

DAY-3

API INTEGRATION REPORT

Introduction:

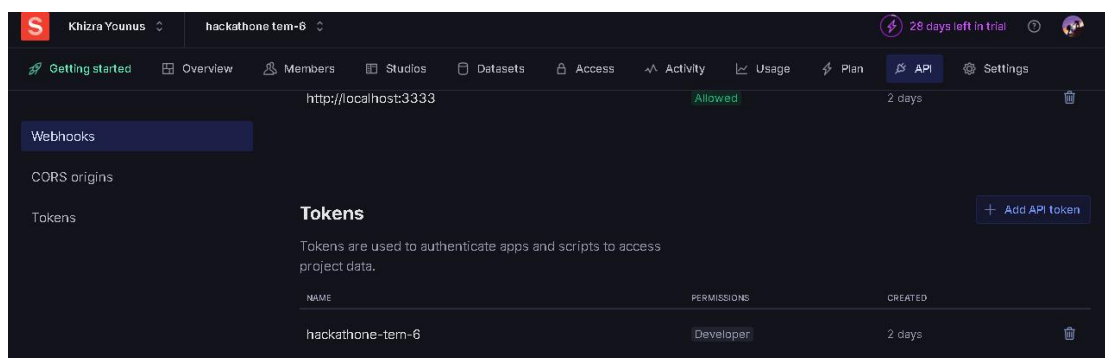
On Day-3 of the Technical Marketplace, we fetching data through API. The objective was fetch product data,including images from API and store it in a sanity CMS. It also includes adjustments made to Sanity's schema, migration steps and tools used.

Data Migration :

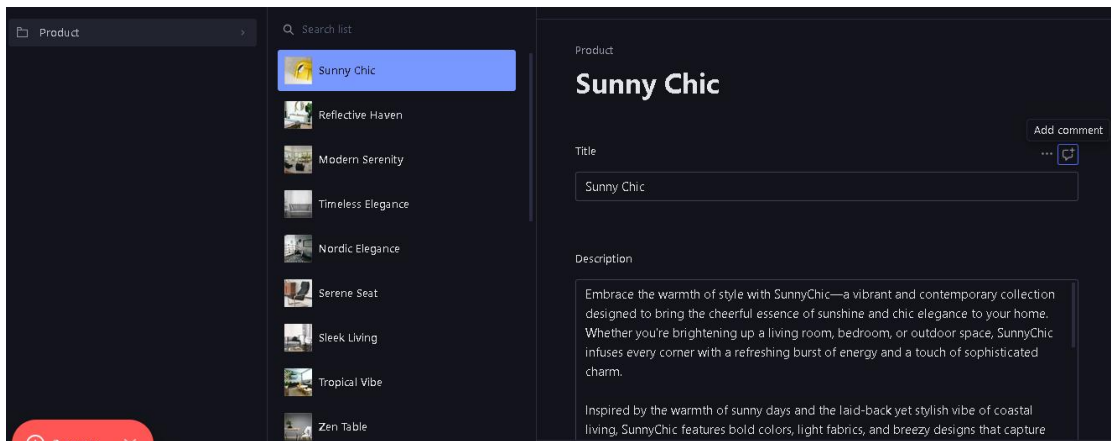
1. **Create a Sanity Project:** Initialize a new project using the Sanity Cli.
2. **Download and Set Up Sanity:** Ensure Sanity is Successfully installed in your project directly.
3. **Obtain Project Credentials:** ProjectID Retrive this from your Sanity dashboard.

API TOKEN:

Generate a token to authentication API request.

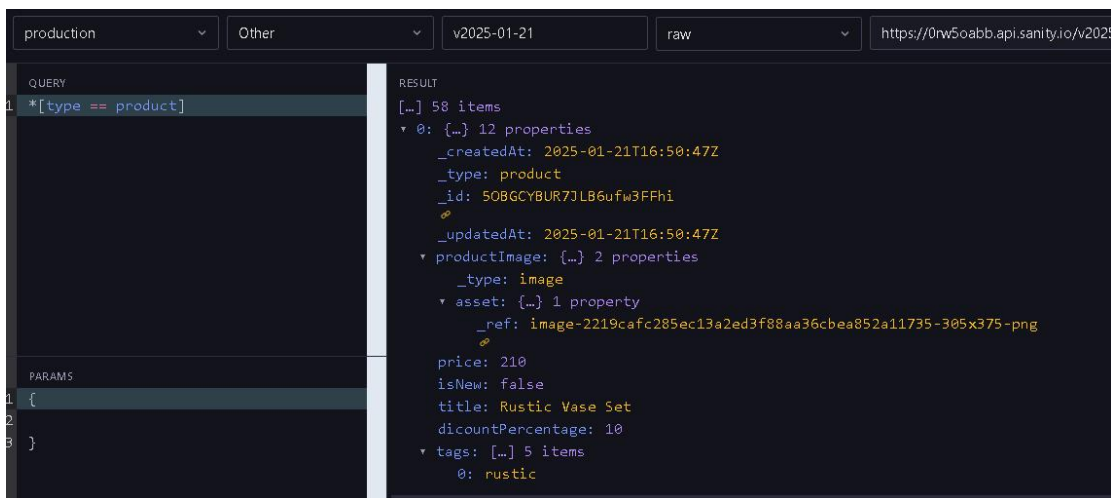


Studio/Products:



- Adjust the schema if additional fields or transformation are necessary for use case.

