

# Hackathon Day 3: Data Integration And Data Migration

## Objective:

1. Import product data from the provided API to **Sanity CMS**.
2. Automate the process of uploading products and images.
3. Build a frontend to display products fetched from **Sanity CMS**.

-----

## Steps Followed:

### 1. Sanity CMS Schema Validation:

- Used the schema provided in **Day 2**.

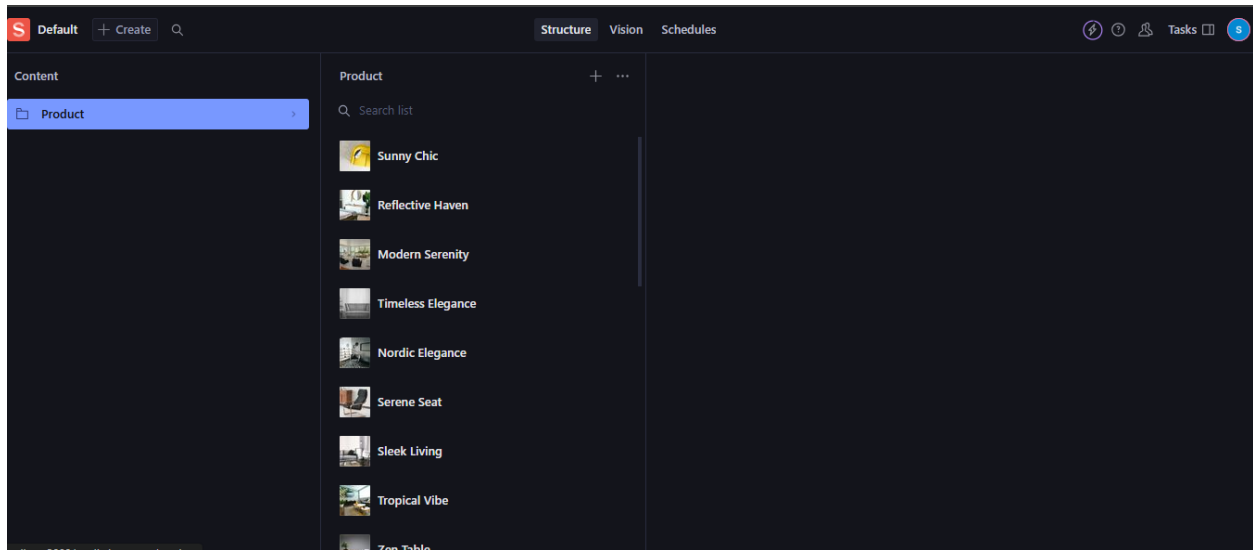
## Final Schema Code:

```
import { defineType } from "sanity"

export const product = defineType({
  name: "product",
  title: "Product",
  type: "document",
  fields: [
    {
      name: "title",
      title: "Title",
      validation: (rule) => rule.required(),
      type: "string"
    },
    {
      name: "description",
      type: "text",
      validation: (rule) => rule.required(),
      title: "Description",
    },
    {
      name: "productImage",
      type: "image",
    },
  ],
})
```

```
        validation: (rule) => rule.required(),
        title: "Product Image"
    },
    {
        name: "price",
        type: "number",
        validation: (rule) => rule.required(),
        title: "Price",
    },
    {
        name: "tags",
        type: "array",
        title: "Tags",
        of: [{ type: "string" }]
    },
    {
        name: "discountPercentage",
        type: "number",
        title: "Discount Percentage",
    },
    {
        name: "isNew",
        type: "boolean",
        title: "New Badge",
    }
]
}))
```

## Sanity CMS Schema Editor:



-----

## 2. API Data Import Implementation:

### Script Overview

A script was created to:

1. Fetch data from the provided API (`https://template6-six.vercel.app/api/products``).
2. Upload product images to **Sanity CMS**.
3. Create product documents in **Sanity CMS** using the fetched data.

