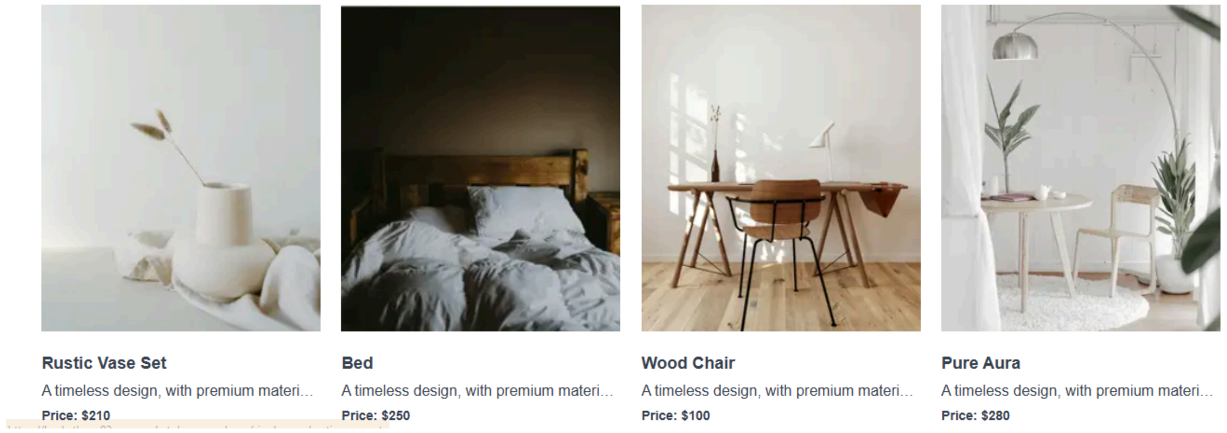
HACKATHON-03
DAY-04
Dynamic Frontend Components
[Costco Club]

1. Functional Deliverables
The product listing page with dynamic data



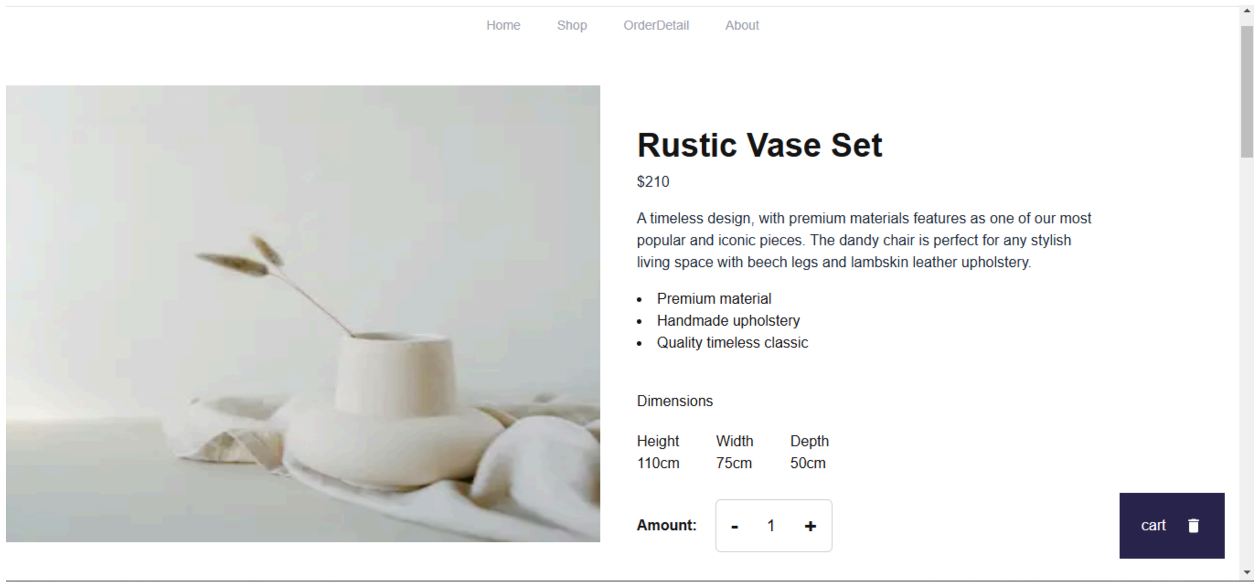
Rustic Vase Set
A timeless design, with premium materi...
Price: $210

