

API Integration Report for E-Commerce Marketplace

Marketplace Type: General E-Commerce

Introduction:

This Report outlines the process of integrating APIs, making necessary adjustments to the schemas. And migrating data to Sanity CMS for **General E-Commerce Marketplace**. The goal was to enable seamless API integration, integrate external product data, and ensure the CMS is populated and functional.

API Integration Process:

1. API Integration:

The API Integration involved fetching product data from an external source and populating it in Sanity CMS. The following steps were followed.

- **Identifying the API Endpoints:**
The external API that provides product data was identified, including endpoints for product information, images, and categories.
- **Creating API Fetch Functions:**
We Created a fetch function using `fetch ()` to retrieve product data from the external API.
- **Integrating API Calls in the Project:**
API calls were made during the data fetching process in Next.js.

Prepared By: Syed Abdullah Zaidi

Roll No: -

Slot: Tuesday 2pm – 5pm

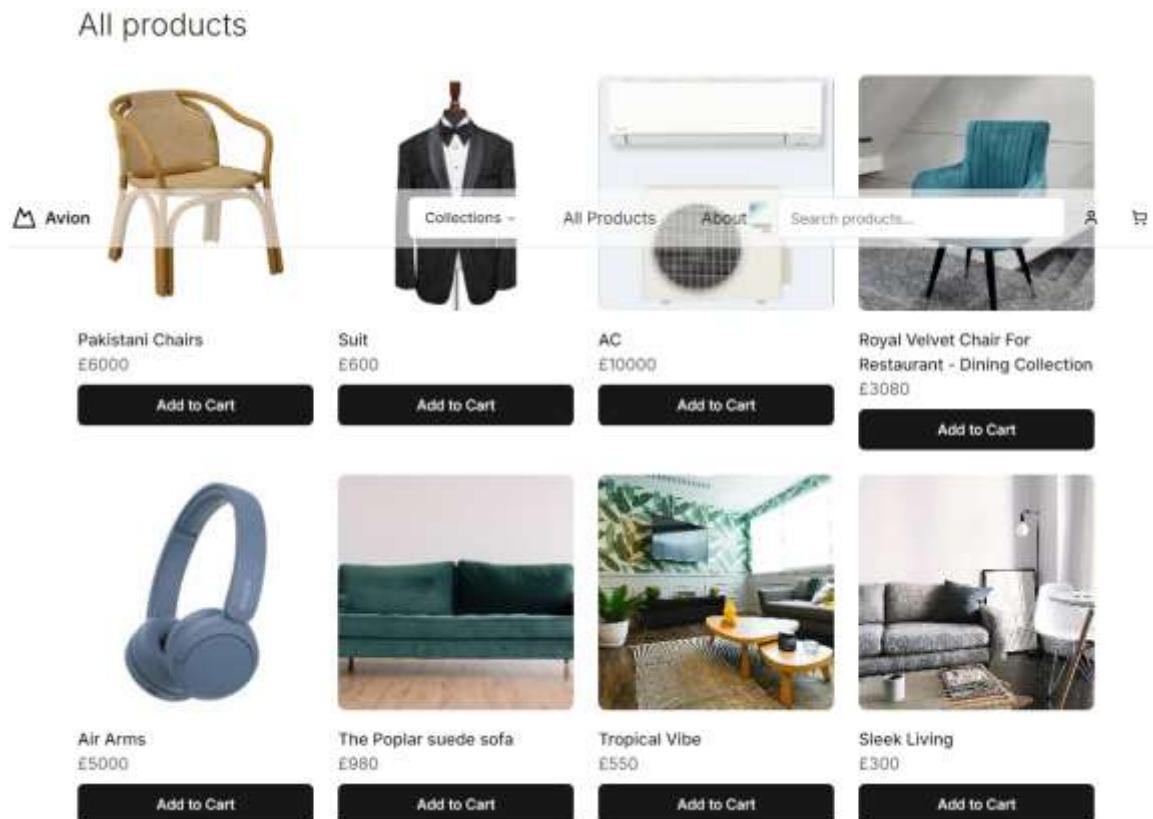
```

1  import { client } from '@lib/sanity'
2  import { NextResponse } from 'next/server'
3
4  export async function GET() {
5    try {
6      const products = await client.fetch(
7        `[_type == "product"] {
8          _id,
9          name,
10         price,
11         "image": image.asset->url,
12         description
13       }
14     `)
15
16     return NextResponse.json(products)
17   } catch (error) {
18     console.error('Error fetching products:', error)
19     return NextResponse.json({ error: 'Failed to fetch products' }, { status: 500 })
20   }
21 }
22

