

# Technical report

## DAY: 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

### *Product listing*

#### Filter and Sorting Products:

##### Filtering:

Filters products based on a user's search query, showing only those with titles that match the query. This enhances search accuracy and user experience.

##### Sorting:

Organizes filtered products by various criteria:

- **A-Z:** Sorts titles alphabetically.
- **Z-A:** Sorts titles in reverse alphabetical order.
- **Price:** Arranges products from lowest to highest price.

These functions streamline product navigation, making it user-friendly and efficient.

```
42 // Filter products based on search query
43 const filteredProducts = products.filter((product) =>
44   product.title?.toLowerCase().includes(searchQuery.toLowerCase())
45 );
46 // Sorting Logic
47 const sortedProducts = [...filteredProducts].sort((a, b) => {
48   if (sortOption === "a-z") {
49     return (a.title ?? "").localeCompare(b.title ?? ""); // Sort in alphabetical order
50   } else if (sortOption === "z-a") {
51     return (b.title ?? "").localeCompare(a.title ?? ""); // Sort in reverse alphabetical order
52   } else if (sortOption === "price") {
53     return a.price - b.price; // Sort by Price (Low to High)
54   }
55   return 0; // Default order (no sorting)
56 });
57
```


Filter

Showing 1-8 of 24 results


Show16

Sort by


A-ZDefaultA-ZZ-APrice




**Amber Haven**  
Step into a world of warmth and tranquility with Amber Hav...  
**\$150**



**Bed**  
Introducing the Bed—your sanctuary for rest and relaxation...  
**\$250**



**Bold Nest**  
Welcome to BoldNest—where fearless design meets comfort an...  
**\$260**



**Bright Space**  
Welcome to BrightSpace—a collection designed to infuse you...  
**\$180**


Filter

Showing 1-8 of 24 results


Show16

Sort by


Z-ADefaultA-ZZ-APrice




**Zen Table**  
Introduce tranquility and balance into your home with ZenT...  
**\$250**



**Wood Chair**  
Introducing the Wood Chair—a beautifully crafted piece tha...  
**\$100**



**Vase Set**  
Elevate your home decor with the timeless beauty of the Va...  
**\$150**



**Tropical Vibe**  
Escape to paradise with TropicalVibe—a collection that bri...  
**\$550**


Filter

Showing 1-8 of 24 results


Show16

Sort by

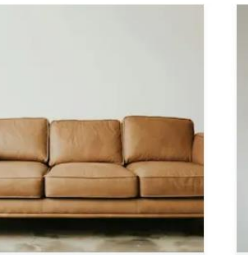
PriceDefaultA-ZZ-APrice



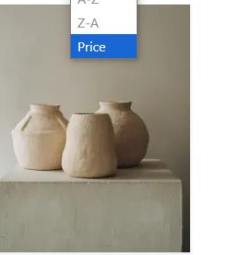
**Wood Chair**  
Introducing the Wood Chair—a beautifully crafted piece tha...  
**\$100**



**The Dandy chair**  
Meet The Dandy Chair—the epitome of comfort, style, and so...  
**\$150**



**Amber Haven**  
Step into a world of warmth and tranquility with Amber Hav...  
**\$150**



**Vase Set**  
Elevate your home decor with the timeless beauty of the Va...  
**\$150**

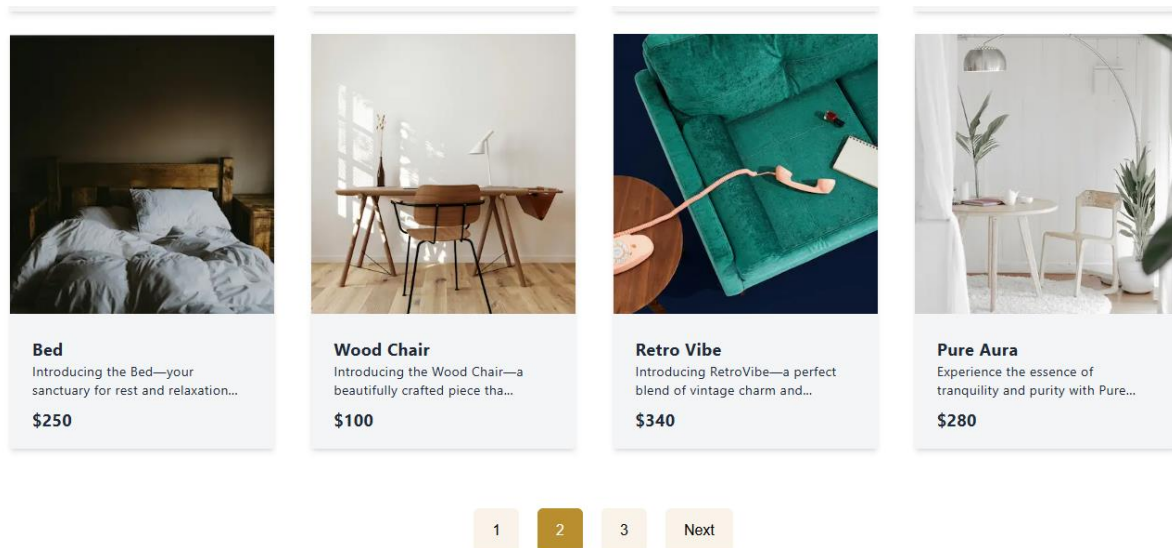
PREPARED BY AYESHA MARYAM

## Pagination:

In this e-commerce website, a total of 24 products are divided into 3 pages, with each page displaying 8 products. This approach ensures faster loading times and better navigation.

To maintain a clean user interface, pagination controls (such as next, and page numbers) are rendered only when pagination is enabled.

```
58 // Apply pagination only if enablePagination is true
59 const displayedProducts = enablePagination
60   ? sortedProducts.slice(
61     (currentPage - 1) * productsPerPage,
62     currentPage * productsPerPage
63   )
64   : sortedProducts;
65
196 /* Render pagination controls only if pagination is enabled */
197 {enablePagination && (
198   <ul className="flex space-x-5 ml-0 sm:ml-[300px] lg:ml-0 xl:ml-[640px] py-10 justify-center font-[sans-serif]">
199     {[...Array(Math.ceil(products.length / productsPerPage))].map(
200       (_, index) => (
201         <li
202           key={index + 1}
203           onClick={() => setCurrentPage(index + 1)}
204           className={`flex items-center justify-center shrink-0 px-[20px] h-12 rounded-md cursor-pointer text-base ${
205             currentPage === index + 1
206               ? "bg-[#B88E2F] text-white"
207               : "bg-[#FAF3EA] text-black hover:bg-[#B88E2F] hover:text-white"
208           }`}>
209             {index + 1}
210           </li>
211         </li>
212       )}
213     <li
214       onClick={() =>
215         setCurrentPage((prev) =>
216           Math.min(prev + 1, Math.ceil(products.length / productsPerPage))
217         )
218       }
219       className={`flex items-center justify-center shrink-0 px-[20px] h-12 rounded-md cursor-pointer text-base ${
220         currentPage === Math.ceil(products.length / productsPerPage)
221           ? "opacity-50 cursor-not-allowed"
222           : "bg-[#FAF3EA] text-black hover:bg-[#B88E2F] hover:text-white"
223       }`}>
224       Next
225     </li>
226   </ul>
227 )}
```



## Product Grid:

Each product is presented within a grid item that typically includes an image, title, price, and short description. The grid adjusts dynamically based on screen size, ensuring a responsive design for different devices.

