Day 3 - API Integration Report – Nike shoes

Introduction:

On Day 3, the goal was to integrate APIs and migrate data into Sanity CMS. This involved fetching product data from an API, importing it into Sanity CMS, and ensuring the schema matched the data structure. The final step was displaying the data dynamically on the frontend.

Key objective

- Fetching product from external API.
- Structuring and migrate data to sanity.
- Displaying the migrated data dynamically on the frontend.

API Integration:

- **Api:**

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

- **Data Fetch:**

  Product: Detail such as product title, price, description, category, status, images etc.

- **Sanity Client:**

  Used to manage and store fetched data in CMS.

Schema File:

```
sanity > TS products.ts > productSchema > fields
export const productSchema = {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'productName',
      title: 'Product Name',
      type: 'string',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'inventory',
      title: 'Inventory',
      type: 'number',
    },
    {
      name: 'colors',
      title: 'Colors',
      type: 'array',
```

Migration Steps:

Set Up Environment:

- Install required dependencies in sanity.
- Configured env. local for secure storage of API Keys.

Sanity Client Setup:

- Create a sanity client for using project Id, dataset & API Token.

```
sanity > lib > TS client.ts > ...
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
})
```

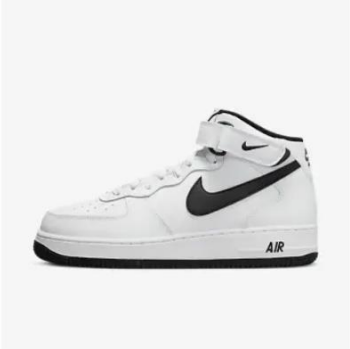
Fetch Data From API:

- Use fetch to retrieve shoes product data from API.

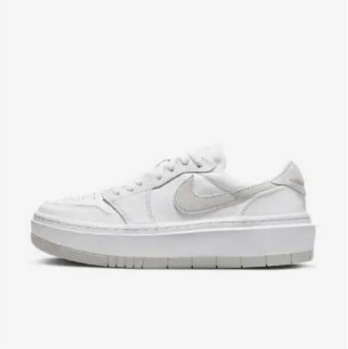
API Call Response:


```
  },
  {
    productName: "Nike Blazer Mid '77 Vintage",
    category: "Men's Shoes",
    price: 8495,
    inventory: 22,
    colors: [ 'White', 'Black' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/5.png',
    description: "The Nike Blazer Mid '77 Vintage brings old-school basketball style to modern fashion. With a hig
h-top design and premium leather, it delivers unmatched versatility and a timeless look."
  },
  {
    productName: 'Nike Metcon 8',
    category: "Men's Training Shoes",
    price: 10295,
    inventory: 25,
    colors: [ 'Black' ],
    status: 'Best Seller',
    image: 'https://template-03-api.vercel.app/products/6.png',
    description: "The Nike Metcon 8 is built for intense training sessions. Featuring a stable base, durable mater
ials, and excellent traction, it's the perfect companion for weightlifting, CrossFit, or HIIT workouts."
  },
  {
    productName: 'Nike Waffle One SE',
    category: "Women's Shoes",
    price: 7895,
    inventory: 30,
    colors: [ 'Pink' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/7.png',
    description: 'The Nike Waffle One SE updates a retro classic with fresh details. Its lightweight mesh, suede a
```

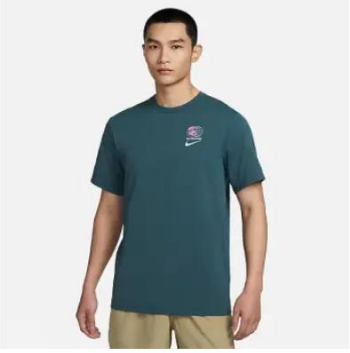
Data Fetch in Frontend:



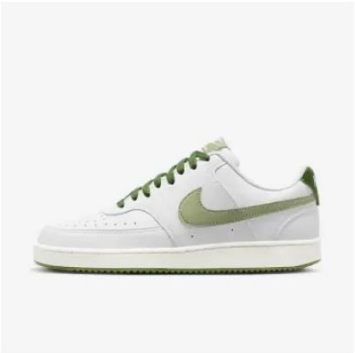
Just In
Men's Shoes
White
$10795



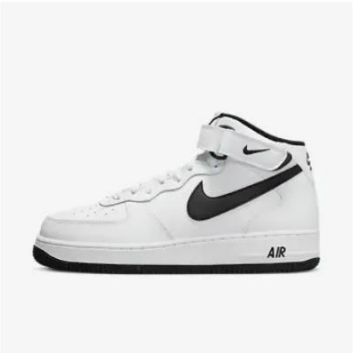
Promo Exclusion
Women's Shoes
White
$11895



