REPORT
HACKATHON Day 3 -MARKETPLACE BUILDER
E-commerce (Bandage)

Name: Abdul Wasay
Slot: Friday 9am to 12pm
Batch: 1 Q2
Roll: 00071305

Overview:

On the third day of the hackathon, I focused on integrating APIs and migrating data into Sanity CMS to build the backend for a functional marketplace. Key tasks completed during this phase included:

- Connecting APIs to my next.js project.
- Moved data APIs into Sanity.
- Use provided APIs of Template 5.

1: <u>API INTERGRATION:</u>

The integration of APIs was essential for fetching product data from an external source (in this case, the provided APIs) and populating it into our Sanity CMS for the marketplace. Below is the step-by-step process I followed for API integration in the Next.js project:

- **API Choosing:** I chose the API from Template 5 provided in the documentation. I used the provided API**: https://template6-six.vercel.app/api/products** was used to fetch product data. This endpoint provided essential details, including product titles, descriptions, prices, and category IDs.

- **API Documentation Review:** I reviewed the API documentation thoroughly to understand the endpoints. The documentation helped identify the necessary fields (e.g., title, description, price discount Percentage, tags, is New, product Image).

- **Schema Revisions:** I updated the existing schema to ensure compatibility with the product data fetched from the API.

2. Adjustments Made to Schemas:
In order to store the product data in Sanity CMS, I had to adjust the existing schema to ensure compatibility with the data fetched from the API.

3. MIGRATION STEPS AND TOOLS:

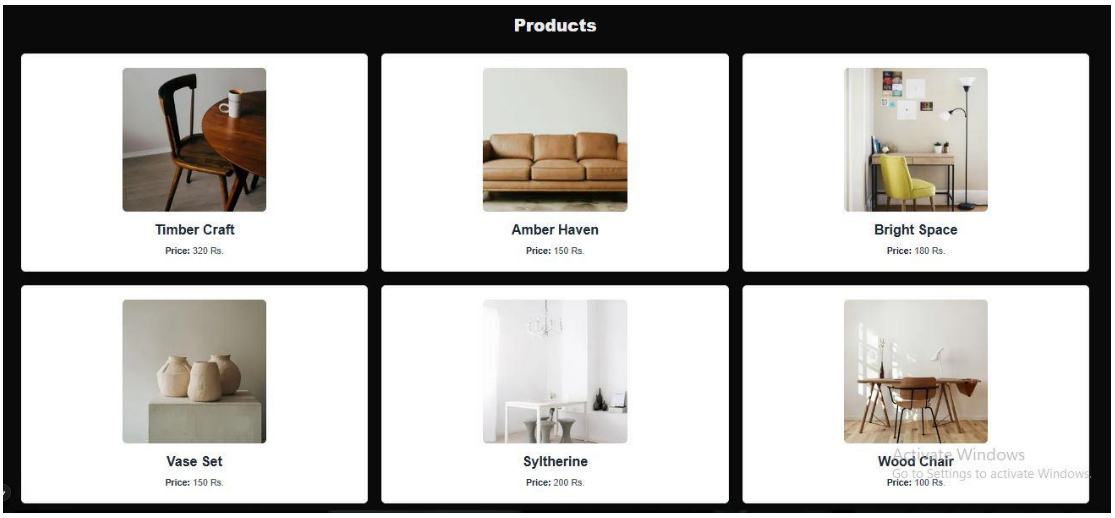
- **Data Migration:**

Migration Process: I utilized the provided migration script to transfer API data into Sanity (CMS). The script fetched product data from the API, transformed it to align with the Sanity schema, and then imported the structured data into the Sanity CMS.

- **Tools Used:** Sanity/Client.

