

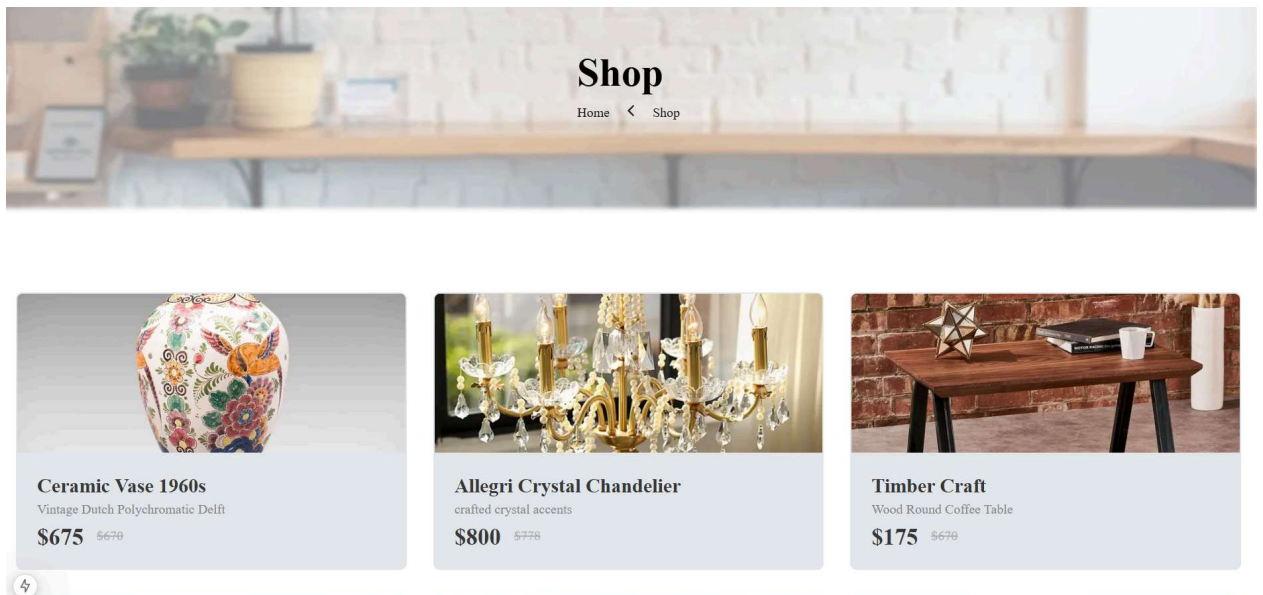
Hackathon_Day_4

Day 4 - Dynamic Frontend Components - [Casa Craft]

Product Listing Page:

A page showcasing dynamic product data including

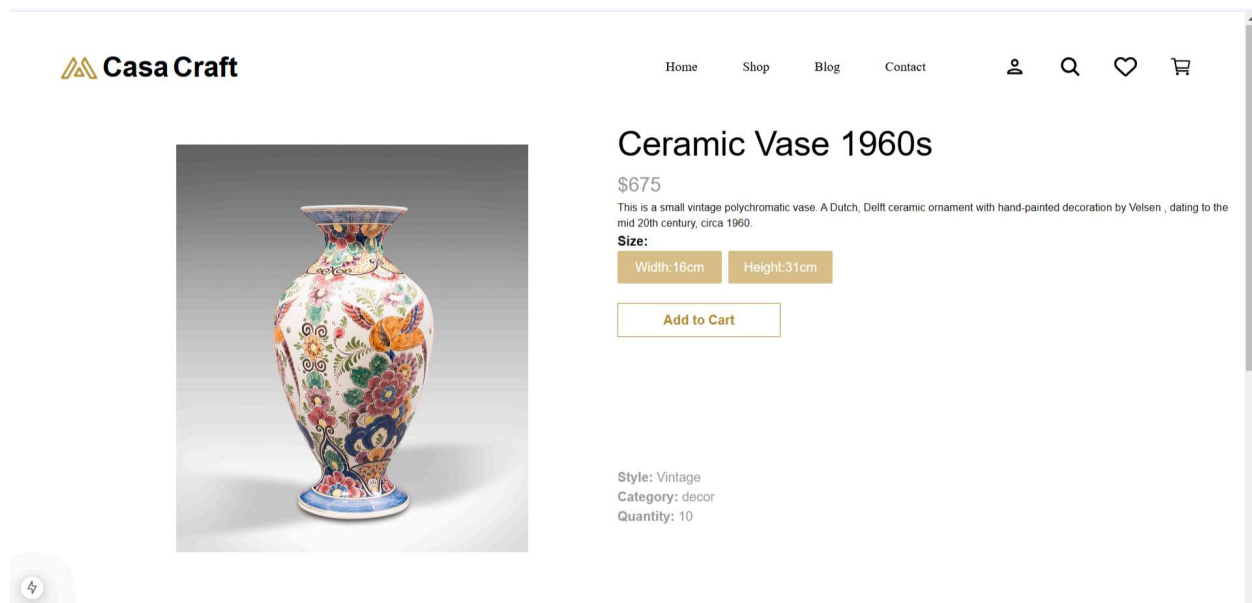
1. Product name
2. Short Description
3. Price
4. Discounted Price
5. Image