By structuring products in a grid format, the website enhances user experience, improves navigation, and allows customers to compare multiple products easily at a glance.

```

19  const ProductGrid: React.FC<ProductsProps> = ({
78    return (
79      <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 gap-6 ml-4 py-8 px-4 sm:mx-auto md:ml-10">
80        {displayedProducts.length > 0 ? (
81          displayedProducts.map((product) => {
82            <div
83              key={product._id}
84              className="relative tracking-wide group w-[285px] h-[446px] bg-white shadow-md overflow-hidden hover:-translate-y-2 transition-all"
85            >
86              <Link href={`/product/${product._id}`} >
87                /* Product Image */
88                <div className="relative">
89                  {product.productImage && (
90                    <Image
91                      src={urlFor(product.productImage).url()}
92                      alt="product image"
93                      width={285}
94                      height={301}
95                      className="w-[285px] h-[301px] object-cover"
96                    />
97                  )}
98                </div>

```

```

155      {/* Product Details */}
156      <div className="p-6 h-40 bg-[#F4F5F7]">
157        <h3 className="text-lg font-bold text-gray-800">
158          {product.title}
159        </h3>
160        <h3 className="text-sm text-gray-800">
161          {product.description?.substring(0, 58)}...
162        </h3>
163        <div className="flex items-center space-x-2 mt-2">
164          <p className="text-lg font-bold text-gray-800">
165            ${product.price}
166          </p>
167        </div>
168      </div>
169      </Link>
170    </div>
171  }

```

## Product Hover Overlay:

In this e-commerce website, the overlay appears as a transparent grey background, providing a subtle yet visually distinct effect.

Within the overlay, key action buttons such as **"Add to Cart"** and interactive icons for **Share, Compare, and Like** are displayed. This allows users to quickly engage with products without leaving the grid view. By incorporating these interactive elements, the hover overlay improves usability, encourages user interaction, and enhances the overall shopping experience.

```

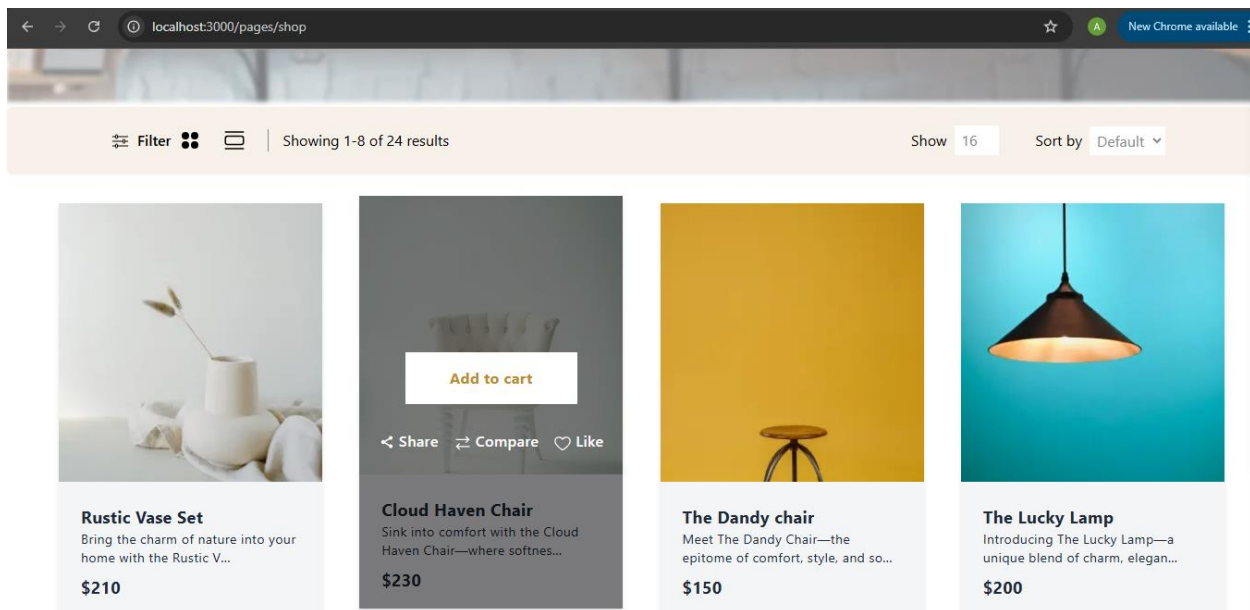
99      {/* Hover Overlay */}
100      <div className="absolute inset-0 bg-black bg-opacity-50 flex flex-col items-center justify-center opacity-0
      group-hover:opacity-100 transition-opacity">
101        <button
102          className="bg-white text-[#888E2F] py-4 px-12 font-lg font-bold mb-3"
103          onClick={(e) => handleAddToCart(e, product)}
104        >
105          Add to cart
106        </button>
107        <div className="flex mt-4 gap-4 font-semibold">
108          <button className="flex gap-1 text-white text-md">
109            <svg
110              xmlns="http://www.w3.org/2000/svg"
111              width={18}
112              height={18}
113              viewBox="0 0 24 24"
114              className="mt-1"
115            >
116              <path
117                fill="currentColor"
118                d="M17.5 3.25a2.75 2.75 0 1 1-1.451 5.086a1 0 0 1-.089.052l-6.713 3.48a3 0 0 1 0 .264l.013.008l6.416 3.802a2.75 2.
119                75 0 1 1-.912 1.784l8.42 13.968a2.75 2.75 0 1 1 .008-3.928l6.415-3.326a2.75 2.75 0 0 1 17.5 3.25"
120              ></path>
121            </svg>
122            Share
123          </button>
124          <button className="flex gap-1 text-white text-md">
125            <svg
126              xmlns="http://www.w3.org/2000/svg"
127              width={20}

```

```

128     viewBox="0 0 32 32"
129     className="mt-1"
130   >
131     <path
132       fill="currentColor"
133       d="M11.41 26.59L7.83 23H28v-2H7.83L3.58-3.59L10 16l-6 6l6 6zM28 10l-6-6l-1.41 1.41L24.17 9H4v2h20.17l-3.58 3.59L22 16z"
134     ></path>
135   </svg>
136   Compare
137 </button>
138 <button className="flex gap-1 text-white text-md">
139   <svg
140     xmlns="http://www.w3.org/2000/svg"
141     width={20}
142     height={20}
143     viewBox="0 0 256 256"
144     className="mt-1"
145   >
146     <path
147       fill="currentColor"
148       d="M178 40c-20.65 0-38.73 8.88-50 23.89C116.73 48.88 98.65 40 78 40a62.07 62.07 0 0 0-62 62c0 70 103.79 126.66 108.21
129a8 8 0 0 7.58 0C136.21 228.66 240 172 240 102a62.07 62.07 0 0 0-62-62m-50 174.8c-18.26-10.64-96-59.11-96-112.8a46.
06 46.06 0 0 1 46-46c19.45 0 35.78 10.36 42.6 27a8 8 0 0 0 14.8 0c6.82-16.67 23.15-27 42.6-27a46.06 46.06 0 0 1 46 46c0
53.61-77.76 102.15-96 112.8"
149     ></path>
150   </svg>
151   Like
152 </button>
153 </div>
154 </div>

```



## Fetching Products from Sanity with limit control

To efficiently display products in the e-commerce website, products are fetched from **Sanity**, a headless CMS. The fetched data includes product details such as title, price, images, and descriptions.

A **limit parameter** is defined to control the number of products displayed on any page. This limit is passed as a **prop** to the product grid component, ensuring flexibility in rendering products dynamically. If a limit is provided, only the specified number of products will be displayed, allowing for better performance and a tailored user experience. This approach ensures scalability and adaptability for different pages, such as featured products on the homepage or full product listings on the shop page.

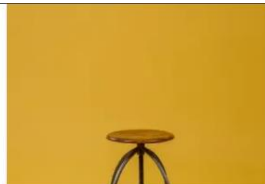
```
29     useEffect(() => {
30       async function fetchProducts() {
31         const fetchedProducts: Product[] = await client.fetch(allProducts);
32
33         // Apply the limit if provided
34         const limitedProducts = limit
35           ? fetchedProducts.slice(0, limit)
36           : fetchedProducts;
37         setProducts(limitedProducts);
38       }
39       fetchProducts();
40     }, [limit]);
41
```



#### Wood Chair

Introducing the Wood Chair—a beautifully crafted piece tha...