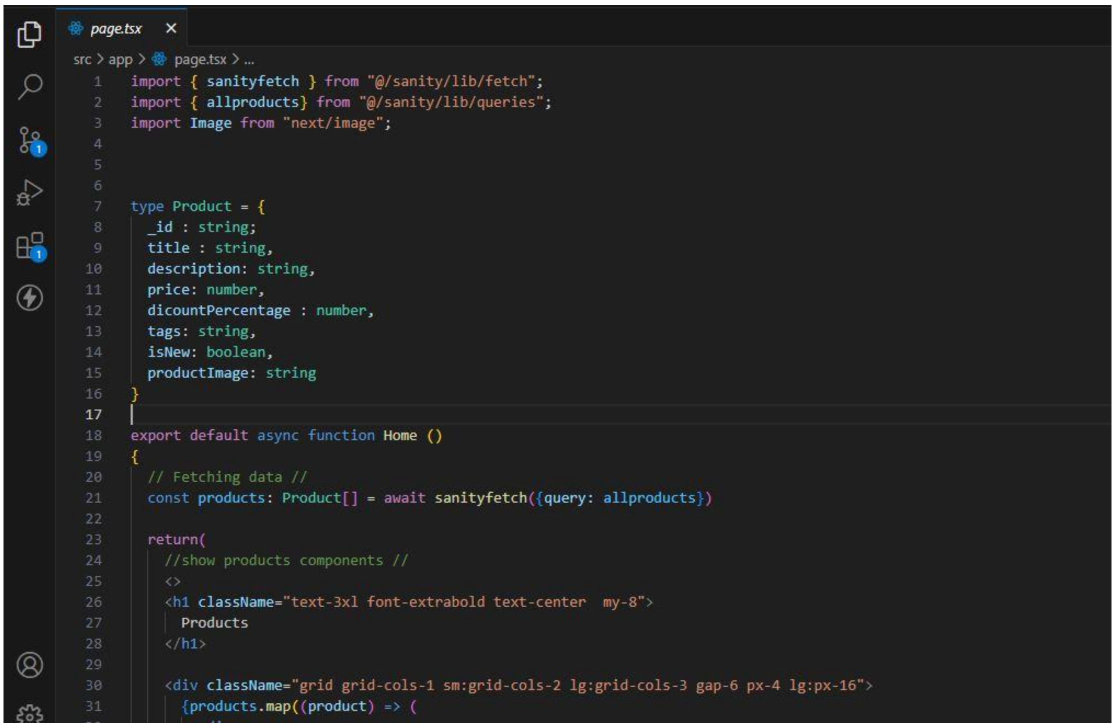
FRONTENT

After migrating the data into Sanity, I built a dynamic and responsive frontend using Next.js to showcase and display the data effectively on the web.



FETCHING PRODUCTS DATA:

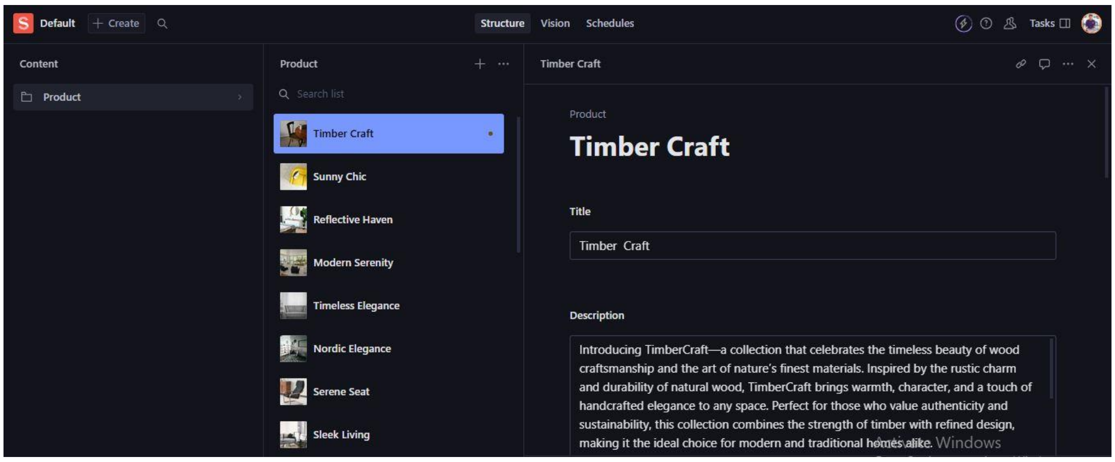
I used sanity queries to fetch data directly from sanity (CMS) to my next.js project.



```
import { sanityfetch } from "@/sanity/lib/fetch";
import { allproducts} from "@/sanity/lib/queries";
import Image from "next/image";



type Product = {
  _id : string;
  title : string,
  description: string,
  price: number,
  dicountPercentage : number,
  tags: string,
  isNew: boolean,
  productImage: string
}

export default async function Home ()
{
  // Fetching data //
  const products: Product[] = await sanityfetch({query: allproducts})

  return(
    //show products components //
    <>
    <h1 className="text-3xl font-extrabold text-center  my-8">
      Products
    </h1>

    <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6 px-4 lg:px-16">
      {products.map((product) => (
```

SANITY STUDIO INTERFACE:

After migrating the data script, the data appears in Sanity studio.