Product Detail page Page:

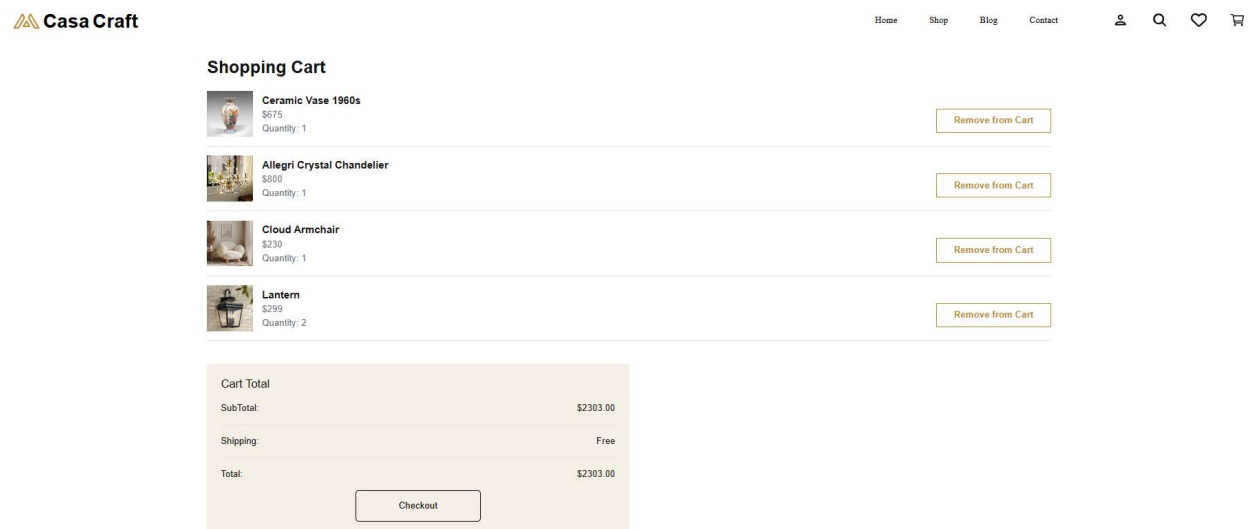
A page showcasing dynamic product detail data including

- Product name
- detail Description
- Price
- Discounted Price
- Image
- Size
- Style
- Category
- Quantity



Add to Cart:

User can add and remove the product dynamically



Code Snippets: ProductCard Component:

```
File Edit Selection View Go Run Terminal Help
Navbar.tsx Setup2.tsx Inspiration.tsx HeroBanner.tsx header.tsx Footer.tsx ProductPage.tsx layout.tsx
app > Shop > page.tsx > Product
1 import Image from "next/image";
2 import { MdCompareArrows } from "react-icons/md";
3 import { CiShare2 } from "react-icons/ci";
4 import { FaRegHeart } from "react-icons/fa6";
5 import { client } from "@sanity/lib/client";
6 import { IoIosArrowBack } from "react-icons/io";
7 import Link from "next/link";
8 import banner from '@image/bgbanner.png'
9 import Hoverbutton from "../component/Hoverbutton";
10
11
12 // Product type define karte hain
13 interface Product {
14   _id: string;
15   name: string;
16   image: any;
17   price: number;
18   discountedPrice?: number;
19   productdescription: string;
20   slug:string
21 }
22
23 // Sanity se data fetch karne ka function
24 async function getData(): Promise<Product[]> {
25   return await client.fetch(`*[_type == "product"]{
26     _id, name, image, price, discountedPrice, productdescription , "slug":slug.current
27   }`);
28 }
29
30 // Server Component
31 const Shop = async () => {
32   const products = await getData();
33   return (
34     <div>
35       { /* Page Heading */ }
36       <div className="relative">
37         <Image src={banner} alt="banner" className="w-full h-[316px]" />
38         <div className="absolute top-1/2 left-1/2 transform -translate-x-1/2 -translate-y-1/2">
39           <h1
40             className="text-black "
41             style={{ fontFamily: "Poppins", fontWeight:700 ,fontSize:48 }}>
42             Shop
43           </h1>
44           <ul className="flex gap-4">
45             <Link href="/"> <li style={{ fontFamily: "Poppins", fontWeight:500 ,fontSize:16 }}>Home</li> </Link>
46             <li><IoIosArrowBack className="w-[20px] h-[20px]" /></li>
47             <Link href="/Shop"> <li style={{ fontFamily: "Poppins", fontWeight:300 ,fontSize:16 }}>Shop</li> </Link>
48           </ul>
49         </div>
50       </div>
51     </div>
52   );
53 }
```

```

/* Product Section */
<section className="text-gray-600 body-font">
  <div className="container px-5 py-24 mx-auto">
    <div className="flex flex-wrap -m-4">
      {products.map((product) => {
        <div key={product.id} className="p-4 md:w-1/3">
          <div className="h-full border-2 bg-gray-200 rounded-lg overflow-hidden group relative">
            /* Sanity Image */

            <Image
              className="lg:h-48 md:h-36 w-full object-cover object-center"
              src={product.image} // Sanity image URL resolve
              alt={product.name}
              width={400}
              height={300}
            />

            /* Overlay effect */
            <div className="absolute top-0 left-0 w-full h-full bg-[#3A3A3A] opacity-0 group-hover:opacity-40 transition-opacity duration-300"></div>

            /* Hover Buttons */
            <Hoverbutton product={product} />

            /* Social Buttons */
            <span className="absolute bottom-4 left-1/2 transform -translate-x-1/2 text-sm text-white opacity-0 group-hover:opacity-100 transition-opacity duration-300 mb-24">
              <ul className="flex flex-row gap-4">
                <li className="flex items-center gap-2">
                  <CiShare2 className="text-lg" />
                  Share
                </li>
                <li>
                  <Link href={`/${Shop}/${product.slug}`}>
                    <li className="flex items-center gap-2">
                      <MdCompareArrows className="text-lg" />
                      view <span>product</span>
                    </li>
                  </Link>
                </li>
                <li className="flex items-center gap-2">
                  <FaRegHeart className="text-lg" />
                  Like
                </li>
              </ul>
            </span>
          </div>
        </div>
      })}
    </div>
  </div>

