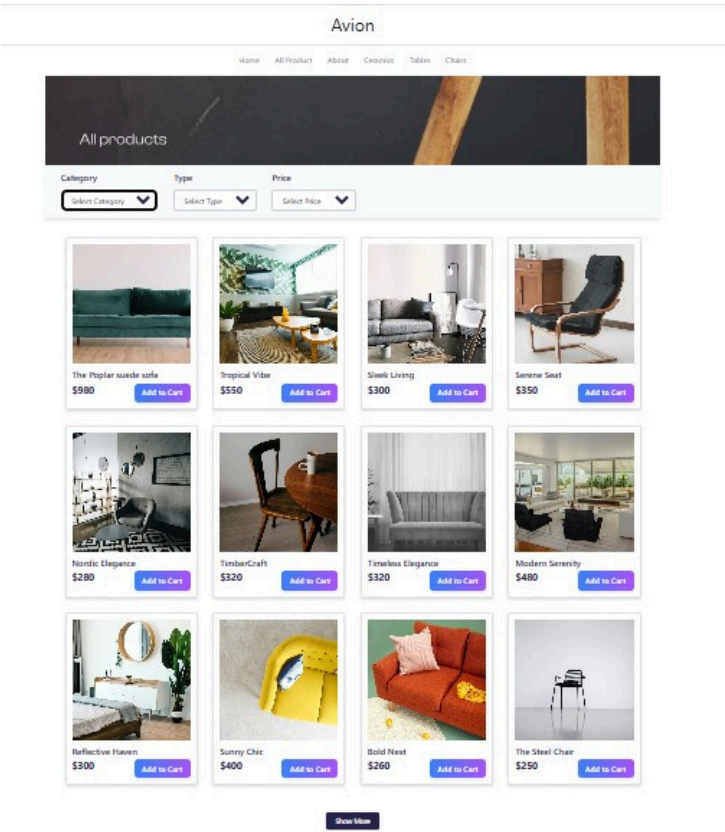


"Day 4 - Dynamic Frontend Components - [Avion E-commerce]"

1. Functional Deliverables:

2. Code Deliverables:

Product listing page screenshot:



AllProducts component code:

```

> app > allproducts > @ pages.tsx > +o IProduct > category
"use client";
import React, { useState, useEffect } from "react";
import Image from "next/image";
import Header from "@components/Header";
import Footer from "@components/Footer";
import { client } from "@sanity/lib/client";
import { urlFor } from "@sanity/lib/image";
import Link from "next/link";
import { Button } from "@components/ui/button";

import Swal from "sweetalert2";
import { addToCart } from "../action/page";

interface ICategory {
  name: string;
  slug: { current: string };
}

interface IProduct {
  _id: string;
  name: string;
  slug: { current: string };
  image: { asset: { url: string } };
  price: number;
  category: { slug: { current: string } };
  type: string;
  inventory: number;
}

const AllProducts = () => {
  const [data, setData] = useState<IProduct[]>([]);
  const [categories, setCategories] = useState<ICategory[]>([]);
  const [loading, setLoading] = useState(true);
  const [selectedCategory, setSelectedCategory] = useState<string>("");
  const [selectedType, setSelectedType] = useState<string>("");
  const [selectedPrice, setSelectedPrice] = useState<string>("");
  const [cart, setCart] = useState<IProduct[]>([]);

  const [visibleCount, setVisibleCount] = useState(12);
  const productsPerPage = 12;

  const [showFilters, setShowFilters] = useState(false);