**\$100**



#### The Dandy chair

Meet The Dandy Chair—the epitome of comfort, style, and so...

**\$150**



#### Amber Haven

Step into a world of warmth and tranquility with Amber Hav...

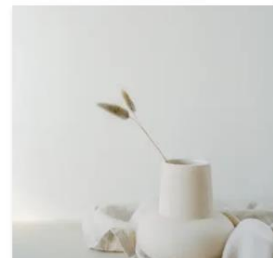
**\$150**



#### Vase Set

Elevate your home decor with the timeless beauty of the Va...

**\$150**



**With limit:**



194

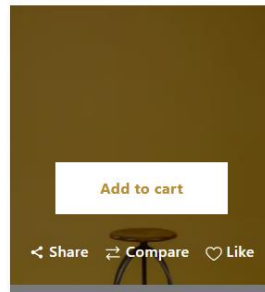
&lt;ProductGrid limit={4} /&gt;

**Rustic Vase Set**

Bring the charm of nature into your home with the Rustic V...

**\$210****Cloud Haven Chair**

Sink into comfort with the Cloud Haven Chair—where softnes...

**\$230****Add to cart**[Share](#) [Compare](#) [Like](#)**The Dandy chair**

Meet The Dandy Chair—the epitome of comfort, style, and so...

**\$150****The Lucky Lamp**

Introducing The Lucky Lamp—a unique blend of charm, elegan...

**\$200**[Show More](#)

localhost:3000/product/YTjpX7gIRz7VphrWQR03Yb

## Add to Cart Functionality:

When a user clicks the "Add to Cart" button on a product, the item is added to the cart state, ensuring it persists throughout the shopping session.

To enhance user experience, a SweetAlert2 notification is displayed upon adding a product. This alert includes:

A success icon to indicate the action was completed.

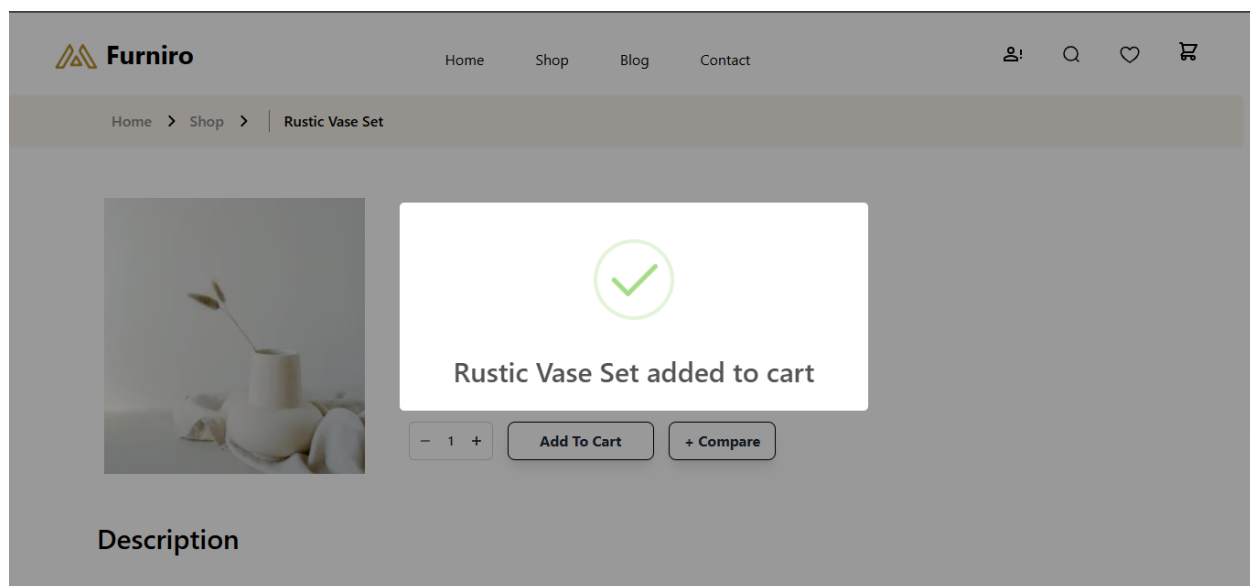
A title saying "Product Added to Cart" to confirm the addition.

This feedback mechanism improves usability by providing real-time confirmation, making the shopping experience more interactive and engaging.



```
65
66 ✓ const handleAddToCart = (e: React.MouseEvent, product: Product) => {
67   e.preventDefault();
68   ✓ Swal.fire({
69     position: "center",
70     icon: "success",
71     title: `${product.title} added to cart`,
72     showConfirmButton: false,
73     timer: 1000,
74   });
75   addToCart(product);
76 };
77

101
102 <button
103   className="bg-white text-[#B88E2F] py-4 px-12 font-lg font-bold mb-3"
104   onClick={(e) => handleAddToCart(e, product)}
105 >
106   Add to cart
</button>
```



## Search Bar Implementation:

The search functionality allows users to quickly find products by entering keywords in the **search input**, which is located in the **Header component**. When a user types a query, the products displayed in the **Product Grid component** are dynamically filtered based on the search term.

To ensure efficient state management, a **Search Context file** is used to handle the search functionality globally. The entire application is wrapped with this context provider, allowing any component to access and utilize the search query. This modular approach ensures seamless filtering across different pages and improves the user experience by providing real-time search results.

```
42 // Filter products based on search query
43 const filteredProducts = products.filter((product) =>
44   product.title?.toLowerCase().includes(searchQuery.toLowerCase())
45 );

172 ) : (
173   // No results found message
174   <div className="col-span-full flex flex-col justify-center items-center py-20">
175     <svg
176       xmlns="http://www.w3.org/2000/svg"
177       width="64"
178       height="64"
179       viewBox="0 0 24 24"
180       className="text-gray-600 m-2"
181     >
182       <path
183         fill="currentColor"
184         d="M7 20.616q-1.671 0-2.835-1.165Q3 18.286 3 16.616q0-1.672 1.165-2.836Q5.329 12.616 7 12.616t2.836 1.165
164T11 16.616q0 1.67-1.164 2.835Q8.67 20.616 7 20.616m13.446-.462l-6.284-6.285q-.27.231-.645.443t-.701.
327q-.108-.21-.234-.413t-.259-.368q1.258-.523 2.083-1.667T15.23 9.5q0-1.971-1.38-3.351t-3.35-1.38T7.149 6.
15T5.769 9.5q0 .304.051.615q.051.31.115.594q-.22.012-.488.095t-.481.167q-.089-.313-.143-.694T4.769 9.5q0-2.
398 1.667-4.064Q8.102 3.769 10.5 3.769t4.065 1.667T16.23 9.5q0 1.075-.376 2.028t-.957 1.657l6.256 6.261zMS.
244 18.917L7 17.161l1.75 1.756l.552-.546l-1.756-1.755l1.756-1.756l-.546-.546L7 16.069l-1.756-1.755l-.546.
545l1.756 1.756l-1.756 1.756z"
185     />
186   </svg>
187   <p className="text-2xl font-semibold text-gray-700">
188     No Results Found
189   </p>
190   <p className="text-gray-500 text-sm mt-2">
191     Try adjusting your search or filter to find what you're looking for.
192   </p>
193 </div>
194 )}
```

src > app > context > searchContext.tsx > ...

