

## DAY 3 - API INTEGRATION AND DATA MIGRATION - [General E-commerce]

### Report Documentation:

#### API Integration Process:

- **API Integration:**  
Data was fetched from Sanity CMS using a fixed (static) ID. The API call was made to <https://template1-neon-nu.vercel.app/api/products>, which retrieves product data for the specific ID.
- **Authentication:**  
The API token was passed in the request headers for authentication.
- **Frontend Display:**  
The fetched data was displayed on the frontend using React `useState` and `useEffect` hooks.

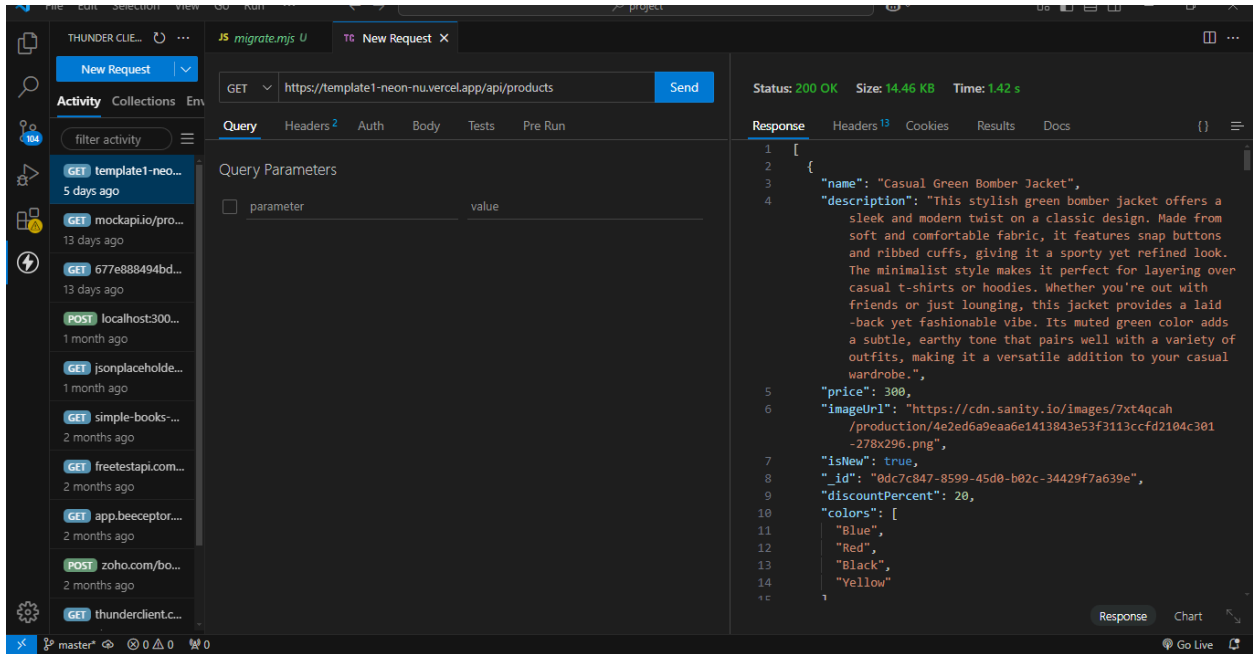
#### Adjustments Made to Schemas:

- **ID Field:**  
A static `id` field was added to each product in Sanity CMS, allowing for the specific product data to be fetched.

#### Migration Steps and Tools Used:

- **Sanity CLI:**  
Sanity CLI was used to migrate the data. After fetching the data, it was updated according to the new schema, including the static `id` field.