```

pp > allproducts > @ page.tsx > IProduct > category

```
const AllProducts = () => {

  useEffect(() => {
    const fetchData = async () => {
      try {
        const productQuery = `[_type == "product"
          && (${selectedCategory ? `category->slug.current == "${selectedCategory}"` : "true"} )
          && (${selectedType ? `type == "${selectedType}"` : "true"} )
          && (${selectedPrice ? `price <= ${selectedPrice}` : "true"})]`;
        const { name, slug, image, price, category, type } = await client.fetch(productQuery);

        const fetchedProducts = await client.fetch(productQuery);
        setData(fetchedProducts);

        const categoryQuery = `[_type == "category"]{name, slug}`;
        const fetchedCategories = await client.fetch(categoryQuery);
        setCategories(fetchedCategories);
      } catch (error) {
        console.error("Error fetching data: ", error);
      } finally {
        setLoading(false);
      }
    };

    fetchData();
  }, [selectedCategory, selectedType, selectedPrice]);

  const handleViewToggle = () => {
    if (visibleCount >= data.length) {
      setVisibleCount(12);
    } else {
      setVisibleCount((prev) => prev + productsPerPage);
    }
  };

  if (loading) {
    return (
      <div className="flex justify-center items-center w-full h-screen">
        <div className="border-8 border-t-8 border-gray-300 border-t-[#2A254B] rounded-full w-16 h-16 animate-spin"></div>
      </div>
    );
  }

  if (data.length === 0) {
    return <p>No products found.</p>;
  }
}
```

```

app > allproducts > page.tsx > IProduct > category
const AllProducts = () => {

  const handleAddtoCart = (e: React.MouseEvent, product: IProduct) => {
    e.preventDefault();
    Swal.fire({
      position: "top-right",
      icon: "success",
      title: `${product.name} added to cart`,
      showConfirmButton: false,
      timer: 1000,
    });
    addToCart(product); // Add product to cart
  };

  return (
    <div>
      <Header />
      <div className="w-full max-w-screen-xl mx-auto text-[#2A254B]">
        <div>
          <Image
            src="/images/productlisting200.svg"
            alt="Product Listing Banner"
            height={209}
            width={1440}
            className="object-cover w-full"
          />
        </div>

        <div>
          <div className="md:hidden flex justify-between items-center py-4 px-8">
            <button
              className="bg-[#F9F9F9] py-2 px-4 text-sm font-semibold rounded-md hover:bg-[#2A254B] hover:text-white transition-colors"
              onClick={() => setShowFilters(!showFilters)}
            >
              {showFilters ? "Hide Filters" : "Show Filters"}
            </button>
          </div>

          <div>
            <div className="md:hidden w-full px-8 pb-8">
              <div className="flex flex-col gap-6">
                <div>
                  <div className="flex flex-col gap-2">
                    <label htmlFor="category" className="text-lg font-medium">
                      Category
                    </label>
                    <select
                      id="category"

```

Individual product detail page screenshot:

product 1:



The Poplar suede sofa

\$980

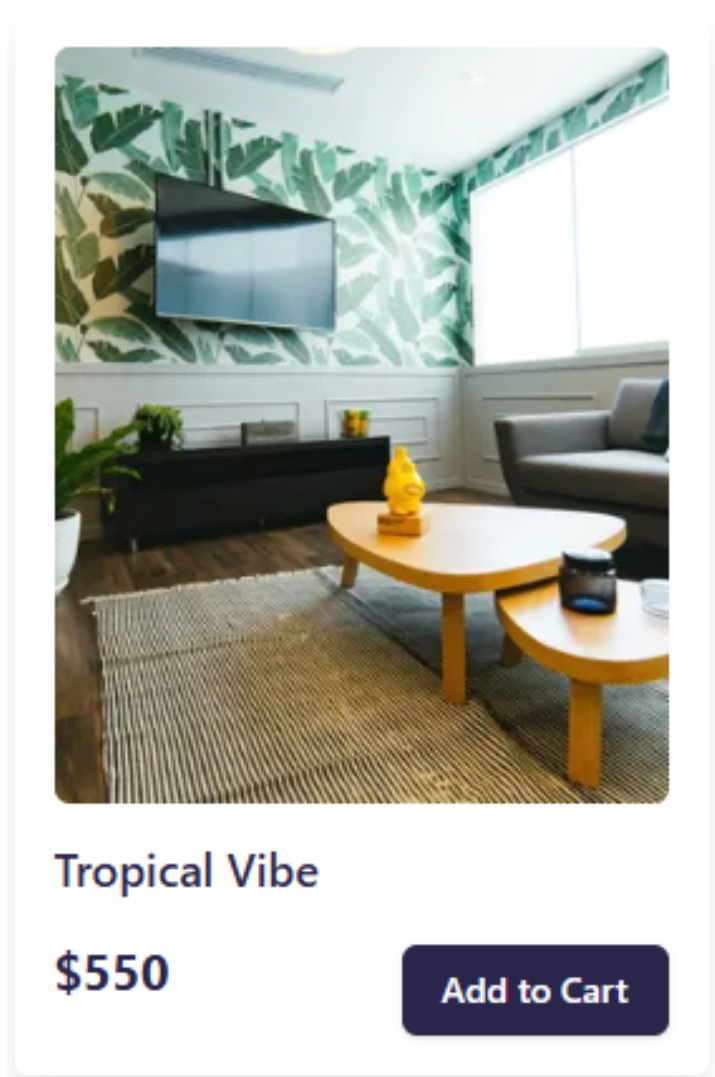
Add to Cart

API integration code for fetching product data:

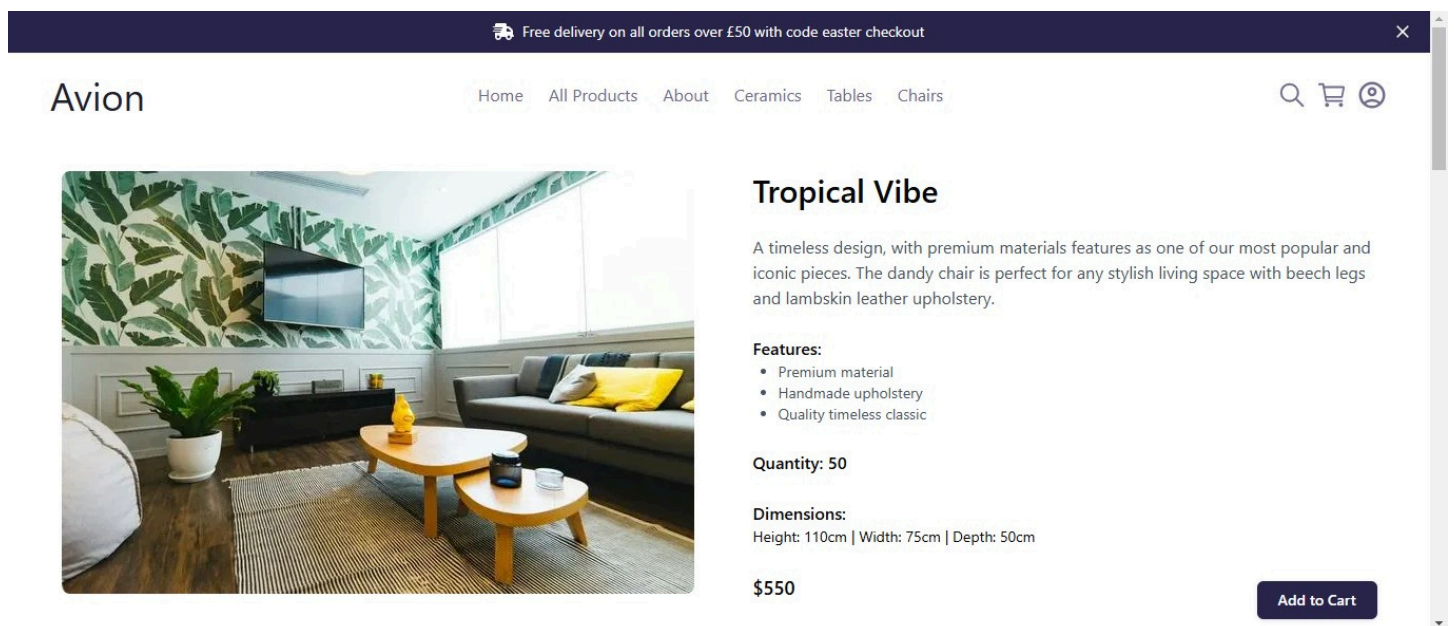
Dynamic routing setup code for /product/[slug] page.tsx:

```
src > app > products > [slug] > page.tsx > fetchProduct
7  import TopBar from "@components/TopBar";
8  import Brand from "@components/Brand";
9  import ClubBenefits from "@components/ClubBenefits";
10 import Footer2 from "@components/Footer2";
11 import { Button } from "@components/ui/button";
12
13
14 async function fetchProduct(slug: string) {
15   return await client.fetch(
16     `*[_type == "product" && slug.current == $slug][0]`,
17     { slug }
18   );
19 }
20
21 const ProductPage = async ({ params }: { params: { slug: string } }) => {
22   const product = await fetchProduct(params.slug);
23
24   if (!product) {
25     notFound();
26   }
27
28   return (
29     <div>
30       <TopBar />
31       <ProductDetailHeader />
32
33       <div className="max-w-screen-xl mx-auto px-4 pt-8">
34         <div className="flex flex-col md:flex-row items-center md:items-start">
35           {/* Product Image (Left Side) */}
36           <div className="flex-shrink-0 mb-6 md:mb-0 md:w-1/2">
37             <Image
38               src={urlFor(product.image?.asset).width(600).height(400).url()}
39             />
40           </div>
41         </div>
42       </div>
43     </div>
44   );
45 }
```

