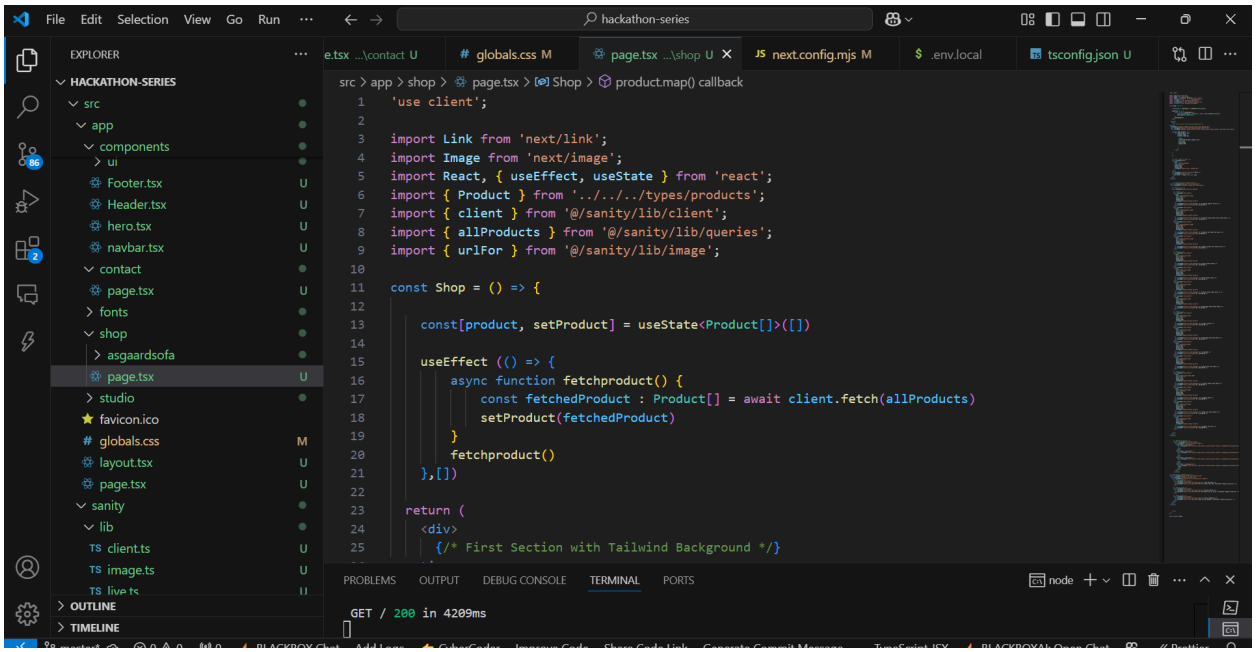
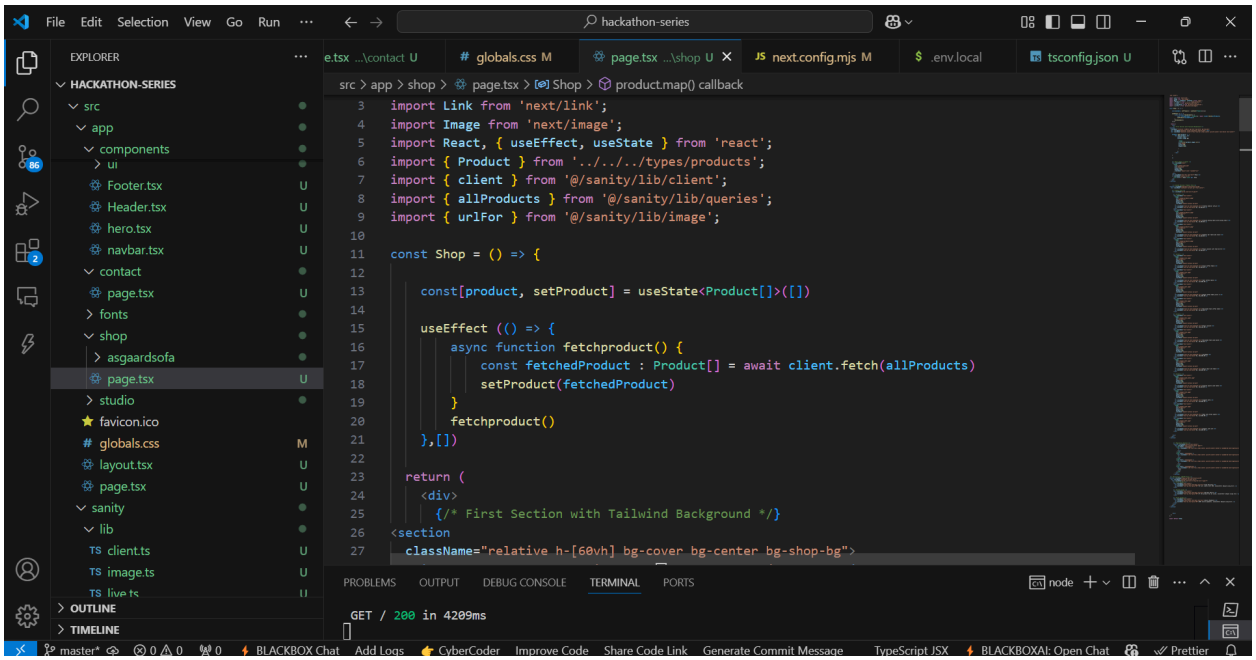
