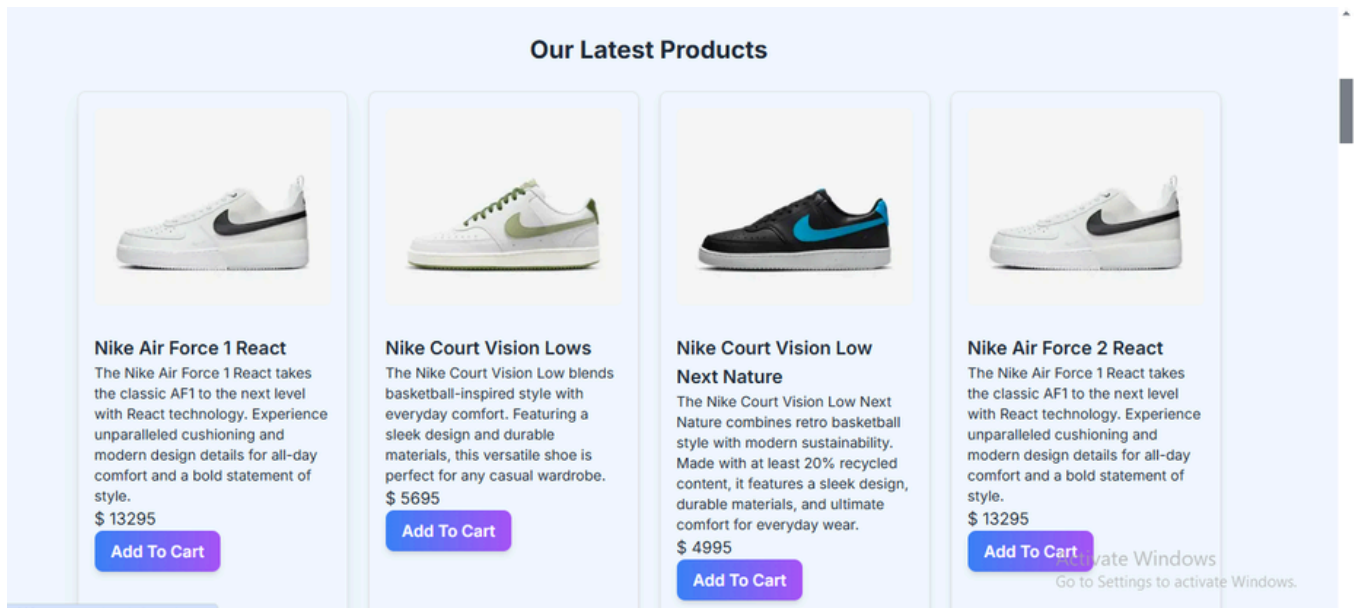


Day 4 IMPLEMENTATION REPORT

1. "This project is a product display page that fetches dynamic data through the Sanity CMS API."
2. "Details like product name, image, price, and description are rendered on the frontend using the API."



API FETCH CODE

```

1 import { client } from "@sanity/lib/client";
2 import { groq } from "next-sanity";
3 import { urlFor } from "@sanity/lib/image";
4 import Image from "next/image";
5 import { product } from "../../../../../type/products";
6
7 interface ProductPageProps {
8   params: Promise<{ slug: string }>;
9 }
10 async function getProduct(slug: string): Promise<product> {
11   return client.fetch(
12     groq`*_type == "product" && slug.current == $slug[0]{
13       _id,
14       productName,
15       _type,
16       image,
17       price,
18       decription,
19     }`,
20     {slug}
21   );
22 }
23
24 export default async function ProductPage({params}: ProductPageProps) {
25   const {slug} = await params;
26   const product = await getProduct(slug);
27   return (
28     <div className="max-w-7xl mx-auto px-4">
29       <div className="grid grid-cols-1 md:grid-cols-2 gap-2">
30         <div className="aspect-square">
31           {product.image && (
32             <Image
33               src={urlFor(product.image).url()}
34               alt={product.productName}
35               width={500}
36               height={500}
37               className="rounded-lg shadow-md"
38             />
39           )}
40         </div>
41         <div className="flex flex-col gap-8 mt-10">
42           <h1 className="text-4xl mt-10 font-bold">{product.productName}</h1>
43           <p className="text-2xl font-sans">${product.price}
44           {product.description}</p>
45         </div>
46       </div>
47     </div>
48   );
49 }
50

