**Day 3 - API Integration Report - SnapEase**

**1. API Integration Process**

**Understanding the API**

* The API provides product listings with fields such as product\_title, price, description, image\_url, and category\_id.
* Categories are embedded within the /products API response rather than having a separate /categories endpoint.

**Integration Steps**

1. **Fetched Data from API** using client.fetch() in Sanity.
2. **Mapped API Fields to Sanity Schema** to align with existing data structures.
3. **Posted Data to Sanity CMS** using client.createOrReplace() for products and categories.
4. **Fetched Data in Next.js Frontend** to display products dynamically.

**2. Adjustments Made to Schemas**

**Sanity Schema Adjustments**

* Renamed product\_title → name
* Renamed category\_id → category (stored as a reference)
* Adjusted price and description for consistency

**Example Sanity Schema for Products**

export default {

  name: 'product',

  type: 'document',

  title: 'Product',

  fields: [

    {

      name: 'name',

      type: 'string',

      title: 'Name',

      validation: (Rule: any) => Rule.required().error('Name is required'),

    },

    {

      name: 'image',

      type: 'image',

      title: 'Image',

      options: {

        hotspot: true,

      },

      description: 'Upload an image of the product.',

    },

    {

      name: 'price',

      type: 'string',

      title: 'Price',

      validation: (Rule: any) => Rule.required().error('Price is required'),

    },

    {

      name: 'description',

      type: 'text',

      title: 'Description',

      validation: (Rule: any) =>

        Rule.max(150).warning('Keep the description under 150 characters.'),

    },

    {

      name: 'discountPercentage',

      type: 'number',

      title: 'Discount Percentage',

      validation: (Rule: any) =>

        Rule.min(0).max(100).warning('Discount must be between 0 and 100.'),

    },

    {

      name: 'isFeaturedProduct',

      type: 'boolean',

      title: 'Is Featured Product',

    },

    {

      name: 'stockLevel',

      type: 'number',

      title: 'Stock Level',

      validation: (Rule: any) => Rule.min(0).error('Stock level must be a positive number.'),

    },

    {

      name: 'category',

      type: 'string',

      title: 'Category',

      options: {

        list: [

          { title: 'Chair', value: 'Chair' },

          { title: 'Sofa', value: 'Sofa' },

        ],

      },

      validation: (Rule: any) => Rule.required().error('Category is required'),

    },

  ],

};

**3. Migration Steps and Tools Used**

**Migration Steps**

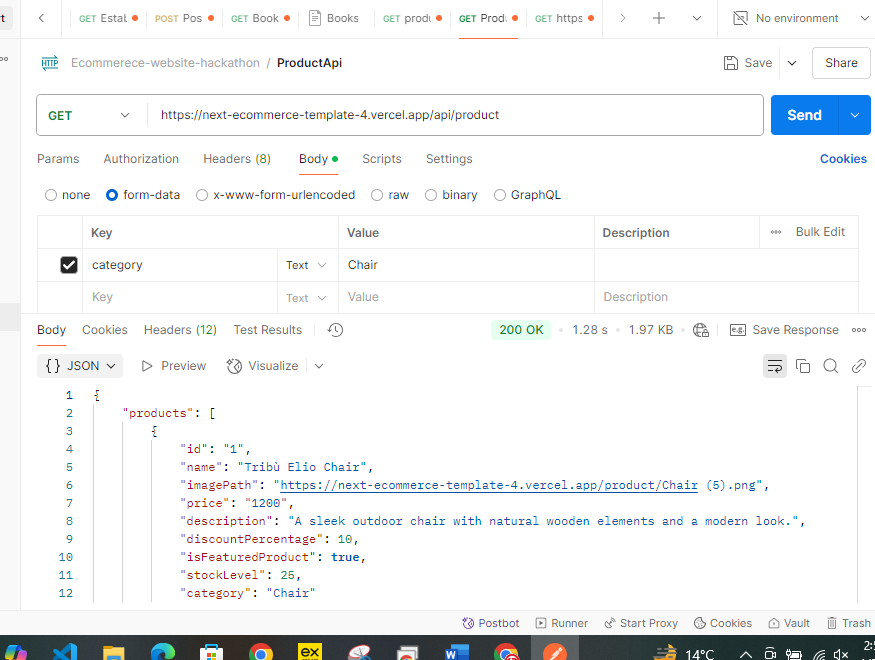
1. **Fetched API Data**
2. **Transformed Data to Match Sanity Schema**
3. **Created/Updated Documents in Sanity CMS**
4. **Validated Data by Querying Sanity Studio**
5. **Fetched Data in Next.js and Rendered it**

