Day 3 - API Integration Report - [Marketplace-Technical-Foundation--Hamza-Furniture-] Repo Name

Prepared by: Muhammad Hamza Samad

API Integration Process

The API integration process involved fetching product data from an external API and importing it into Sanity CMS. The steps are outlined below:

Fetching Product Data: Data was retrieved from https://next-ecommerce-template-4.vercel.app/api/product using Axios.

Uploading Images to Sanity: Images associated with products were downloaded and uploaded to Sanity as assets.

Storing Data in Sanity: The transformed data was structured according to the Sanity schema and uploaded to the database.

Migration Steps and Tools Used

Migration Process

API Data Fetching: Retrieved data from next-ecommerce-template-4 API.

Image Upload: Downloaded images and stored them in Sanity CMS.

Data Transformation: Adjusted the data structure to match the Sanity schema.

Data Insertion: Used the Sanity client to insert data into the CMS.

Tools Used

Tailwind Css

Next.js

Sanity CMS

Axios

Node.js

Dotenv

Anasseth Repo

Code Implementation

The following code snippet demonstrates the API integration:

```
import ( createClient ) from '@sanity/client';
import axios from 'axios';
import dateny from 'doteny'
import { fileURLToPath } from 'url';
import path from 'path';
const_filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '_/_/env') });
const client = createClient{{
    projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID.
    dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
    token: process.env.SANITY_APL_TOKEN,
  apiVersion: '2025-01-13',
 useCdn: false,
}):
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: imageUrLsplit()').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 importData() {
 try {
  console.log('Fetching Product Data From API ...');
  const response = await axios.get("https://next-ecommerce-template-
4.vercel.app/api/product")
  const products = respo
  for (const item of products) [
   console.log('Processing Item: ${item.name}');
   let imageRef = null;
   if (item.imagePath) {
   imageRef = await uploadImageToSanihty(item.imagePath);
   const sanityItem = {
    _type: 'product'
     name; item.name
     category: item.category || null,
     price: item.price,
description: item.description || ",
     discountPercentage: item.discountPercentage || 0;
stockLevel: item.stockLevel || 0,
     isFeaturedProduct: item.isFeaturedProduct.
```

Adjustments Made to Schemas

The product schema was adjusted to accommodate data from the external API while ensuring required fields and data validation.

```
defineField({
    name: 'description',
    type: 'text',
    title: 'Description',
    validation: (Rule: any) =>
    Rule.max(150).waming('Keep the description under 150
                                                                                                                                                                                                                                defineField({
import { defineField, defineType } from "sanity";
                                                                                                                                                                                                                                 name: 'category',
type: 'string',
title: 'Category',
export const productType = defineType({
   name: 'product'.
                                                                                                                                                                                                                                  options: {
   type: 'document',
title: 'Product',
                                                                                                                                                                                                                                   list: {
    { title: 'Chair', value: 'Chair' },
    { title: 'Sofa', value: 'Sofa' },
                                                                                                  characters.'),
  fields: {
defineField({
                                                                                                        defineField({
                                                                                                         dennel-ieldi(
name: 'discountPercentage',
type: 'number',
title: 'Discount Percentage',
validation: (Rule: any) =>
      type: 'string',
title: 'Name',
validation: (Rule: any) => Rule.required().error('Name is
                                                                                                                                                                                                                                  validation: (Rule: any) => Rule.required().error('Category is required'),
                                                                                                                                                                                                                                1).
                                                                                                           Rule.min(0).max(100).warning("Discount must be between 0 and
required").
                                                                                                  100.)
                                                                                                                                                                                                                            });
                                                                                                       }).
defineField({
                                                                                                                                                                                                                                  validation: (Rule: any) => Rule.required().error('Category is required'),
     defineField({
       name: 'image',
                                                                                                                                                                                                                                1).
                                                                                                         name: 'isFeaturedProduct',
type: 'boolean',
title: 'is Featured Product',
      type: 'image',
title: 'Image',
                                                                                                                                                                                                                            }):
      options: {
                                                                                                       }).
defineField({
        hotspot: true,
                                                                                                          name: 'stockLevel'.
    )).
defineField({
                                                                                                         name: sucketes,
type: 'number',
title: 'Stock Level',
validation: (Rule: any) => Rule min(0).error|'Stock level must be a
      name: 'price',
      type: 'string',
title: 'Price',
                                                                                                   positive number.'),
                                                                                                       }).
validation: (Rule: any) => Rule required().error('Category is required'),
      validation: (Rule: any) => Rule.required().error('Price is
required'),
                                                                                                       11.
    3).
                                                                                                    ].
]):
```

Required Steps

Populated Sanity CMS Fields:

Verify that all product fields are correctly stored in Sanity CMS.

Ensure that uploaded images appear correctly in the CMS.

Code Snippets for API Integration and Migration Scripts:

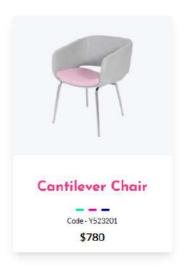
Ensure that all integration scripts are documented properly.

