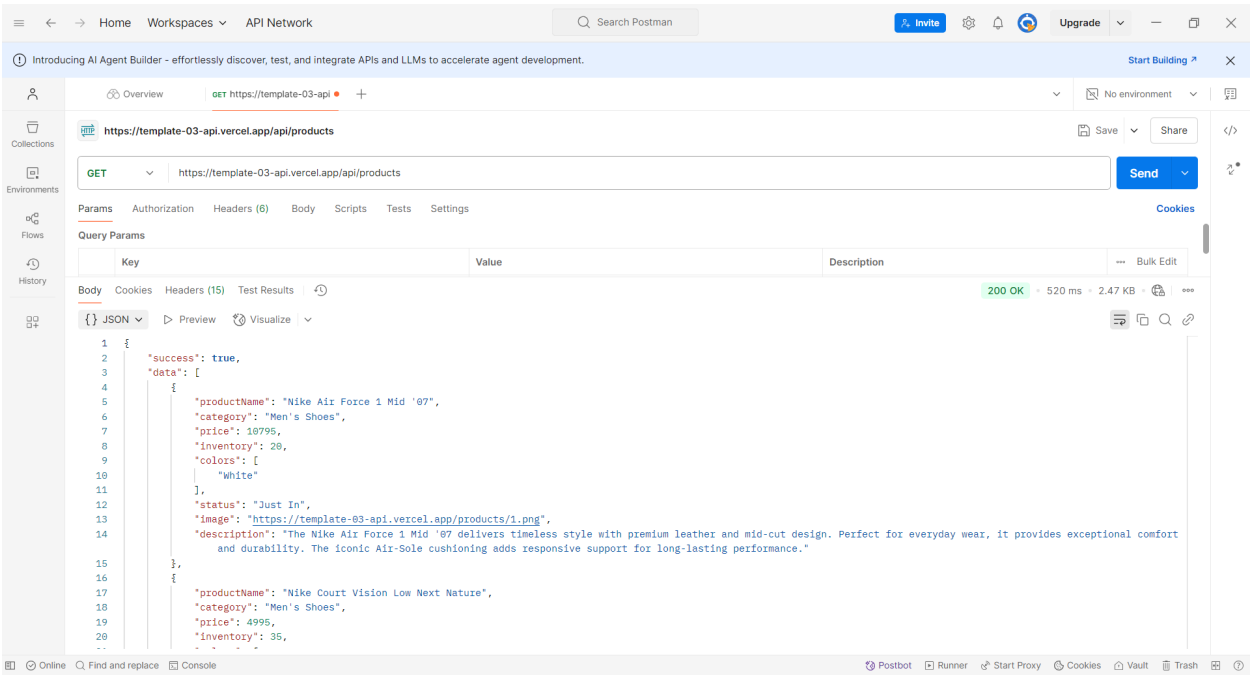
Day 3 - API Integration and Data Migration (Template 3)

Objective:

The goal for Day 3 was to integrate the API provided for Template 3 and migrate its data into Sanity CMS to create a functional backend for a marketplace. This process aligns closely with real-world practices, focusing on schema validation, data transformation, and seamless API integration.

