Day 3 - API Integration Report - E-Commerce Marketplace

API Integration Process

1. Overview

On Day 3, we focused on integrating external APIs into our marketplace platform. This included fetching data from [API Name] and integrating it with our Sanity CMS. The integration ensures seamless synchronization between the backend data and the frontend display.

2. Steps Followed

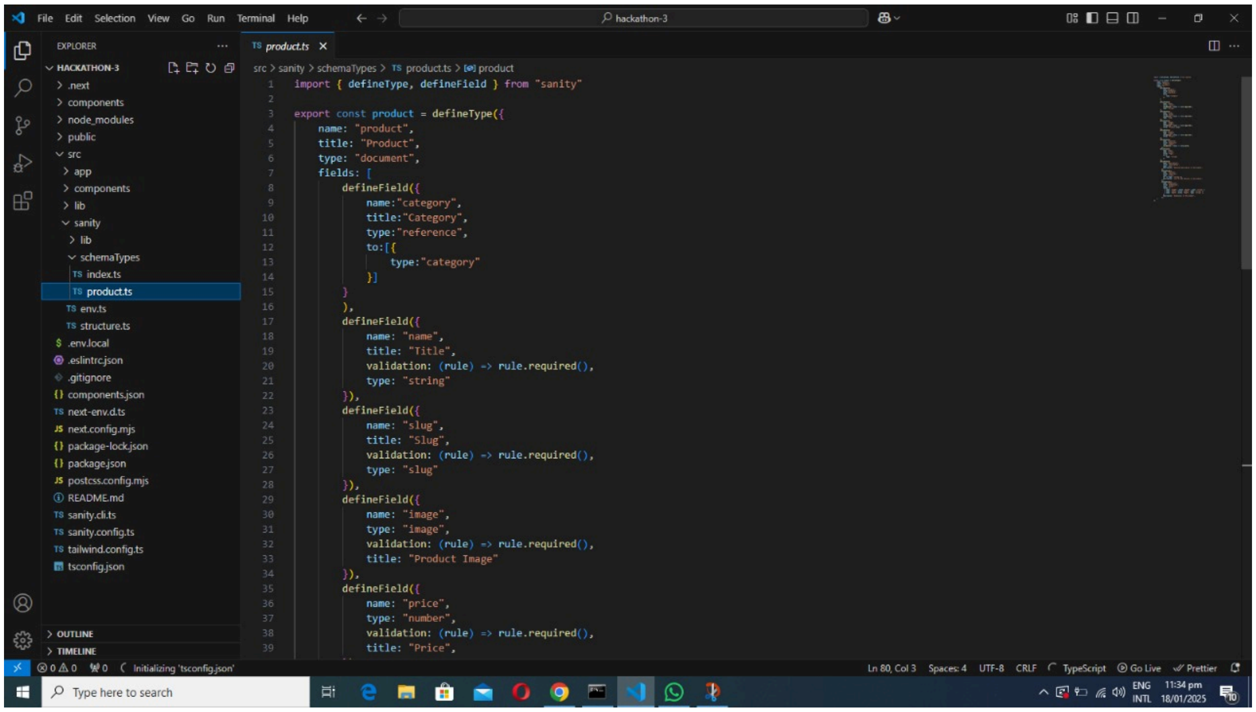
1. **Understanding API Documentation:** Reviewed API endpoints, authentication mechanisms, and rate limits.
2. **Setting Up API Calls:** Implemented API requests using Next.js `fetch` and Axios.
3. **Parsing and Storing Data:** Mapped API response data to Sanity schemas.
4. **Rendering Data on the Frontend:** Used React components to display the fetched data dynamically.
5. **Error Handling & Optimization:** Implemented error handling and caching strategies.