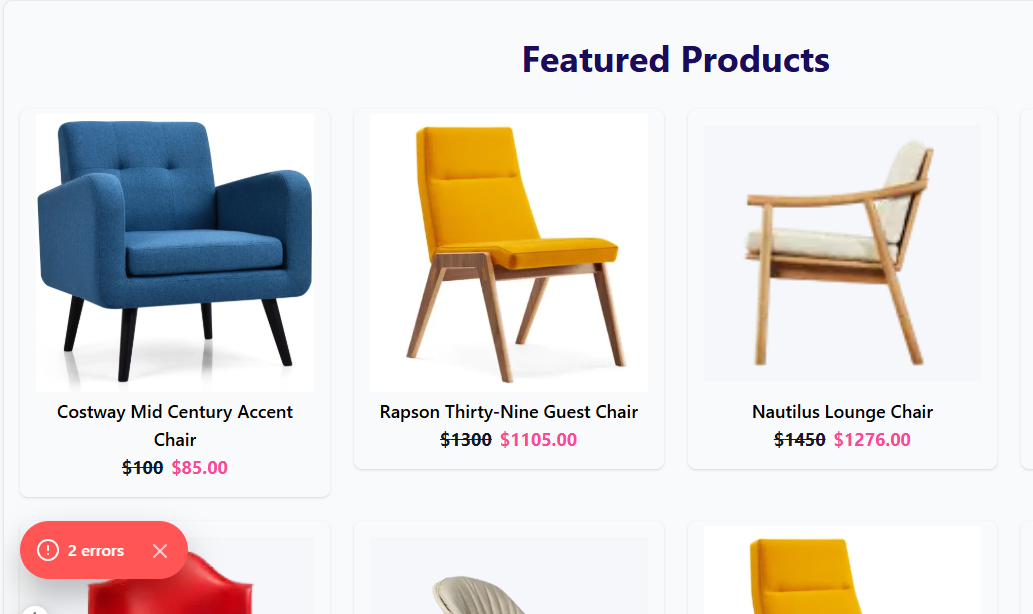
**Tools Used**

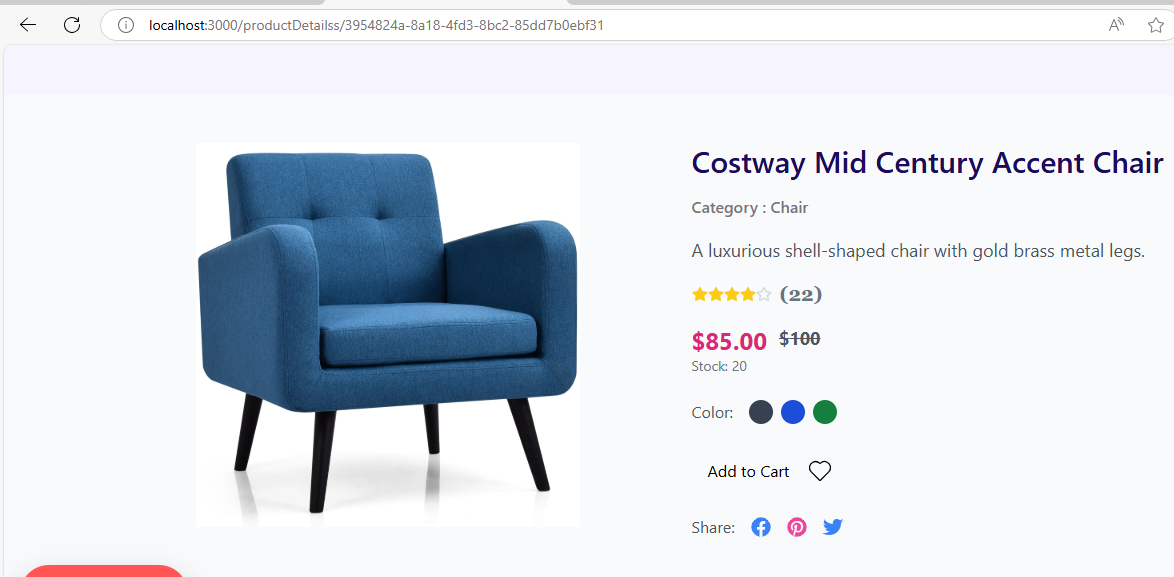
* **Sanity Client** for posting and fetching data.
* **Next.js** for frontend integration.
* **Postman** for testing API calls.