Bed
A timeless design, with premium materi...
Price: $250

Wood Chair
A timeless design, with premium materi...
Price: $100

Pure Aura
A timeless design, with premium materi...
Price: $280

Individual product detail pages with accurate routing and data rendering.



Rustic Vase Set

$210

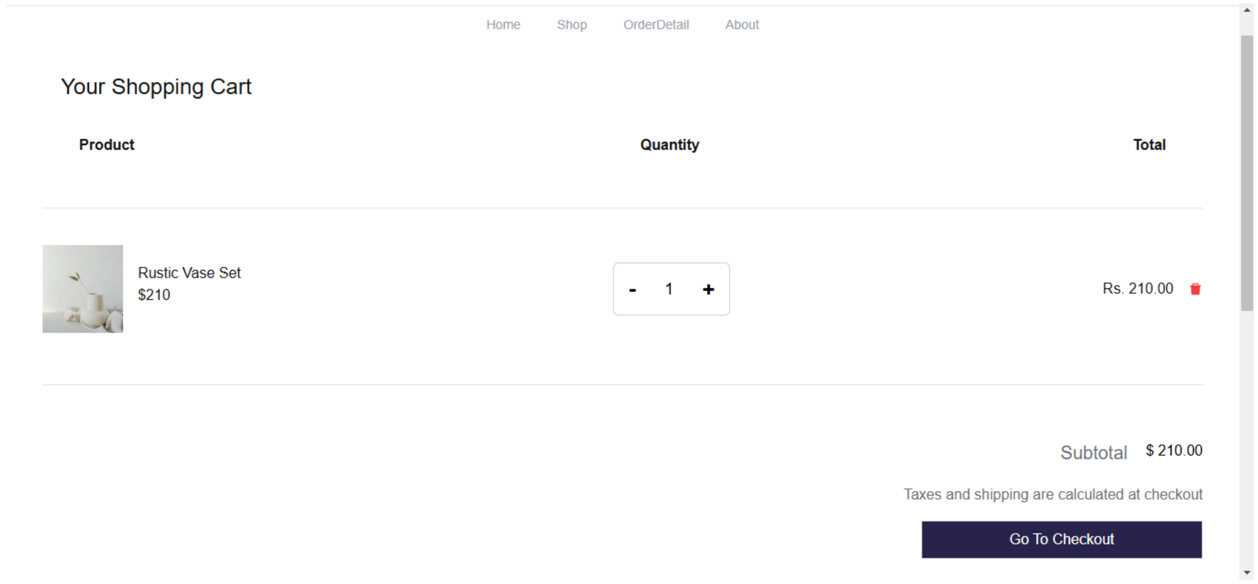
A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather upholstery.

- Premium material
- Handmade upholstery
- Quality timeless classic

Dimensions

Height	Width	Depth
110cm	75cm	50cm

Cart page with dynamic price and cart totals items and price.