Adjustments Made to Schemas

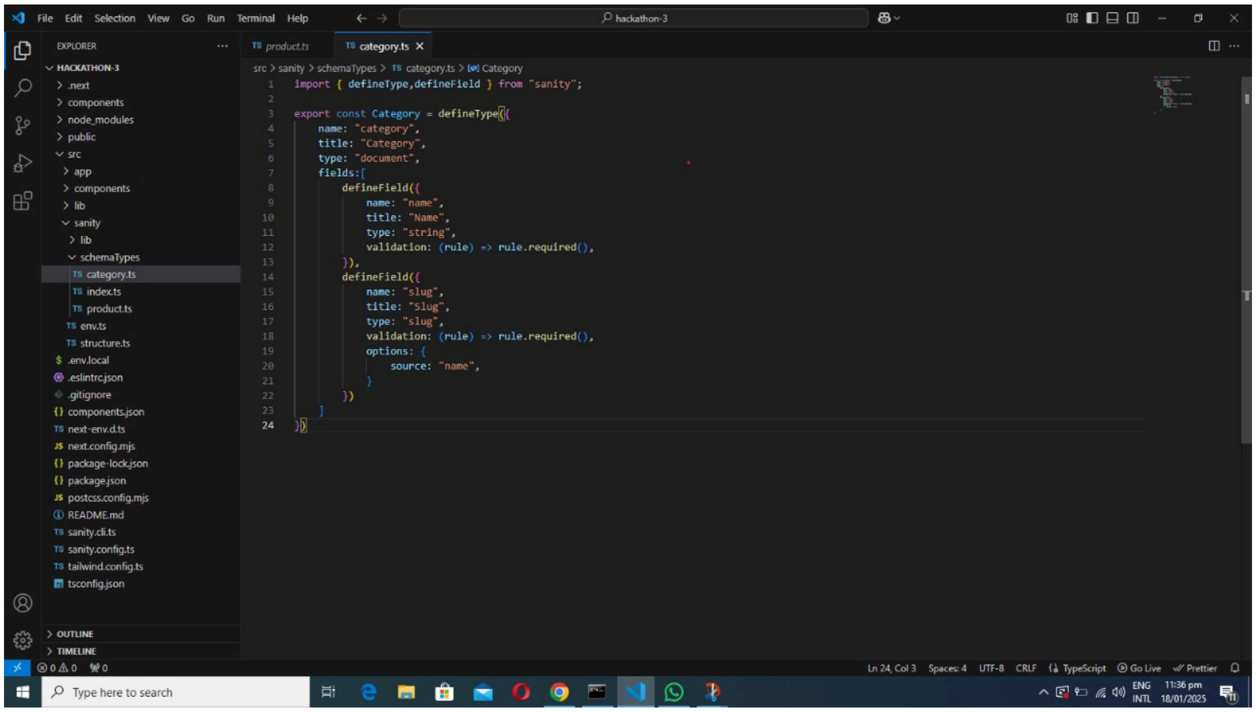
1. Sanity Schema Updates

- **Added New Fields:** Introduced additional fields to accommodate API response data.
- **Updated Existing Fields:** Modified field structures to align with the API data format.
- **Data Validation:** Ensured schema validations to maintain data integrity.

Schema Adjustment:



```
import { defineType, defineField } from "sanity"

export const product = defineType({
    name: "product",
    title: "Product",
    type: "document",
    fields: [
        defineField({
            name:"category",
            title:"Category",
            type:"reference",
            to:[{
                type:"category"
            }]
        }
        ),
        defineField({
            name: "name",
            title: "Title",
            validation: (rule) => rule.required(),
            type: "string"
        }),
        defineField({
            name: "slug",
            title: "Slug",
            validation: (rule) => rule.required(),
            type: "slug"
        }),
        defineField({
            name: "image",
            type: "image",
            validation: (rule) => rule.required(),
            title: "Product Image"
        }),
        defineField({
            name: "price",
            type: "number",
            validation: (rule) => rule.required(),
            title: "Price",
```



```
import { defineType,defineField } from "sanity";

export const Category = defineType({
    name: "category",
    title: "Category",
    type: "document",
    fields:[
        defineField({
            name: "name",
            title: "Name",
            type: "string",
            validation: (rule) => rule.required(),
        }),
        defineField({
            name: "slug",
            title: "Slug",
            type: "slug",
            validation: (rule) => rule.required(),
            options: {
                source: "name",
            }
        })
    ]
})
```