**4. Screenshots**

✅ **API Calls in Postman** (Fetching /products API)

  
✅ **Frontend Display of Data** (Product listings with categories)

  
✅ **Populated Sanity CMS Fields** (Products stored in Sanity)



**5. Code Snippets for API Integration and Migration Scripts**

**FeaturedProducts:(Chair,Sofa)**

import Image from "next/image";

import { useEffect, useState } from "react";

import { client } from "../../../../studio-template4-ecommerce-website/lib/client";

import Link from "next/link";

// Define the Product type

type Product = {

  id: string;

  name: string;

  imagePath: string;

  price: number;

  description: string;

  discountPercentage: number;

  stockLevel: number;

  category: string;

};

const ProductGrid = () => {

  const [products, setProducts] = useState<Product[]>([]);

  const [loading, setLoading] = useState<boolean>(true);

  const [error, setError] = useState<string | null>(null);

  // Fetch featured products from Sanity

  useEffect(() => {

    const fetchProducts = async () => {

      const query = `\*[\_type == "product" && isFeaturedProduct == true && category=="Chair" ]{

        \_id,

        "id": \_id,

        name,

        "imagePath": image.asset->url,

        "price": price,

        "description": description,

        "discountPercentage": discountPercentage,

        "stockLevel": stockLevel,

        "category": category

      }`;

      try {

        const data: Product[] = await client.fetch(query);

        setProducts(data);

      } catch (error) {

        setError("Error fetching featured products.");

        console.error("Error fetching featured products:", error);

      } finally {

        setLoading(false);

      }

    };

    fetchProducts();

  }, []);

  return (

    <div className="bg-gray-50 py-12">

      <h2 className="text-4xl font-bold text-center mb-8 text-[#1A0B5B]">

        Featured Products

      </h2>

      <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-6 px-4">

        {loading ? (

          <p className="text-center col-span-4">Loading...</p>

        ) : error ? (

          <p className="text-center col-span-4 text-red-500">{error}</p>

        ) : products.length > 0 ? (

          products.map((product) => (

            <Link

              key={product.id}

              href={`/productDetailss/${product.id}`} // Use dynamic route

              passHref

            >

              <div className="group relative p-4 hover:bg-[#2F1AC4] hover:text-white rounded-lg shadow hover:shadow-lg transition cursor-pointer">

                <div className="flex items-center justify-center h-64 bg-[#F6F7FB]">

                  <Image

                    height={2000}

                    width={1000}

                    src={product.imagePath}

                    alt={product.name}

                    className="object-contain h-full"

                    layout="intrinsic"

                  />

                </div>

                <div className="mt-4 text-center hover:text-white">

                  <h3 className="text-lg font-medium">{product.name}</h3>

                  <div className="flex justify-center items-center space-x-2">

                    <p className="line-through text-lg font-bold text-gray-900">

                      ${product.price}

                    </p>

                    {product.discountPercentage > 0 && (

                      <p className="text-lg text-[#FB4997] font-bold">

                        ${(

                          product.price -

                          (product.discountPercentage / 100) \* product.price

                        ).toFixed(2)}

                      </p>

                    )}

                  </div>

                </div>

              </div>

            </Link>

          ))

        ) : (

          <p className="text-center col-span-4">No featured products found.</p>

        )}

      </div>

    </div>

  );

};

export default ProductGrid;

**Product Details:**

// src/app/productDetailss/[id]/page.tsx

import Image from "next/image";

import { client } from "../../../../../studio-template4-ecommerce-website/lib/client";

import { FaStar, FaRegStar, FaFacebook, FaTwitter, FaPinterest } from "react-icons/fa";

import { CiHeart } from "react-icons/ci";

import SectionHeader from "@/app/components/SectionHeader";

import Footer from "@/app/components/Footer/Footer";

import ProductDetailsTabs from "@/app/home/ProductDetailsTabs";

// Define the product type

type Product = {

  id: string;

  name: string;

  imagePath: string;

  price: number;

  description: string;

  discountPercentage: number;

  stockLevel: number;

  category: string;