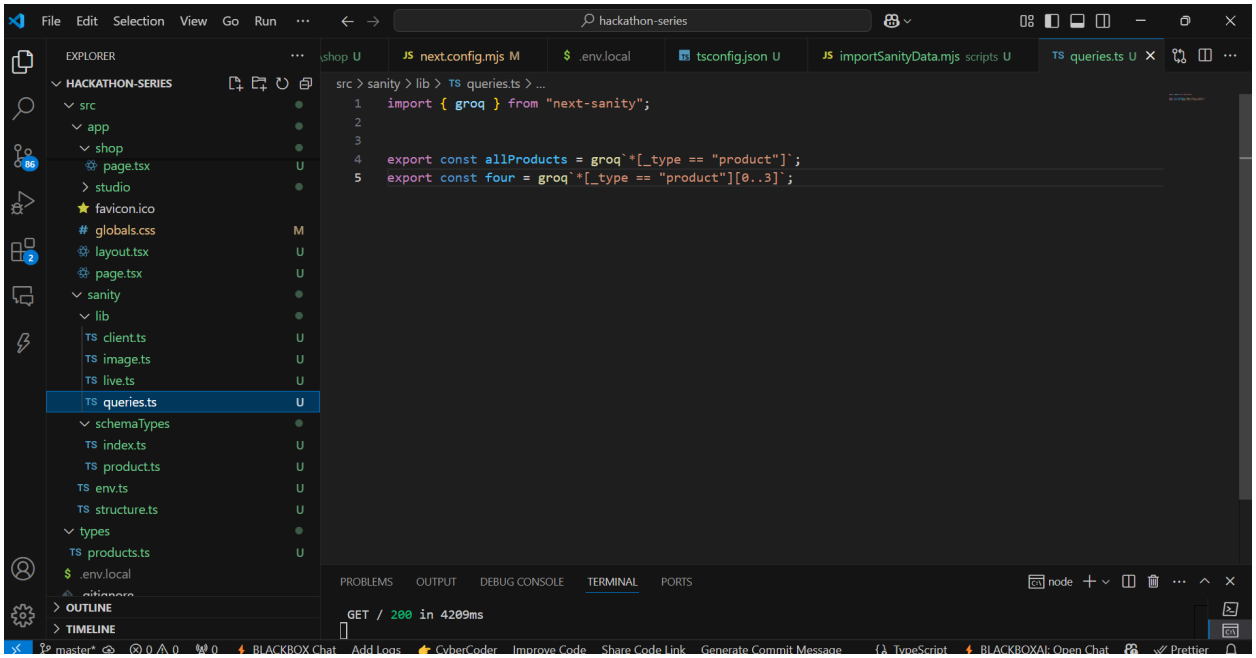
DAY 4