Migration Steps and Tools Used

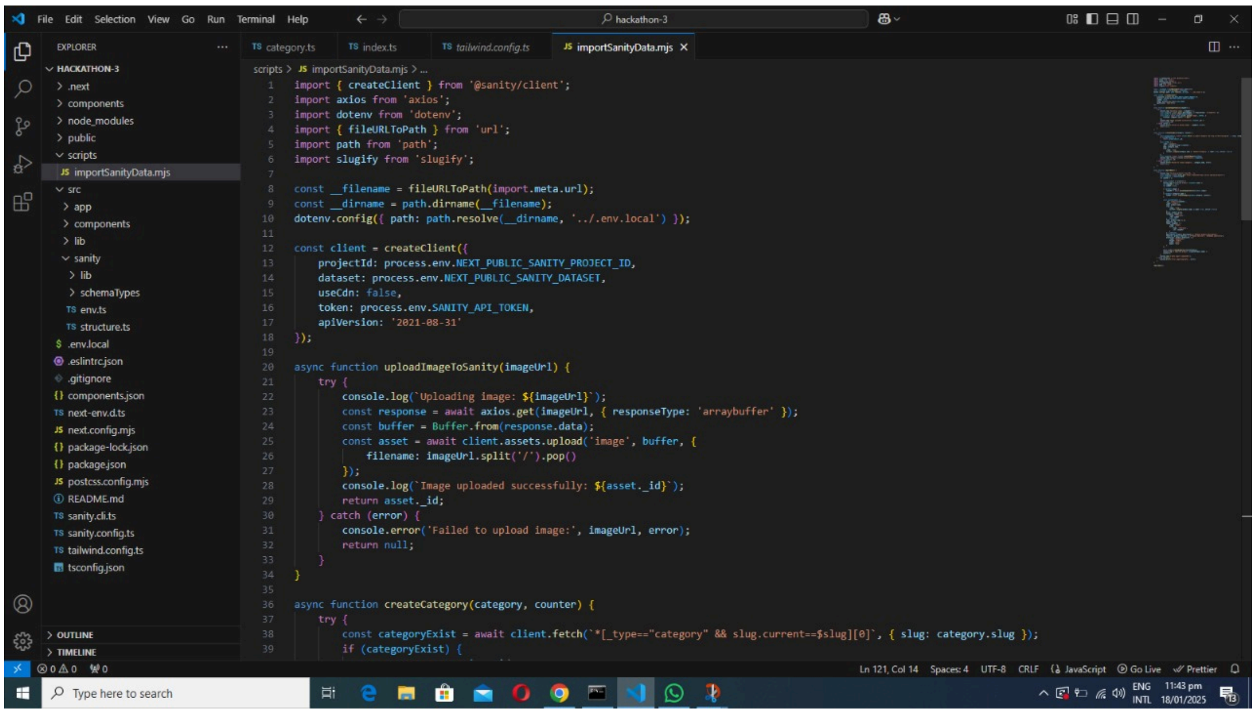
1. Data Migration Strategy

- **Exported Existing Data:** Used Sanity's CLI to backup existing content.
- **Scripted Data Migration:** Created migration scripts to transform API data into Sanity-compatible format.
- **Re-imported Data:** Used Sanity's import tool to populate the database.

2. Tools Utilized

- **Sanity CLI (`sanity dataset export/import`)** for data migration.
- **Node.js Script** for transforming data.
- **Postman** for testing API endpoints.

