GIAIC

HACKATHON 3 DAY 3 (Nextjs)

API Integration Report - [NIKE]

Objective:

The aim of Day 3 is to:

1. Integrate APIs into the Next.js project.
2. Migrate product data and related content into Sanity CMS.
3. Ensure compatibility by validating the schemas.
4. Display data fetched from APIs on the frontend dynamically.

1.API Integration

Product API:

I used API for my project, which was given by my teacher, to fetch product data and integrate it into my application. Each product contains the necessary details such as the name, price, description, image, and inventory, making it suitable for the dynamic nature of my e-commerce site.

API: https://template-03-api.vercel.app/api/products

This endpoint provides data for products, including:

- **id:** Unique identifier for each product.
- **name:** Name of the product.
- **price:** Price of the product.
- **description:** Product description.

- **image:** URL of the product image.
- **inventory:** Available inventory quantity.
- **size:** Available sizes for the product (if applicable).

2.Schema Validation:

Explored the Sanity CMS to ensure compatibility with the Product API and adjusted the schema as needed.

- **Product Schema**:
  This is the Product Schema for the API.

```
export const productSchema = {
    name: 'product',
    title: 'Product',
    type: 'document',
    fields: [
      {
        name: 'productName',
        title: 'Product Name',
        type: 'string',
      },
      {
        name: 'category',
        title: 'Category',
        type: 'string',
      },
      {
        name: 'slug',
        type: 'slug',
        title: 'Slug',
        options: {
          source: 'productName',
          maxLength: 96,
          slugify: (input: any) =>
            input
              .toLowerCase()
              .replace(/[^\w\s-]+/g, '')
              .replace(/\s+/g, '-')
              .replace(/--+/g, '-')
              .replace(/^-+|-+$/g, ''),
        },
      },
      {
        name: 'tags',
        type: 'array',
        title: 'Tags',
        of: [{ type: 'string' }],
      },
      {
        name: 'price',
        title: 'Price',
        type: 'number',
      },
      {
        name: 'inventory',
        title: 'Inventory',
        type: 'number'
```

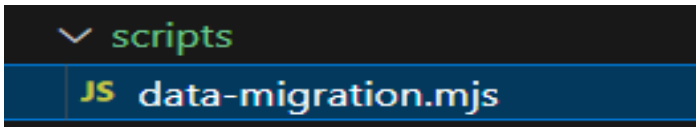
3.Data Migration Steps:

I proceeded with migrating the product data from the API to Sanity CMS, ensuring the proper mapping of fields and accurate data insertion.

```javascript
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",
});

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType: "arraybuffer" });
    const buffer = Buffer.from(response.data);
    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);
    return null;
  }
}

async function importData() {
  try {
    console.log("migrating data please wait...");

    // API endpoint containing car data
    const response = await axios.get(
      "https://template-03-api.vercel.app/api/products"
    );
    const products = response.data.data;
```

```javascript
async function importData() {
  try {
    console.log("migrating data please wait...");

    // API endpoint containing car data
    const response = await axios.get(
      "https://template-03-api.vercel.app/api/products"
    );
    const products = response.data.data;
    console.log("products ==>> ", products);

    for (const product of products) {
      let imageRef = null;
      if (product.image) {
        imageRef = await uploadImageToSanity(product.image);
      }

      const sanityProduct = {
        _type: "product",
        productName: product.productName,
        category: product.category,
        price: product.price,
        inventory: product.inventory,
        colors: product.colors || [], // Optional, as per your schema
        status: product.status,
        description: product.description,
        image: imageRef
          ? {
              _type: "image",
              asset: {
                _type: "reference",
                _ref: imageRef,
              },
            }
          : undefined,
      };

      await client.create(sanityProduct);
    }

    console.log("Data migrated successfully!");
  } catch (error) {
    console.error("Error in migrating data ==>> ", error);
  }
}
```

