Name	Muhammad Bilal Hussain
Roll Number	**00425090**
Class Day	**Tuesday**
Timing	**2pm to 5pm**

DAY 3 – API INTEGRATION

AND

DATA MGRATION

Hackathon Day 3: Furniture API Integration and Data Migration

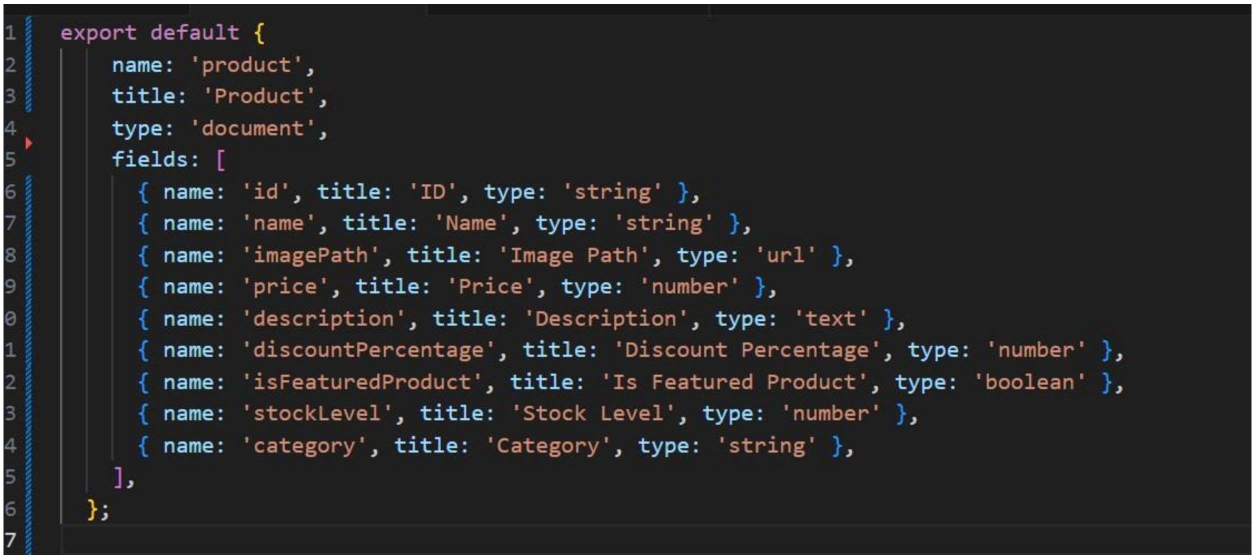
On Day 3 of the hackathon, the focus was on integrating a furniture-related API and migrating data into **Sanity CMS** to set up the backend for a fully functional furniture e-commerce website. Below is a detailed breakdown of the steps I followed to complete this task.

Step 1: Understanding the Furniture API and Data Structure

I began by analyzing the documentation of the furniture API, which provided details about products such as **sofas**, **tables**, **chairs**, and other furniture items. The key endpoints included data for product details, categories (e.g., living room, office furniture), pricing, stock levels, and images. This initial review was essential to understand the structure of the API and ensure it aligned with my e-commerce website's requirements.

Step 2: Validating and Customizing the Sanity CMS Schema

I validated my existing Sanity CMS schema to confirm compatibility with the furniture API data. During this process, I ensured that fields such as **name**, **category**, **price**, **imagePath**, and **description** matched the incoming data.