### Key Code

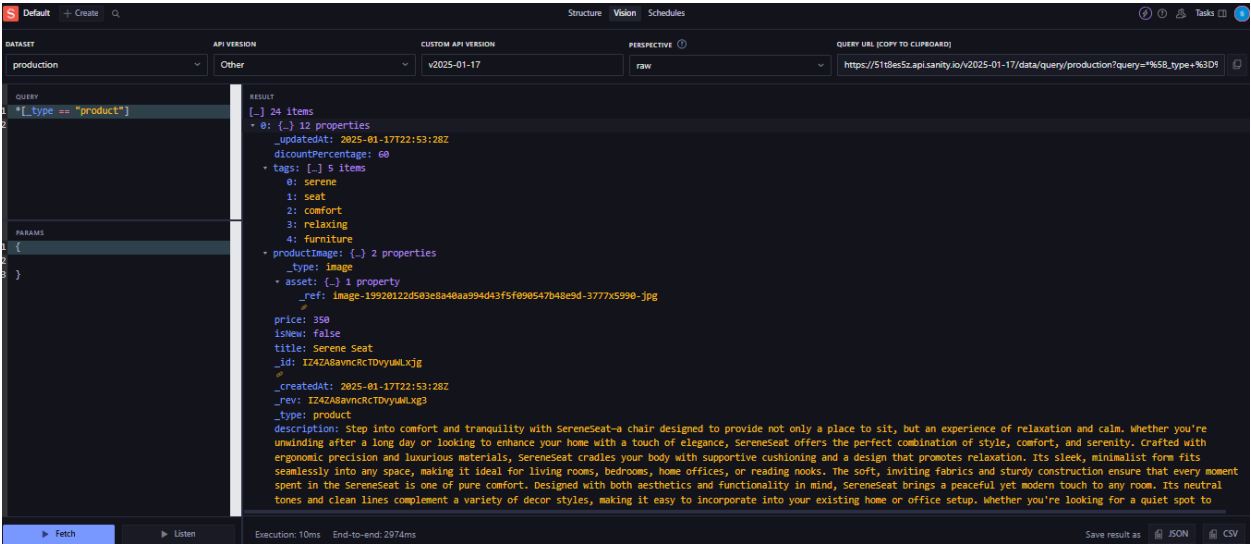
#### Sanity Client Setup

```
import { createClient } from 'next-sanity'

import { apiVersion, dataset, projectId } from '../env'

export const client = createClient({
  projectId ,
  dataset,
  apiVersion,
  useCdn: true, // Set to false if statically generating pages, using ISR or tag-
based revalidation
})
```

# Main Function to Import Products:



-----

## 3. Testing & Validation

### Steps

1. Executed the script to import product data into **Sanity CMS**.
2. Verified all products were successfully imported by checking the Sanity dashboard.
3. Cross-checked the data consistency between the API and Sanity CMS.