};

const relatedProducts = [

    { id: 1, image: "/product5.png", name: "Men's Fashion Wear", price: "$43.00", rating: 4 },

    { id: 2, image: "/product6.png", name: "Women’s Fashion", price: "$67.00", rating: 5 },

    { id: 3, image: "/product7.png", name: "Wolx Dummy Fashion", price: "$67.00", rating: 3 },

    { id: 4, image: "/product8.png", name: "Top Wall Digital Clock", price: "$51.00", rating: 4 },

  ];

const ProductDetails = async ({ params }: { params: { id: string } }) => {

  // Fetch product data server-side

  const query = `\*[\_type == "product" && \_id == $id]{

    \_id,

    "id": \_id,

    name,

    "imagePath": image.asset->url,

    "price": price,

    "description": description,

    "discountPercentage": discountPercentage,

    "stockLevel": stockLevel,

    "category": category

  }`;

  const product :Product[]= await client.fetch(query, { id: params.id });

  // Check if product is found

  if (!product || product.length === 0) {

    return <p className="text-center text-red-500">Product not found.</p>;

  }

  const productDetails = product[0];

  // Calculate discounted price

  const discountedPrice = productDetails.discountPercentage

    ? (productDetails.price - (productDetails.discountPercentage / 100) \* productDetails.price).toFixed(2)

    : productDetails.price;

  return (

    <div className="bg-gray-50">

      {/\* Section Header \*/}

      <SectionHeader

        title="Product Details"

        breadcrumb="Home . Pages"

        breadcrumbHighlight="Product Details"

        backgroundColor="bg-[#F6F5FF]"

        textColor="text-gray-800"

      />

      {/\* Product Section \*/}

      <div className="max-w-6xl mx-auto p-4 grid grid-cols-1 md:grid-cols-2 gap-8 mt-8">

        {/\* Product Images \*/}

        <div className="relative w-full h-96">

          <Image

            src={productDetails.imagePath}

            alt={productDetails.name}

            layout="fill"

            objectFit="contain"

            className="rounded"

          />

        </div>

        {/\* Product Details \*/}

        <div className="flex flex-col justify-center">

          <h1 className="text-3xl font-semibold text-[#1A0B5B]">{productDetails.name}</h1>

          <h6 className=" font-semibold text-[#757479] mt-[3%] mb-[3%]"> Category : {productDetails.category}</h6>

          <p className="text-lg text-gray-600 mb-4">{productDetails.description}</p>

          {/\* Rating Section \*/}

          <div className="flex items-center mb-4">

            {[...Array(5)].map((\_, index) => (

              index < 4 ? <FaStar key={index} className="text-yellow-400" /> : <FaRegStar key={index} className="text-gray-300" />

            ))}

            <span className="ml-2 text-gray-500 font-serif font-bold text-xl">(22)</span>

          </div>

          {/\* Price and Stock \*/}

          <div className="flex gap-3">

            <p className="text-2xl font-bold text-pink-600">${discountedPrice}</p>

            <p className="text-lg font-bold text-gray-600 line-through">${productDetails.price}</p>

          </div>

          <p className="text-sm text-gray-500">

            Stock: {productDetails.stockLevel > 0 ? productDetails.stockLevel : "Out of stock"}

          </p>

          {/\* Color Selection (For now, placeholder colors) \*/}

          <div className="flex items-center mt-6">

            <p className="text-gray-600 mr-4">Color:</p>

            <div className="flex">

              <button className="w-6 h-6 rounded-full bg-gray-700 mr-2"></button>

              <button className="w-6 h-6 rounded-full bg-blue-700 mr-2"></button>

              <button className="w-6 h-6 rounded-full bg-green-700"></button>

            </div>

          </div>

           {/\* Add to Cart Button \*/}

           <div className="flex items-center mt-[5%]">

           <button className=" text-black px-4 py-2 rounded hover:bg-pink-700">

              Add to Cart

            </button>

            <CiHeart size={30} />

           </div>

          {/\* Share Section \*/}

          <div className="mt-6 flex gap-4">

            <p className="text-gray-600">Share:</p>

            <button className="text-blue-500"><FaFacebook size={20} /></button>

            <button className="text-pink-500"><FaPinterest size={20} /></button>

            <button className="text-blue-500"><FaTwitter size={20} /></button>

          </div>

        </div>

      </div>

      {/\* Product Details Tabs \*/}

      <div className="mt-12">

        <ProductDetailsTabs  />

      </div>

       {/\* Related Products \*/}

            <div className="max-w-6xl mx-auto px-4 mt-12">

              <h2 className="text-xl font-bold mb-4">Related Products</h2>

              <div className="grid grid-cols-4 gap-4">

                {relatedProducts.map((product) => (

                  <div key={product.id} className="text-center p-4 bg-white rounded shadow">

                    <Image width={1000} height={120} src={product.image} alt={product.name} className="w-full mb-2 rounded" layout="responsive" />