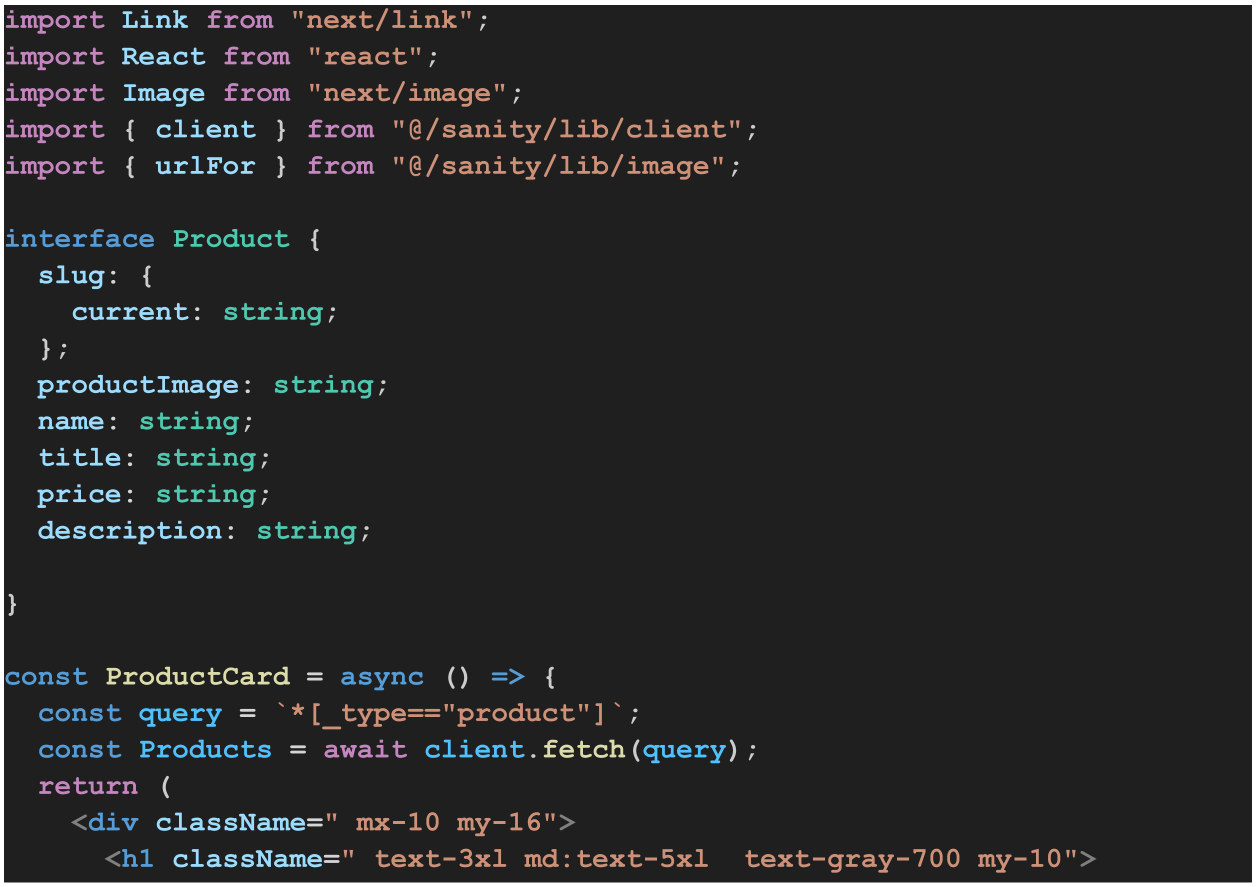
Your Shopping Cart

Product	Quantity	Total
Rustic Vase Set $210	- 1 +	Rs. 210.00

Subtotal $ 210.00

Taxes and shipping are calculated at checkout

Code Deliverables:
Product listing page



```
import Link from "next/link";
import React from "react";
import Image from "next/image";
import { client } from "@/sanity/lib/client";
import { urlFor } from "@/sanity/lib/image";

interface Product {
  slug: {
    current: string;
  };
  productImage: string;
  name: string;
  title: string;
  price: string;
  description: string;

}

const ProductCard = async () => {
  const query = `*[_type=="product"]`;
  const Products = await client.fetch(query);
  return (
    <div className=" mx-10 my-16">
      <h1 className=" text-3xl md:text-5xl  text-gray-700 my-10">
```

```
        New ceramics
      </h1>
      <div className="flex  flex-wrap gap-5 justify-around  ">
        {Products.map((product: Product, index: number) => (  index <= 3
&&
          <Link
            key={index}
            href={`/singlepage/${product.slug.current}`}
            className=""
          >
            <div className="w-[250px] sm:w-[250px] md:w-[300px]
lg:w-[300px]">
              <div className="relative">
          <Image
            src={urlFor(product.productImage).url()}
            alt={product.name}
            width={200}
            height={350}
            className="w-full h-[350px] object-cover"
          />
              </div>
              <div className="-10">
          <h3 className="text-lg text-gray-700 font-bold mb-1 my-5">
            {product.title}
          </h3>
          <h3 className=" text-gray-700  mb-1 line-clamp-1">
            {product.description}
          </h3>
          <p className="text-sm font-bold text-gray-700 mb-2">Price:
${product.price}</p>
              </div>
            </div>
          </Link>
        ))}
      </div>

      <div className="flex justify-center mt-10">
        <Link href="/shop">
          <button className="bg-gray-100  text-gray-700 font-bold py-3
px-10 rounded">
            New Collection{" "}
          </button>
        </Link>
      </div>
    </div>
  );
};
```

```
export default ProductCard;
```

SinglePage

Page.tsx

```
import Image from "next/image";
import ProductCard from "@/components/products";
import Feature from "@/components/feature";
import Banner from "@/components/banner";
import { client } from "@/sanity/lib/client";
import CartIcon from "@/components/cardicon";
import { urlFor } from "@/sanity/lib/image";
async function page({ params }: { params: { slug: string } }) {
  const { slug } = params;
  const product = await client.fetch(
    `*[_type=="product" && slug.current == $slug][0]`,
    { slug }
  );
  if (product) {
    return (
      <div className="">
        <div className="flex flex-col lg:flex-row">