Product

Timber Craft

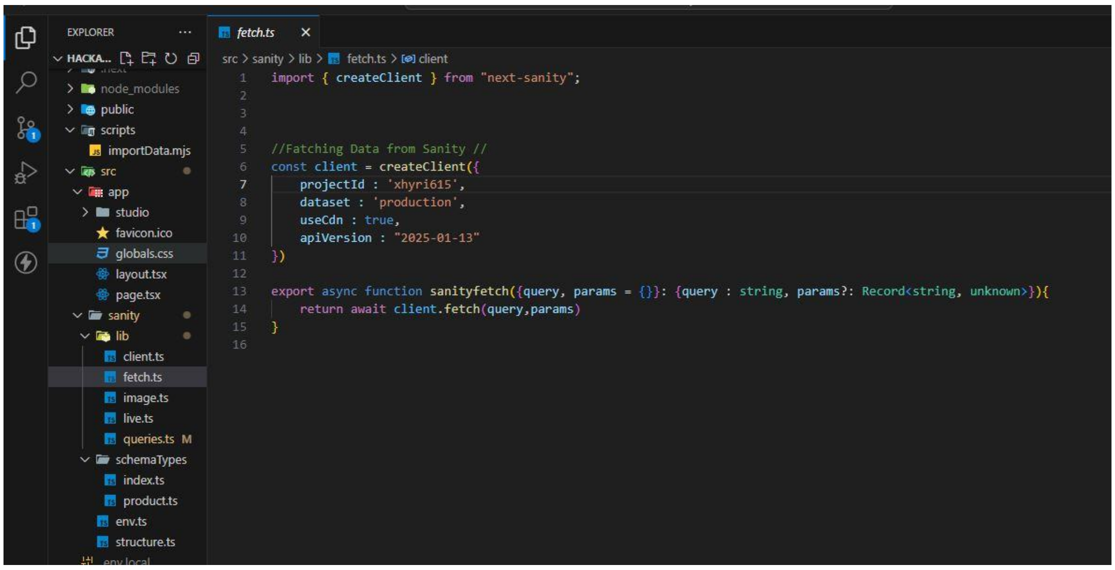
Title

Timber Craft

Description

Introducing TimberCraft—a collection that celebrates the timeless beauty of wood craftsmanship and the art of nature's finest materials. Inspired by the rustic charm and durability of natural wood, TimberCraft brings warmth, character, and a touch of handcrafted elegance to any space. Perfect for those who value authenticity and sustainability, this collection combines the strength of timber with refined design, making it the ideal choice for modern and traditional homes alike.

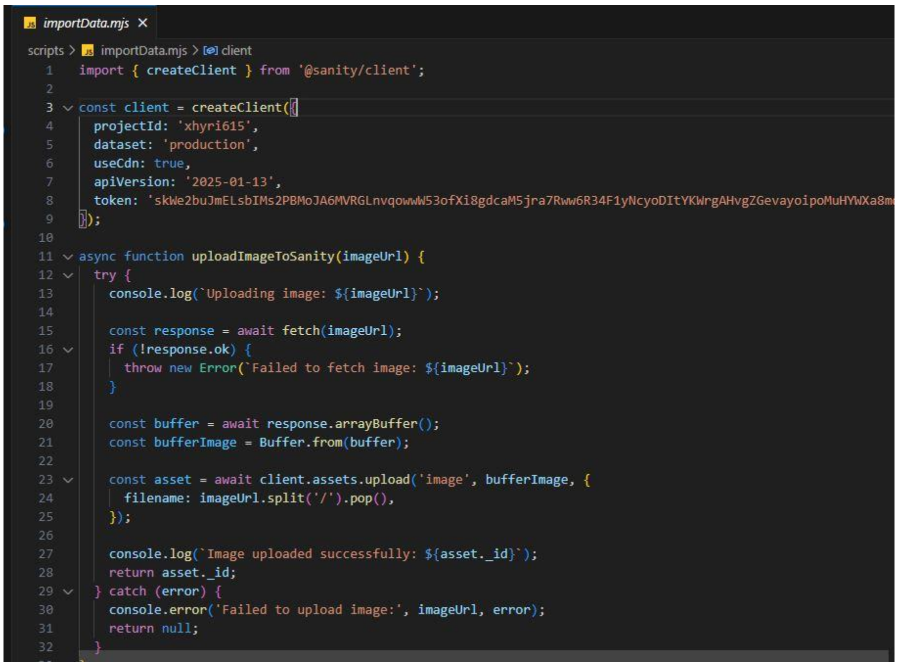
API CALLS:



```
import { createClient } from "next-sanity";



//Fatching Data from Sanity //
const client = createClient({
    projectId : 'xhyri615',
    dataset : 'production',
    useCdn : true,
    apiVersion : "2025-01-13"
})

export async function sanityfetch({query, params = {}}: {query : string, params?: Record<string, unknown>}){
    return await client.fetch(query,params)
}

```

MIGRATING SCRIPT:

I created a folder for scripts and added a file named importData.mjs. Inside this file, I pasted the provided migration code. This setup is likely used to import or migrate data as part of your project.



```
import { createClient } from '@sanity/client';

const client = createClient({
  projectId: 'xhyri615',
  dataset: 'production',
  useCdn: true,
  apiVersion: '2025-01-13',
  token: 'skWe2buJmELsbIMs2PBMoJA6MVRGLnvqowwW53ofXi8gdcaM5jra7Rww6R34F1yNcyoDItYKWrgAHvgZGevayoipoMuHYWXa8m
});

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);

    const asset = await client.assets.upload('image', bufferImage, {
      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;
  }
```

Conclusion:

This report outlines the seamless integration of an external API into a Next.js project, the customization of the Sanity CMS schema, and the successful migration of product data into the CMS. The project effectively fetched API data, displayed it dynamically on the frontend, and ensured the Sanity CMS was populated with the transformed data.