                    <h3 className="font-semibold">{product.name}</h3>

                    <p className="text-pink-600 font-bold">{product.price}</p>

                    {/\* Rating \*/}

                    <div className="flex justify-center mb-2">

                      {Array.from({ length: 5 }, (\_, index) => (

                        index < product.rating ? (

                          <FaStar key={index} className="text-yellow-400" />

                        ) : (

                          <FaRegStar key={index} className="text-gray-300" />

                        )

                      ))}

                    </div>

                    <button className="mt-2 text-pink-600 border border-pink-600 px-2 py-1 rounded hover:bg-pink-600 hover:text-white">

                      View Product

                    </button>

                  </div>

                ))}

              </div>

            </div>

      {/\* Footer \*/}

      <Footer />

    </div>

  );

};

export default ProductDetails;

**Latest Products:**

import { useEffect, useState } from "react";

import Image from "next/image";

import { client } from "../../../../studio-template4-ecommerce-website/lib/client";

import Link from "next/link";

// Define Product type

type Product = {

  id: string;

  name: string;

  price: string;

  discountPercentage: string;

  image: string;

};

export default function LatestProducts() {

  const [products, setProducts] = useState<Product[]>([]);

  const [loading, setLoading] = useState<boolean>(true); // Loading state

  // Fetch the latest products from Sanity

  useEffect(() => {

    const fetchLatestProducts = async () => {

      const query = `\*[\_type == "product"] | order(\_createdAt desc) [0...6] {

        \_id,

        name,

        price,

        discountPercentage,

        "image": image.asset->url

      }`;

      try {

        const data: Product[] = await client.fetch(query);

        console.log(data);  // Check the fetched products

        setProducts(data);

      } catch (error) {

        console.error("Error fetching products: ", error);

      } finally {

        setLoading(false);

      }

    };

    fetchLatestProducts();

  }, []);

  // Product Card Component

  const ProductCard = ({ image, name, price, discountPercentage }: ProductCardProps) => {

    const discountedPrice = discountPercentage

      ? (parseFloat(price) \* (1 - parseFloat(discountPercentage) / 100)).toFixed(2)

      : price;

    return (

      <div className="border rounded-lg shadow-md p-4 bg-[#f7f7f7]">

        <div className="w-360 h-270 mx-auto">

          <Image

            height={250}

            width={1000}

            src={image}

            alt={name}

            className="w-full h-250 object-cover rounded-t-lg mb-4"

          />

        </div>

        <div className="bg-white p-4 rounded-b-lg">

          <h2 className="text-lg font-semibold mb-2">{name}</h2>

          <div className="flex items-center">

            <span className="text-pink-500 font-bold mr-2">${discountedPrice}</span>

            {discountPercentage && (

              <span className="text-gray-400 line-through">${price}</span>

            )}

          </div>

        </div>

      </div>

    );

  };

  interface ProductCardProps {

    image: string;

    name: string;

    price: string;

    discountPercentage: string;

  }

  return (

    <div className="container mx-auto px-4 py-8" style={{ backgroundColor: "white" }}>

      <h1 className="text-3xl font-bold text-center mb-8">Latest Products</h1>

      <div className="flex justify-center space-x-8 text-[#151875]">

        <a href="#" className="text-[#FB4997] font-semibold">

          New Arrival

        </a>

        <a href="#" className="hover:text-[#FB4997]">Best Seller</a>

        <a href="#" className="hover:text-[#FB4997]">Featured</a>

        <a href="#" className="hover:text-[#FB4997]">Special Offer</a>

      </div>

      {/\* Show loading state if data is being fetched \*/}

      {loading ? (

        <p className="text-center text-xl font-semibold">Loading...</p>

      ) : (

        <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 gap-8 mt-8">

          {products.map((product) => (

            <div key={product.id}>

              <Link href={`/productDetails/${product.id}`} passHref>

                  <ProductCard

                    image={product.image}

                    name={product.name}

                    price={product.price}

                    discountPercentage={product.discountPercentage}

                  />

              </Link>

            </div>

          ))}

        </div>

      )}

    </div>

  );