```

QUERY CODE

File Edit Selection View Go Run ...

tep-5

EXPLORER

OPEN EDITORS

TEP-5

my-app

src

components

lib

sanity

lib

TS client.ts U

TS image.ts U

TS live.ts U

TS queries.ts U

schemaTypes

page.tsx ...\Shopcard U

page.tsx ...\Shop U

navbar.tsx M

TS product.ts U

my-app > src > sanity > lib > TS queries.ts > four

1 import { groq } from "next-sanity"

2

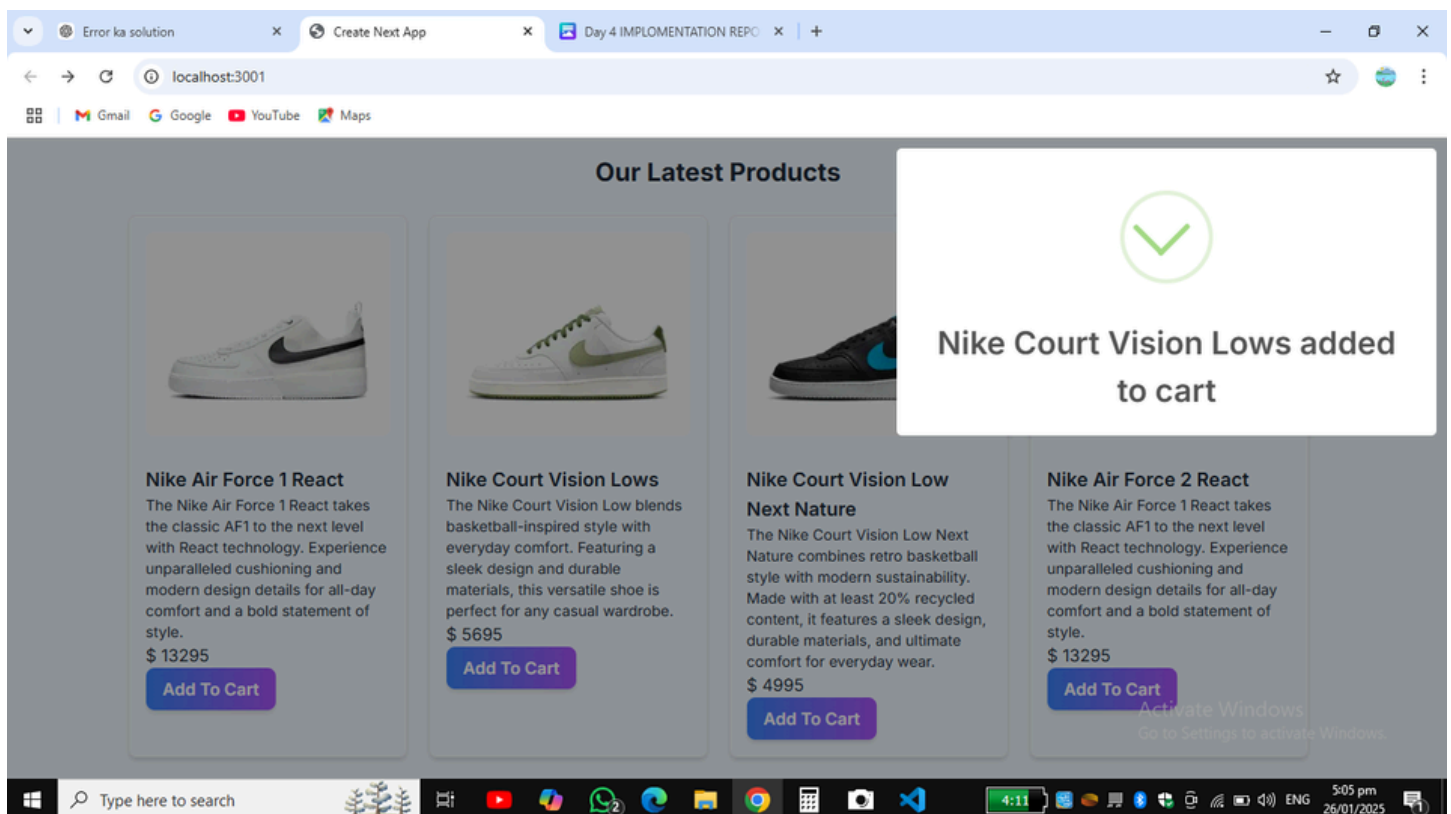
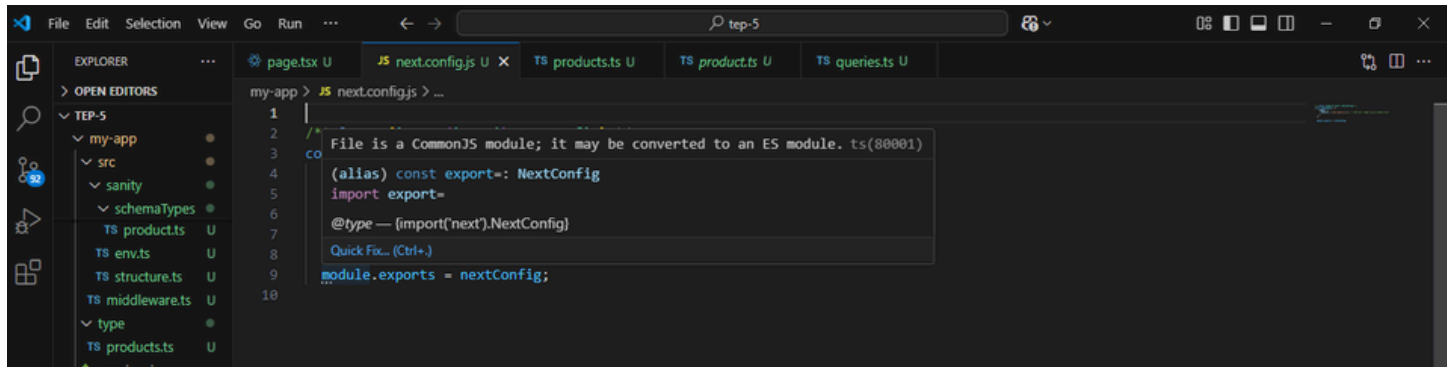
3 export const allProducts = groq`*_type == "product"`;

