Day 3 - API Integration and Data Migration

Objective

The goal was to integrate APIs into a **Next.js** project, set up **Sanity CMS**, and migrate data into Sanity CMS to create a functional marketplace backend. This task involved installing necessary tools, configuring the environment, writing schemas, and migrating data using scripts.

Steps Followed

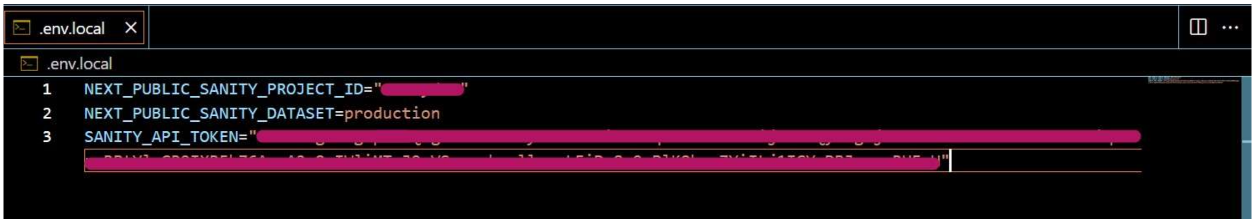
1. Installing Next.js and Sanity CMS

- Initialized a new **Next.js** project.
- Installed **Sanity CMS** and selected the eCommerce template during setup.
- Configured the project ID and dataset in the setup process.

---

2. Setting Up Environment Variables

- Created a `.env.local` file to securely store environment variables, such as the Sanity project ID, dataset name, and API token.



```
NEXT_PUBLIC_SANITY_PROJECT_ID="[redacted]"
NEXT_PUBLIC_SANITY_DATASET=production
SANITY_API_TOKEN="[redacted]"
```