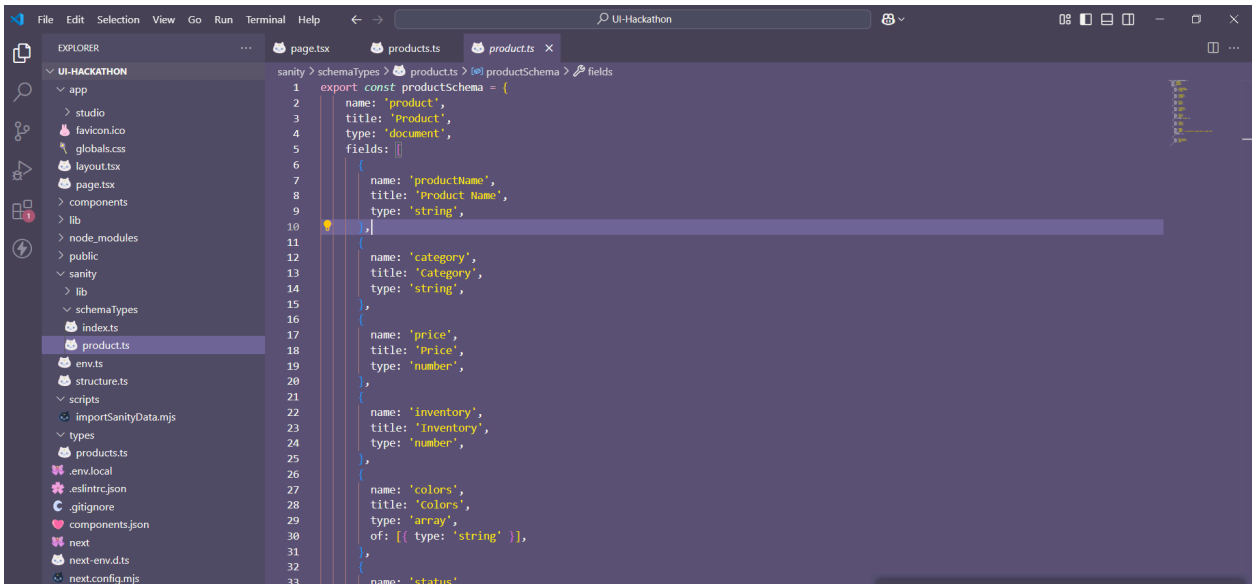
Key Outcomes from Template 3

1. **API Overview**:
    - **API URL**: https://template-03-api.vercel.app/api/products
    - This API provided the product data required to populate the marketplace.