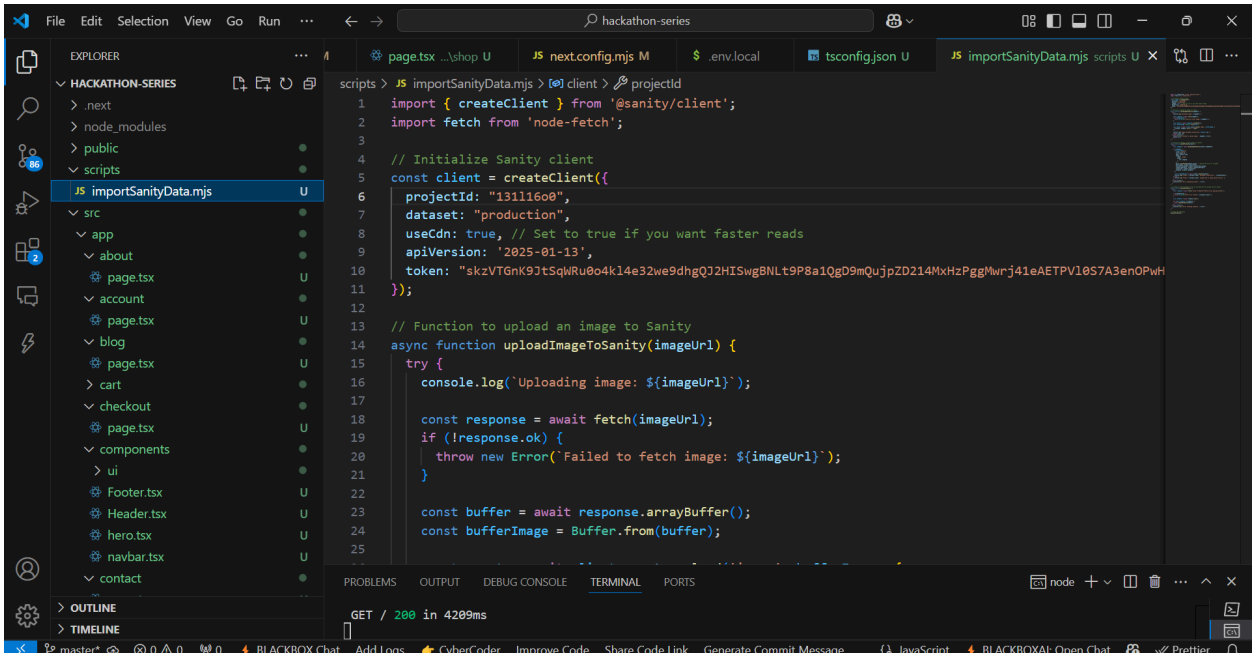

File Edit Selection View Go Run
hackathon-series
EXPLORER
HACKATHON-SERIES
src
app
components
ui
Footer.tsx
Header.tsx
hero.tsx
navbar.tsx
contact
page.tsx
fonts
shop
asgaardsofa
page.tsx
studio
favicon.ico
globals.css
layout.tsx
page.tsx
sanity
lib
client.ts
image.ts
OUTLINE
TIMELINE
globals.css M
page.tsx ...\shop U
next.config.mjs M
.env.local
tsconfig.json U
src > app > shop > page.tsx > Shop > product.map() callback
'use client';
import Link from 'next/link';
import Image from 'next/image';
import React, { useEffect, useState } from 'react';
import { Product } from '../../../types/products';
import { client } from '@/sanity/lib/client';
import { allProducts } from '@/sanity/lib/queries';
import { urlFor } from '@/sanity/lib/image';
const Shop = () => {
const[product, setProduct] = useState<Product[]>([])
useEffect (() => {
async function fetchproduct() {
const fetchedProduct : Product[] = await client.fetch(allProducts)
setProduct(fetchedProduct)
}
fetchproduct()
},[])
return (