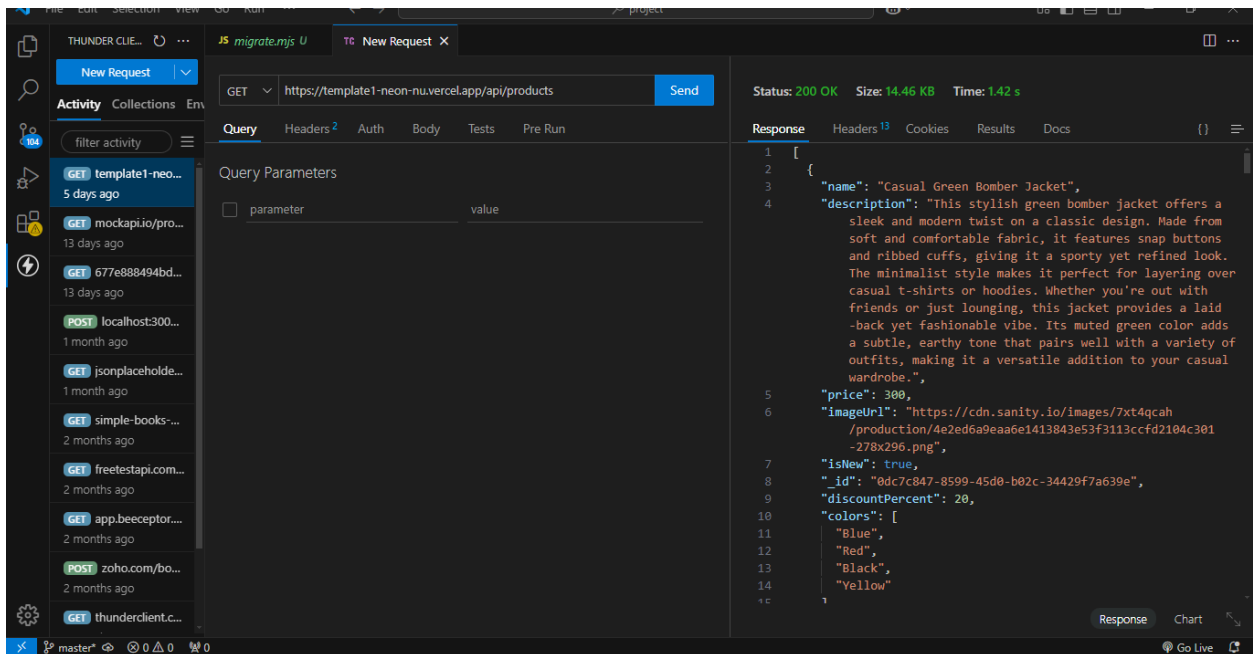
# API get



Thunder Client interface showing a GET request to `https://template1-neon-nu.vercel.app/api/products`. The response is a JSON array with one item: a 'Casual Green Bomber Jacket' with a price of 300, an image URL, and various attributes like 'isNew', '\_id', 'discountPercent', and 'colors'.

```
1 [
2   {
3     "name": "Casual Green Bomber Jacket",
4     "description": "This stylish green bomber jacket offers a
5       sleek and modern twist on a classic design. Made from
6       soft and comfortable fabric, it features snap buttons
7       and ribbed cuffs, giving it a sporty yet refined look.
8       The minimalist style makes it perfect for layering over
9       casual t-shirts or hoodies. Whether you're out with
10      friends or just lounging, this jacket provides a laid
11      -back yet fashionable vibe. Its muted green color adds
12      a subtle, earthy tone that pairs well with a variety of
13      outfits, making it a versatile addition to your casual
14      wardrobe.",
15     "price": 300,
16     "imageUrl": "https://cdn.sanity.io/images/7xt4qcah
17       /production/4e2ed6a9eaa6e1413843e53f3113ccfd2104c301
18       -278x296.png",
19     "isNew": true,
20     "_id": "0dc7c847-8599-45d0-b02c-34429f7a639e",
21     "discountPercent": 20,
22     "colors": [
23       "Blue",
24       "Red",
25       "Black",
26       "Yellow"
27     ]
28   }
29 ]
```