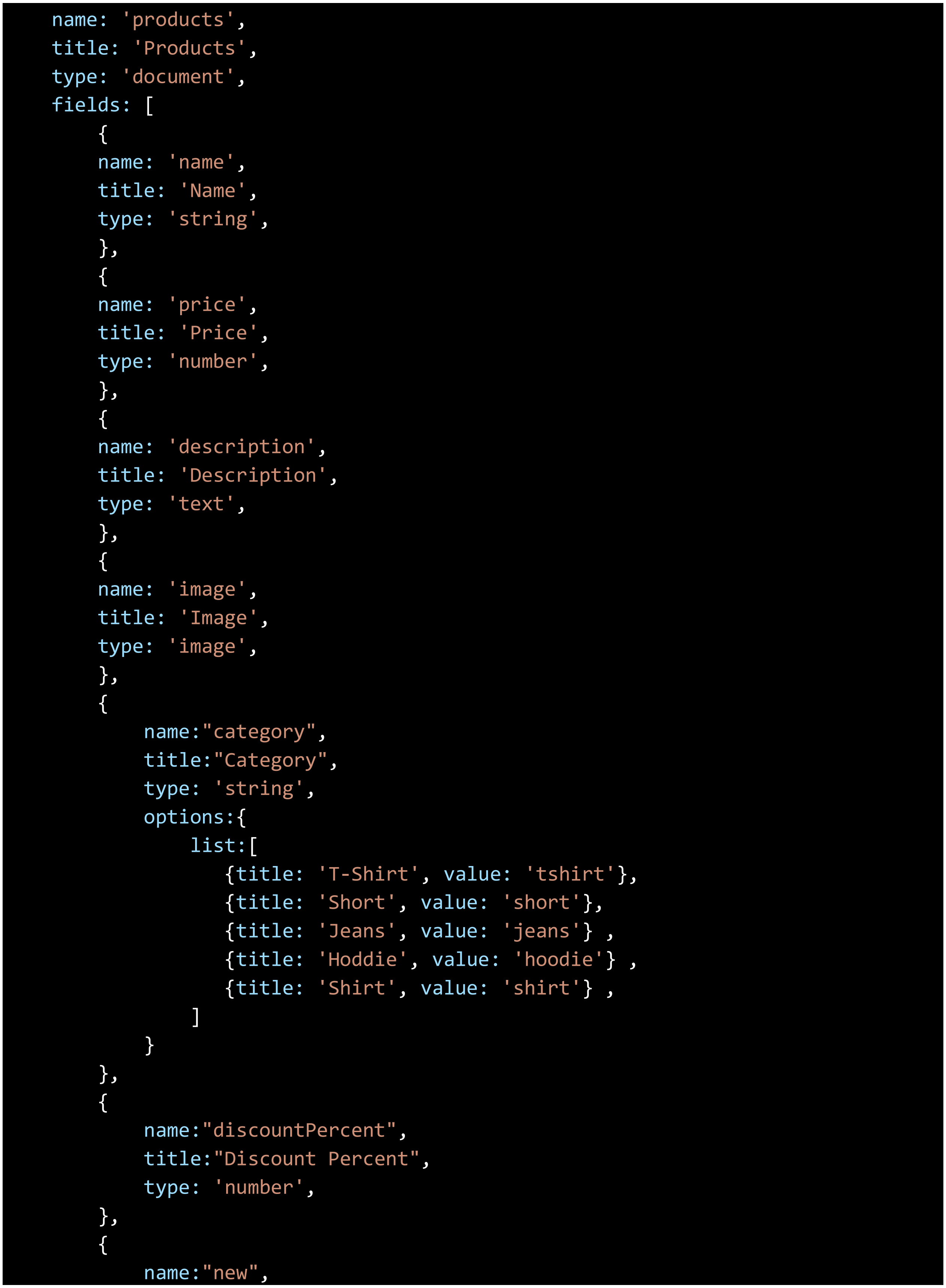
---

3. Writing Sanity Schema

- Defined schemas for products and categories in the project.
- Ensured that the fields matched the data structure from the API.

```
import { defineType } from "sanity"

export default defineType({
```



```
    name: 'products',
    title: 'Products',
    type: 'document',
    fields: [
        {
        name: 'name',
        title: 'Name',
        type: 'string',
        },
        {
        name: 'price',
        title: 'Price',
        type: 'number',
        },
        {
        name: 'description',
        title: 'Description',
        type: 'text',
        },
        {
        name: 'image',
        title: 'Image',
        type: 'image',
        },
        {
            name:"category",
            title:"Category",
            type: 'string',
            options:{
                list:[
                   {title: 'T-Shirt', value: 'tshirt'},
                   {title: 'Short', value: 'short'},
                   {title: 'Jeans', value: 'jeans'} ,
                   {title: 'Hoddie', value: 'hoodie'} ,
                   {title: 'Shirt', value: 'shirt'} ,
                ]
            }
        },
        {
            name:"discountPercent",
            title:"Discount Percent",
            type: 'number',
        },
        {
            name:"new",
            type: 'boolean',
            title:"New",
        },
        {
            name:"colors",
            title:"Colors",
            type: 'array',
            of:[
                {type: 'string'}
            ]
        },
        {
            name:"sizes",
            title:"Sizes",
            type: 'array',
            of:[
                {type: 'string'}
            ]
        }
    ],
})
```