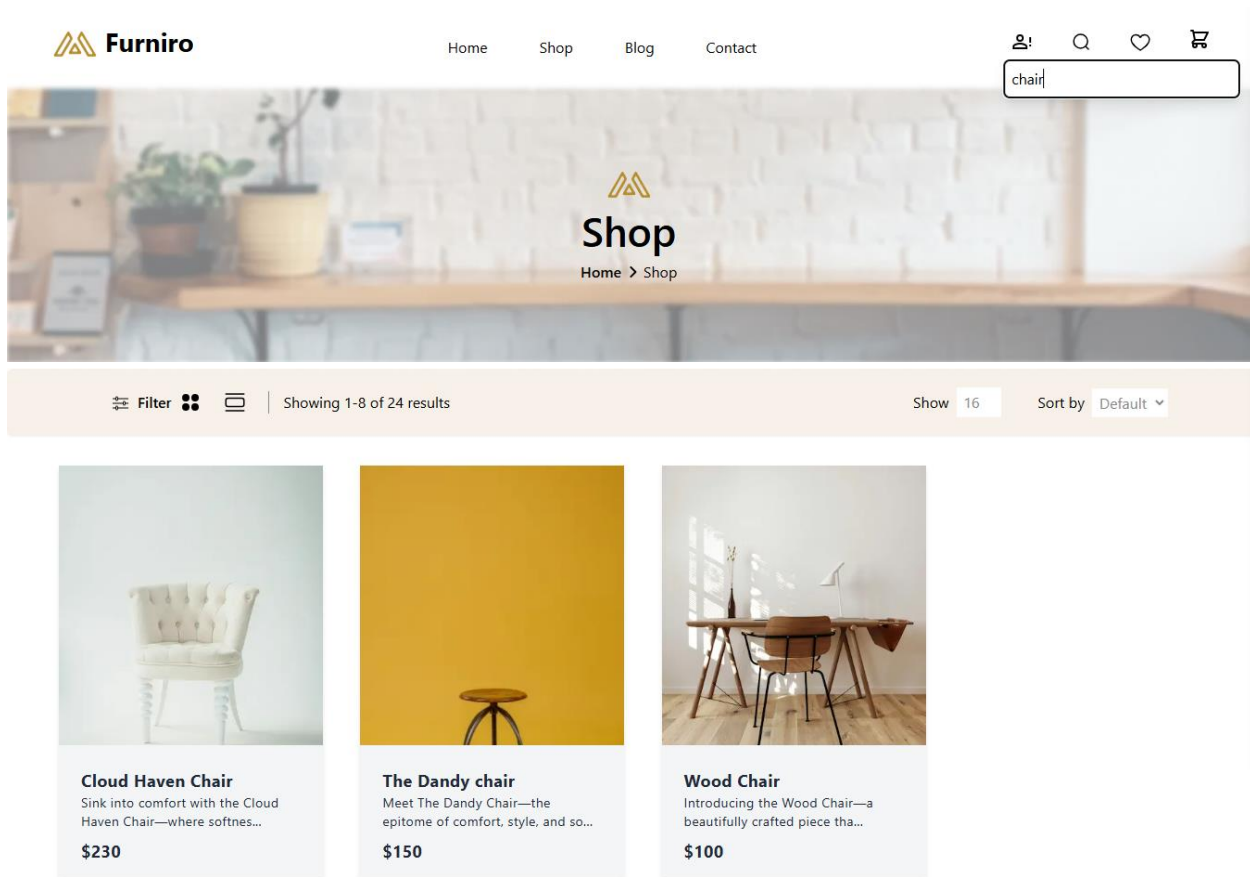
```
1  "use client"
2
3  import { createContext, useContext, useState } from "react";
4
5  // Create Context
6  const SearchContext = createContext({
7    searchQuery: "",
8    setSearchQuery: (query: string) => {},
9  });
10
11 // Provider Component
12 export const SearchProvider = ({ children }: { children: React.ReactNode }) => {
13   const [searchQuery, setSearchQuery] = useState("");
14
15   return (
16     <SearchContext.Provider value={{ searchQuery, setSearchQuery }}>
17       {children}
18     </SearchContext.Provider>
19   );
20 };
21
22 // Custom Hook for easy access
23 export const useSearch = () => useContext(SearchContext);
24
```

```
89  /* Search Icon */
90  <button>
91    <svg
92      xmlns="http://www.w3.org/2000/svg"
93      width={24}
94      height={24}
95      viewBox="0 0 24 24"
96      onClick={handleSearchToggle}
97    >
98      <path
99        fill="none"
100       stroke="currentColor"
101       strokeLinecap="round"
102       strokeLinejoin="round"
103       strokeWidth={1.5}
104       d="M19 11.5a7.5 7.5 0 1 1-15 0a7.5 7.5 0 0 1 15 0m-2.107 5.421l3.08"
105     />
106   </svg>
107
108   /* Conditionally render search bar */
109   {isSearchVisible && (
110     <input
111       type="text"
112       placeholder="Search products..."
113       value={searchQuery}
114       onChange={(e) => setSearchQuery(e.target.value)}
115       className="absolute top-12 right-0 mt-2 sm:mt-4 mr-4 w-32 sm:w-48 lg:w-64 p-2 border border-gray-300 rounded-md shadow-lg"
116     />
117   )}
118 </button>
```

```

34 | <SearchProvider>
35 | <Header/>
36 | {children}
37 | <Footer/>
38 | </SearchProvider>

```



## Dynamic routing

Dynamic routing is implemented to generate individual product detail pages when a user clicks on a product in the **Product Grid**. Each product has a unique identifier (such as a slug or ID), which is used to dynamically fetch and display detailed information.

On the **Product Detail Page**, the following elements are shown:

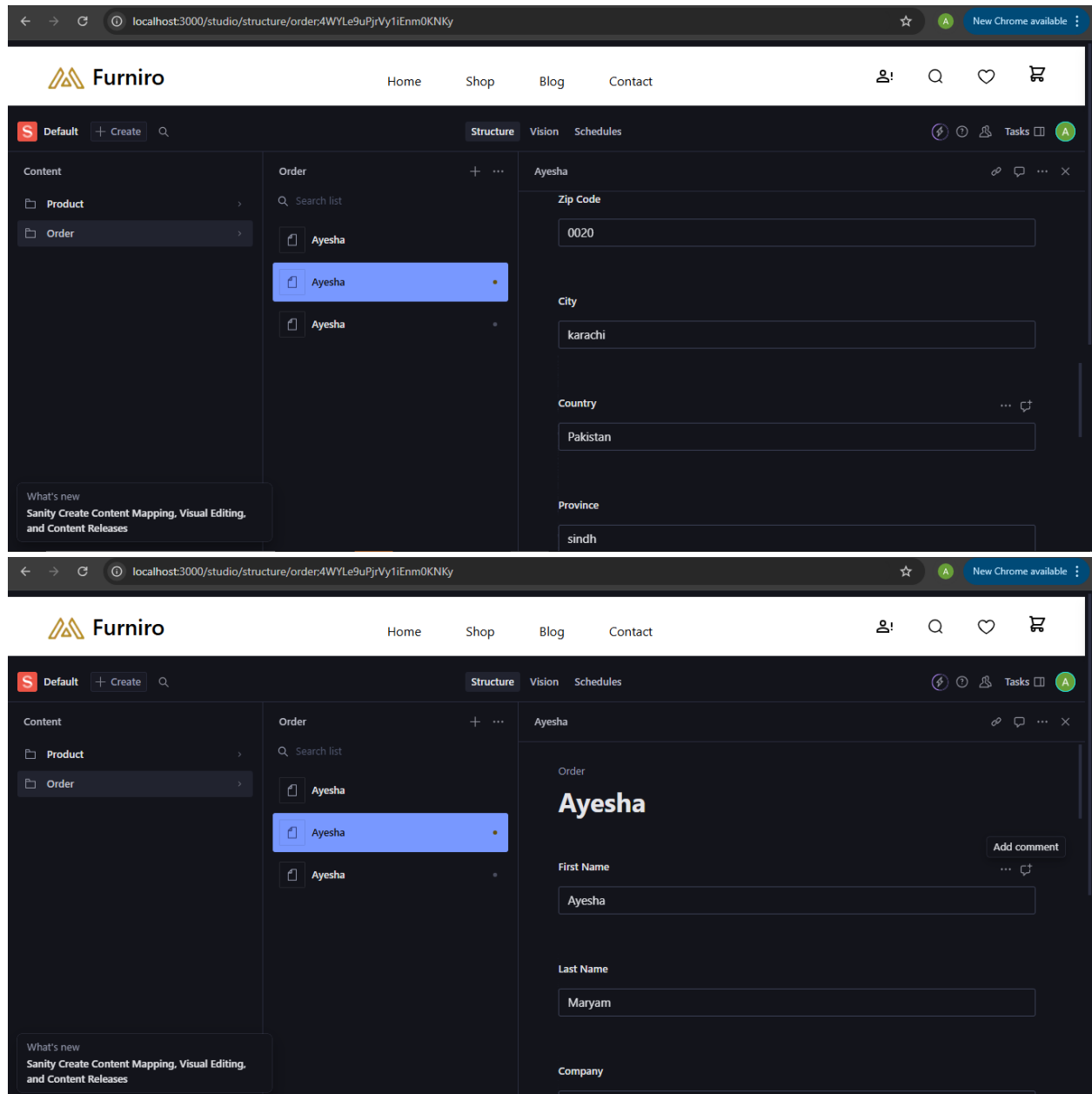
- **Product Image**

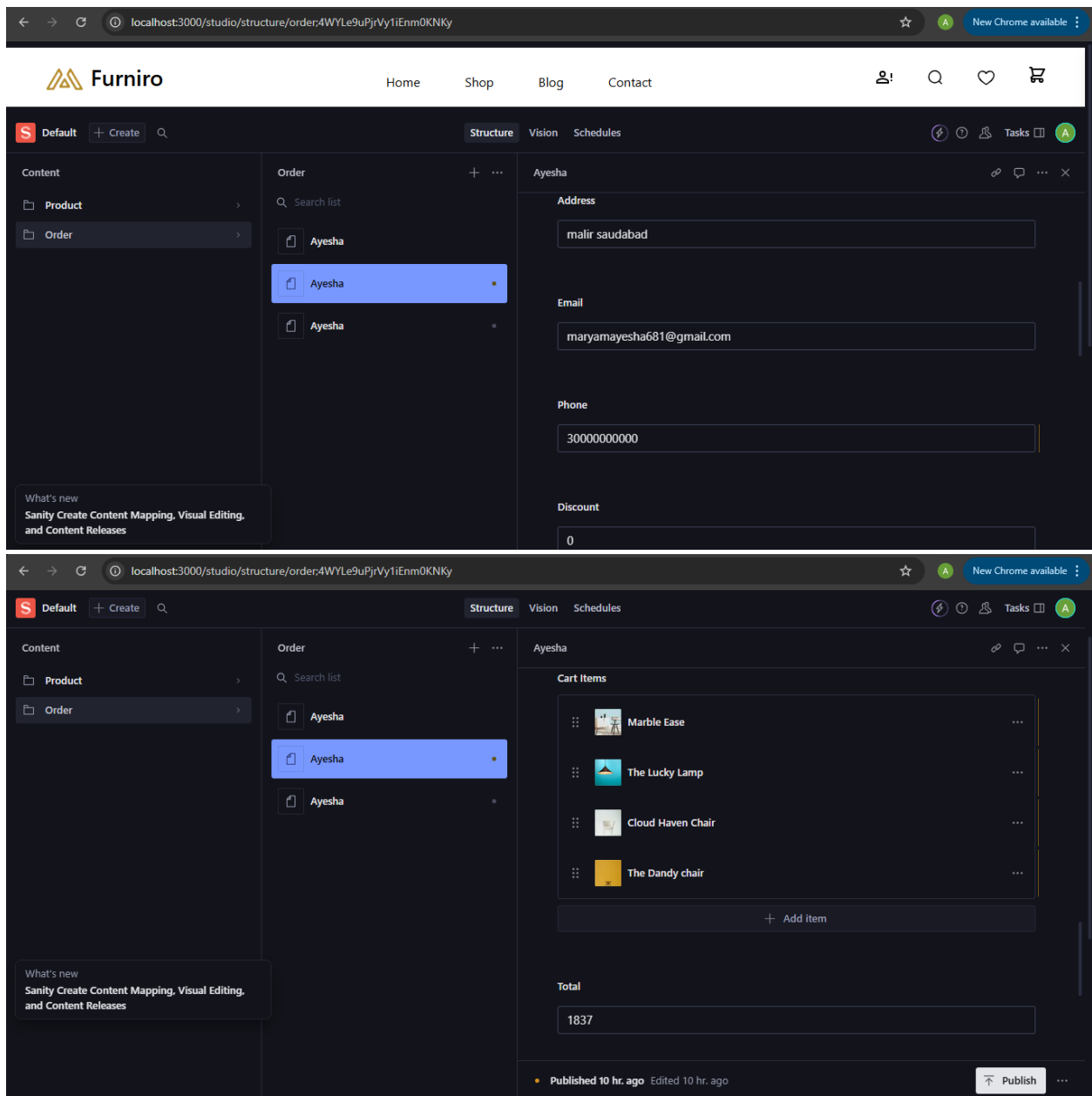
- **Size and Color Options**
- **Product Description**
- **Price**
- **Add to Cart Button** (with SweetAlert2 confirmation)
- **Related Products Section**

This approach ensures a seamless navigation experience, allowing users to explore products in detail while maintaining a structured and user-friendly e-commerce interface.