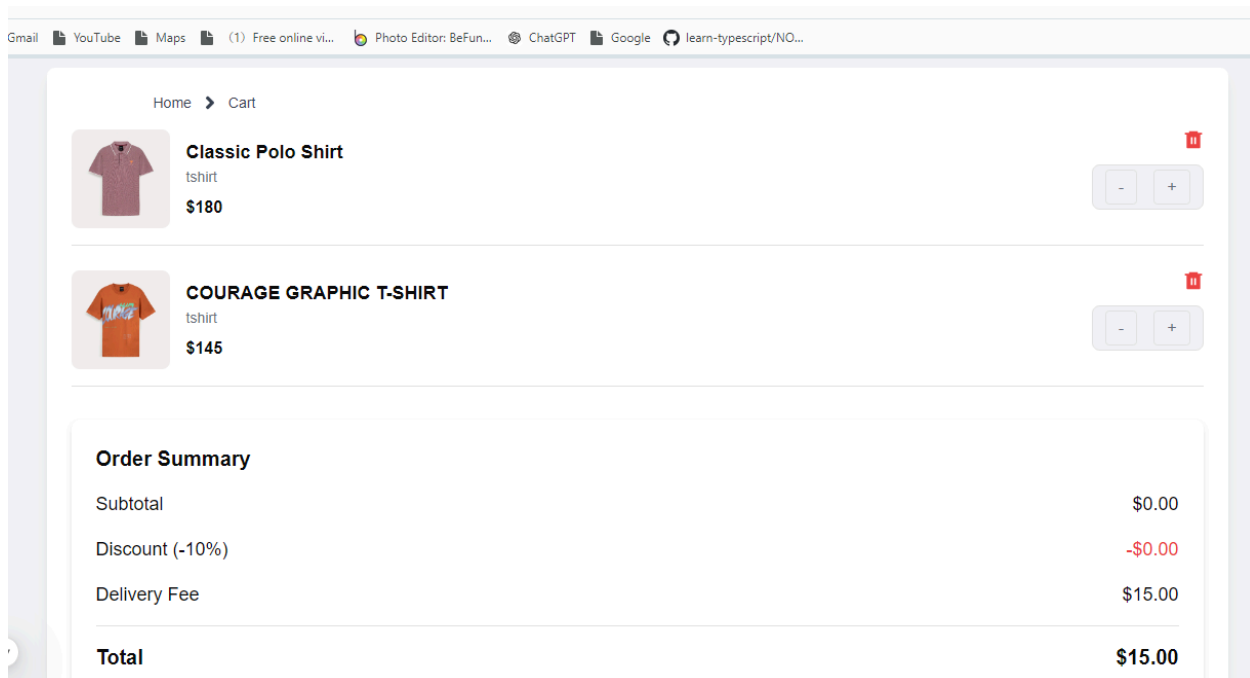
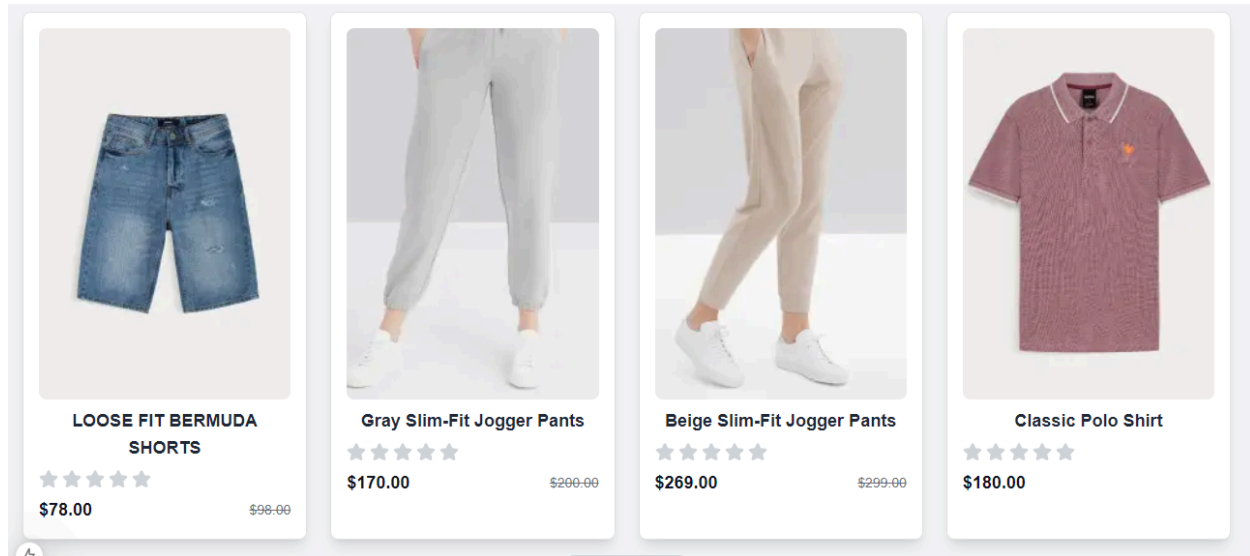
## API get status