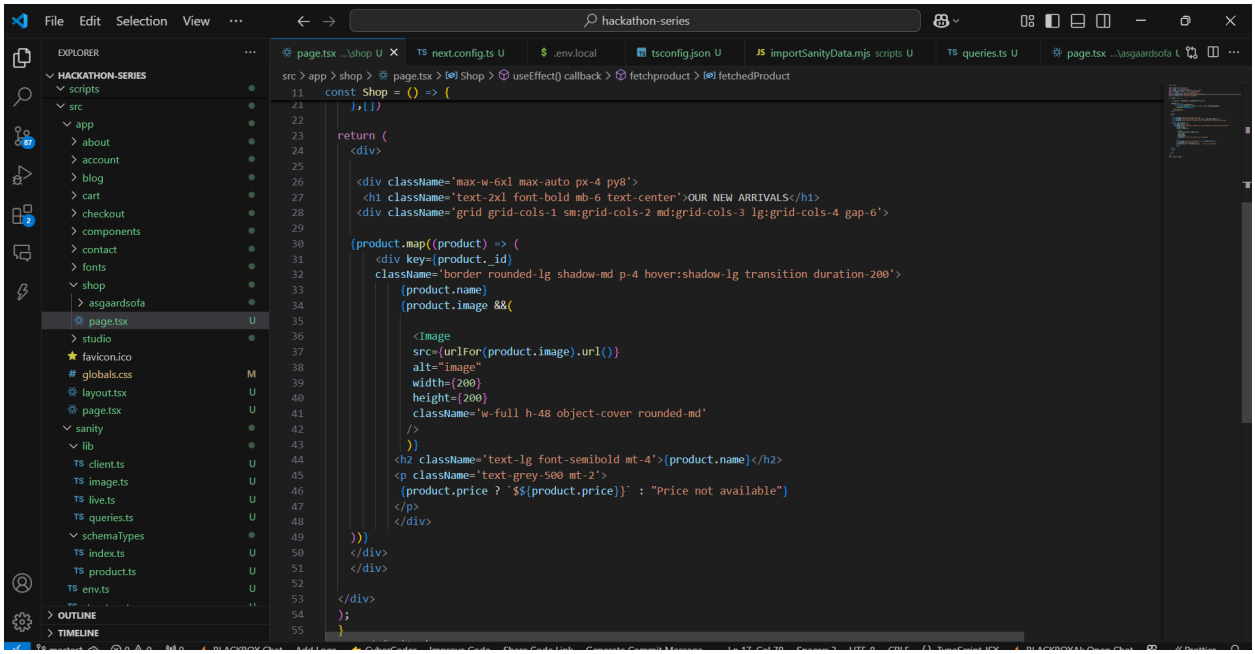
{/* First Section with Tailwind Background */}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node
GET / 200 in 4209ms


File Edit Selection View Go Run
hackathon-series
EXPLORER
HACKATHON-SERIES
src
app
components
ui
Footer.tsx
Header.tsx
hero.tsx
navbar.tsx
contact
page.tsx
fonts
shop
asgaardsofa
page.tsx
studio
favicon.ico
globals.css
layout.tsx
page.tsx
sanity
lib
client.ts
image.ts
OUTLINE
TIMELINE
globals.css M
page.tsx ...\shop U
next.config.mjs M
.env.local
tsconfig.json U
src > app > shop > page.tsx > Shop > product.map() callback
import Link from 'next/link';
import Image from 'next/image';
import React, { useEffect, useState } from 'react';
import { Product } from '../../../types/products';
import { client } from '@/sanity/lib/client';
import { allProducts } from '@/sanity/lib/queries';
import { urlFor } from '@/sanity/lib/image';
const Shop = () => {
const[product, setProduct] = useState<Product[]>([])
useEffect (() => {
async function fetchproduct() {
const fetchedProduct : Product[] = await client.fetch(allProducts)
setProduct(fetchedProduct)
}
fetchproduct()
},[])
return (