- **Sanity CMS Dashboard** showing imported products.
- **Console Output** showing successful upload logs.

```

D:\next\my-hackathon>npm run dev

> my-hackathon@0.1.0 dev
> next dev

▲ Next.js 14.2.2
- Local:      http://localhost:3000
- Environments: .env.local

/ Starting...
/ Ready in 6.5s
o Compiling / ...
/ Compiled / in 18.6s (515 modules)
/ Compiled in 3.4s (246 modules)
GET / 200 in 21830ms
o Compiling /studio/[...tool] ...
/ Compiled /product in 46.3s (6041 modules)
GET /product 200 in 38643ms

{
  tags: [ 'serene ', 'seat ', 'comfort ', 'relaxing ', 'furniture' ],
  discountPercentage: 60,
  title: 'Serene Seat',
  price: 350,
  description: "Step into comfort and tranquility with SereneSeat—a chair designed to provide not only a place to sit, but an experience of relaxation and calm. Whether you're unwinding after a long day or looking to enhance your home with a touch of elegance, SereneSeat offers the perfect combination of style, comfort, and serenity.\n" +
    '\n' +
    'Crafted with ergonomic precision and luxurious materials, SereneSeat cradles your body with supportive cushioning and a design that promotes relaxation. Its sleek, minimalist form fits seamlessly into any space, making it ideal for living rooms, bedrooms, home offices, or reading nooks. The soft, inviting fabrics and sturdy construction ensure that every moment spent in the SereneSeat is one of pure comfort.\n' +
    '\n' +
    "Designed with both aesthetics and functionality in mind, SereneSeat brings a peaceful yet modern touch to any room. Its neutral tones and clean lines complement a variety of decor styles, making it easy to incorporate into your existing home or office setup. Whether you're looking for a quiet spot to read, meditate, or simply relax, the SereneSeat is your go-to choice for a serene and stylish seating experience.\n" +
    '\n' +
    'Key Features:\n' +
    '\n' +
    'Ergonomic design for optimal comfort and support\n' +
    'High-quality materials for durability and lasting comfort\n' +
    'Sleek, minimalist style that complements various decor styles\n' +
    'Ideal for creating a peaceful and relaxing atmosphere\n' +
    'Perfect for use in living rooms, bedrooms, or offices\n' +
    'Transform your home into a sanctuary of relaxation and style with SereneSeat—where comfort meets elegance in every seat.',
  productImage: { asset: [Object] }
},
{
  price: 280,
  description: 'Elevate your space with Nordic Elegance—a collection that brings the minimalist beauty and understated sophistication of Scandinavian design into your home. Inspired by the serene landscapes and functional aesthetics of Nordic regions, this collection offers timeless pieces that blend clean lines, natural materials, and a refined simplicity to create a peaceful and inviting atmosphere.\n' +
    '\n' +
    'Crafted with precision and attention to detail, Nordic Elegance celebrates the beauty of light, airy spaces, and the harmony between form and function. Each piece is designed to be both practical and visually appealing, combining functionality with refined, minimalist elegance. Whether you are outfitting your living room, bedroom, or home office, the Nordic Elegance collection creates a sophisticated yet relaxed vibe that suits any modern interior.\n' +
    '\n' +
    'With natural wood finishes, neutral tones, and soft textures, Nordic Elegance fosters a sense of calm and tranquility, making it the perfect choice for those who value simplicity and quality. Ideal for creating a serene, clutter-free space, this collection is a true reflection of Scandinavian design principles: clean, functional, and beautifully crafted.\n' +
    '\n' +
    'Key Features:\n' +

```

=====

## 4. Frontend Integration

### Frontend Setup

- Used **React/Next.js** to build the frontend.
- Fetched products from **Sanity CMS** using GROQ queries.
- Displayed product listings on the homepage.

### GROQ Query for Fetching Products

```

async function getdata() {
  const fetchdata = client.fetch(`*[_type == "product" ]{
    title,

```

```

    price,
    description,
    productImage{
      asset->{
        _id,
        url
      }
    },
    tags,
    dicountPercentage
  }
`
)
return fetchdata;
}

const ProductDescription = ({ description }: { description: string }) => {
  const [isExpanded, setIsExpanded] = useState(false);

  const toggleDescription = () => {
    setIsExpanded((prev) => !prev); // Toggles the expanded state
  };

  return (
    <p className="text-xs mb-4 text-gray-600">
      {isExpanded ? description : `${description.slice(0, 100)}...`}
      <span
        onClick={toggleDescription}
        className="text-blue-500 cursor-pointer ml-1"
      >
        {isExpanded ? "Show Less" : "Read More"}
      </span>
    </p>
  );
};

async function ProductPage() {

  const data = await getdata()
  console.log(data);
}

```

```

// Slice the first 4 products
const firstSectionProducts = data.slice(0, 4);

const secondSectionProducts = data.slice(4, 8);

const thirdSectionProducts = data.slice(8, 12);

const fourthSectionProducts = data.slice(12, 18);

return (
  <div className="bg-gray-100 min-h-screen">

    {/* Product Section */}
    <main className="p-8">
      <h1 className="text-2xl font-bold mb-6">All Products</h1>
      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-6">
        {firstSectionProducts.map((val:any, i:any) => (

          <div key={i} className="rounded-lg shadow-md bg-white p-4">
            <div className="relative h-64 mb-4">
              <img
                src={val.productImage.asset.url}
                alt={val.title}
                className="h-full w-full object-cover rounded-lg"
              />
              {val.tags && val.tags[0] && (
                <span
                  className={`absolute top-2 left-2 px-2 py-1 text-xs font-medium
text-white rounded ${
                    val.tags[0].toLowerCase() === 'new'
                      ? 'bg-green-500'
                      : 'bg-red-500'
                  }`}
                >
                  {val.tags[0]}
                </span>
              )}
            </div>
            <p className="text-sm font-semibold mb-2">{val.title}</p>

            <ProductDescription description={val.description} />