```

Frontend Display of API Data:

Once the data is fetched, it is displayed in the frontend. For Example, All Products page, we display the product details like title and price fetched from the API.

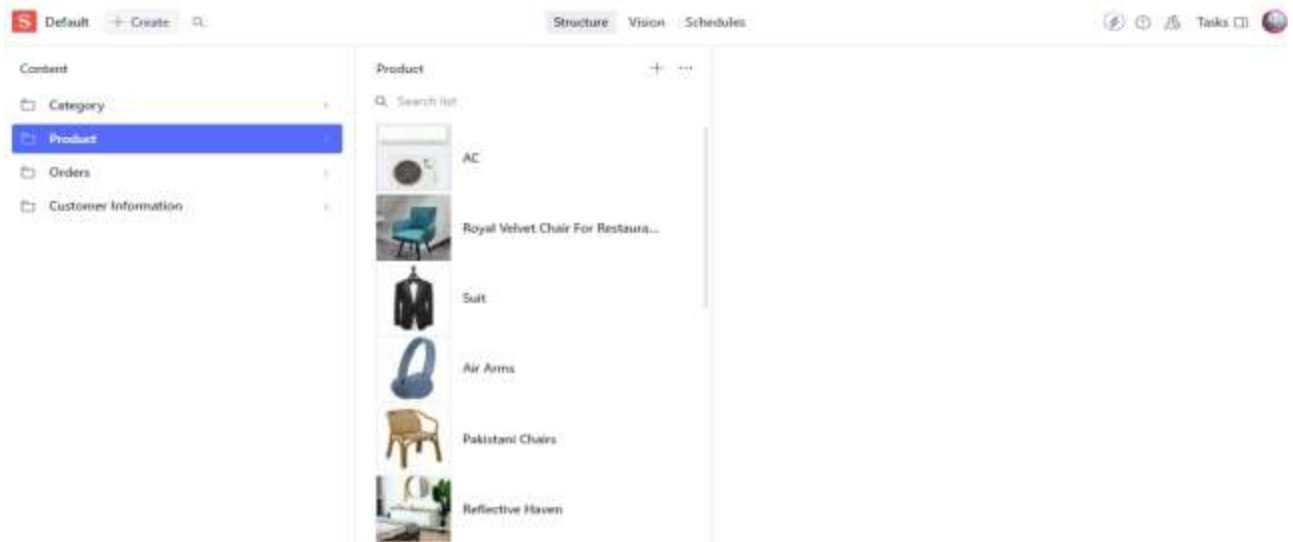


Prepared By: Syed Abdullah Zaidi
Roll No: -
Slot: Tuesday 2pm – 5pm

Handling Data:

Upon successful API calls, product data was parsed and displayed in the frontend.

Sanity Schemas:



2. Schemas Adjustments:

To integrate and store data effectively, the following adjustments were made to the Sanity CMS Schemas.

Product Schema:

New fields were added to the product schema to handle API data.