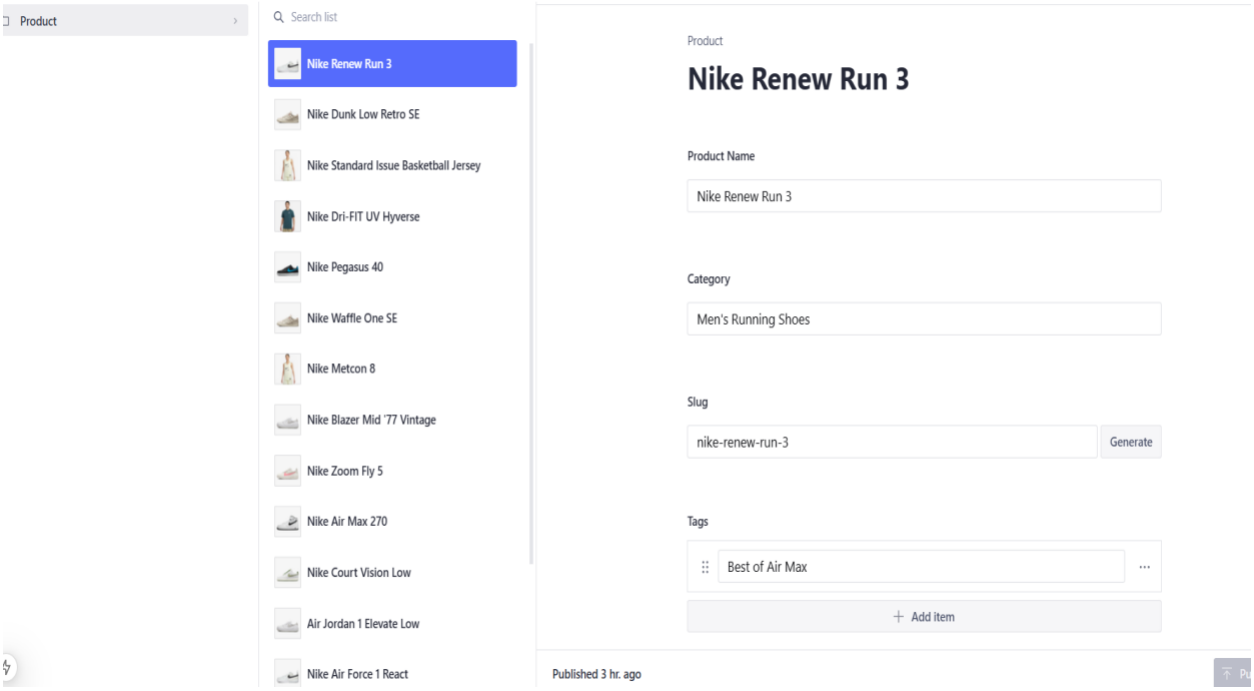
Run this Script to Migrate the data



Populated Sanity CMS fields:

The Sanity CMS schema fields match the product data, displaying details like name, price, description, image, inventory, and size correctly.

Sanity Fields:

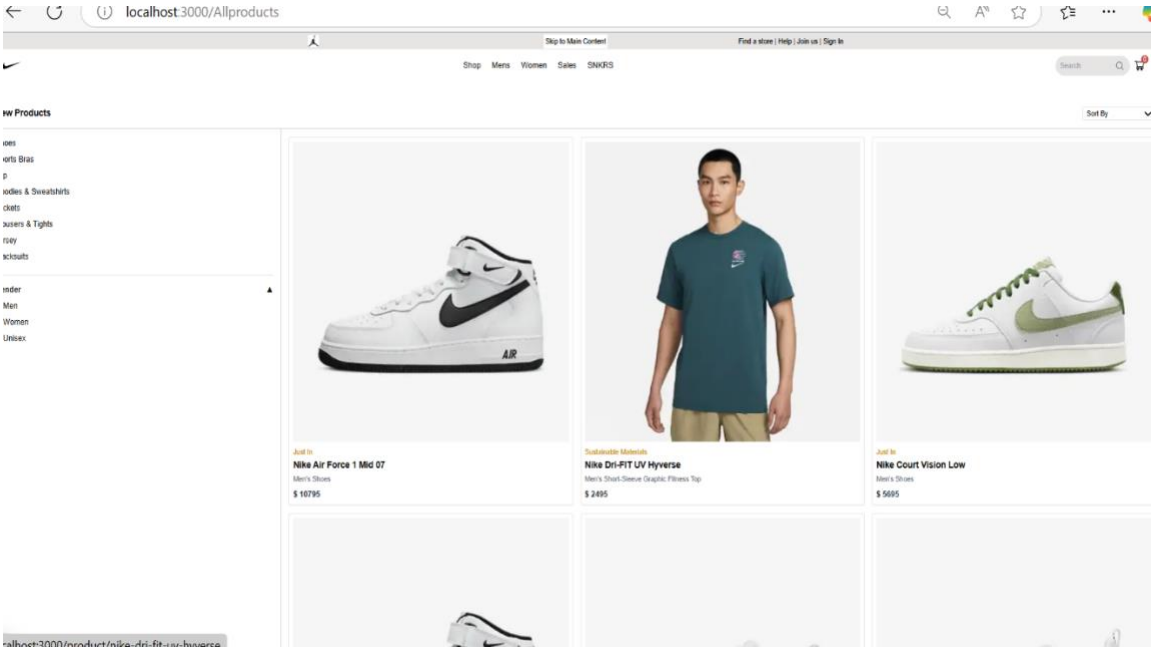


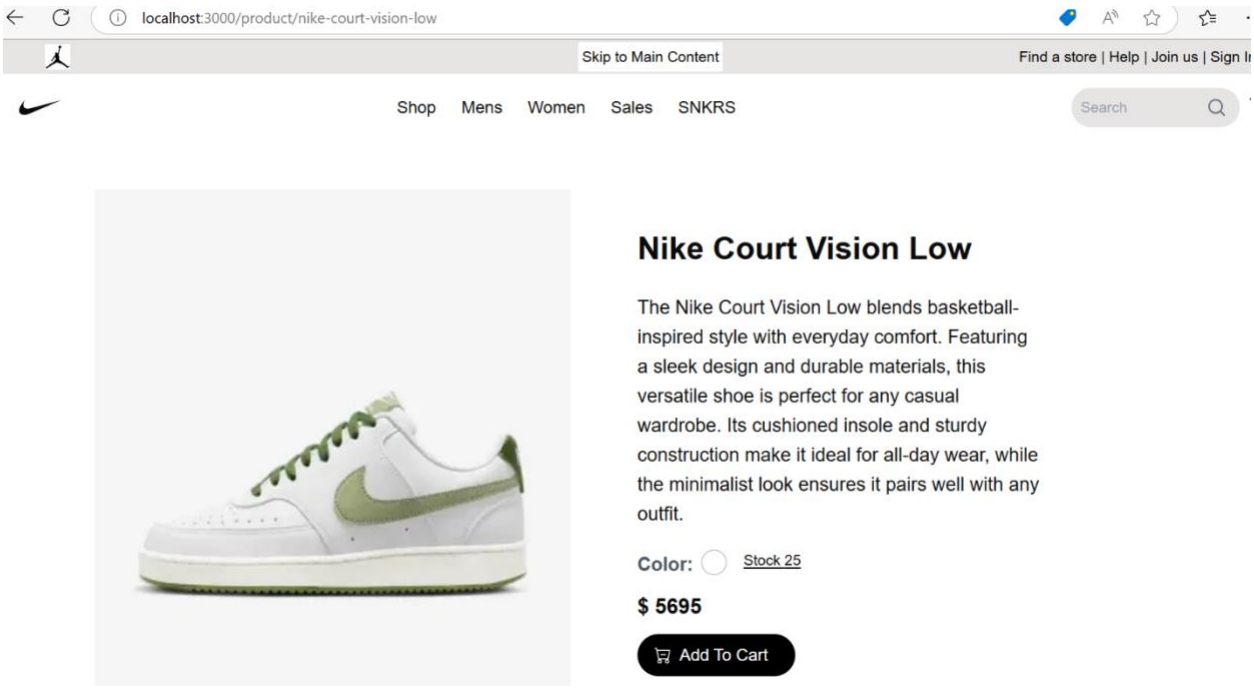
Tools Used:

I used the following tools for the API integration and data migration process:

- **Next.js**: For frontend development and API handling.
- **Sanity**: As the CMS for managing and storing product data.
- **dotenv**: For managing environment variables securely.
- **Axios**: For making HTTP requests to fetch product data from the API.

Data successfully displayed in the frontend:


localhost:3000/Allproducts
Skip to Main Content
Find a store | Help | Join us | Sign In
Shop Mens Women Sales SNKRS
Search
Sort By
Men
Women
Unisex
Nike Air Force 1 Mid 07
Men's Shoes
$ 10795
Nike Dri-FIT UV Hyverse
Men's Short-Sleeve Graphic Fitness Top
$ 2495
Nike Court Vision Low
Men's Shoes
$ 5695


localhost:3000/product/nike-court-vision-low
Skip to Main Content
Find a store | Help | Join us | Sign In
Shop Mens Women Sales SNKRS
Search
Nike Court Vision Low
The Nike Court Vision Low blends basketball-inspired style with everyday comfort. Featuring a sleek design and durable materials, this versatile shoe is perfect for any casual wardrobe. Its cushioned insole and sturdy construction make it ideal for all-day wear, while the minimalist look ensures it pairs well with any outfit.
Color:
Stock 25
$ 5695
Add To Cart