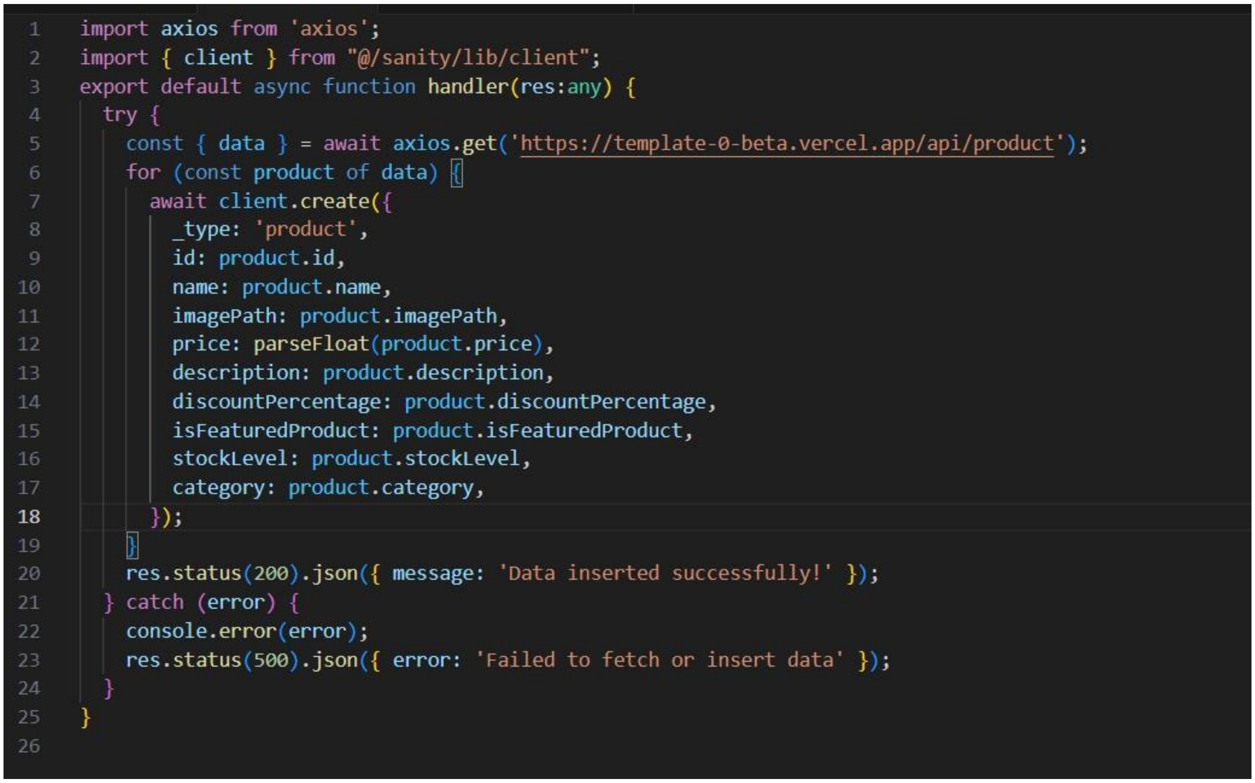


```
export default {
    name: 'product',
    title: 'Product',
    type: 'document',
    fields: [
      { name: 'id', title: 'ID', type: 'string' },
      { name: 'name', title: 'Name', type: 'string' },
      { name: 'imagePath', title: 'Image Path', type: 'url' },
      { name: 'price', title: 'Price', type: 'number' },
      { name: 'description', title: 'Description', type: 'text' },
      { name: 'discountPercentage', title: 'Discount Percentage', type: 'number' },
      { name: 'isFeaturedProduct', title: 'Is Featured Product', type: 'boolean' },
      { name: 'stockLevel', title: 'Stock Level', type: 'number' },
      { name: 'category', title: 'Category', type: 'string' },
    ],
  };
```

Step 3: Fetching Furniture Data and Inserting into Sanity

Using **Axios**, I fetched data from the furniture API and wrote a script to migrate it into the Sanity CMS. Below is the API integration and migration logic implemented in my Next.js project:

API Route for Data Insertion:



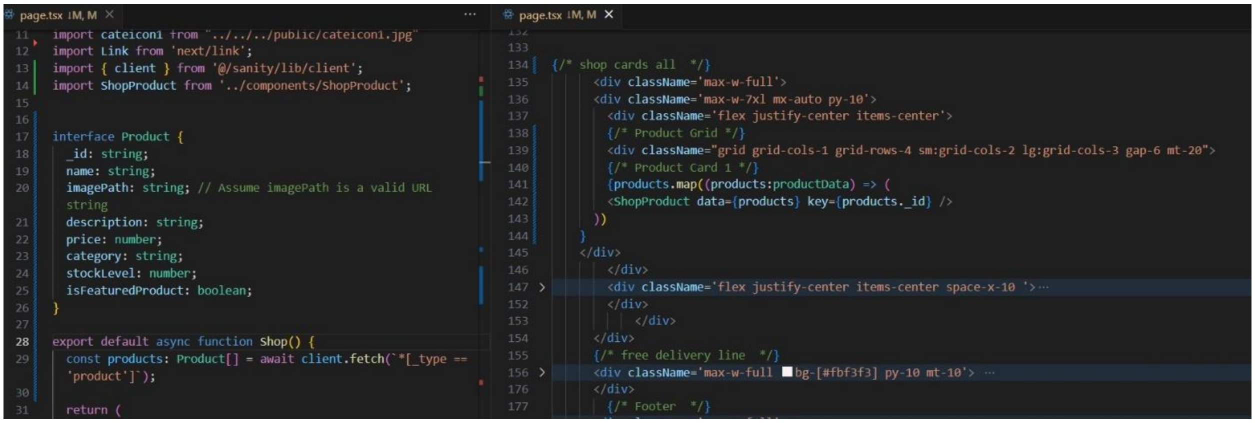
```
import axios from 'axios';
import { client } from "@/sanity/lib/client";
export default async function handler(res:any) {
  try {
    const { data } = await axios.get('https://template-0-beta.vercel.app/api/product');
    for (const product of data) {
      await client.create({
        _type: 'product',
        id: product.id,
        name: product.name,
        imagePath: product.imagePath,
        price: parseFloat(product.price),
        description: product.description,
        discountPercentage: product.discountPercentage,
        isFeaturedProduct: product.isFeaturedProduct,
        stockLevel: product.stockLevel,
        category: product.category,
      });
    }
    res.status(200).json({ message: 'Data inserted successfully!' });
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'Failed to fetch or insert data' });
  }
}

```

By running this API route, all furniture data was successfully fetched and inserted into the Sanity CMS.

Step 4: Verifying and Displaying Data

After migrating the data, I verified it in the Sanity Studio under the "Furniture Product" section. To ensure the data was displayed correctly on the website, I implemented a page to fetch and render it dynamically:

Fetch & Displaying Furniture Data in Next.js:



```
import cateicon1 from "../../../public/cateicon1.jpg"
import Link from 'next/link';
import { client } from '@/sanity/lib/client';
import ShopProduct from '../components/ShopProduct';


interface Product {
  _id: string;
  name: string;
  imagePath: string; // Assume imagePath is a valid URL
  string
  description: string;
  price: number;
  category: string;
  stockLevel: number;
  isFeaturedProduct: boolean;
}

export default async function Shop() {
  const products: Product[] = await client.fetch(`*[_type ==
  'product']`);

  return (
```

```

{/* shop cards all  */}
      <div className='max-w-full'>
      <div className='max-w-7xl mx-auto py-10'>
        <div className='flex justify-center items-center'>
        {/* Product Grid */}
        <div className="grid grid-cols-1 grid-rows-4 sm:grid-cols-2 lg:grid-cols-3 gap-6 mt-20">
        {/* Product Card 1 */}
        {products.map((products:productData) => (
        <ShopProduct data={products} key={products._id} />
      ))
    }
    </div>
        </div>
        <div className='flex justify-center items-center space-x-10 '>⋯
        </div>
            </div>
      </div>
      {/* free delivery line  */}
      <div className='max-w-full ⬜bg-[#fbf3f3] py-10 mt-10'> ⋯
      </div>
        {/* Footer  */}
```

Step 5: Testing and Feedback

I tested the implementation by running the Next.js app and verified the functionality on HTTP://LOCALHOST:3000/SHOP. All furniture products, along with their details, were displayed successfully, providing a seamless user experience for browsing sofas, tables, chairs, and other furniture.

Final Thoughts:

This process successfully set up the backend for a furniture e-commerce website using Sanity CMS and integrated data from a furniture API. The flexibility of Sanity CMS allowed me to customize the schema for furniture-specific attributes, while Next.js made it easy to fetch and display the data dynamically.