

Hackathon : 03

Day 3 - API Integration Report – Oak&Teak

Objective

The goal of Day 3 is to focus on:

1. API integration with the Next.js project.
2. Migrating JSON data into Sanity CMS.
3. Validating schemas to ensure compatibility.
4. Displaying API data in the frontend.

1.API Integration

I developed custom APIs for migrating JSON data into Sanity CMS. The endpoints created for this purpose are:

Custom API Endpoints:

1. **Products API:** <https://hackathon-3-mynew.vercel.app/api/products>
 - Endpoint to fetch product data.
 - Fields include:
 - id: Unique identifier.
 - name: Product name.
 - price: Price of the product.
 - category: Category of the product.
 - description: Product description.

2. Categories API: <https://hackathon-3-mynew.vercel.app/api/categories>

- Endpoint to fetch category data.
- Fields include:
 - id: Unique identifier.
 - title: Category title.
 - slug: URL-friendly identifier.

2. Schema Validation

I reviewed and modified the Sanity CMS schema to ensure compatibility with the API data structure.

Category Schema:

```
export default {
  name: 'category',
  title: 'Category',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Category Name',
      type: 'string',
      description: 'Name of the category (e.g., Chairs, Sofas)',
    },
    {
      name: 'slug',
      title: 'Slug',
      type: 'slug',
      description: 'Slug generated from category name',
      options: {
        source: 'name',
        maxLength: 96,
      },
    },
  ],
}
```

Product Schema:

```
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'product_id',
      title: 'Product ID',
      type: 'string',
    },
    {
      name: 'productName',
      title: 'Product Name',
      type: 'string',
    },
    {
      name: 'slug', type: 'slug', title: 'Slug', options: {
        source: 'productName',
        maxLength: 96,
        slugify: (input:any) => input
          .toLowerCase()
          .replace(/\/s+/g, '-')
          .replace(/[\^\\w\~]+/g, '')
          .replace(/\/\~\~+/g, '-')
          .replace(/\/\~+/g, '')
          .replace(/\/-+$/g, '')
      }},
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'reference',
      to: [{ type: 'category' }],
    },
    {
      name: 'tags',
      title: 'Tags',
      type: 'array',
      of: [{ type: 'string' }],
    },
    {
      name: 'discountPercentage',
      title: 'Discount Percentage',
      type: 'number',
    },
    {
      name: 'colors',
      title: 'Colors',
      type: 'array',
      of: [{ type: 'string' }],
    },
    {
      name: 'sizes',
      title: 'Sizes',
      type: 'array',
      of: [{ type: 'string' }],
    },
  ],
}
```

3.Data Migration Steps:

- Used dotenv to manage environment variables for Sanity configuration
- Fetched category data from a custom API endpoint (/api/categories) using axios.

```
import { createClient } from "@sanity/client";
import axios from "axios";
import dotenv from "dotenv";
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.local") });

const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: "2025-01-17",
});

async function getCategoryMap() {
  try {
    const response = await axios.get(
      "https://hackathon-3-mynew.vercel.app/api/categories"
    );
    const categories = response.data.data;
    const categoryMap = {};

    // Check for each category if it exists in Sanity
    for (const category of categories) {
      // Check if category exists in Sanity
      const existingCategory = await client.fetch(
        `*[_type == "category" && name == $name][0]`,
        { name: category.name }
      );

      if (existingCategory) {
        categoryMap[category.name] = existingCategory._id;
      } else {
        // Create category if it doesn't exist
        const newCategory = await client.create({
          _type: "category",
          name: category.name,
          slug: {
            _type: "slug",
            current: category.name.toLowerCase().replace(/\s+/g, "-"),
          },
        });
        categoryMap[category.name] = newCategory._id;
      }
    }

    return categoryMap;
  } catch (error) {
    console.error("Error fetching categories:", error);
    return {};
  }
}
```

```

async function importData() {
  try {
    const response = await axios.get(
      "https://hackathon-3-mynew.vercel.app/api/products"
    );
    const products = response.data.data;

    for (const product of products) {
      let imageRefs = [];
      if (product.images && product.images.length > 0) {
        for (const imageUrl of product.images) {
          const imageRef = await uploadImageToSanity(imageUrl);
          if (imageRef) {
            imageRefs.push({
              _key: `${product.product_id}-${imageUrl}-${Date.now()}`,
              _type: "image",
              asset: {
                _type: "reference",
                _ref: imageRef,
              },
            });
          }
        }
      }

      const sanityProduct = {
        _type: "product",
        product_id: product.product_id,
        productName: product.name,
        description: product.description,
        price: product.price,
        tags: product.tags || [],
        discountPercentage: product.discountPercentage || 0,
        colors: Array.isArray(product.color)
          ? product.color
          : product.color
            ? [product.color]
            : [],
        sizes: product.sizes || [],
        inventory: product.stock || 0,
        material: product.material,
        dimensions: product.dimensions,
        weight: product.weight,
        images: imageRefs.length > 0 ? imageRefs : undefined,
      };

      await client.create(sanityProduct);
    }

    console.log("Data migrated successfully!");
  } catch (error) {
    console.error("Error migrating data: ", error);
  }
}

