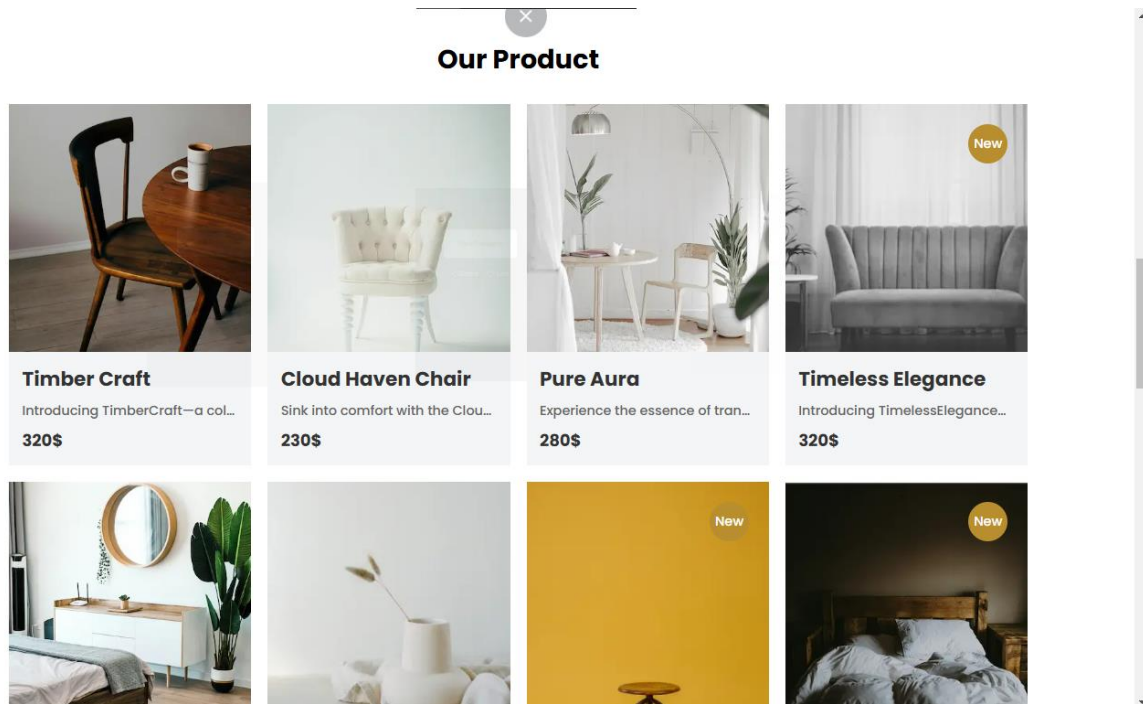


Day 4 - Dynamic Frontend Components – Furniro

Product Listing Page:

The Home Page and Shop Page both utilizes the same API Call to fetch data from Sanity CMS, the only difference is that at the home page, we only display the first 12 out of the 24 products for consistency.



Product Details Page (Dynamic):

Displays all information about a particular product, including the feature to add it to the cart which is managed by a global state management tool called jotai.



Rustic Vase Set

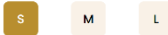
\$210



5 Customer Review

Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those who appreciate timeless beauty and a warm, inviting atmosphere, this set of vases adds a touch of rustic elegance to any space. Crafted with care and attention to detail, these vases are designed to evoke the essence of vintage craftsmanship while seamlessly complementing both modern and traditional decor styles. The Rustic Vase Set features a collection of three uniquely designed vases, each with its own character. Their earthy tones, textured finishes, and artisanal touch capture the essence of the countryside, making them ideal for showcasing fresh flowers, dried arrangements, or simply as stand-alone decor pieces. Whether placed on a mantel, coffee table, or dining area, these vases effortlessly enhance the ambiance of your home. Made from high-quality materials, the Rustic Vase Set offers both style and durability. The natural, imperfect surfaces of the vases give them a distinct, hand-crafted appeal, ensuring that each set is one-of-a-kind. With their timeless design, these vases make a perfect gift for housewarmings, weddings, or any special occasion. Key Features: Set includes three uniquely designed rustic vases Crafted from high-quality materials with a natural, hand-crafted finish Perfect for displaying flowers, greenery, or as standalone decorative pieces Versatile design complements both modern and traditional interiors Ideal for gifting or personal use in any living space Add warmth and character to your home with the Rustic Vase Set—where classic design meets natural beauty.

Size



Color



Related Products:

Displays similar products on every dynamic route

Related Products



Timber Craft

Introducing TimberCraft—a colle...