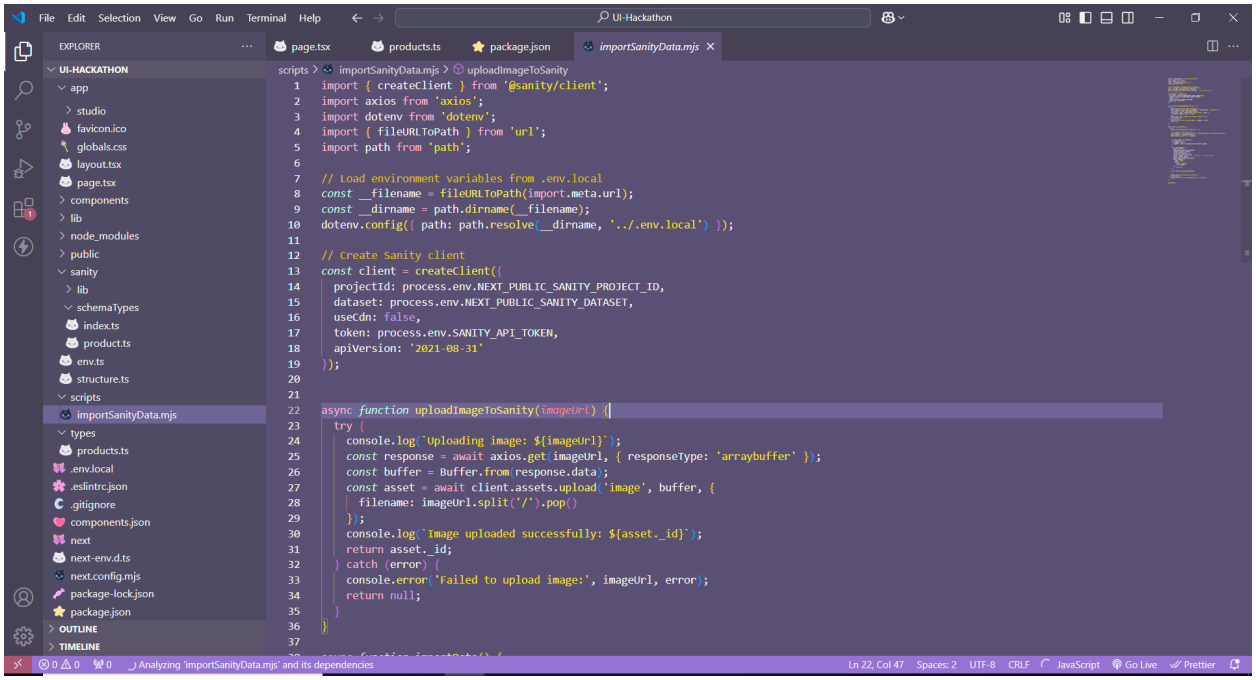


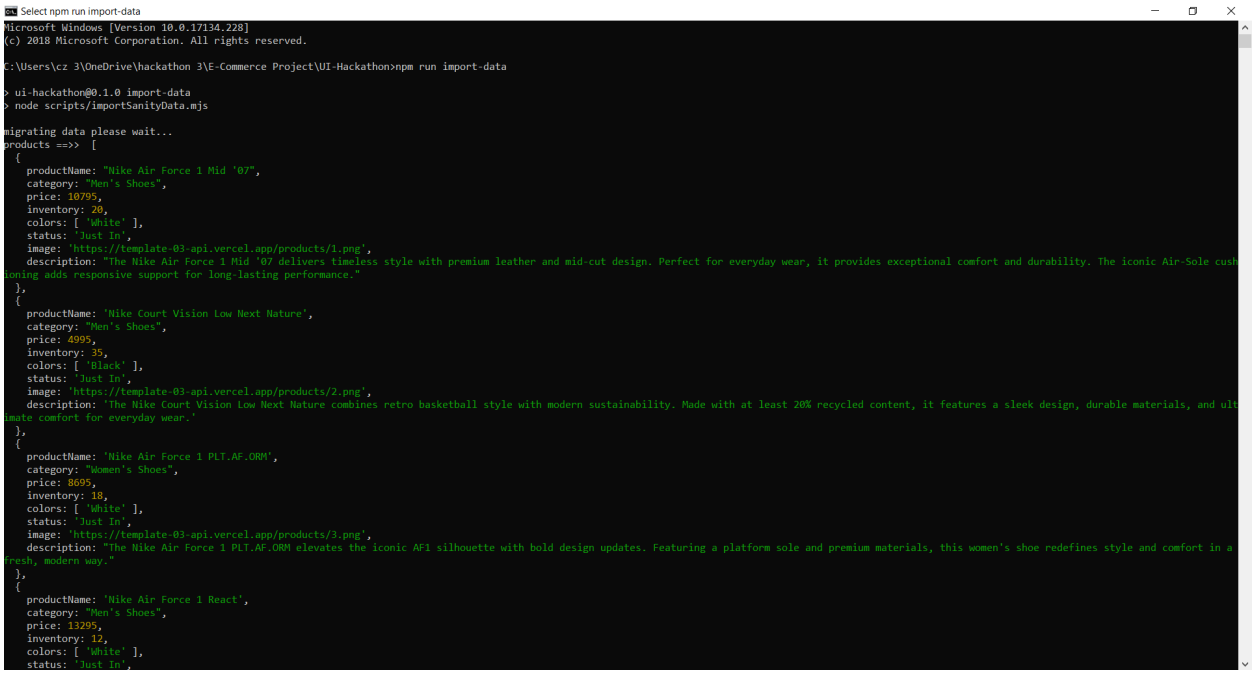
2. **Schema Validation**:
    - The Sanity CMS schema needed adjustments to match the API data.
    - For instance:

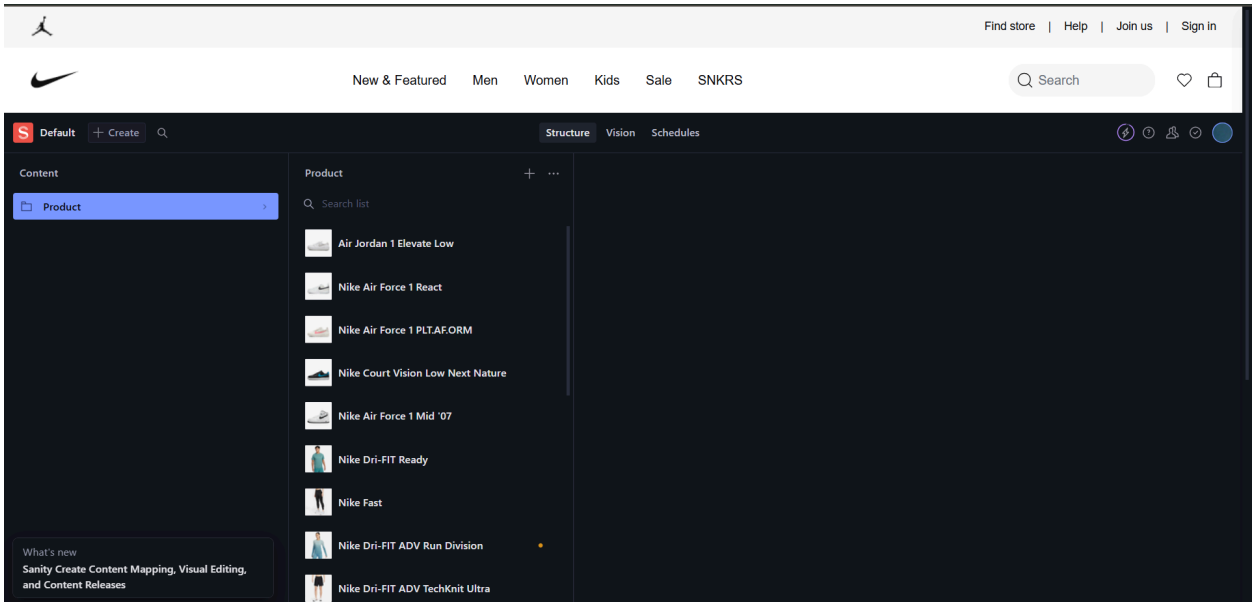
- API field: `product_title`
- Schema field: `name`
- Mapping these fields ensured data compatibility during migration.