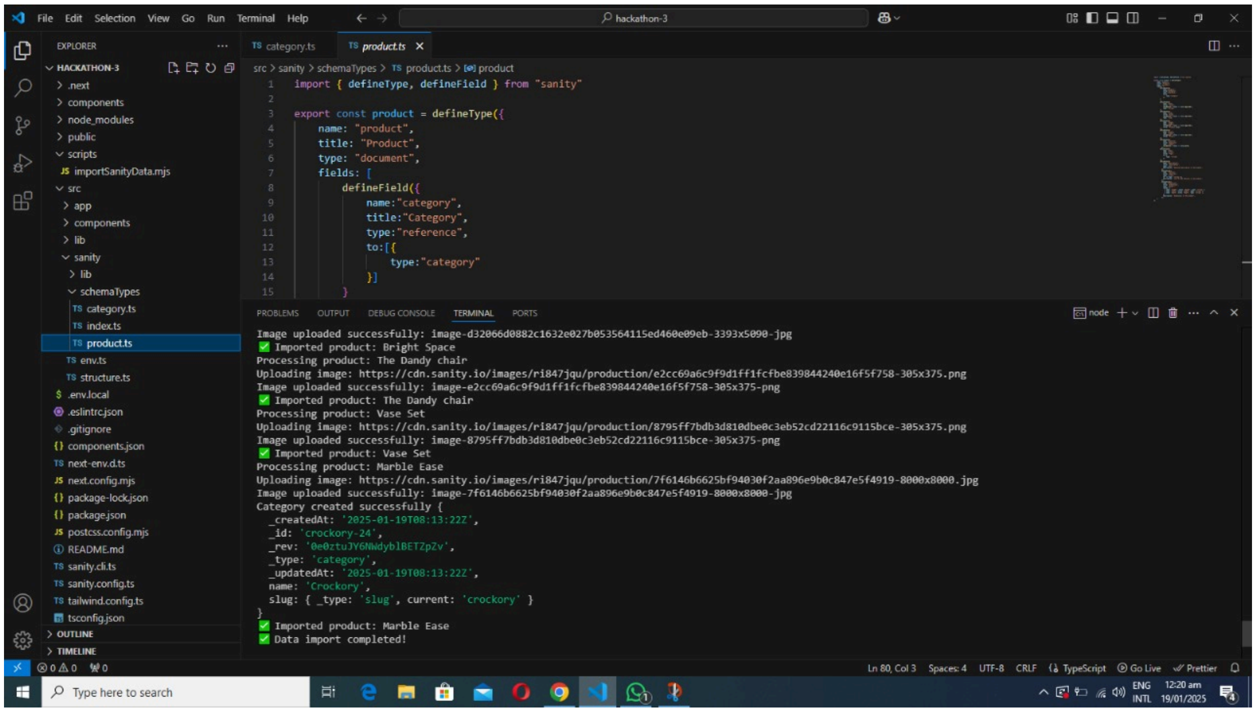
Migration Script:



```
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';
import slugify from 'slugify';

const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

const client = createClient({
    projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
    dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
    useCdn: false,
    token: process.env.SANITY_API_TOKEN,
    apiVersion: '2021-08-31'
});

async function uploadImageToSanity(imageUrl) {
    try {
        console.log(`Uploading image: ${imageUrl}`);
        const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
        const buffer = Buffer.from(response.data);
        const asset = await client.assets.upload('image', buffer, {
            filename: imageUrl.split('/').pop()
        });
        console.log(`Image uploaded successfully: ${asset._id}`);
        return asset._id;
    } catch (error) {
        console.error('Failed to upload image:', imageUrl, error);
        return null;
    }
}

async function createCategory(category, counter) {
    try {
        const categoryExist = await client.fetch(`*[_type=="category" && slug.current==$slug][0]`, { slug: category.slug });
        if (categoryExist) {
```