320\$



Cloud Haven Chair

Sink Into comfort with the Cloud ...

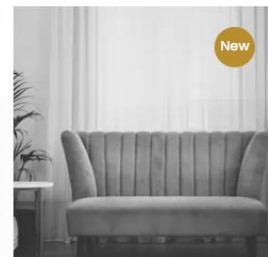
230\$



Pure Aura

Experience the essence of tranqu...

280\$



Timeless Elegance


Introducing TimelessElegance—...

320\$





Show More

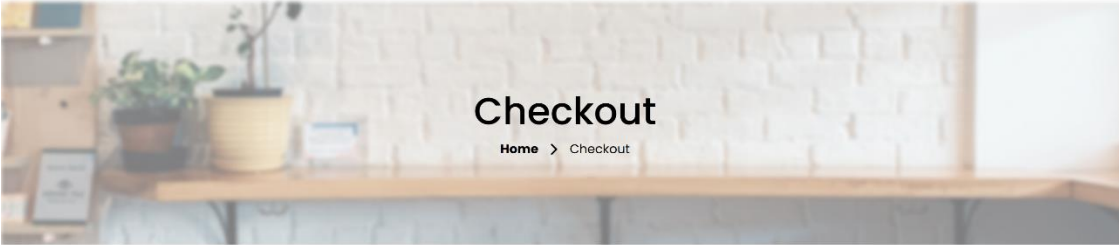
Cart/Checkout:

A Global state is managed by a Third party tool called jotai, which manages cart operations such as adding products, calculating sub totals and removal of product



HomeShopBlogContact





Checkout

Home > Checkout

Billing details

First Name

Last Name

Company Name (Optional)