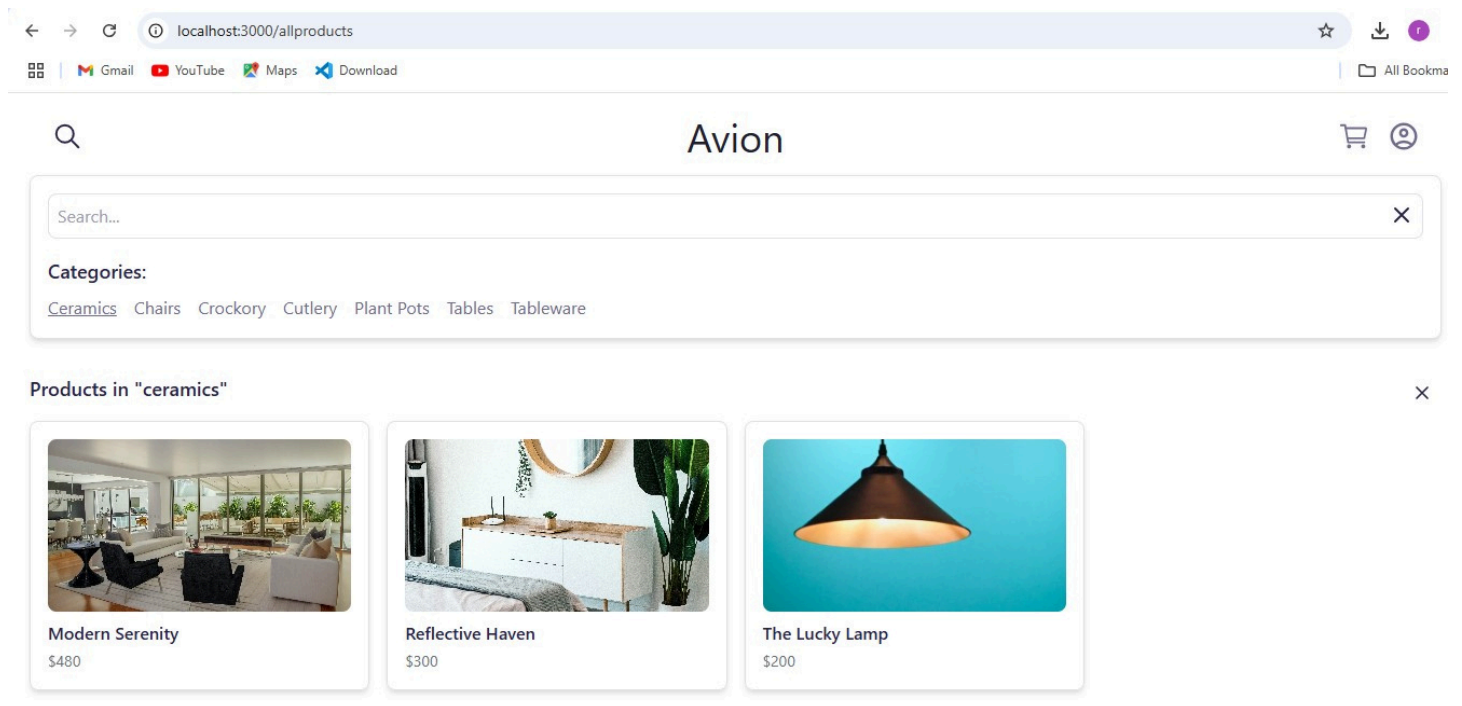