4 export const four =groq `*_type == "product"[0..3]`;

FACE ERRORS

1. "This code handles fetching environment variables securely for the Sanity CMS API integration."
2. "It ensures required variables like *APIVersion*, *Dataset*, and *Project ID* are validated and available."

ERROR



ALER ADD TO CART

Our Latest Products



Nike Air Force 1 React

The Nike Air Force 1 React takes the classic AF1 to the next level with React technology. Experience unparalleled cushioning and modern design details for all-day comfort and a bold statement of style.

\$ 13295

Add To Cart



Nike Court Vision Lows

The Nike Court Vision Low blends basketball-inspired style with everyday comfort. Featuring a sleek design and durable materials, this versatile shoe is perfect for any casual wardrobe.

\$ 5695

Add To Cart



Nike Court Vision Low Next Nature

The Nike Court Vision Low Next Nature combines retro basketball style with modern sustainability. Made with at least 20% recycled content, it features a sleek design, durable materials, and ultimate comfort for everyday wear.

\$ 4995

Add To Cart

Nike Air Force 2 React

The Nike Air Force 1 React takes the classic AF1 to the next level with React technology. Experience unparalleled cushioning and modern design details for all-day comfort and a bold statement of style.

\$ 13295

Add To Cart

Activate Windows
Go to Settings to activate Windows.



Nike Court Vision Lows added
to cart