Screenshots

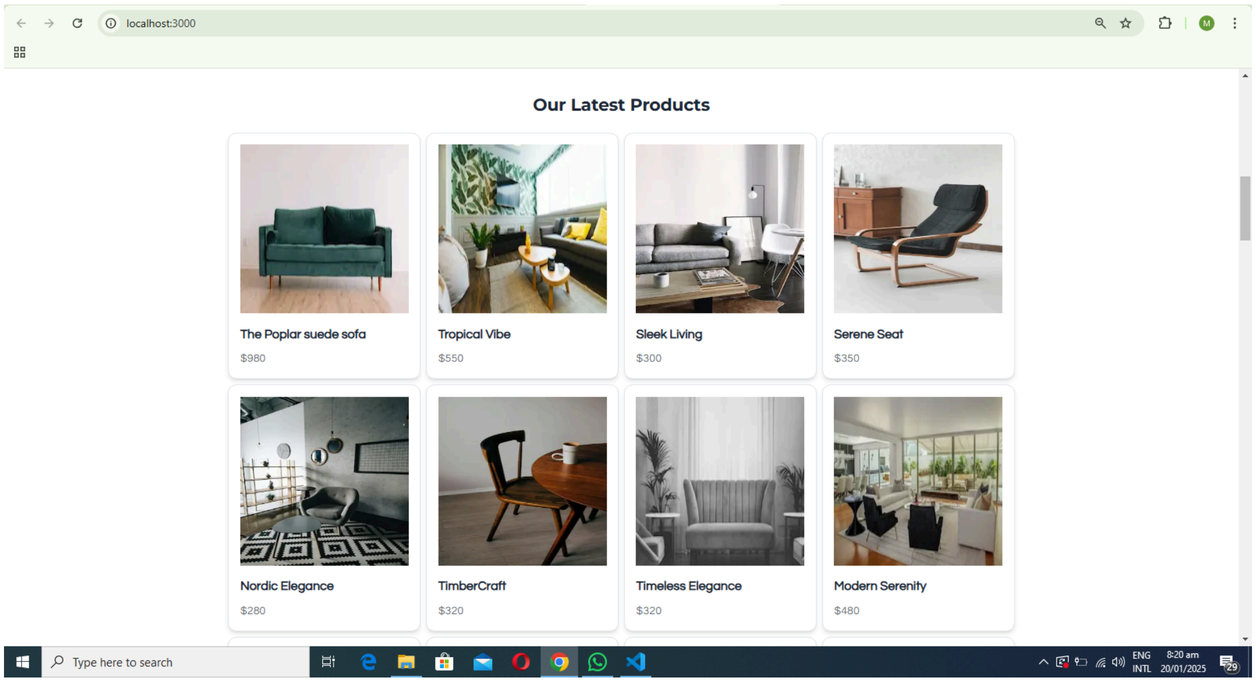
1. API Calls



```
import { defineType, defineField } from "sanity"

export const product = defineType({
    name: "product",
    title: "Product",
    type: "document",
    fields: [
        defineField({
            name:"category",
            title:"Category",
            type:"reference",
            to:[{
                type:"category"
            }]
        }
```

```
Image uploaded successfully: image-d32066d0882c1632e027b053564115ed460e09eb-3393x5090-jpg
✅ Imported product: Bright Space
Processing product: The Dandy chair
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/e2cc69a6c9f9d1ff1fcfbe839844240e16f5f758-305x375.png
Image uploaded successfully: image-e2cc69a6c9f9d1ff1fcfbe839844240e16f5f758-305x375-png
✅ Imported product: The Dandy chair
Processing product: Vase Set
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/8795ff7bdb3d810dbe0c3eb52cd22116c9115bce-305x375.png
Image uploaded successfully: image-8795ff7bdb3d810dbe0c3eb52cd22116c9115bce-305x375-png
✅ Imported product: Vase Set
Processing product: Marble Ease
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/7f6146b6625bf94030f2aa896e9b0c847e5f4919-8000x8000.jpg
Image uploaded successfully: image-7f6146b6625bf94030f2aa896e9b0c847e5f4919-8000x8000-jpg
Category created successfully {
  _createdAt: '2025-01-19T08:13:22Z',
  _id: 'crockory-24',
  _rev: '0e0ztuJY6NWdyblBETZpZv',
  _type: 'category',
  _updatedAt: '2025-01-19T08:13:22Z',
  name: 'Crockory',
  slug: { _type: 'slug', current: 'crockory' }
}
✅ Imported product: Marble Ease
✅ Data import completed!
```