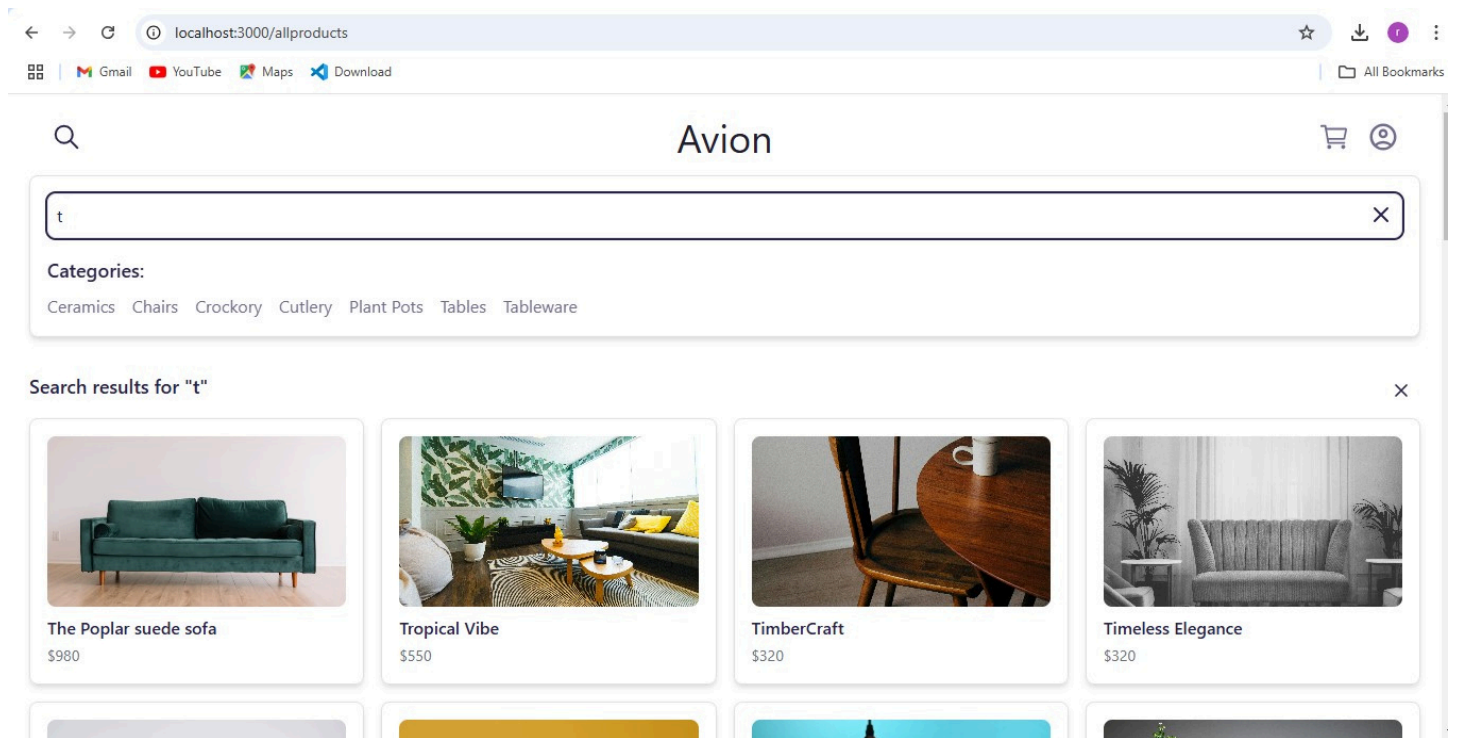
product 2:



product 2 Detailpage:



Search bar in action:

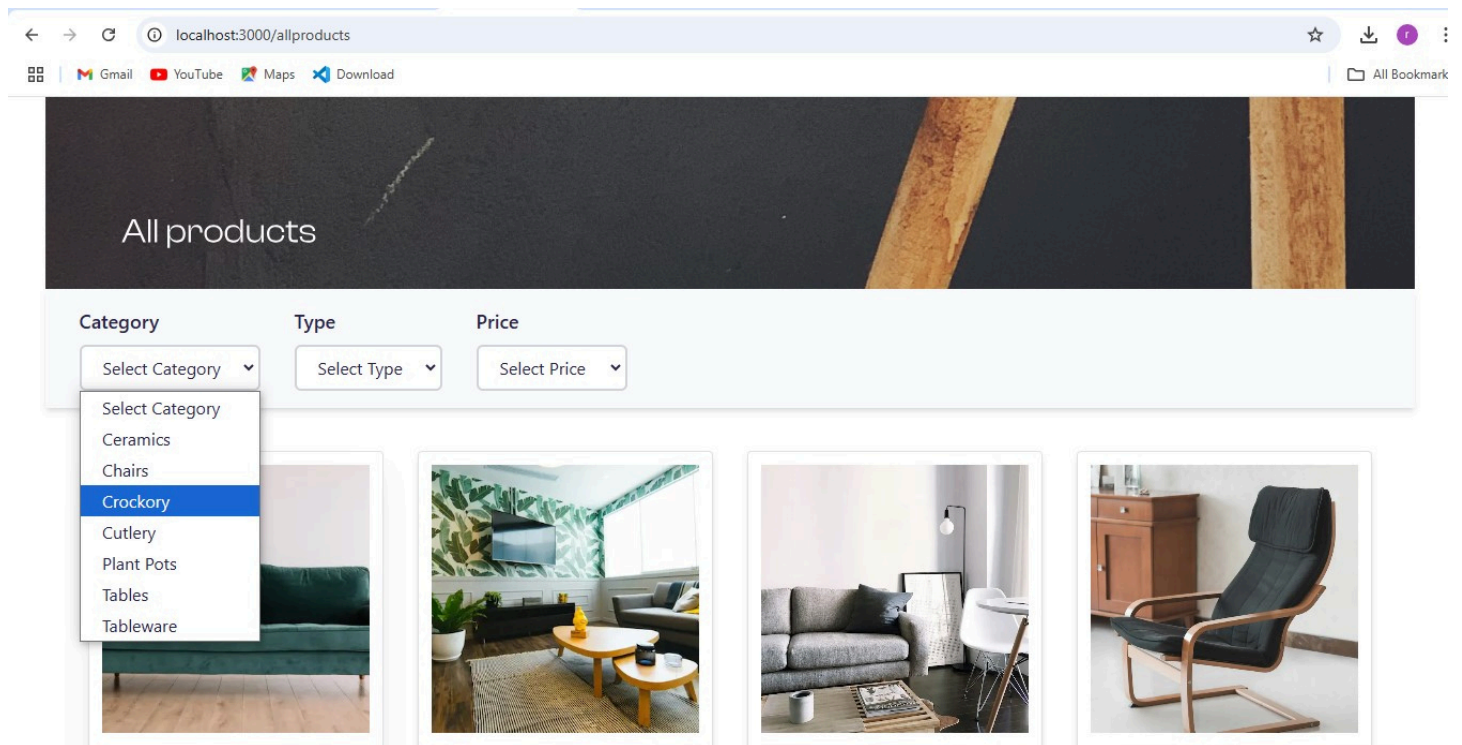


Search Bar Code:

```
25 const Header = () => {
42   }, []);
43
44   // Fetch products based on search term or category selection
45   useEffect(() => {
46     if (searchTerm.length > 0) {
47       const fetchProducts = async () => {
48         const query = `*_type == "product" && name match "${searchTerm}"`;
49         const {
50           name,
51           price,
52           image,
53           slug
54         } = await client.fetch(query);
55         const productsData = await client.fetch(query);
56         setProducts(productsData);
57       };
58       fetchProducts();
59     } else if (selectedCategory) {
60       const fetchCategoryProducts = async () => {
61         const query = `*_type == "product" && references(*[_type == "category" && slug.current == "${selectedCategory}"]..
62         {
63           name,
64           price,
65           image,
66           slug
67         } = await client.fetch(query);
68         const categoryProductsData = await client.fetch(query);
69         setProducts(categoryProductsData);
70       };
71       fetchCategoryProducts();
72     } else {
73       setProducts([]); // Clear products if no search or category selected
74     }
75   }, [searchTerm, selectedCategory]);
76 }
```

```
25 const Header = () => {
92
93   <div className="flex items-center lg:hidden">
94     <IoIosSearch
95       className="text-xl mr-4 text-[#726E8D] cursor-pointer"
96       size={30}
97       onClick={() => setIsSearchOpen((prev) => !prev)}
98     />
99     <HiOutlineMenu
100       className="text-xl cursor-pointer"
101       onClick={() => setIsMenuOpen(true)}
102     />
103   </div>
104 }
```

Filter in action:



Filter code:

```

File Edit Selection View Go Run ... hackathon-3-template-02
EXPLORER
  HACKATHON-3-TEMPLATE-02
  > .next
  > node_modules
  > public
  > src
    > app
      > about
        > page.tsx
      > allproducts
        > page.tsx
      > cart
        > page.tsx
      > checkout
        > page.tsx
      > fonts
      > products\[slug]
        > page.tsx
      > studio
      > favicon.ico
      > globals.css
      > layout.tsx
      > page.tsx
      > components
        > ui
          > button.tsx
          > AboutHeader.tsx
    > OUTLINE
    > TIMELINE
    > IDENTIFY

src > app > allproducts > page.tsx [10] AllProducts
37  const AllProducts = () => {
148
149
150  /* Mobile Filter Button */