---

4. Writing the Migration Script

- Wrote a script to fetch data from the provided API and migrate it into Sanity CMS.
- Mapped API fields to schema fields and handled data transformation where needed.

```
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

// Load environment variables from .env.local
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

// Create Sanity client
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: '2021-08-31',
});

// Function to upload an image to Sanity
async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);

    // Validate the image URL
    if (!imageUrl || typeof imageUrl !== 'string') {
      console.error('Invalid image URL:', imageUrl);
      return null;
    }

    // Fetch the image as a buffer
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);

    // Upload the image to Sanity
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop(),
    });

    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error.message || error);
    return null;
  }
}

// Function to import data
async function importData() {
  try {
    console.log('Fetching products from API...');
    const response = await axios.get('https://template1-neon-nu.vercel.app/api/products');
    const products = response.data;

    if (!Array.isArray(products)) {
      throw new Error('Invalid data format: Expected an array of products');
    }

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

    for (const product of products) {
      console.log(`Processing product: ${product.name}`);

      let imageRef = null;
      if (product.imageUrl) { // If image URL is provided
        console.log(`Uploading image for product: ${product.name}`);
        imageRef = await uploadImageToSanity(product.imageUrl);

        if (!imageRef) {
          console.warn(`Failed to upload image for product: ${product.name}`);
        }
      }

      const sanityProduct = {
        _type: 'products',
        name: product.name,
        price: product.price,
        description: product.description,
        image: imageRef
          ? {
              _type: 'image',
              asset: {
                _type: 'reference',
                _ref: imageRef,
              },
            }
          : undefined,
        category: product.category,
        discountPercent: product.discountPercent || 0,
        new: product.isNew || false,
        colors: product.colors || [],
        sizes: product.sizes || [],
      };

      console.log('Uploading product to Sanity:', sanityProduct.name);
      const result = await client.create(sanityProduct);
      console.log(`Product uploaded successfully: ${result._id}`);
    }

    console.log('Data import completed successfully!');
  } catch (error) {
    console.error('Error importing data:', error.message || error);
  }
}

importData();

```

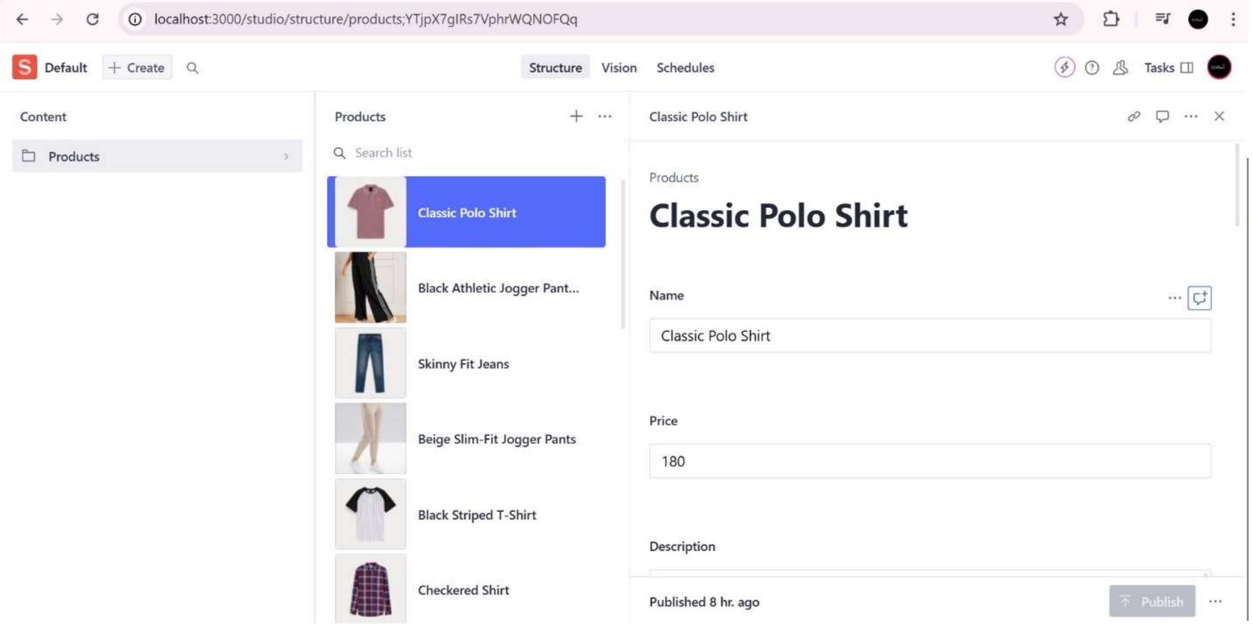
5. Importing Data into Sanity CMS

- Executed the migration script to upload data into Sanity CMS.
- By run this command `npm run import-data`
- Confirmed that the data was imported successfully.