2. Data Display on Frontend



Our Latest Products

The Poplar suede sofa
$980

Tropical Vibe
$550

Sleek Living
$300

Serene Seat
$350

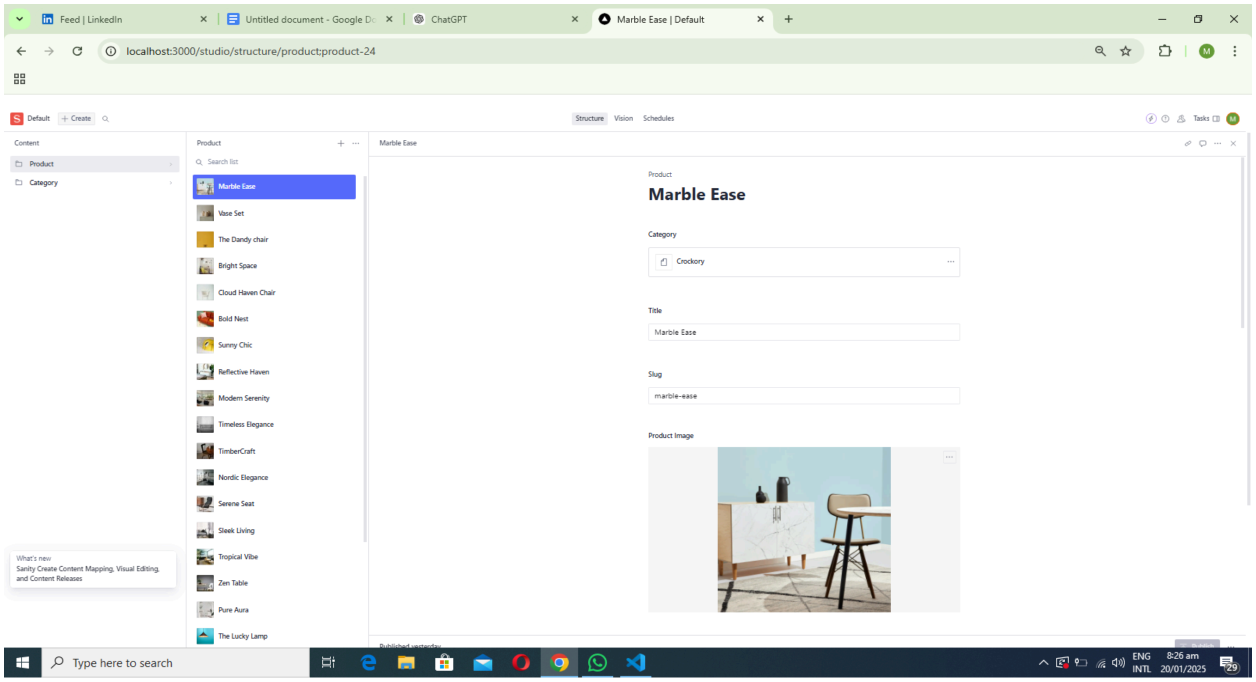
Nordic Elegance
$280

TimberCraft
$320

Timeless Elegance
$320

Modern Serenity
$480

3. Sanity CMS Populated Fields



Product

Marble Ease
Vase Set
The Dandy chair
Bright Space
Cloud Haven Chair
Bold Nest
Sunny Chic
Reflective Haven
Modern Serenity
Timeless Elegance
TimberCraft
Nordic Elegance
Serene Seat
Sleek Living
Tropical Vibe
Zen Table
Pure Aura
The Lucky Lamp

