div className="flex flex-col md:flex-row justify-center md:gap-20 mt-20 px-4 md:px-0">
59     /* Product Image */
60     <div className="flex justify-center mb-10 md:mb-0">
61       <img
62         className="w-full md:w-[600px] h-auto object-contain"
63         src={product.imageUrl}
64         alt={product.name}
65       />
66     </div>
67
68     /* Product Information */
69     <div className="text-center md:text-left">
70       <h1 className="text-3xl md:text-5xl font-bold mb-4">
71         {product.name}
72       </h1>
73       <p className="text-lg font-semibold bg-[#005f63] text-white py-2 px-4 rounded-full inline-block">
74         ${product.salePrice || product.price}.00 USD
75       </p>
76       <hr className="my-6" />
77
78       <p className="text-sm md:text-lg text-gray-700 font-bold flex items-center">
79         <span
80           className={`inline-block w-2 h-2 rounded-full mr-2 ${
81             product.isNew ? "bg-green-500" : "bg-blue-400"
82           }`}
83         ></span>
84         {product.isNew ? "NEW ARRIVAL" : "AVAILABLE"}
85       </p>
86
87       <p className="mt-4 text-gray-600 text-lg leading-relaxed">
88         Upgrade your living space with the ${product.name}, a perfect <br />{" "}
89         blend of comfort and style. Ideal for any modern home, this product{" "}
90         <br />ensures luxury and functionality with a sleek design.
91       </p>
92
93       /* Color Options */
94       <div className="flex gap-x-5 mt-5 mb-3">
95         <div className="w-10 h-10 bg-pink-200 rounded-full "></div>
96         <div className="w-10 h-10 bg-sky-200 rounded-full "></div>
97         <div className="w-10 h-10 bg-emerald-200 rounded-full "></div>
98         <div className="w-10 h-10 bg-rose-300 rounded-full "></div>
99         <div className="w-10 h-10 bg-blue-400 rounded-full "></div>
100        <div className="w-10 h-10 bg-slate-500 rounded-full "></div>
101        <div className="w-10 h-10 bg-black rounded-full "></div>
102        <div className="w-10 h-10 bg-violet-200 rounded-full "></div>
103      </div>
104
105       /* Sold */
106
107       <p className="text-lg mt-5">★ 500+ Products Sold</p>
108    </div>
109  }
```

DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

```
    { /* Sold */ }

    <p className="text-lg mt-5">★ 500+ Products Sold</p>

    { /* Add to Cart Button */ }
    <button className="mt-6 w-full md:w-1/2 flex items-center justify-center gap-2 py-3 bg-[#005f63] hover:bg-[#006063de] text-white font-semibold rounded-md transition">
      
      ADD TO CART
    </button>
  </div>
</div>

{ /* Featured Products Section */ }

<FeaturedSection/>

<Footer />
</main>
);
}
```

DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

SEARCH BAR

```
header.tsx M  page.tsx ...\app M  page.tsx ...\products M  route.ts M  search.tsx U X  page.tsx ...\id U
src > app > components > search.tsx > Search > useEffect() callback
16 const Search = () => {
17   const [query, setQuery] = useState("");
18   const [results, setResults] = useState<Product[]>([])
19   const [loading, setLoading] = useState(false)
20
21   useEffect(() => {
22     const fetchResults = async () => {
23       if (!query) {
24         setResults([]);
25         return;
26       }
27
28       setLoading(true);
29       try {
30         const res = await fetch(`/api/products?q=${query}`);
31         const data = await res.json();
32         setResults(data);
33       } catch (err) {
34         console.error("Error fetching products:", err);
35       } finally {
36         setLoading(false);
37       }
38     };
39
40     const delayDebounce = setTimeout(() => {
41       fetchResults();
42     }, 300);
43
44     return () => clearTimeout(delayDebounce);
45   }, [query]);
46
47   return (
48     <div className="relative hidden md:flex items-center p-2 px-4 gap-x-5 text-gray-700 text-sm focus:outline-none border-none">
49       <input
50         type="text"
51         placeholder="What are you looking for?"
52         className="border rounded-full focus:outline-none md:w-72 p-3 bg-gray-100"
53         value={query}
54         onChange={(e) => setQuery(e.target.value)}
55       />
56       
57       {loading && <div className="absolute top-full mt-2">Loading...</div>}
58       {results.length > 0 && (
59         <ul className="absolute top-full mt-2 bg-white border shadow-lg z-50 p-2 px-2 w-52">
60           {results.map((product) => (
61             <Link key={product.id} href={`/products/${product.id}`} passHref>
62               <li
63                 className="p-3 hover:bg-gray-100 cursor-pointer z-10 text-[14px]"
64               >
65                 {product.name}
66               </li>
67             </Link>

```

DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

CATEGORIES.TS