Country / Region

Select Country

Street address

Town / City

Province

ZIP code

Phone

Email address

Additional Information

Save Billing Info

Product

Rustic Vase Set **x 3**

Subtotal

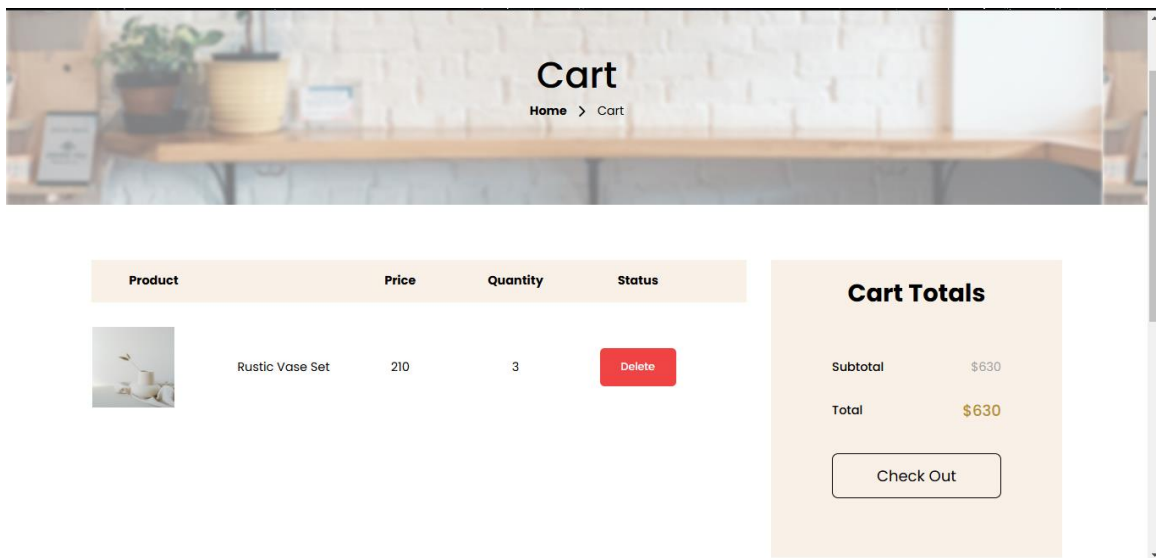
630

Total

Rs. 630

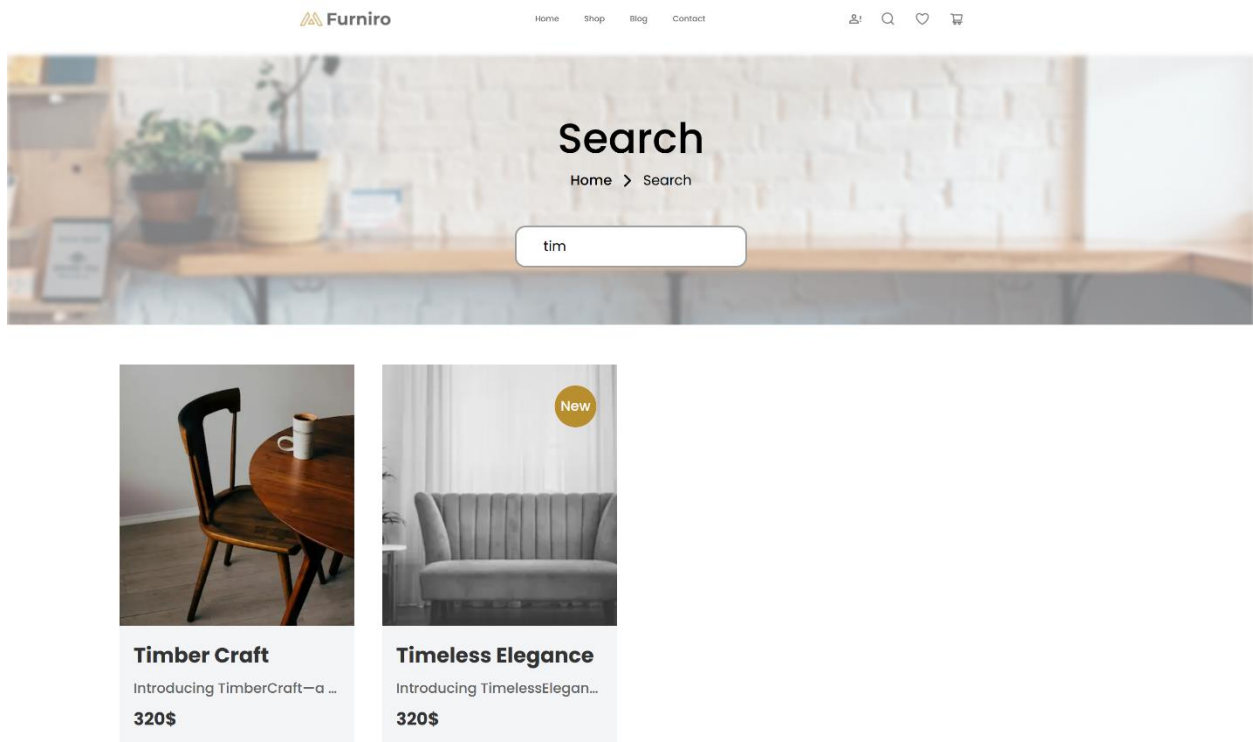
Your personal data will be used to support your experience throughout this website, to manage access to your account, and for other purposes described in our [privacy policy](#).

Place Order



Search Feature:

Fully functional bar that makes an api call based on the search term (which is a local state)



Code Snippets

Product Card:

```
1 <motion.div
2   initial="initial"
3   animate="initial"
4   whileHover="animate"
5   viewport={{ once: false }}
6   transition={{ duration: 0.5 }}
7   className="relative cursor-pointer"
8   onClick={() => router.push(`/shop/product/${item._id}`)}
9 >
10   <div className="relative">
11     <Image
12       src={item.imageUrl}
13       alt="product"
14       className="h-[301px] w-full object-cover"
15       height={300}
16       width={300}
17     />
18     <div className="flex gap-2 absolute top-[24px] right-6">
19       {item.isNew && (
20         <div className="flex items-center justify-between gap-2">
21           <span className="w-12 h-12 rounded-full flex items-center justify-center bg-primary text-white font-medium">New</span>
22         </div>
23       )}
24       {item.discountPercentage > 0 && (
25         <div className="flex items-center justify-between gap-2">
26           <span className="w-12 h-12 rounded-full flex items-center justify-center bg-[#E97171] text-white font-medium">{item.discountPercentage}</span>
27         </div>
28       )}
29     </div>
30   </div>
31   <div className="bg-[#F4F5F7] p-4">
32     <p className="text-customBlack text-24 font-semibold">{item.title}</p>
33     <p className="text-customGray font-medium text-normal py-[8px] truncate">
34       {item.description}
35     </p>
36     <div className="flex justify-between items-center">
37       <p className="text-customBlack text-20 font-semibold">{item.price}$</p>
38     </div>
39   </div>
40   <motion.div
41     className={cn(
42       "absolute p-4 left-0 right-0 top-0 bottom-0 bg-[#3A3A3A]/80"
43     )}
44     variants={cardVariant}
45   >
46     <div className="pt-[30%]">
47       <div className="flex justify-center">
48         <Button
49           className="bg-white text-primary p-8 font-bold hover:bg-white"
50           onClick={() => router.push(`/shop/product/${item._id}`)}
51         >
52           View Products
53         </Button>
54       </div>
55       <div className="flex justify-center gap-5 mt-[24px]">
56         {icons.map((icon, index) => (
57           <div
58             className="flex gap-1 items-center hover:cursor-pointer"
59             key={index}
60             onClick={icon.action}
61           >
62             <div>
63               <img src={icon.imageUrl} alt="icon" />
64             </div>
65             <p className="text-white">{icon.title}</p>
66           </div>
67         ))}
68       </div>
69     </div>
70   </motion.div>
71 </motion.div>
```