Product

Marble Ease

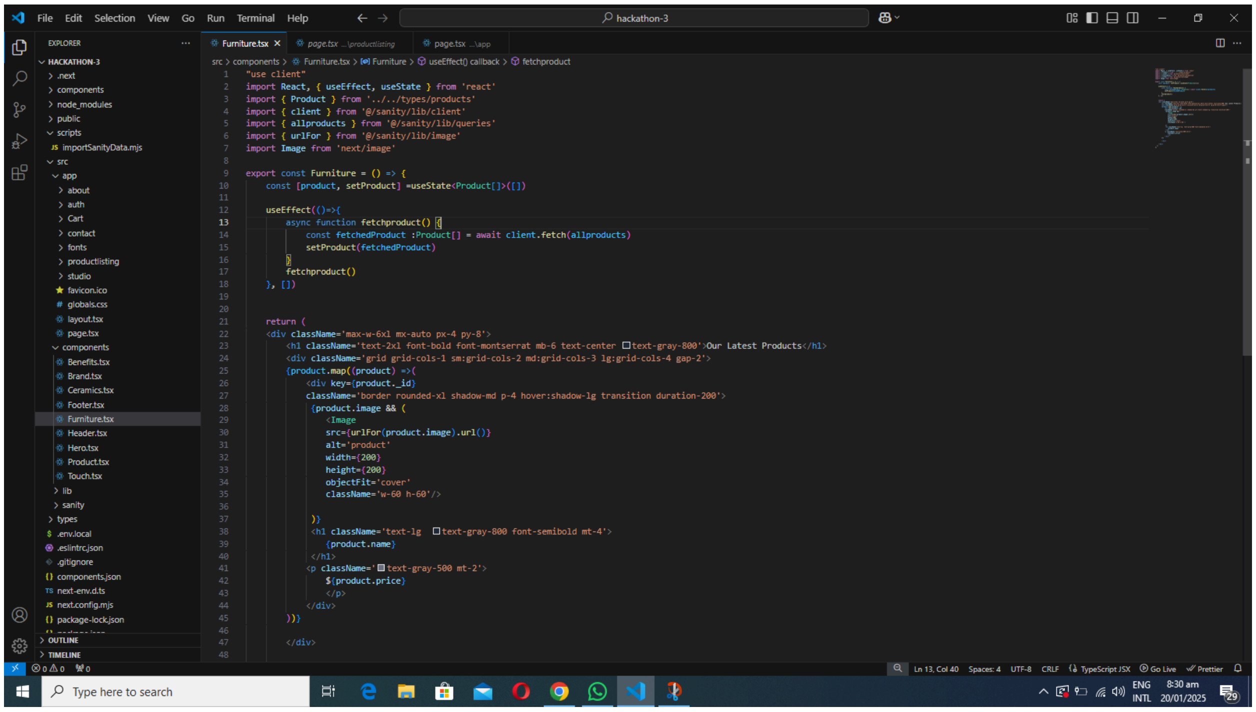
Category: Crockery

Title: Marble Ease

Slug: marble-ease

Product Image

Code Snippets for API Integration (Fetching Data in Next.js)



src > components > Furniture.tsx > Furniture > useEffect() callback > fetchproduct

```tsx
"use client"
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'
import Image from 'next/image'

export const Furniture = () => {
    const [product, setProduct] =useState<Product[]>([])

    useEffect(()=>{
        async function fetchproduct() {
            const fetchedProduct :Product[] = await client.fetch(allproducts)
            setProduct(fetchedProduct)
        }
        fetchproduct()
    }, [])


    return (
    <div className='max-w-6xl mx-auto px-4 py-8'>
        <h1 className='text-2xl font-bold font-montserrat mb-6 text-center text-gray-800'>Our Latest Products</h1>
        <div className='grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-2'>
        {product.map((product) =>(
            <div key={product._id}
            className='border rounded-xl shadow-md p-4 hover:shadow-lg transition duration-200'>
             {product.image && (
                <Image
                src={urlFor(product.image).url()}
                alt='product'
                width={200}
                height={200}
                objectFit='cover'
                className='w-60 h-60'/>

             )}
             <h1 className='text-lg  text-gray-800 font-semibold mt-4'>
                {product.name}
             </h1>
            <p className='text-gray-500 mt-2'>
                ${product.price}
                </p>
            </div>
        ))}

        </div>

```

Conclusion

Day 3 focused on integrating APIs, refining schemas, and migrating data. The successful completion of this phase ensures that our e-commerce platform seamlessly synchronizes external data with Sanity CMS while displaying it on the frontend dynamically. Next steps include refining UI components and optimizing API performance.

[AYESHA EJAZ] Q2 Marketplace Builder Hackathon 2025