6. Testing and Verifying the Data

- Verified the imported data in **Sanity Studio** to ensure all fields were populated correctly.
- Displayed the data on the **Next.js** frontend to test the integration.

Sanity Studio showing the imported data.



Classic Polo Shirt

Name

Classic Polo Shirt

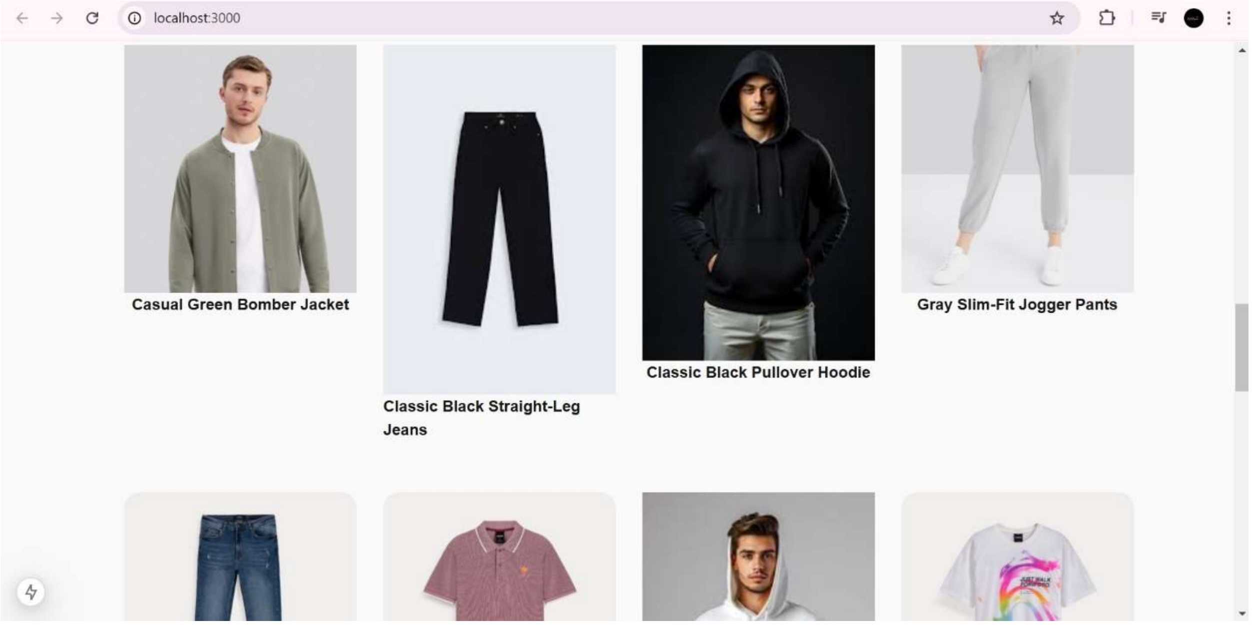
Price

180

Description

Published 8 hr. ago

Next.js frontend displaying the integrated data.


localhost:3000
Casual Green Bomber Jacket
Classic Black Straight-Leg Jeans
Classic Black Pullover Hoodie
Gray Slim-Fit Jogger Pants

Conclusion

This task successfully covered:

1. Setting up **Next.js** and **Sanity CMS**.
2. Defining schemas for data organization.
3. Migrating external API data into Sanity CMS.
4. Verifying and displaying the data in both the CMS and frontend.

By following these steps, a functional backend was created for a marketplace project, demonstrating a practical application of modern web technologies.