Validate that the scripts function correctly in the production environment.

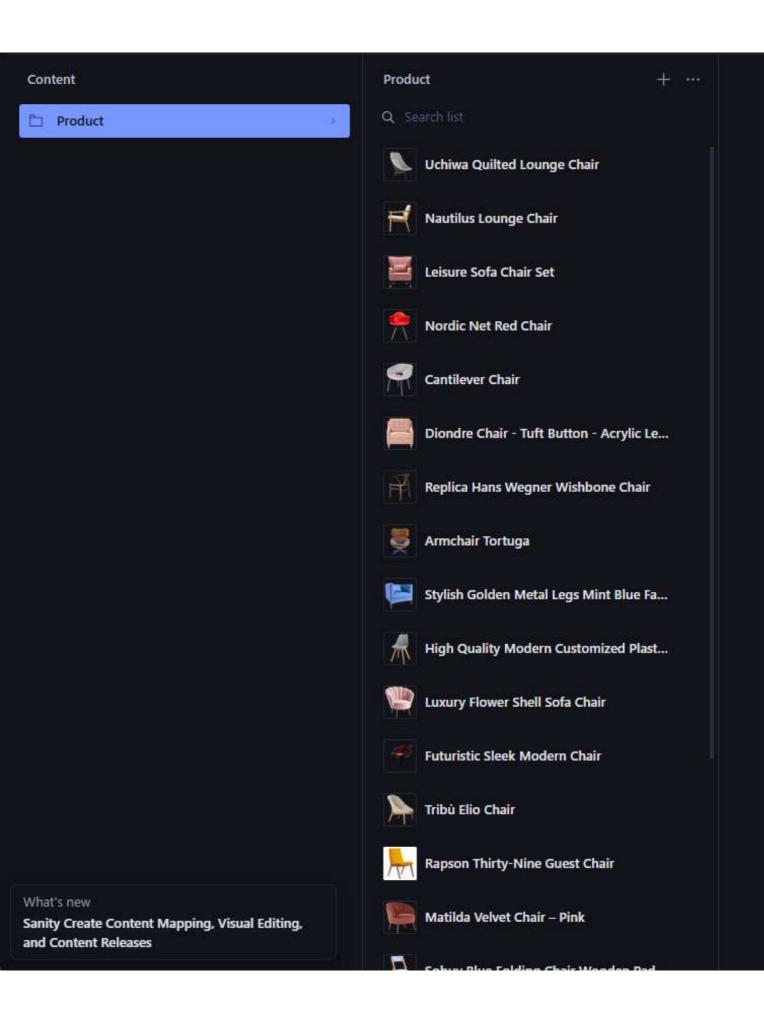
Featured Products











```
"products": [
                                    "id": "1",
"name": "Tribù Elio Chair",
"imagePath": "https://next-ecommerce-template-4.vercel.app/product/Chair (5).png",
"price": "1200",
"description": "A sleek outdoor chair with natural wooden elements and a modern look.",
                                    "discountPercentage": 10,
"isFeaturedProduct": true,
"stockLevel": 25,
"category": "Chair"
                                   "id": "2",
"name": "Armchair Tortuga",
"imagePath": "https://next-ecommerce-template-4.vercel.app/product/Chair (35).png",
"price": "850",
"price": "850",
                                    "imagePath": "https://next-ecommerce-template-4.vercel.app/product/Chair (35).png",
"price": "850",
"description": "An elegant armchair with plush cushions and a curved design for comfort.",
"discountPercentage": 0,
"isFeaturedProduct": false,
"stockLevel": 10,
"category": "Chair"
                                  "stockLevel": 5,
"category": "Chair"
                                    "id": "4",
"name": "Stylish Golden Metal Legs Mint Blue Fabric Velvet Sofa Leisure Armchair",
"imagePath": "https://next-ecommerce-template-4.vercel.app/product/Chair (3).png",
"price": "780",
"price": "780",
                                    "imagePath": "https://next-ecommerce-template-4.vercel.app/product/Chair (3).png" price": "780", "description": "A luxurious armchair with velvet fabric and golden metal legs.", "discountPercentage": 0, "isFeaturedProduct": false, "stockLevel": 12, "category": "Sofa"
                  },
                                    "id": "5",
"name": "Hans Wegner Style Three-Legged Shell Chair",
"imagePath": "https://next-ecommerce-template-4.vercel.app/product/Chair (4).png",
"price": "990",
"description": "Iconic three-legged chair with faux leather and ash plywood frame.",
                                     "discountPercentage": 10,
"isFeaturedProduct": false,
                                     "stockLevel": 8,
"category": "Chair"
                                   "id": "6",
"name": "Rapson Thirty-Nine Guest Chair",
"imagePath": "https://next-ecommerce-template-4.vercel.app/product/Chair (7).png",
"price": "1300",
"description": "A mid-century modern chair with clean lines and durable materials.",
"" construction of the content of the 
                                    "discountPercentage": 15,
"isFeaturedProduct": true,
"stockLevel": 20,
"category": "Chair"
                  },
```