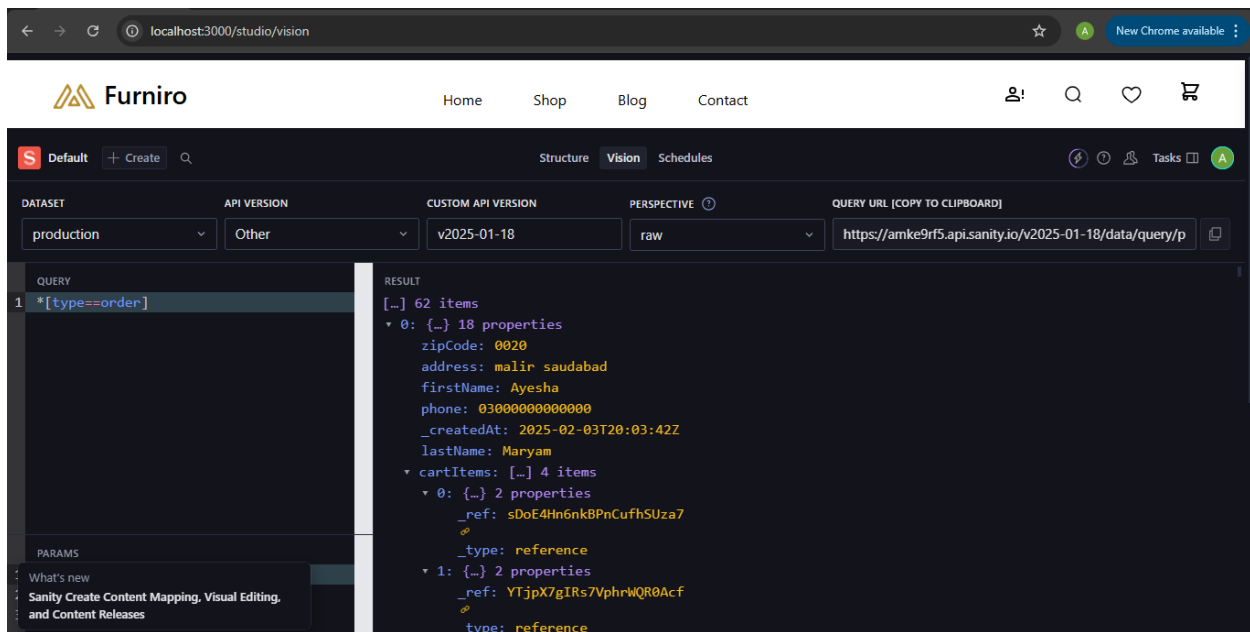
```
src > app > product > [id] > page.tsx > ...
1  "use client"
2
3  import { client } from "@sanity/lib/client";
4  import { groq } from "next-sanity";
5  import { urlFor } from "@sanity/lib/image";
6  import Image from "next/image";
7  import ProductGrid from "@app/components/ProductGrid";
8  import Link from "next/link";
9  import Swal from "sweetalert2";
10 import { addToCart } from "@app/actions/actions";
11 import { Product } from "../../types/product";
12
13 interface ProductPageProps {
14   params: { id: string };
15 }
16
17 const getProduct = async (id: string) => {
18   const query = groq`*[_type == "product" && _id == $id][0]{
19     _id,
20     title,
21     description,
22     price,
23     productImage
24   }`;
25   return await client.fetch(query, { id });
26 };
27
28 const handleAddToCart = (e: React.MouseEvent, product: Product) => {
29   e.preventDefault();
30   Swal.fire({
31     position: "center",
32     icon: "success",
33
34     const handleAddToCart = (e: React.MouseEvent, product: Product) => {
35       e.preventDefault();
36       Swal.fire({
37         position: "center",
38         icon: "success",
39         title: `${product.title} added to cart`,
40         showConfirmButton: false,
41         timer: 1000,
42       });
43       addToCart(product);
44     };
45
46 export default async function ProductPage({ params }: ProductPageProps) {
47   const product = await getProduct(params.id);
48
49   if (!product) {
50     return <div>Product not found</div>;
51   }
52
53   return (
54     <div className="font-sans">
55       <div className="flex items-center justify-between bg-[#f8f1e9] p-4 rounded-md shadow-sm">
56         <div className="flex items-center lg:pl-24">
57           <div className="flex items-center font-semibold text-black">
58             <li className="flex">
59               <p className="font-semibold text-[#898989]">Home &nbsp;</p>
60               <svg
61                 xmlns="http://www.w3.org/2000/svg"
62                 width={24}
63                 height={24}
64                 viewBox="0 0 12 24">
65                 </svg>
66             </li>
67           </div>
68         </div>
69       </div>
70     </div>
71   );
72 }
```