151  <div className="md:hidden flex justify-between items-center py-4 px-8">
152    <button
153      className="bg-[#F9F9F9] py-2 px-4 text-sm font-semibold rounded-md hover:bg-[#2A254B] hover:text-white transition"
154      onClick={() => setShowFilters(!showFilters)}
155    />
156    {showFilters ? "Hide Filters" : "Show Filters"}
157  </div>
158
159  /* Filter section for small screens */
160  {showFilters && (
161    <div className="md:hidden w-full px-8 pb-8">
162      <div className="flex flex-col gap-6">
163        /* Category Filter UI */
164        <div className="flex flex-col gap-2">
165          <label htmlFor="category" className="text-lg font-medium">
166            Category
167          </label>
168          <select
169            id="category"
170            value={selectedCategory}
171            onChange={(e) => setSelectedCategory(e.target.value)}
172            className="border-2 border-gray-300 rounded-md px-4 py-2"
173          >
174            <option value="">Select Category</option>
175            {categories.map((category) => (
176              <option
177                key={category.slug.current}
178                value={category.slug.current}

```

Notification:



Added to Cart

Tropical Vibe has been added to your cart.

OK

Notificatin code:

```
File Edit Selection View Go Run ... hackathon-3-template-02 0% [Icons]
```

EXPLORER

- HACKA...
- > .next
- > node_modules
- > public
- src
 - app
 - about
 - page.tsx
 - allproducts
 - page.tsx
 - cart
 - page.tsx
 - checkout
 - page.tsx
 - fonts
 - products\[slug]
 - page.tsx
 - studio
 - favicon.ico
 - global.css

src > app > allproducts > page.tsx > AllProducts

```
37 const AllProducts = () => {
96   const handleAddToCart = (product: IProduct) => {
97     setCart((prevCart) => {
101       return [...prevCart, product];
102     });
103   };
104   // Show SweetAlert notification
105   Swal.fire({
106     title: "Added to Cart",
107     text: `${product.name} has been added to your cart.`,
108     icon: "success",
109     confirmButtonText: "OK",
110     background: "#f9f9f9",
111     color: "#2A254B",
112     confirmButtonColor: "#2A254B",
113   });
114 };
115 if (loading) {
116   return (
117     <div className="flex justify-center items-center w-full h-screen">
118       <div className="border-8 border-t-8 border-gray-300 border-t-[#2A254B] rounded-full w-16 h-16 animate-spin"></div>
119     </div>
120   );
121 }
```