

Day 3 - API Integration Report – Furniro

Objectives:

Day 03 focused on API integration and data migration using Sanity CMS as the backend. The backend schemas were updated to meet project needs, ensuring smooth integration. All API data was successfully moved to Sanity CMS and used on the frontend for dynamic and efficient content display. This step created a strong connection between the backend and frontend, making content management easier and more scalable.

API Integration

To integrate the provided API, I will fetch product information from the endpoint `https://template6-six.vercel.app/api/products`

The API provides the following product details:

- **Product Image**
- **Title**
- **Description**
- **Price**
- **Discounted Percentage**
- **IsNew**
- **Tags**

Schema

Schemas define the content structure in Sanity CMS. Key fields include:

- **Image:** Product image.
- **Title:** Product name.
- **Description:** Product details.
- **Price:** Product cost.
- **Discount:** Discount percentage.
- **New:** Indicates if the product is new.

Schema Code:

```
1
2 import { defineType } from "sanity";
3
4 export const product = defineType({
5   name: "product",
6   title: "Product",
7   type: "document",
8   fields: [
9     {
10       name: "title",
11       title: "Title",
12       type: "string",
13       validation: (rule) => rule.required()
14     },
15     {
16       name: "description",
17       title: "Description",
18       type: "text",
19       validation: (rule) => rule.required()
20     },
21     {
22       name: "productImage",
23       type: "image",
24       validation: (rule) => rule.required(),
25       title: "Product Image"
26     },
27     {
28       name: "price",
29       title: "Price",
30       type: "number",
31       validation: (rule) => rule.required()
32     },
33     {
34       name: "tags",
35       title: "Tags",
36       type: "array",
37       of: [{ type: "string" }]
38     },
39     {
40       name: "dicountPercentage",
41       type: "number",
42       title: "Discount Percentage",
43     },
44     {
45       name: "isNew",
46       title: "New Badge",
47       type: "boolean"
48     },
49   ],
50 }
51 ];
52 });
53
```

Migration Steps and Tools

- **Setup Environment:**

- Load environment variables using `dotenv`.

- **Create Sanity Client:**

- Use `@sanity/client` to create a Sanity client.

- **Upload Image:**

- Fetch images using `fetch` and upload them using `uploadImageToSanity()`.

- **Import Data:**

- Fetch product data with `axios`.
- Upload product images and create Sanity documents using `createClient()`.

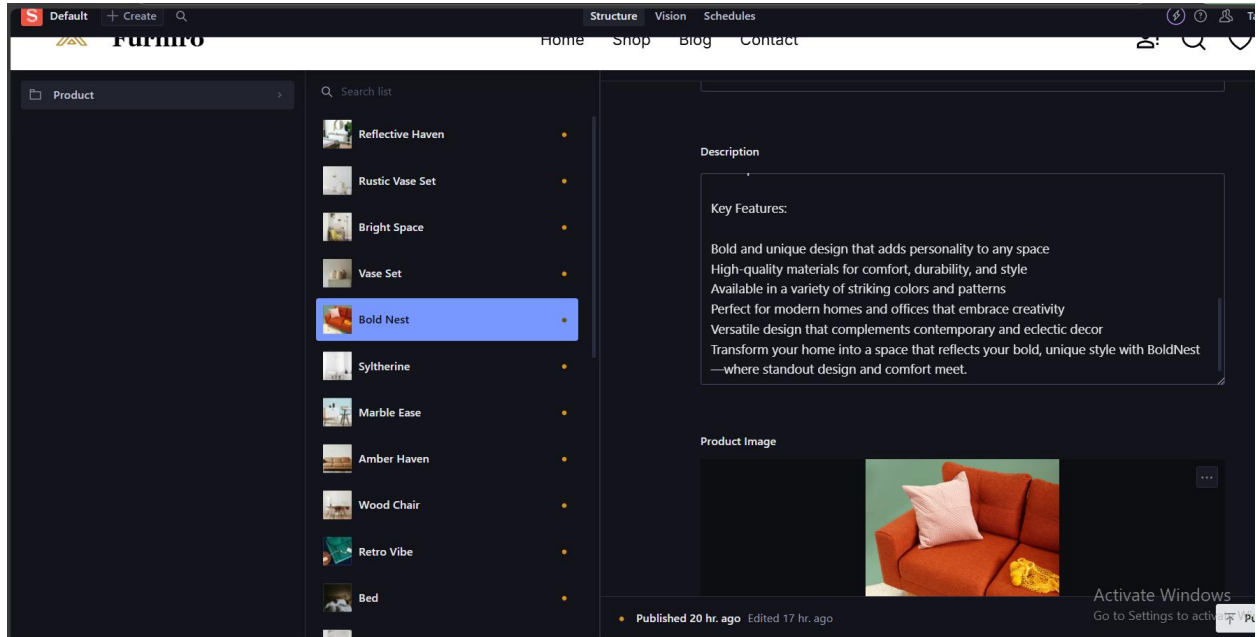
- **Execute:**

- Run `importData()` to migrate product data to Sanity.