# OUTPUTS

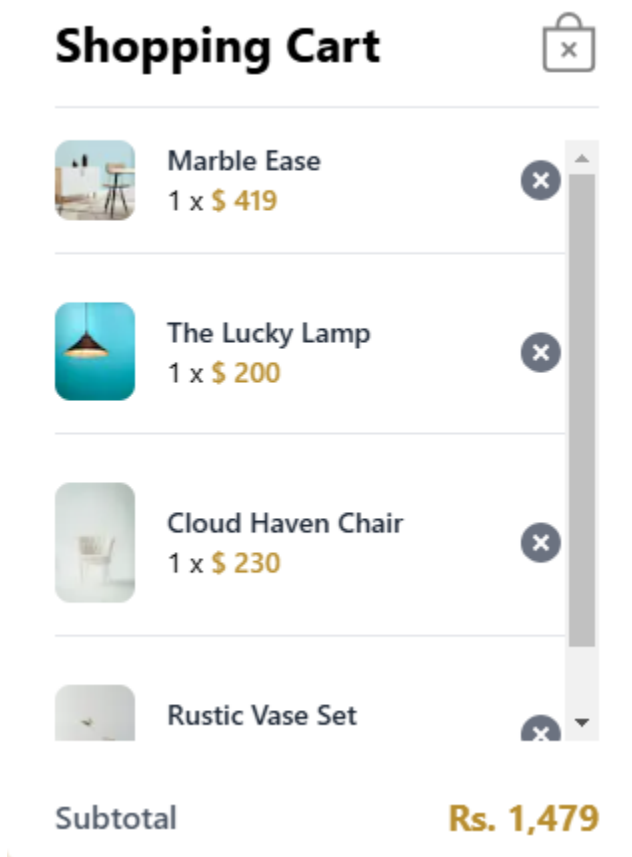
















## Cart Sidebar:

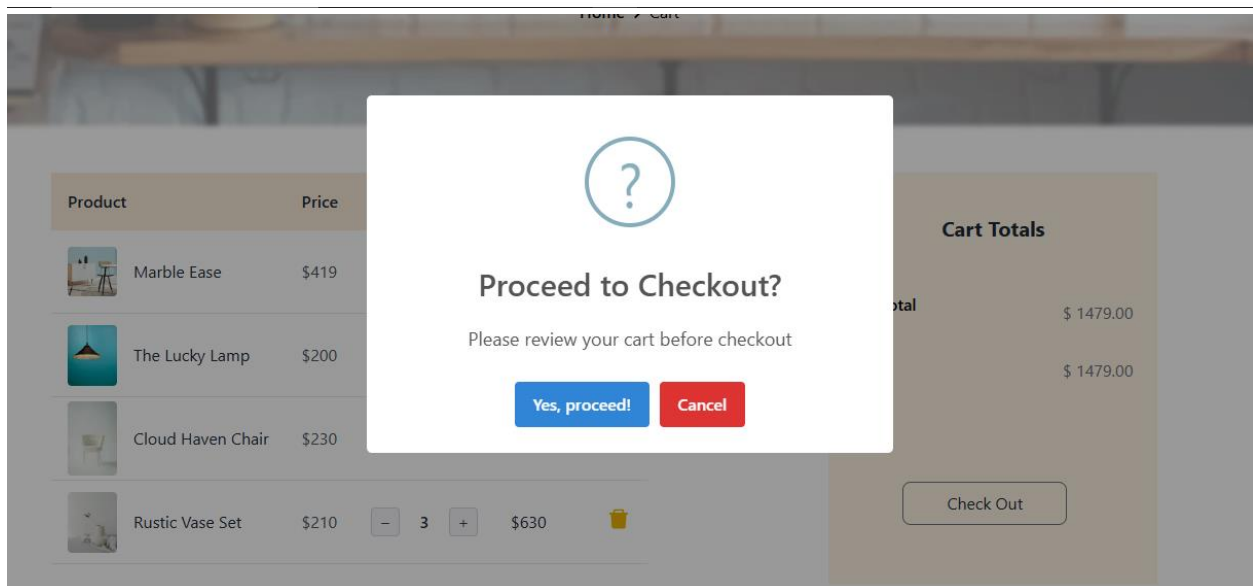


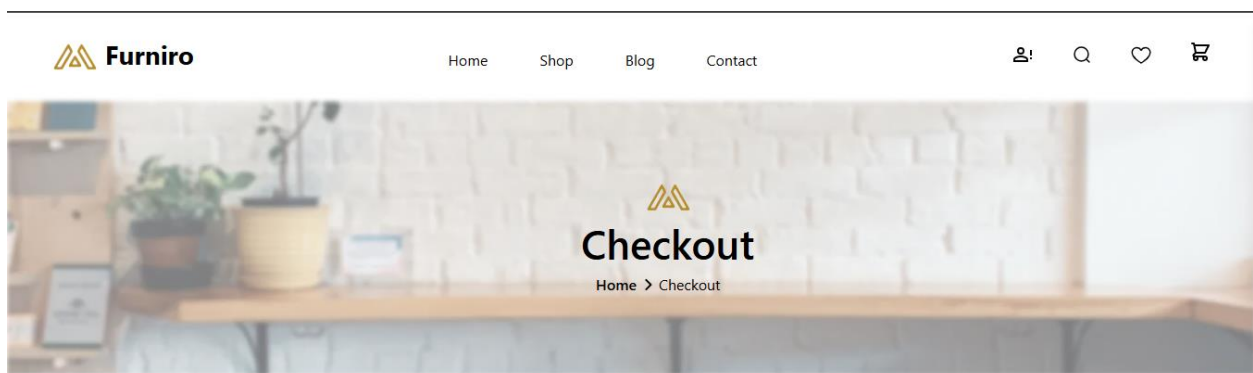
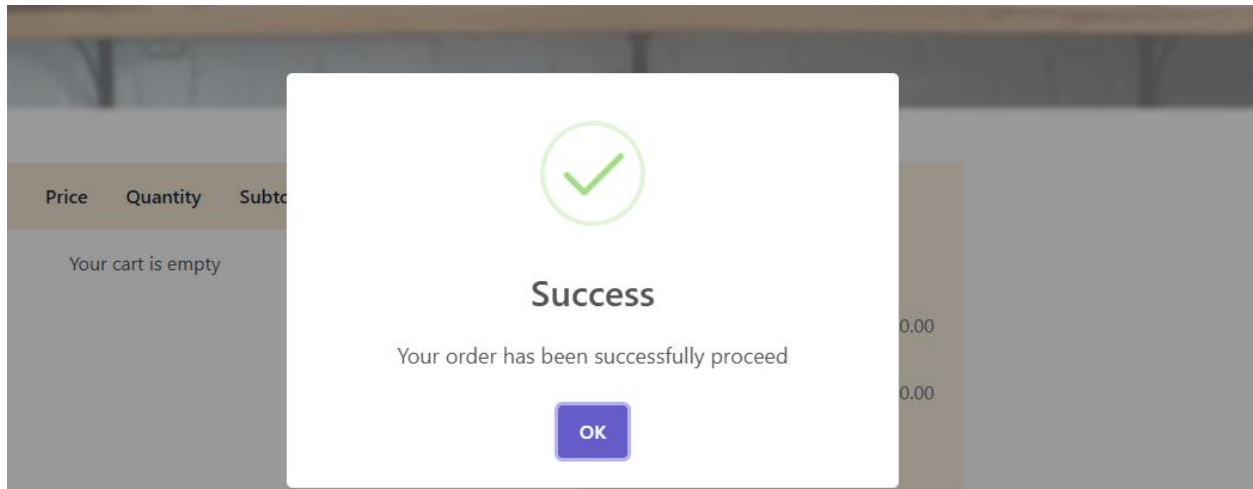
## Cart to Checkout Page:



Product	Price	Quantity			Subtotal	
 Marble Ease	\$419	-	1	+	\$419	
 The Lucky Lamp	\$200	-	1	+	\$200	
 Cloud Haven Chair	\$230	-	1	+	\$230	
 Rustic Vase Set	\$210	-	3	+	\$630	

Cart Totals	
Subtotal	\$ 1479.00
Total	\$ 1479.00
<button>Check Out</button>	





## Billing details

First Name	Last Name
<input type="text" value="Ayesha"/>	<input type="text" value="Maryam"/>
Company Name (Optional)	
<input type="text" value="none"/>	
Contry / Region	
<input type="text" value="Pakistan"/>	
Street Address	
<input type="text" value="malir saudabad"/>	
Town / City	
<input type="text" value="karachi"/>	
Province	

Product		Subtotal
Marble Ease	x 1	\$419
The Lucky Lamp	x 1	\$200
Cloud Haven Chair	x 1	\$230
Rustic Vase Set	x 3	\$630
Subtotal		\$ 1479
Discount		\$ 0
Total		\$ 1479.00

### • Direct Bank Transfer

Make your payment directly into our bank account. Please use your Order ID as the payment reference. Your order will not be shipped until the funds have cleared in our account.

- Direct Bank Transfer
- Cash On Delivery

karachi

Province

sindh

ZIP code

0020

Phone

0333333333

Email address

ayesha123@gmail.com

Additional Information

shipped until the funds have cleared in our account.

- Direct Bank Transfer
- Cash On Delivery

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

Province

sindh

ZIP code

0020

Phone


0333333333

Email address

ayesha123@gmail.com

• Cash On Delivery


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**.



**Processing your order...**

Please wait

[Proceed](#) [Cancel](#)



**Success**

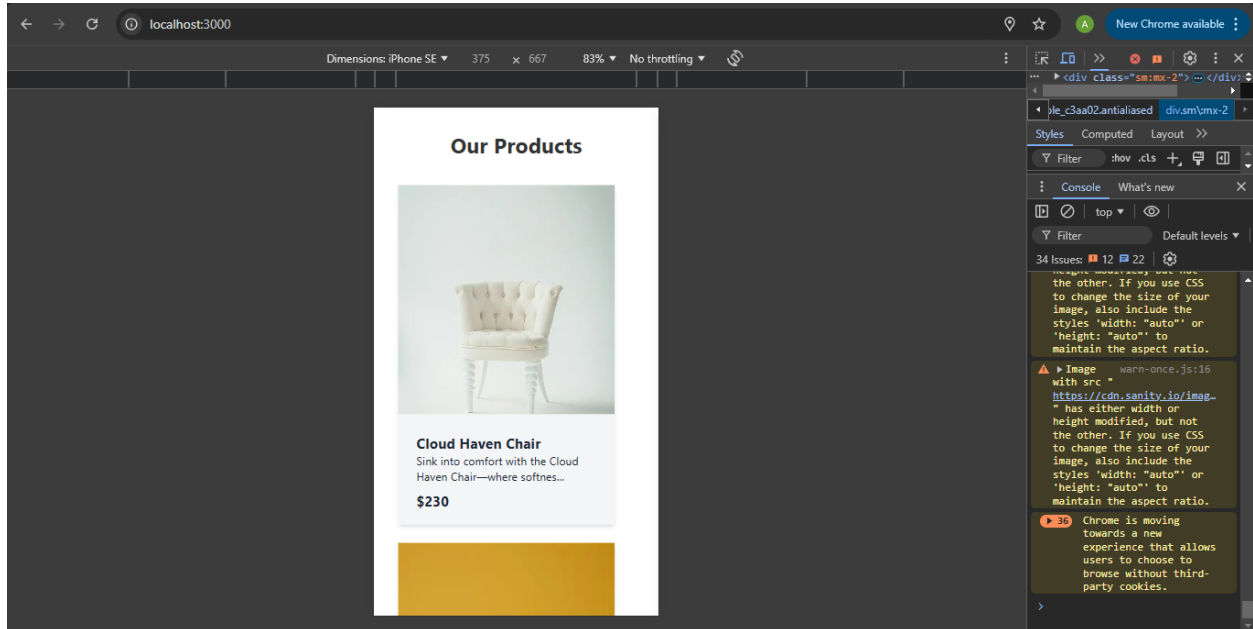
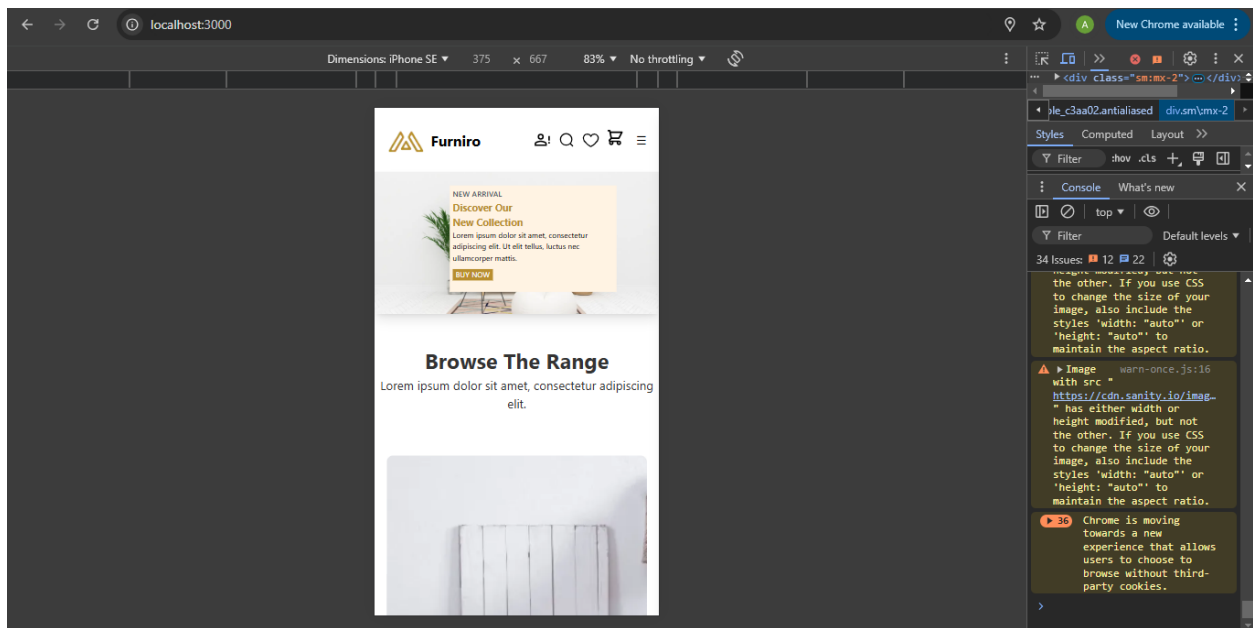
Your order has been successfully processed!