}

**Showing only Chair Data:**

import Image from 'next/image';

import { useEffect, useState } from 'react';

import { client } from '../../../../studio-template4-ecommerce-website/lib/client';

const features = [

  {

    image: "/aboutlogo1.png",

    title: "Free Delivery",

    description: "Enjoy free delivery on all orders above $50.",

  },

  {

    image: "/aboutlogo2.png",

    title: "100% Cash Back",

    description: "Guaranteed refunds if not satisfied.",

  },

  {

    image: "/aboutlogo3.png",

    title: "Quality Product",

    description: "We provide high-quality products for all needs.",

  },

  {

    image: "/aboutlogo4.png",

    title: "24/7 Support",

    description: "Contact our team anytime for assistance.",

  },

];

// Define the Product type

type Product = {

  id: string;

  name: string;

  imagePath: string;

  price: number;

  description: string;

  discountPercentage: number;

  stockLevel: number;

  category: string;

};

export default function RemainSection() {

  const [chairs, setChairs] = useState <Product[]>([]);

  useEffect(() => {

    // Define the async function inside useEffect

    const fetchChairs = async () => {

      const query = `

        \*[\_type == "product" && category == "Chair"] | order(\_createdAt asc) [0...4] {

          \_id,

          "id": \_id,

          name,

          "imagePath": image.asset->url,

          price,

          description,

          discountPercentage,

          stockLevel,

          category

        }

      `;

      try {

        const data:Product[] = await client.fetch(query);

        setChairs(data); // Update the state with fetched data

      } catch (error) {

        console.error("Error fetching chairs:", error);

      }

    };

    // Call the fetchChairs function

    fetchChairs();

  }, []); // Empty dependency array to only run on component mount

  return (

    <div className="">

      {/\* Features Section \*/}

      <section className="py-12 px-6">

        <h3 className="text-center text-2xl font-bold mb-8">What Shopex Offer!</h3>

        <div className="grid grid-cols-1 md:grid-cols-4 gap-8 max-w-6xl mx-auto">

          {features.map((feature, index) => (

            <div key={index} className="text-center p-4 shadow-md rounded-lg">

              <Image

                src={feature.image}

                alt={feature.title}

                width={50}

                height={50}

                className="mx-auto mb-4"

              />

              <h4 className="text-xl font-semibold mb-2">{feature.title}</h4>

              <p className="text-gray-600">{feature.description}</p>

            </div>

          ))}

        </div>

      </section>

      {/\* Hero Section \*/}

      <section className="bg-[#F1F0FF] py-10">

        <div className="max-w-5xl mx-auto px-4">

          <div className="flex flex-col md:flex-row items-center justify-between space-y-10 md:space-y-0">

            {/\* Left Column: Content \*/}

            <div className="flex-1 flex justify-center">

              <div className="text-center">

                <Image

                  src="/uniqueFeature.png"

                  alt="B&B Italian Sofa"

                  width={300}

                  height={300}

                  className="object-contain"

                />

              </div>

            </div>

            <div className="flex-1 text-center md:text-left">

              <h1 className="text-4xl font-bold mb-4 leading-tight">

                Unique Features Of Latest &<br />

                Trending Products

              </h1>

              <ul className="text-gray-700 mb-6 list-disc list-inside">

                <li>

                  All frames constructed with hardwood solids and laminates.

                </li>

                <li>

                  Reinforced with double wood dowels, glue, screw-nails, corner

                  blocks, and machine nails.