Vision:



Import Product Schema:

Define and add product Schema into your sanity project index file.

```

1  import { type SchemaTypeDefinition } from 'sanity'
2  import { product } from './product'
3
4  export const schema: { types: SchemaTypeDefinition[] } = {
5    types: [product],
6  }
7

```

Import Data into Sanity:

Use commands or tools provided by Sanity to import your data set.

```

template-6 > {} package.json
1  {
2    "name": "template-6",
3    "version": "0.1.0",
4    "private": true,
5    "type": "module",
6    "scripts": {
7      "dev": "next dev",
8      "build": "next build",
9      "start": "next start",
10     "lint": "next lint",
11     "import-data": "node script/importData.js"
12   },

```

Npm run import-data

```

Key Features:\n' +
'\n' +
'Bright, bold colors and breezy designs inspired by sunny, coastal living\n' +
'High-quality materials that are both durable and comfortable\n' +
'A versatile collection perfect for living rooms, bedrooms, or outdoor spaces\n' +
'Combines chic style with a relaxed, welcoming atmosphere\n' +
'Ideal for those looking to add a refreshing, cheerful touch to their home\n' +
'Fill your home with the light and energy of SunnyChic—where sunshine meets style for a lively
and sophisticated living experience. Make every day feel like a sunlit escape.',
dicountPercentage: 50,
isNew: false,
price: 400,
productImage: {
  _type: 'image',
  asset: {
    _ref: 'image-ed5874fb4498ee4ff897075bad504056cb87388c-4000x3907-jpg'
  }
},
tags: [ 'sunny ', 'chic ', 'modern ', 'elegant ', 'furniture' ],
title: 'Sunny Chic'
}

```

Validation:

Confirm that data has been successfully import by query the Sanity backend.

```
const fetchProducts = async () => {
  try {
    const query = `
      *[type == "product"] {
        _id,
        title,
        price,
        discription,
        disountPercentage,
        "imageUrl": productImage.assest->url,
        tags
      }
    `;

    const data = await sanity.fetch(query);
    setProducts(data);
  } catch (error) {
    console.error("Error Fetching products:", error);
  }
};
```

API Integration in Next.js:

Create Utility Function:

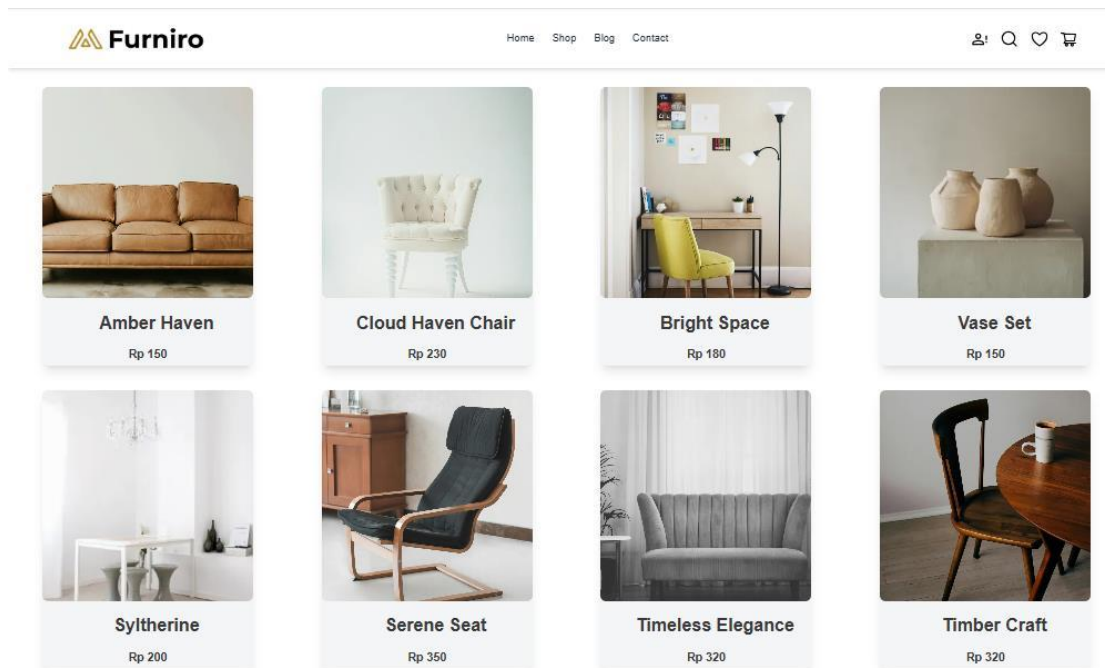
Abstract and centralize API calls for better maintainability and scalability.

```
template-6 > src > components > product > page.tsx > sanity > apiVersion

1  "use client"
2  import React, { useEffect, useState } from "react";
3  import sanityClient from "@sanity/client";
4  import Image from "next/image";
5
6  const sanity = sanityClient({
7    projectId: "0rw5oabb",
8    dataset: "production",
9    apiVersion: "2025-01-21",
10    useCdn: true,
11  });
12
13  interface Product {
14    _id: string;
15    title: string;
16    price: number;
17    description: string;
18    dicountPercentage: number;
19    imageUrl: string;
20    productImage: {
21      assest: {
22        _ref: string;
23      };
24    };
25    tags: string[];
26  }
```

Render Data in Website:

Fetch and render the data in your components.



Conclusion:

This report documents the API integration process, schema adjustments, migration steps, and tools used for importing product data into Sanity CMS. The project successfully integrates product data from an external API, handles image uploads, and render the data in a Next.js frontend.