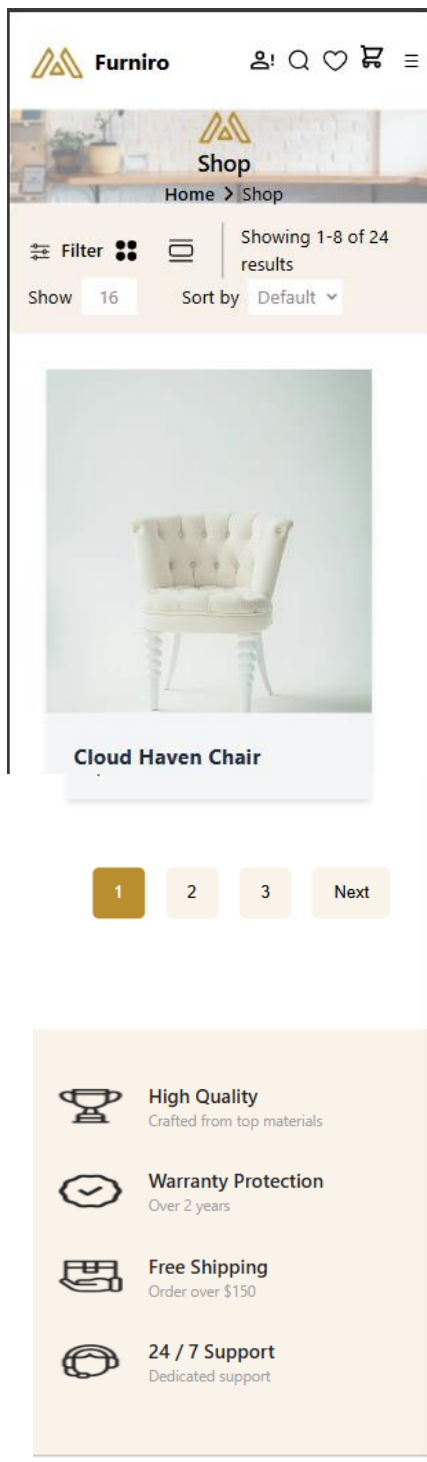
[OK](#)

## Responsive Design Implementation:

This e-commerce website is fully responsive across all screen sizes, ensuring an optimal user experience on desktops, tablets, and mobile devices. Given the website's extensive content and multiple pages, responsiveness has been a key focus in its development.

To demonstrate its adaptability, **screenshots of the website's responsiveness on an iPhone SE** have been included. These screenshots highlight how various elements, such as navigation, product grids, header, footer, shop page, blog page, cart sidebar, cart page, checkout page, sweetalerts and adjust seamlessly to smaller screens while maintaining usability and visual appeal. The responsive design ensures accessibility and a smooth browsing experience for all users, regardless of their device.





## Furniro

400 University Drive Suite 200  
Coral Gables, FL 33134 USA

### Links

Home

Shop

Blog

Contact

### Help

Payment Options

Returns

Privacy Policies

### Newsletter

Enter Your Email Address

SUBSCRIBE

2023 Furniro. All rights reserved.



## Going all-in with millennial design

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Mus mauris vitae ultricies leo integer malesuada nunc. In nulla posuere sollicitudin aliquam ultrices. Morbi blandit cursus risus at ultrices mi tempus imperdiet. Libero enim sed faucibus turpis in. Cursus mattis molestie a iaculis at erat. Nibh cras pulvinar mattis nunc sed blandit libero. Pellentesque elit ullamcorper dignissim cras tincidunt. Pharetra et ultrices neque ornare aenean euismod elementum.



Furniro



Blog

Home > Blog



Admin

14



Wood

Oct

2022



### Categories

Crafts

2

## Recent Posts



**Going all-in with  
millennial design**  
03 Aug 2022



**Exploring new ways of  
decorating**  
03 Aug 2022



**Handmade pieces that  
took time to make**  
03 Aug 2022



**Modern home in  
Milan**  
03 Aug 2022



**Colorful office  
redesign**  
03 Aug 2022

## Exploring new ways of

## Exploring new ways of decorating

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Mus mauris vitae ultricies leo integer malesuada nunc. In nulla posuere sollicitudin aliquam ultrices. Morbi blandit cursus risus at ultrices mi tempus imperdiet. Libero enim sed faucibus turpis in. Cursus mattis molestie a iaculis at erat. Nibh cras pulvinar mattis nunc sed blandit libero. Pellentesque elit ullamcorper dignissim cras tincidunt. Pharetra et ultrices neque ornare aenean euismod elementum.



 Admin  14  
Oct  
2022  Wood

## Handmade pieces that took time to make

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Mus mauris vitae ultricies leo integer malesuada nunc. In nulla posuere sollicitudin aliquam ultrices. Morbi blandit cursus

elementum.

[Read more](#)

1

2

3

Next



### High Quality

Crafted from top materials



### Warranty Protection

Over 2 years



### Free Shipping

Order over \$150



### 24 / 7 Support

Dedicated support



## Get In Touch With Us

For More Information About Our Product & Services,  
Please Feel Free To Drop Us An Email. Our Staff  
Always Be There To Help You Out. Do Not Hesitate!



### Address

236 5th SE Avenue, New  
York NY10000, United  
States



### Phone

Mobile: +(84) 546-6789  
Hotline: +(84) 456-6789



### Working Time

Monday-Friday: 9:00 -  
22:00  
Saturday-Sunday: 9:00 -  
21:00

Your Name

abc

Email address

Abc@def.com

Subject


This is an optional

Message


Hi! i'd like to ask about

Hi! I'd like to ask about


Submit




**High Quality**  
Crafted from top materials



**Warranty Protection**  
Over 2 years









**Free Shipping**  
Order over \$150




**24 / 7 Support**  
Dedicated support


**Furniro**

**Furniro**





**Cart**  
Home > Cart




**Marble Ease**  
Price: \$419


-

1

+

\$419






**The Lucky Lamp**  
Price: \$200


-

1

+

\$200






**Cloud Haven Chair**  
Price: \$230

-

1

+

\$230



PREPARED BY AYESHA MARYAM