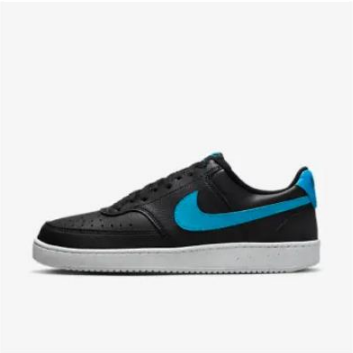
Sustainable Materials
Men's Short-Sleeve Graphic Fitness Top
Teal
$2495



Just In
Men's Shoes
White
$5695

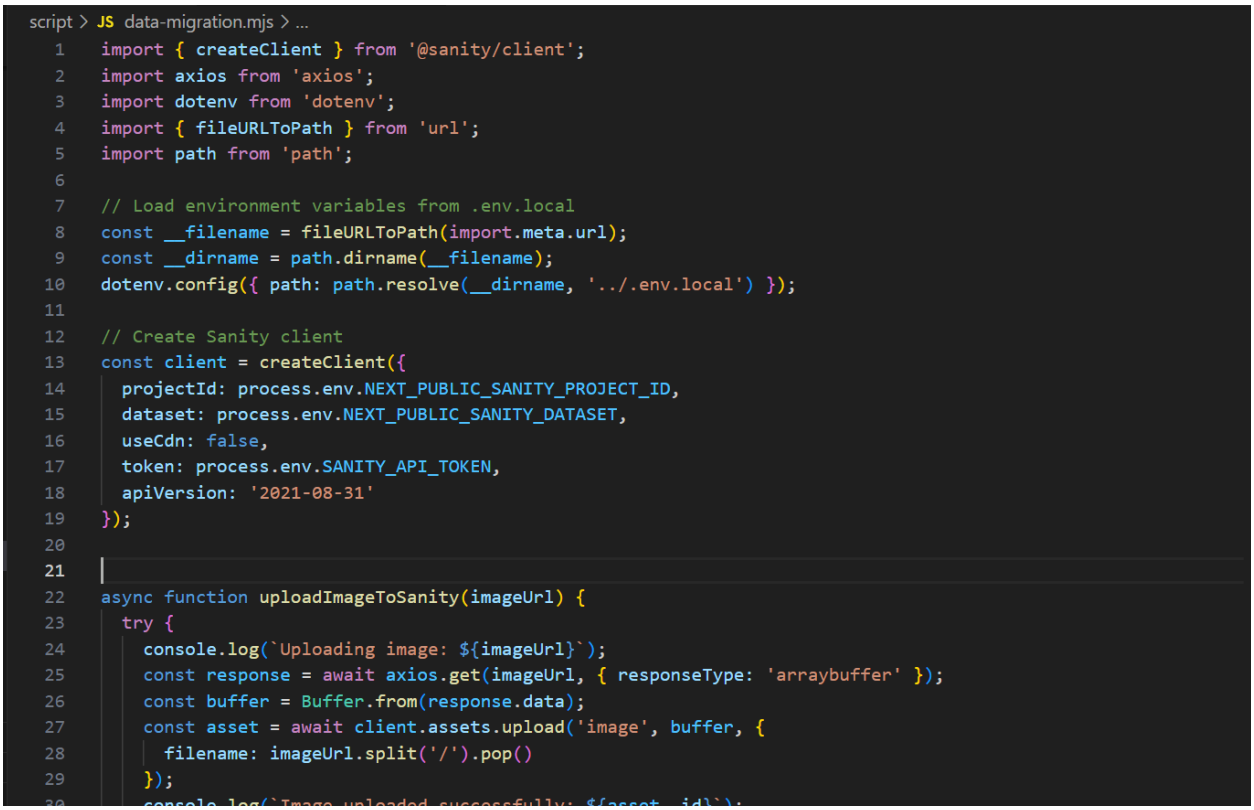


Trending
Men's Shoes
Black
$13295



Best Seller
Men's Running Shoes
Gray
$9795

Migration Script:



```
script > JS data-migration.mjs > ...
1  import { createClient } from '@sanity/client';
2  import axios from 'axios';
3  import dotenv from 'dotenv';
4  import { fileURLToPath } from 'url';
5  import path from 'path';
6
7  // Load environment variables from .env.local
8  const __filename = fileURLToPath(import.meta.url);
9  const __dirname = path.dirname(__filename);
10 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12 // Create Sanity client
13 const client = createClient({
14   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
15   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
16   useCdn: false,
17   token: process.env.SANITY_API_TOKEN,
18   apiVersion: '2021-08-31'
19 });
20
21
22 async function uploadImageToSanity(imageUrl) {
23   try {
24     console.log(`Uploading image: ${imageUrl}`);
25     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
26     const buffer = Buffer.from(response.data);
27     const asset = await client.assets.upload('image', buffer, {
28       filename: imageUrl.split('/').pop()
29     });
```

Final Check List:

API Understanding	Schema Validation	Data Migration	Api Integration in Next.js	Submission Preparation
✓	✓	✓	✓	✓