3. **Data Migration**:
    - Utilized the migration script provided for Template 3:
        - [Data Migration Script]
    - Fetched and transformed data from the API into a format compatible with the schema.


File Edit Selection View Go Run Terminal Help
UI-Hackathon
EXPLORER
UI-HACKATHON
app
studio
favicon.ico
globals.css
layout.tsx
page.tsx
components
lib
node_modules
public
sanity
lib
schemaTypes
index.ts
product.ts
env.ts
structure.ts
scripts
importSanityData.mjs
types
products.ts
.env.local
.eslintrc.json
.gitignore
components.json
next
next-env.d.ts
next.config.mjs
package-lock.json
package.json
OUTLINE
TIMELINE
page.tsx
products.ts
package.json
importSanityData.mjs
scripts > importSanityData.mjs > uploadImageToSanity
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;
}
}
Analyzing 'importSanityData.mjs' and its dependencies
Ln 22, Col 47
Spaces: 2
UTF-8
CRLF
JavaScript
Go Live
Prettier


Select npm run import-data
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\cz 3\OneDrive\hackathon 3\E-Commerce Project\UI-Hackathon>npm run import-data
> ui-hackathon@0.1.0 import-data
> node scripts/importSanityData.mjs
migrating data please wait...
products ==>> [
{
productName: "Nike Air Force 1 Mid '07",
category: "Men's Shoes",
price: 10795,
inventory: 20,
colors: ['White'],
status: 'Just In',
image: 'https://template-03-api.vercel.app/products/1.png',
description: "The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and mid-cut design. Perfect for everyday wear, it provides exceptional comfort and durability. The iconic Air-Sole cushioning adds responsive support for long-lasting performance."
},
{
productName: 'Nike Court Vision Low Next Nature',
category: "Men's Shoes",
price: 4995,
inventory: 35,
colors: ['Black'],
status: 'Just In',
image: 'https://template-03-api.vercel.app/products/2.png',
description: 'The Nike Court Vision Low Next Nature combines retro basketball style with modern sustainability. Made with at least 20% recycled content, it features a sleek design, durable materials, and ultimate comfort for everyday wear.'
},
{
productName: 'Nike Air Force 1 PLT.AF.ORM',
category: "Women's Shoes",
price: 8695,
inventory: 18,
colors: ['White'],
status: 'Just In',
image: 'https://template-03-api.vercel.app/products/3.png',
description: "The Nike Air Force 1 PLT.AF.ORM elevates the iconic AF1 silhouette with bold design updates. Featuring a platform sole and premium materials, this women's shoe redefines style and comfort in a fresh, modern way."
},
{
productName: 'Nike Air Force 1 React',
category: "Men's Shoes",
price: 13295,
inventory: 12,
colors: ['White'],
status: 'Just In',



4. **API Integration in Next.js**:
    - Created reusable utility functions to fetch data from the API.
    - Rendered product listings and categories on the frontend.
    - Tested endpoints using Postman and validated the API responses.

Steps I Followed

1. Understanding the Template 3 API

- Reviewed API documentation and identified key endpoints:
    - `/products`: For fetching the list of products.

2. Validating and Adjusting the Schema

- Compared the Sanity schema to the API response structure.
- Updated the schema in Sanity CMS:
    - Adjusted field names and data types to align with the API.

3. Migrating Data

- Ran the migration script to pull data from the API.
- Verified the data was correctly imported into Sanity CMS.

- Example:
    - Migrated product data with fields like `name`, `price`, and `category`.

4. Integrating the API in Next.js

- Built utility functions to handle API calls and data fetching.
- Rendered the fetched data in frontend components.
- Tested the integration to ensure all data was displayed correctly.

Deliverables

1. **Sanity CMS**:
    - Populated with Template 3 product data.
    - Adjusted schema validated against the API data structure.
2. **Frontend Integration**:
    - Displayed product listings fetched via API.
    - Included error handling for invalid responses or empty data sets.