```

Tools Used:

1. Next.js:

- Used for building the frontend and handling the API integration.

2. Sanity CMS:

- A headless CMS used to store and manage content (products and categories in this case).
- **Sanity Schema:** Custom schemas for the product and category data, validated and structured to ensure compatibility with the API data.

3. Axios:

- A popular HTTP client for making API requests to fetch product and category data from your custom API endpoints.

4. dotenv:

- A library used to manage environment variables (e.g., your Sanity project credentials or other configuration values).

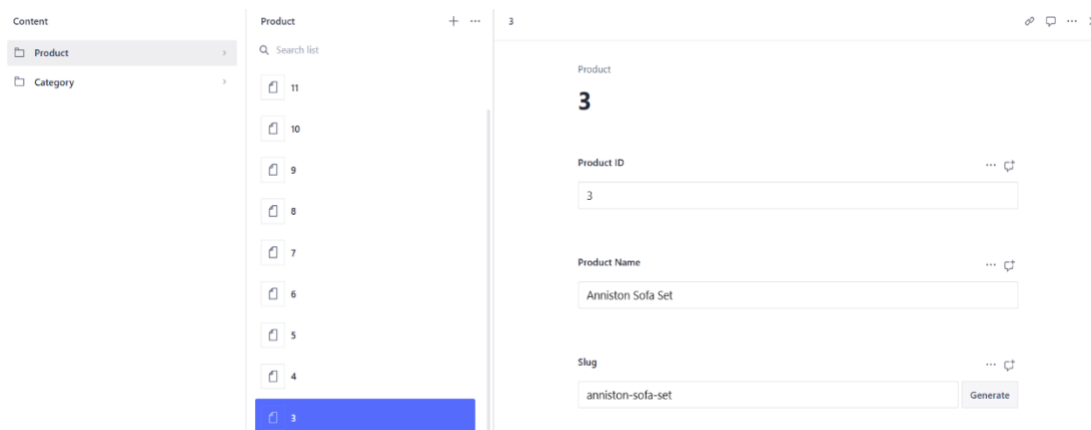
5. Sanity Client:

- A Node.js client provided by Sanity to interact with the Sanity CMS backend, used to create and manage documents in Sanity.

6. Console Logging:

- For debugging and tracking the status of data migration, helping to log success or failure during data fetching and insertion.