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

export const product = defineType({
  name: "product",
  title: "Product",
  type: "document",
  fields: [
    defineField({
```

Prepared By: Syed Abdullah Zaidi

Roll No: -

Slot: Tuesday 2pm – 5pm

```

        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",
}),
defineField({
    name: "quantity",
    title: "Quantity",
    type: "number",
    validation: (rule) => rule.min(0),
}),
defineField({
    name: "tags",
    type: "array",
    title: "Tags",
    of: [{
        type: "string"
    }]
})

```

Prepared By: Syed Abdullah Zaidi

Roll No: -

Slot: Tuesday 2pm – 5pm

```

    }},
    defineField({
      name: 'description',
      title: 'Description',
      type: 'text',
      description: 'Detailed description of the product',
    }),
    defineField({
      name: 'features',
      title: 'Features',
      type: 'array',
      of: [{ type: 'string' }],
      description: 'List of key features of the product',
    }),
    defineField({
      name: 'dimensions',
      title: 'Dimensions',
      type: 'object',
      fields: [
        { name: 'height', title: 'Height', type: 'string' },
        { name: 'width', title: 'Width', type: 'string' },
        { name: 'depth', title: 'Depth', type: 'string' },
      ],
      description: 'Dimensions of the product',
    }),
  ],
})),

```

Category Schema:

A Category reference was added in the product schema to link products to specific categories.

```

import { defineType, defineField } from "sanity";

export const Category = defineType({
  name: "category",
  title: "Category",
  type: "document",
  fields: [

```

Prepared By: Syed Abdullah Zaidi

Roll No: -

Slot: Tuesday 2pm – 5pm

```
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",
  }
})
]
})
```

Image Handling:

The Product images fetched via the API were connected with sanity image asset reference field to ensure proper handling of images data.

Migration Steps and Tools Used:

Data Migration:

The Product data was mirrored from the external API into the Sanity CMS using the following steps.

Preparing the Migration Script:

A Node.js script was written to automate the migration process. The Script used Sanity client to create documents in the CMS with data from the external API.

Script to Migrate Product Data:

The script iterated over the product data and pushed it to sanity CMS using the following code.

```
"use strict";
import axios from "axios";
import { client } from "../sanityClient.js";
import slugify from "slugify";

// Function to upload image to Sanity with retry logic
async function uploadImageToSanity(imageUrl) {
  try {
    const response = await axios.get(imageUrl, { responseType:
"arraybuffer", timeout: 10000 });
    const buffer = Buffer.from(response.data);

    const asset = await client.assets.upload("image", buffer, {
      filename: imageUrl.split("/").pop(),
    });

    console.log("✔ Image uploaded successfully:", asset);
    return asset._id;
  } catch (error) {
    console.error("✖ Failed to upload image:", imageUrl, error.message);
  }
}
```

Prepared By: Syed Abdullah Zaidi

Roll No: -

Slot: Tuesday 2pm – 5pm

```

    return null; // Return null if image upload fails
  }
}

// Function to create or fetch category in Sanity
async function createCategory(category, counter) {
  try {
    const slug = slugify(category.name, { lower: true, strict: true });
    const categoryExist = await client.fetch(`*[_type=="category" && slug.current==${slug}[0]]`, { slug });

    if (categoryExist) {
      console.log(`Category already exists: ${categoryExist.name}`);
      return categoryExist._id;
    }

    const categoryData = {
      _type: "category",
      _id: `${slug}-${counter}`,
      name: category.name,
      slug: {
        _type: "slug",
        current: slug,
      },
    };

    const response = await client.createOrReplace(categoryData);
    console.log("✔ Category created successfully:", response);
    return response._id;
  } catch (error) {
    console.error("✗ Failed to create category:", category.name, error.message);
    return null;
  }
}

// Function to import data into Sanity
async function importData() {
  try {
    console.log("Fetching products from external API...");
    const { data: products } = await axios.get("https://hackathon-apis.vercel.app/api/products");

    console.log(`Fetched ${products.length} products.`);
  }
}

```

Prepared By: Syed Abdullah Zaidi

Roll No: -

Slot: Tuesday 2pm – 5pm