```
header.tsx M  page.tsx ..\app M  categories.tsx U x  route.ts M  search.tsx U  page.tsx ..\id U
src > app > components > categories.tsx > CategoryComponent > filteredProducts.map() callback
10
11 export default function CategoryComponent() {
12   const [categories, setCategories] = useState<string[]>([
13     "Desk Chair",
14     "Wooden Chair",
15     "Wing Chair",
16   ]);
17   const [products, setProducts] = useState<Product[]>([]);
18   const [filteredProducts, setFilteredProducts] = useState<Product[]>([]);
19   const [selectedCategory, setSelectedCategory] = useState<string | null>(null);
20
21   useEffect(() => {
22     // Fetch products data from your API
23     fetch("/api/products")
24       .then((response) => response.json())
25       .then((data) => {
26         setProducts(data);
27         setFilteredProducts(data); // Initially, show all products
28       });
29   }, []);
30
31   // Filter products by selected category
32   const handleCategoryChange = (category: string | null) => {
33     setSelectedCategory(category);
34     if (category === null) {
35       setFilteredProducts(products); // Show all products if no category is selected
36     } else {
37       const filtered = products.filter(
38         (product) => product.category === category
39       );
40       setFilteredProducts(filtered);
41     }
42   };
43
44   return (
45     <div className="">
46       <div className="category-filter ml-[270px] mt-10 mb-10 flex gap-x-5">
47         <h3 className="font-bold text-lg">Filter by Category</h3>
48         <select
49           className="border-none"
50           onChange={(e) => handleCategoryChange(e.target.value)}
51           value={selectedCategory || ""}>
52           >
53             <option value="">All Categories</option>
54             {categories.map((category) => (
55               <option key={category} value={category}>
56                 {category}
57               </option>
58             ))}
59           </select>
60         </div>
61       </div>
62     );
63   }
64 }
```

DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

LOGIC OF SEARCH IN ROUTE.TS

```
130
131 export const GET = async (req: Request) => {
132   const url = new URL(req.url);
133   const query = url.searchParams.get("q")?.toLowerCase() || "";
134
135   // Filter products based on the query
136   const filteredProducts = query
137     ? products.filter((product) =>
138       product.name.toLowerCase().includes(query)
139     )
140     : products;
141
142   return NextResponse.json(filteredProducts);
143 };
144
```

TECHNICAL SUMMARY:

I implemented a search bar, categories filter, and dynamic routing for the product details. The search bar dynamically updates products as you type, and the categories filter allows users to narrow down products by category. The dynamic routing lets users navigate to the product detail page.

There were issues with filtering the dynamically fetched products and creating dynamic routes for the product detail page. I couldn't filter out the products and link them properly to their detail pages, so I decided to leave the product navigation as it is for now. I added the categorized products to the same page instead, though I know this isn't professional. This part was just really confusing for me at the moment.

There wasn't any problem with the search or category filter features. The main issue was combining the dynamic data fetching with routing. For now, I kept the navigation simple and listed the categorized products on the same page.

I followed some best practices like breaking down components, handling errors properly, and making sure the page stays responsive during data fetching. These steps helped me get through the development process.

DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

I didn't do all of this by myself and had some confusion along the way, but with some help, I was able to make progress.