Data Displayed on Frontend and Populated in Sanity CMS



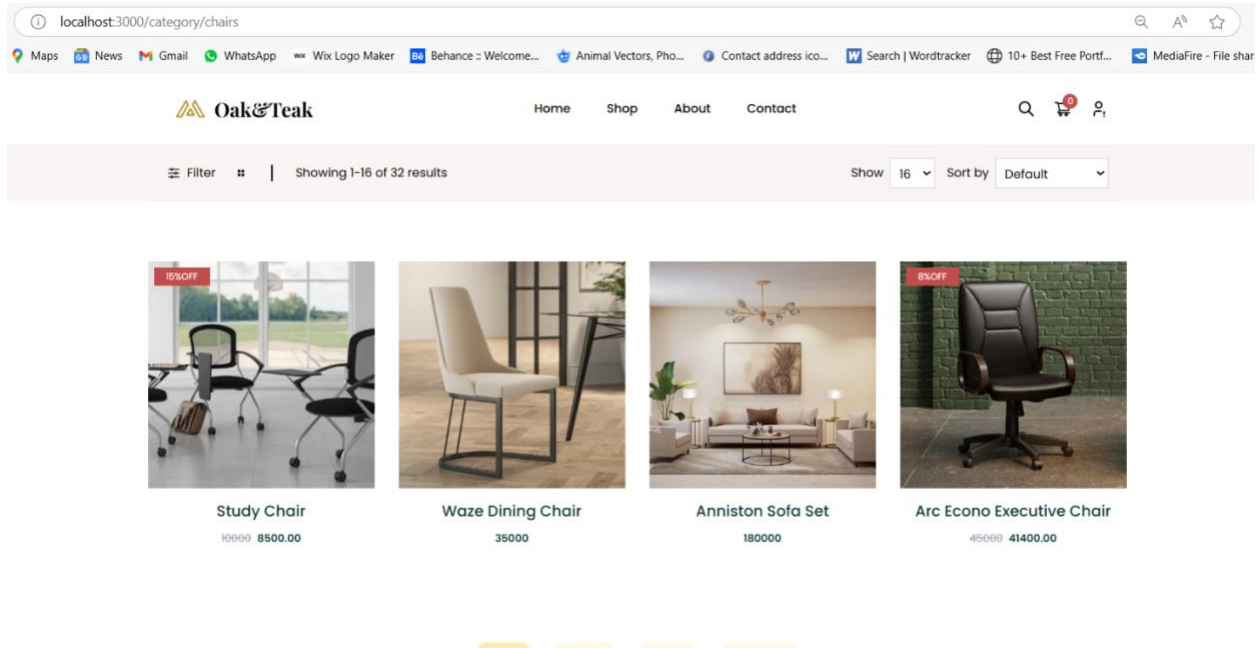
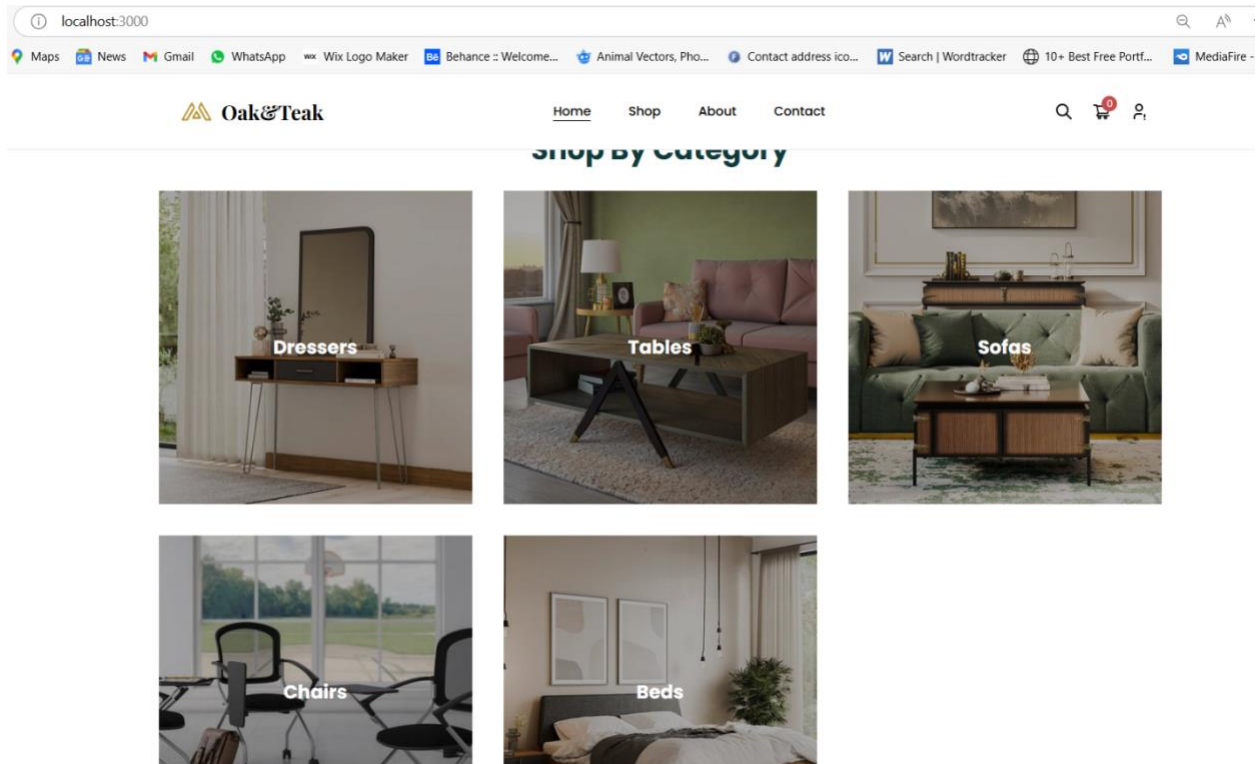