```

let counter = 1;

// Iterate over products with batching for efficiency
for (const product of products) {
  try {
    console.log(`Processing product: ${product.name}`);

    // Upload image and fetch category references
    const imageRef = product.image ? await
uploadImageToSanity(product.image) : null;
    const categoryRef = product.category?.name ? await
createCategory(product.category, counter) : null;

    // Prepare product object for Sanity
    const sanityProduct = {
      _id: `product-${counter}`,
      _type: "product",
      name: product.name || "Unnamed Product",
      slug: {
        _type: "slug",
        current: slugify(product.name || `product-${counter}`, {
lower: true, strict: true })),
      },
      price: product.price || 0,
      category: categoryRef ? { _type: "reference", _ref: categoryRef
} : undefined,
      tags: product.tags || [],
      quantity: product.quantity || 50,
      image: imageRef
        ? {
          _type: "image",
          asset: { _type: "reference", _ref: imageRef },
        }
        : undefined,
      description: product.description || "No description provided.",
      features: product.features || ["Default feature 1", "Default
feature 2"],
      dimensions: product.dimensions || {
        _type: "dimensions",
        height: "Unknown",
        width: "Unknown",
        depth: "Unknown",
      },
    };
  }
}

```

Prepared By: Syed Abdullah Zaidi

Roll No: -

Slot: Tuesday 2pm – 5pm

```

        // Upload product to Sanity
        console.log("Uploading product:", sanityProduct.name);
        await client.createOrReplace(sanityProduct);

        console.log(`✔ Product uploaded successfully:
${sanityProduct.name}`);
        counter++;
    } catch (productError) {
        console.error(`✖ Failed to process product: ${product.name}`,
productError.message);
    }
}

console.log("✔ All products processed successfully!");
} catch (error) {
    console.error("✖ Error importing data:", error.message);
}
}

// Start the import process
importData();

```

Sanity Studio:

Sanity Studio was used to verify that the data was correctly migrated and populated. We could check the product documents and their fields to ensure that all relevant information was stored.

Verification:

After migration, the data was verified in the Sanity Studio interface, ensuring that all product data, including pricing, description and categories were correctly added.

Screenshots:

API Calls:

Screenshot of successful API calls were made to fetched product data.

Prepared By: Syed Abdullah Zaidi

Roll No: -

Slot: Tuesday 2pm – 5pm

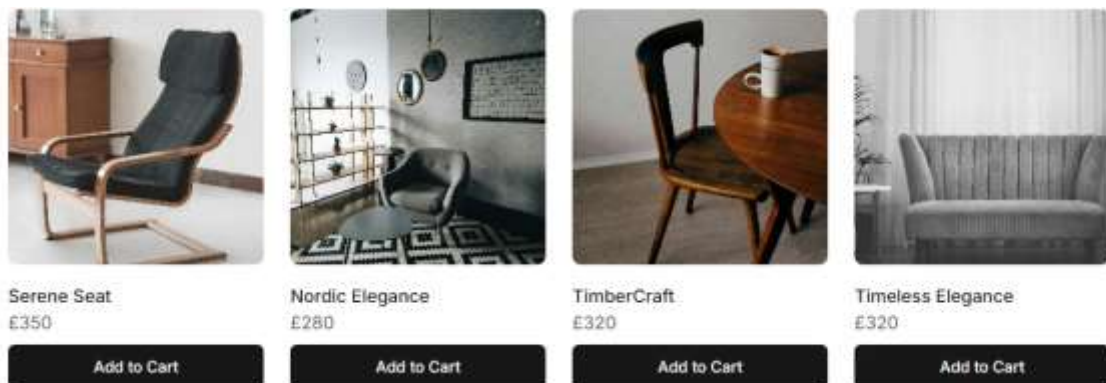
```
import { client } from '@lib/sanity'
import { NextResponse } from 'next/server'