{/* First Section with Tailwind Background */}
<section
className="relative h-[60vh] bg-cover bg-center bg-shop-bg">
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node
GET / 200 in 4209ms
master*
BLACKBOX Chat
Add Logs
CyberCoder
Improve Code
Share Code Link
Generate Commit Message
TypeScript JSX
BLACKBOXAI: Open Chat
Prettier


File Edit Selection View Go Run
hackathon-series
EXPLORER
HACKATHON-SERIES
src
app
shop
page.tsx
studio
favicon.ico
globals.css
layout.tsx
page.tsx
sanity
lib
client.ts
image.ts
live.ts
queries.ts
schemaTypes
index.ts
product.ts
env.ts
structure.ts
types
products.ts
.env.local
OUTLINE
TIMELINE
next.config.mjs M
.env.local
tsconfig.json U
importSanityData.mjs scripts U
queries.ts U
src > sanity > lib > queries.ts > ...
import { groq } from "next-sanity";
export const allProducts = groq`*[_type == "product"]`;
export const four = groq`*[_type == "product"][0..3]`;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node
GET / 200 in 4209ms


File Edit Selection View Go Run
hackathon-series
EXPLORER
HACKATHON-SERIES
.next
node_modules
public
scripts
importSanityData.mjs
src
app
about
page.tsx
account
page.tsx
blog
page.tsx
cart
checkout
page.tsx
components
ui
Footer.tsx
Header.tsx
hero.tsx
navbar.tsx
contact
OUTLINE
TIMELINE
page.tsx ...\shop U
next.config.mjs M
.env.local
tsconfig.json U
importSanityData.mjs scripts U
scripts > importSanityData.mjs > client > projectId
import { createClient } from '@sanity/client';
import fetch from 'node-fetch';
// Initialize Sanity client
const client = createClient({
projectId: "131116o0",
dataset: "production",
useCdn: true, // Set to true if you want faster reads
apiVersion: '2025-01-13',
token: "skzVTGnK9JtSqWRu0o4kl4e32we9dhgQJ2HISwgBNLt9P8a1QgD9mQujpZD214MxHzPggMwrj41eAETPVl0S7A3enOPwH
});
// Function to upload an image to Sanity
async function uploadImageToSanity(imageUrl) {
try {
console.log(`Uploading image: ${imageUrl}`);
const response = await fetch(imageUrl);
if (!response.ok) {
throw new Error(`Failed to fetch image: ${imageUrl}`);
}
const buffer = await response.arrayBuffer();
const bufferImage = Buffer.from(buffer);
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node
GET / 200 in 4209ms


File Edit Selection View
hackathon-series
EXPLORER
HACKATHON-SERIES
scripts
src
app
about
account
blog
cart
checkout
components
contact
fonts
shop
asgaardsofa
page.tsx
studio
favicon.ico
globals.css
layout.tsx
page.tsx
sanity
lib
client.ts
image.ts
live.ts
queries.ts
schemaTypes
index.ts
product.ts
env.ts
OUTLINE
TIMELINE
page.tsx ...\shop U
next.config.ts U
.env.local
tsconfig.json U
importSanityData.mjs scripts U
queries.ts U
page.tsx ...\asgaardsofa
src > app > shop > page.tsx > Shop > useEffect() callback > fetchproduct > fetchedProduct
const Shop = () => {
},[])
return (

<h1 className='text-2xl font-bold mb-6 text-center'>OUR NEW ARRIVALS</h1>

{product.map((product) => (
<div key={product._id}
className='border rounded-lg shadow-md p-4 hover:shadow-lg transition duration-200'>
{product.name}
{product.image &&(
<Image
src={urlFor(product.image).url()}
alt="image"
width={200}
height={200}
className='w-full h-48 object-cover rounded-md'
/>
)}
<h2 className='text-lg font-semibold mt-4'>{product.name}</h2>

{product.price ? `$${product.price}` : "Price not available"}

))}

);
}