```


ProductDetail Page:

```
interface Props {
  params: {
    slug: string;
  };
}

export default async function Page({ params }: Props) {
  const { slug } = params;

  // Fetch data from Sanity using the slug
  const data = await client.fetch(
    `*[_type == "product" && slug.current == $slug]{
      name,
      description,
      image,
      price,
      style,
      dimensions,
      category,
      quantity,
      "slug": slug.current
    }`,
    { slug }
  );

  const product = data[0];
  return (
    <div style={{ display: 'flex', justifyContent: 'space-between', alignItems: 'center' }}>
      <div style={{ flex: 1 }}>
        {product.image && (
          <img
            src={product.image}
            alt={product.name}
            style={{ width: '432px', height: '500px', objectFit: 'cover' }}
            className='ml-52 mt-16'
          />
        )}
      </div>
    </div>
  );
}
```

```

export default async function Page({ params }: Props) {
  //
  </div>
  <div style={{ flex: 1, paddingRight: '20px' }}>
    <h1 style={{fontWeight:400, fontSize:42}} className='text-black'>{product.name}</h1>
    <h1 style={{fontWeight:400, fontSize:24}} className='text-[#9F9F9F]'>${product.price}</h1>
    <p style={{fontSize:13}}>{product.description}</p>
    /* Display additional details */
    {product.dimensions && (product.dimensions.width || product.dimensions.height) && (
  </>
  <h3 className='font-bold'>Size:</h3>
  <div className='flex gap-2'>
    {product.dimensions.width &&
    <div className='text-center text-white w-32 h-10 pt-2 rounded-sm bg-[#d9c087]'>Width:{product.dimensions.width}cm</div>
    {product.dimensions.height &&
    <div className='text-center text-white w-32 h-10 pt-2 rounded-sm bg-[#d9c087]'>Height:{product.dimensions.height}cm</div>}
  </div>
  </>
  </div>
  <div>
    <AddToCartButton product={product} />

    <ul className='text-[#9F9F9F] mt-40'>
      {product.style && <li><strong>Style:</strong> {product.style}</li>}
      {product.category && <li><strong>Category:</strong> {product.category}</li>}
      {product.quantity && <li><strong>Quantity:</strong> {product.quantity}</li>}
    </ul>
  </div>
</div>
</div>
);
}

```


Add to Cart Functionality:

```
interface Product {
  name: string;
  price: number;
  slug: string;
  image?: string;
  quantity?: number;
}

export default function Hoverbutton ({ product }: { product: Product }) {
  const handleAddToCart = () => {
    let cart = JSON.parse(localStorage.getItem("Cart") || "[]");

    const existingItem = cart.find((item: any) => item.slug === product.slug);

    if (existingItem) {
      existingItem.quantity += 1;
    } else {
      cart.push({ ...product, quantity: 1 });
    }

    localStorage.setItem("Cart", JSON.stringify(cart));
  };

  return (
    <div className="relative group">
      { /* Your existing button with add to cart functionality */ }
      <button
        className="absolute bottom-4 left-1/2 transform -translate-x-1/2 bg-white text-[#f00]
        onClick={handleAddToCart}
      >
        Add to Cart
      </button>
    </div>
  );
};
```

Remove From Cart Functionality:

```
"use client";

interface Product {
  name: string;
  price: number;
  slug: string;
  image?: string;
  quantity?: number;
}

interface RemoveFromCartButtonProps {
  product: Product;
  onRemove: (productSlug: string) => void; // Accepts a function that takes a product slug
}

export default function RemoveFromCartButton({ product, onRemove }: RemoveFromCartButtonProps) {
  return (
    <button
      className="bg-white border border-[#B88E2F] text-[#B88E2F] font-bold px-4 py-2 w-[200px] mt-6"
      onClick={() => onRemove(product.slug)}
    >
      Remove from Cart
    </button>
  );
}
```