                </li>

                <li>Arms, backs, and seats are structurally reinforced.</li>

              </ul>

             <div className='flex gap-10'>

             <button className="bg-pink-500 text-white p-5 rounded-md hover:bg-pink-600">

                Add To Cart

              </button>

              <div>

                <p className="text-gray-800 font-bold text-xl mt-4">

                  B&B Italian Sofa

                </p>

                <p className="text-pink-500 text-lg">$32.00</p>

              </div>

             </div>

            </div>

          </div>

        </div>

      </section>

      {/\* Top Categories Section \*/}

      <section className="py-10 text-center">

        <h2 className="text-2xl font-bold mb-[5%]">

          <span className="px-4 py-2 border-b-4 border-blue-500">Top Categories</span>

        </h2>

        <div className="flex justify-center gap-6 px-4 max-w-6xl mx-auto">

          {chairs.map((item, index) => (

            <div key={index} className="flex flex-col items-center">

              <div className="w-64 h-80 rounded-lg bg-gray-100 flex items-center justify-center shadow-md">

                <Image

                  src={item.imagePath}  // Use the fetched image path

                  alt={item.name}

                  width={270}

                  height={350}

                  className="object-contain"

                />

              </div>

              <h3 className="mt-4 font-semibold text-gray-700">{item.name}</h3>

              <p className="text-pink-500 font-bold mt-2">${item.price}</p>

            </div>

          ))}

        </div>

      </section>

      {/\* Subscribe Section \*/}

      <section className="bg-gray-200 py-10 text-center relative">

        <div className="absolute inset-0 z-0">

          <Image src={"/final.png"} width={2000} alt="Shop Now Image" layout='responsive' height={500} />

        </div>

        <div className="relative z-10 max-w-4xl mx-auto px-4 mb-10 mt-[10%]">

          <h2 className="text-2xl font-bold mb-4 text-black">Get Latest Update By Subscribe Our Newsletter</h2>

          <button className="bg-pink-500 text-white px-4 py-2 rounded-lg hover:bg-pink-600 mb-6">

            Shop Now

          </button>

        </div>

      </section>

      {/\* Latest Blog Section \*/}

      <section className="py-10">

        <div className="text-center mb-6">

          <h2 className="text-3xl font-bold">Latest Blog</h2>

        </div>

        <div className="grid grid-cols-1 md:grid-cols-3 gap-6 px-4 max-w-6xl mx-auto">

          {[{title: 'Blog Post 1', date: 'Jan 10, 2024', image: "/latblog1.png", content: 'Lorem ipsum dolor sit amet.'},

            {title: 'Blog Post 2', date: 'Jan 12, 2024', image: "/latblog2.png", content: 'Consectetur adipiscing elit.'},

            {title: 'Blog Post 3', date: 'Jan 14, 2024', image: "/latblog3.png", content: 'Exploring new technology.'}]

            .map((post, index) => (

              <div key={index} className="border p-4 rounded-lg shadow-md">

                <Image src={post.image} width={255} height={270} className='w-full mb-5' alt={`Blog post image for ${post.title}`} />

                <h3 className="text-xl font-semibold">{post.title}</h3>

                <p className="text-gray-600 mt-2">{post.date}</p>

                <p className="text-gray-700 mt-2">{post.content}</p>

                <button className="text-blue-500 mt-4">Read More</button>

              </div>

            ))}

        </div>

      </section>

    </div>

  );

}

**Fetching Data from API & Posting to Sanity CMS**

import sanityClient from '@sanity/client';

const client = sanityClient({

projectId: 'your\_project\_id',

dataset: 'production',

useCdn: false,

token: 'your\_sanity\_token'

});

async function migrateProducts() {

const response = await fetch('https://api.example.com/products');

const products = await response.json();

for (const product of products) {

await client.createOrReplace({

\_type: 'product',

\_id: product.product\_id,

name: product.product\_title,

price: product.price,

description: product.description,

category: { \_ref: product.category\_id }

});

}

}

migrateProducts();

**Fetching Data in Next.js Frontend**

export async function fetchProducts() {

const query = '\*[\_type == "product"] { name, price, description, category-> { title }, image }';

const products = await client.fetch(query);

return products;

}

**Rendering Products in Next.js**

import { fetchProducts } from './apiUtils';

import { useEffect, useState } from 'react';

function ProductList() {

const [products, setProducts] = useState([]);

useEffect(() => {

const getProducts = async () => {

const data = await fetchProducts();

setProducts(data);

};

getProducts();

}, []);

return (

<div className="product-list">

{products.map((product) => (

<div key={product.name} className="product-item">

<h3>{product.name}</h3>

<p>{product.price}</p>

<p>{product.category?.title}</p>

</div>

))}

</div>

);

}

export default ProductList;

**Final Outcome:**

✅ API data successfully posted to **Sanity CMS**  
✅ Product listings displayed correctly in **Next.js frontend**  
✅ Category filtering works based on Sanity schema