Search Bar:

```
1 "use client"
2 import ProductCard from '@components/ProductCard';
3 import { client } from '@sanity/lib/client';
4 import { Product } from '@types/productType';
5 import { useState, useEffect } from 'react';
6 import Search from './Search';
7 import SearchHero from './SearchHero';
8
9 const SearchPage = () => {
10   const [products, setProducts] = useState<Product[]>([]);
11   const [searchTerm, setSearchTerm] = useState('');
12
13   useEffect(() => {
14     const fetchData = async () => {
15       try {
16         const query = `*_type == "product" &&
17           lower(title) match "${searchTerm.toLowerCase()}"
18         ] {
19           _id,
20           title,
21           "imageUrl": productImage.asset->url,
22           price,
23           tags,
24           dicountPercentage,
25           description,
26           isNew
27         } | order(title asc)`;
28         const data = await client.fetch(query, { searchTerm });
29         setProducts(data);
30         console.log(products)
31       } catch (error) {
32         console.error('Error fetching products:', error);
33       }
34     };
35     fetchData();
36   }, [searchTerm]);
37
38   const handleSearch = (searchTerm: string) => {
39     setSearchTerm(searchTerm);
40   };
41
42   return (
43     <>
44       <SearchHero>
45         <Search onSearch={handleSearch} />
46       </SearchHero>
47       <div className='flex flex-col gap-[56px] mx-4 md:mx-[130px]'\>
48         {products.length === 0 || searchTerm === '' ? (
49           <p className='mt-10 text-[24px] text-red-500 font-bold text-center'\>No products found.</p>
50         ):(
51           <section className='w-full overflow-x-hidden'\>
52             <div className="grid grid-cols-1 md:grid-cols-2 xl:grid-cols-4 gap-[32px] mt-[46px]">
53               {products.map((item: Product) => (
54                 <ProductCard item={item} key={item._id} />
55               ))}
56             </div>
57           </section>
58         )}
59       </div>
60     </>
61   );
62 };
63
64 export default SearchPage;
```

API Route for all products:

```
1 import { client } from "@sanity/lib/client";
2 import { NextResponse } from "next/server";
3
4 export async function GET() {
5   try {
6     const data = await client.fetch(`*[_type=="product"]{
7       _id,
8       title,
9       "imageUrl" :productImage.asset -> url,
10      price,
11      tags,
12      dicountPercentage,
13      description,
14      isNew
15    }`);
16
17     return NextResponse.json(data, { status: 200 });
18   } catch (error) {
19     console.error('Error fetching data from Sanity:', error);
20     return new NextResponse('Error fetching data', { status: 500 });
21   }
22 }
```


API Route for Single products:

```
1 import { NextRequest, NextResponse } from "next/server";
2 import { client } from "@sanity/lib/client";
3
4 export async function GET( req: NextRequest, { params }: { params: Promise<{ productId: string }> }) {
5
6     const productId = (await params).productId;
7
8     if (!productId) {
9         return NextResponse.json(
10             { message: "Product ID is missing" },
11             { status: 400 }
12         );
13     }
14
15     try {
16         const product = await client.fetch(
17             `*[_type == "product" && _id == $id][0] {
18                 _id,
19                 title,
20                 imageUrl: productImage.asset->url,
21                 price,
22                 tags,
23                 discountPercentage,
24                 description,
25                 isNew
26             }`,
27             { id: productId }
28         );
29
30         if (!product) {
31             return NextResponse.json(
32                 { message: "Product not found" },
33                 { status: 404 }
34             );
35         }
36
37         return NextResponse.json(product, { status: 200 });
38     } catch (error) {
39         console.error("Error fetching product:", error);
40         return NextResponse.json(
41             { message: "Error fetching product" },
42             { status: 500 }
43         );
44     }
45 }
46
```