export async function GET() {
  try {
    const products = await client.fetch(`
      *[_type == "product"] {
        _id,
        name,
        price,
        "image": image.asset->url,
        description
      }
    `)

    return NextResponse.json(products)
  } catch (error) {
    console.error('Error fetching products:', error)
    return NextResponse.json({ error: 'Failed to fetch products' }, {
      status: 500 })
  }
}
```

Frontend Display:

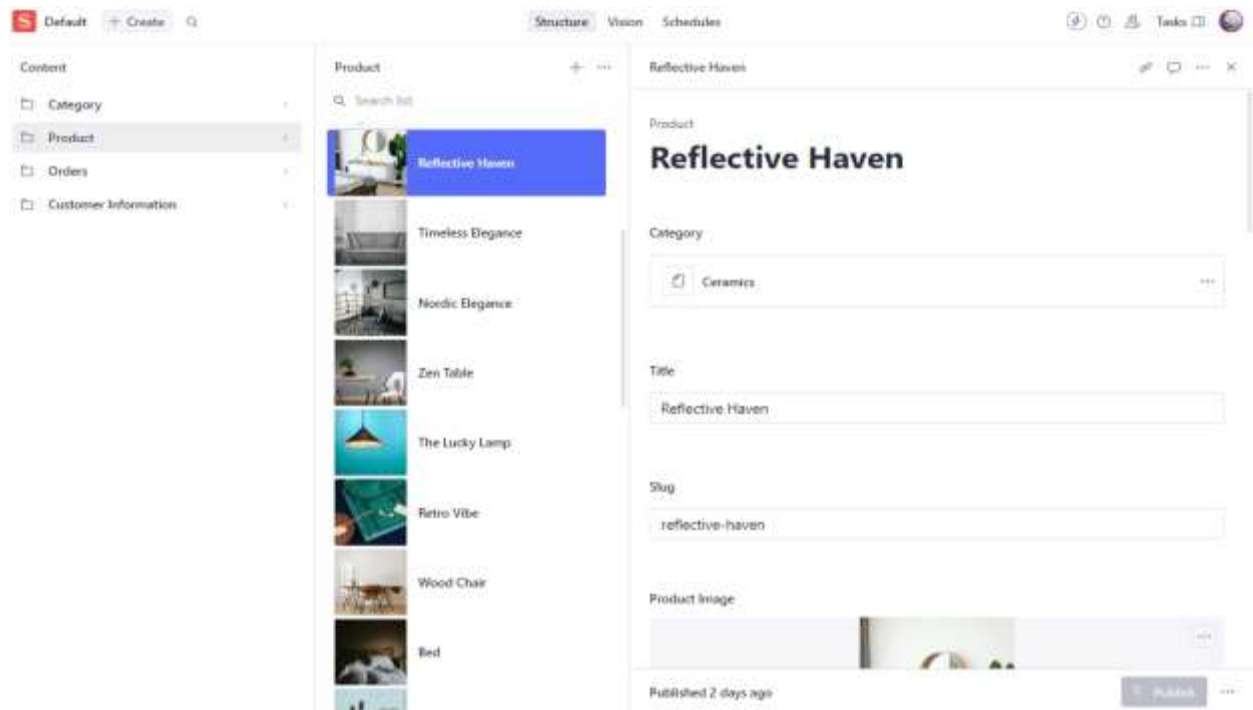
A screenshot showing the data displayed on the frontend (e.g: All Product page with information from the API).



Prepared By: Syed Abdullah Zaidi
Roll No: -
Slot: Tuesday 2pm – 5pm

Populated Sanity CMS Fields:

A screenshot of sanity studio showing the product document with populated fields, such as title, prices and description.



Conclusion:

The API integration process was successfully completed with the product data being fetched, stored and displayed on the frontend. Necessary adjustments were made to the sanity CMS schema to accommodate the new product information. Data migration was automated using a custom script, ensuring a smooth transition from the external API to the CMS.