          <div className="flex items-center justify-between">
            <div className="flex "></div>
            <Image
              src={urlFor(product.productImage).url()}
              alt={product.name}
              width={500}
              height={500}
              className="w-[650px] h-[500px] "
            />
          </div>
          <div className="lg:w-1/2 mt-20 lg:pl-8 mx-2">
            <h1 className="sm:text-4xl text-2xl font-bold mb-2">
              {product.title}
            </h1>
            <p className="text-gray-800 mb-4 pr-32">${product.price}</p>
            <p className="text-gray-800 mb-4
pr-32">{product.description}</p>
            <li>Premium material </li>
            <li>Handmade upholstery</li>
            <li>Quality timeless classic</li>
            <h1  className="mt-10">Dimensions</h1>
            <div className="my-5 flex gap-10">
                <p>Height <br /> 110cm</p>
                <p>Width <br /> 75cm</p>
                <p>Depth <br /> 50cm</p>

            </div>

            <div className="mb-4 justify-between flex flex-wrap
mitems-center gap-10">
              <div className="flex gap-5 items-center">
                <h1 className="text-1xl font-bold   ">Amount:</h1>
                <div className="border py-3 w-32 flex justify-around
items-center border-gray-300 rounded-md">
                  <button className="text-2xl font-bold">-</button>
                  <span className="text-lg">{product.quantity}</span>
                  <button className="text-2xl font-bold">+</button>
                </div>
              </div>
              <button className="bg-[#2a254b]  px-4 py-4 ">
                <CartIcon id={product._id} />
              </button>
            </div>
          </div>
        </div>
        <hr className="my-10" />
        <ProductCard />
        <Feature />
        <Banner />
      </div>
    );
  } else {
    return <h1>Product not found</h1>;
  }
}

export default page;
```

REPORTS

Best Practices:

Focused on creating reusable components to keep the code clean and easy to maintain.
Organized the project with a modular folder structure to make it easier to scale in the future.
Implemented lazy loading to improve performance and speed up the initial page load time.
Added toast notifications for the shopping cart to enhance the user experience with helpful alerts.

My Conclusion:

By following these best practices, the project ended up with a clean and efficient front-end setup. Reusable components and a well-organized folder structure made it easier to maintain and scale the project. Performance improvements, like lazy loading and API caching, helped make the site faster and more user-friendly. Overall, these strategies not only made development smoother but also set the stage for future improvements.