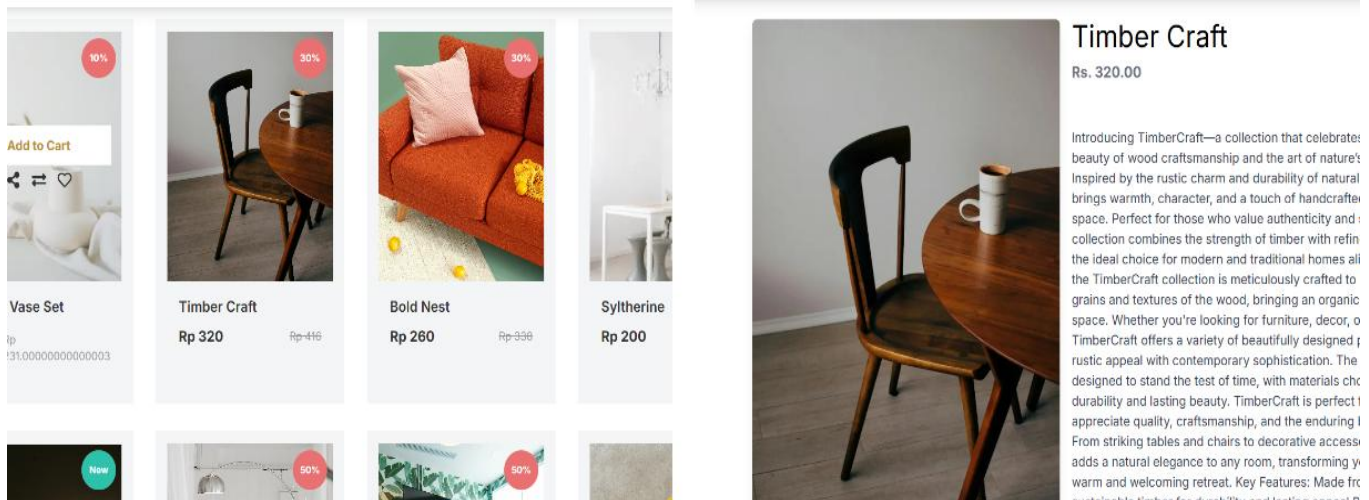
Migration Code:

```
1 import { createClient } from '@sanity/client';
2 import axios from 'axios';
3 import dotenv from 'dotenv';
4 import path from 'path';
5 import { fileURLToPath } from 'url';
6 const __filename = fileURLToPath(import.meta.url);
7 const __dirname = path.dirname(__filename);
8 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
9 const client = createClient({
10   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
11   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
12   useCdn: false,
13   token: process.env.SANITY_API_TOKEN,
14   apiVersion: '2021-08-31',
15 });
16 async function uploadImageToSanity(imageUrl) {
17   try {
18     console.log('Uploading image: ${imageUrl}');
19     const response = await fetch(imageUrl);
20     if (!response.ok) {
21       throw new Error('Failed to fetch image: ${imageUrl}');
22     }
23     const buffer = await response.arrayBuffer();
24     const bufferImage = Buffer.from(buffer);
25     const asset = await client.assets.upload('image', bufferImage, {
26       filename: imageUrl.split('/').pop(),
27     });
28     console.log('Image uploaded successfully: ${asset._id}');
29     return asset._id;
30   } catch (error) {
31     console.error('Failed to upload image:', imageUrl, error);
32     return null;
33   }
34 }
35 async function importData() {
36   try {
37     const response = await axios.get('https://template6-six.vercel.app/api/products');
38     if (response.status === 200) {
39       const products = response.data;
40       console.log('Fetched ${products.length} products');
41       if (products.length > 0) {
42         for (const product of products) {
43           console.log('Processing product: ${product.title}');
44           let imageRef = null;
45           if (product.image) {
46             imageRef = await uploadImageToSanity(product.image);
47           }
48           const imageId = await uploadImageToSanity(product.imageUrl);
49           if (imageId) {
50             const sanityProduct = {
51               _type: 'product',
52               title: product.title,
53               description: product.description,
54               productImage: {
55                 _type: 'image',
56                 asset: {
57                   _ref: imageId,
58                 },
59               },
60               tags: product.tags,
61               discountPercentage: product.discountPercentage,
62               price: product.price,
63               isNew: product.isNew || false,
64               slug: {
65                 _type: 'slug',
66                 current: product.title.toLowerCase().replace(/ /g, '-'),
67               },
68             };
69             console.log('Uploading product to Sanity:', sanityProduct.title);
70             const result = await client.create(sanityProduct);
71             console.log('Product uploaded successfully: ${result._id}');
72           }
73         }
74       }
75     }
76   } catch (error) {
77     console.error('Error importing data:', error);
78   }
79 }
```

Populated Sanity CMS fields:



Data successfully displayed in the frontend:



Conclusion:

The migration for Furniro was successful, improving scalability and dynamic content display. API data was successfully moved to Sanity and displayed seamlessly on the frontend, ensuring an efficient experience.