Image uploaded successfully: image-7bab58a108f8d7f1c60e17107043d5c48c36cc7b-160x160-jpg
Uploading image: https://interwood.pk/cdn/shop/products/fg050000482-f677122-interwood-mobel_580x_crop_center.jpg?v=1737118571
Image uploaded successfully: image-3f83139c80cb4f58edfc56ab7499a02a51a5120a-580x435-jpg
Uploading image: https://interwood.pk/cdn/shop/products/folding-chair---fg050000482-6caba53-interwood-mobel_580x_crop_center.jpg
Image uploaded successfully: image-6210e96cd1e7a8ef422d67bbf3c5a28aa93c6d8e-580x435-jpg
Uploading image: https://interwood.pk/cdn/shop/files/Anniston_6c5deb66-bd7a-4352-9083-cbcd3ef14f16_580x_crop_center.jpg?v=1737118571
Image uploaded successfully: image-87d3b56edec261d1eaca82ff7119b1d5c0841ecf-580x441-jpg
Uploading image: <https://hackathon-3-mynew.vercel.app/Anniston-Sofa-Set-new.png>
Image uploaded successfully: image-ea63cf601ad3086b3ddb9c8aa12b7eb46e2f8a26-574x434-png
Uploading image: https://interwood.pk/cdn/shop/files/Anniston-closeup_580x_crop_center.jpg?v=1737118001
Image uploaded successfully: image-227413071160dcac27743e3382ced91ffa719cb6-580x441-jpg
Uploading image: https://interwood.pk/cdn/shop/files/SofaAnniston3Seater_FabricWaffleBeige_2_4c25c89d-4008-47d0-93d7-7e749dd5aecd-580x_crop_center.jpg?v=1737118001
01
Image uploaded successfully: image-12136bb5236900578a9b6947cd34924b4d0f47d3-580x441-jpg
Uploading image: https://interwood.pk/cdn/shop/files/Sydney_Bed_Set_F_1_b520efa3-6c94-4ff1-b2e5-bb92701af1e9_580x_crop_center.jpg
Image uploaded successfully: image-53ef0bddb340578f99f2175e74bcf43eea02dc37-580x441-jpg
Uploading image: https://interwood.pk/cdn/shop/files/Untitleddesign_9_580x_crop_center.jpg?v=1737118049
Image uploaded successfully: image-231474fe82fd3b4e970825829f3ce1cde758aff6-580x441-jpg
Uploading image: https://interwood.pk/cdn/shop/files/Untitleddesign_10_580x_crop_center.jpg?v=1737118049
Image uploaded successfully: image-9ded24b4cd302830b39356300001471b118d09a9-580x441-jpg
Uploading image: https://interwood.pk/cdn/shop/files/WhatsApp-Image-2024-1_580x_crop_center.jpg?v=1737118050