```

```

        <div className="flex justify-between items-center">
          <p className="text-lg font-bold">${val.price}</p>
          <FaShoppingCart className="text-blue-500 text-2xl cursor-
pointer" />

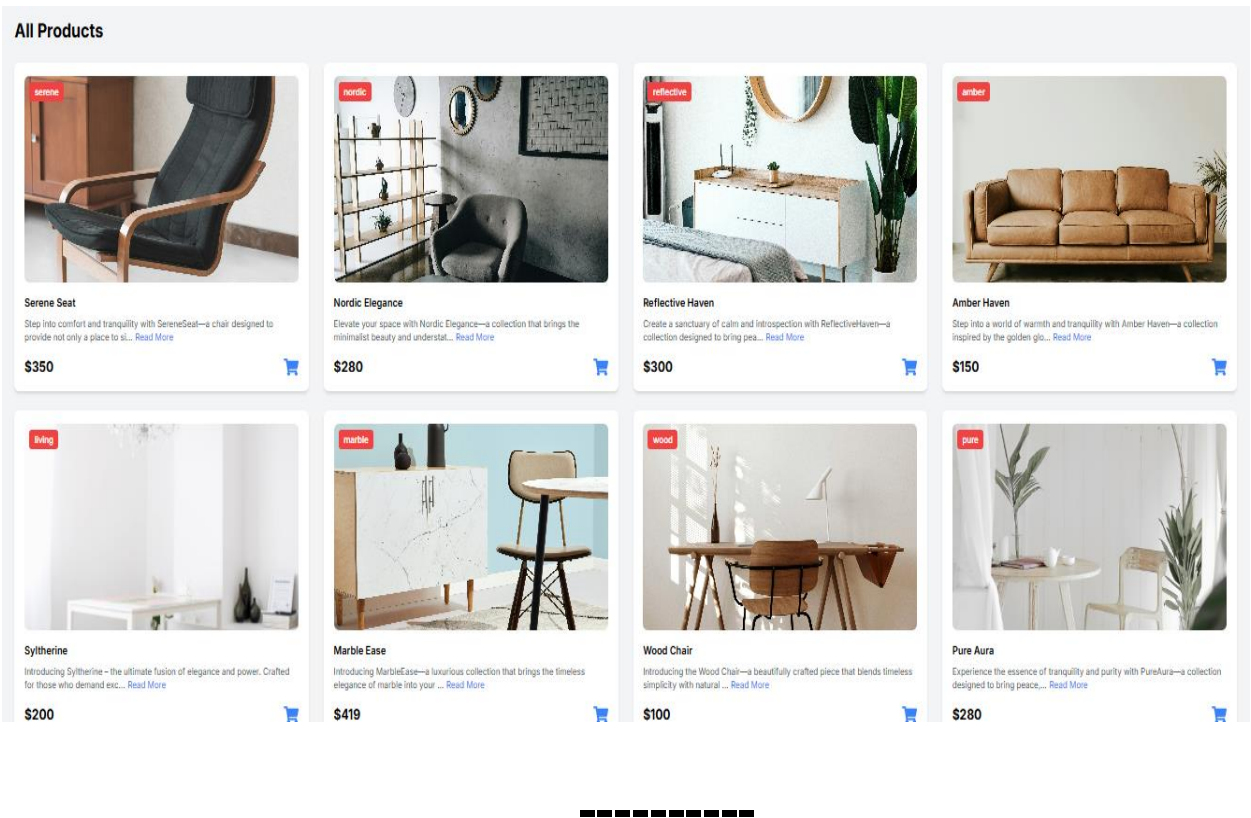
        </div>
      </div>

    )))
  </div>
</main>

```

5. Deliverables

- 1. Sanity CMS Dashboard showing imported products.
- 2. Console log confirming successful import.
- 3. Frontend product listing page.





## **Conclusion:**

### **Successfully completed the following:**

- Imported API data into Sanity CMS.
- Built a functional frontend to display products fetched from Sanity.
- Validated the entire workflow, ensuring seamless integration between API, CMS, and frontend.

-----