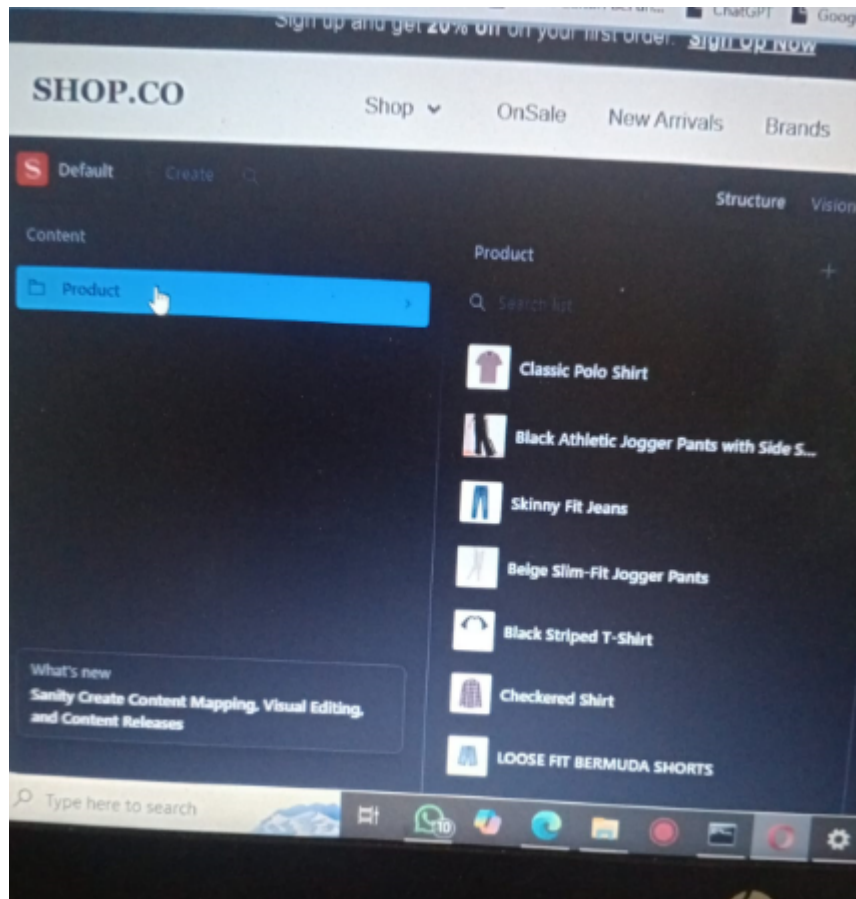
Thunder Client interface showing a GET request to `https://template1-neon-nu.vercel.app/api/products`. The response is a JSON array with one item: a 'Casual Green Bomber Jacket' with a price of 300, an image URL, and various attributes like 'isNew', '\_id', 'discountPercent', and 'colors'.

```
1 [
2   {
3     "name": "Casual Green Bomber Jacket",
4     "description": "This stylish green bomber jacket offers a
5       sleek and modern twist on a classic design. Made from
6       soft and comfortable fabric, it features snap buttons
7       and ribbed cuffs, giving it a sporty yet refined look.
8       The minimalist style makes it perfect for layering over
9       casual t-shirts or hoodies. Whether you're out with
10      friends or just lounging, this jacket provides a laid
11      -back yet fashionable vibe. Its muted green color adds
12      a subtle, earthy tone that pairs well with a variety of
13      outfits, making it a versatile addition to your casual
14      wardrobe.",
15     "price": 300,
16     "imageUrl": "https://cdn.sanity.io/images/7xt4qcah
17       /production/4e2ed6a9eaa6e1413843e53f3113ccfd2104c301
18       -278x296.png",
19     "isNew": true,
20     "_id": "0dc7c847-8599-45d0-b02c-34429f7a639e",
21     "discountPercent": 20,
22     "colors": [
23       "Blue",
24       "Red",
25       "Black",
26       "Yellow"
27     ]
28   }
29 ]
```

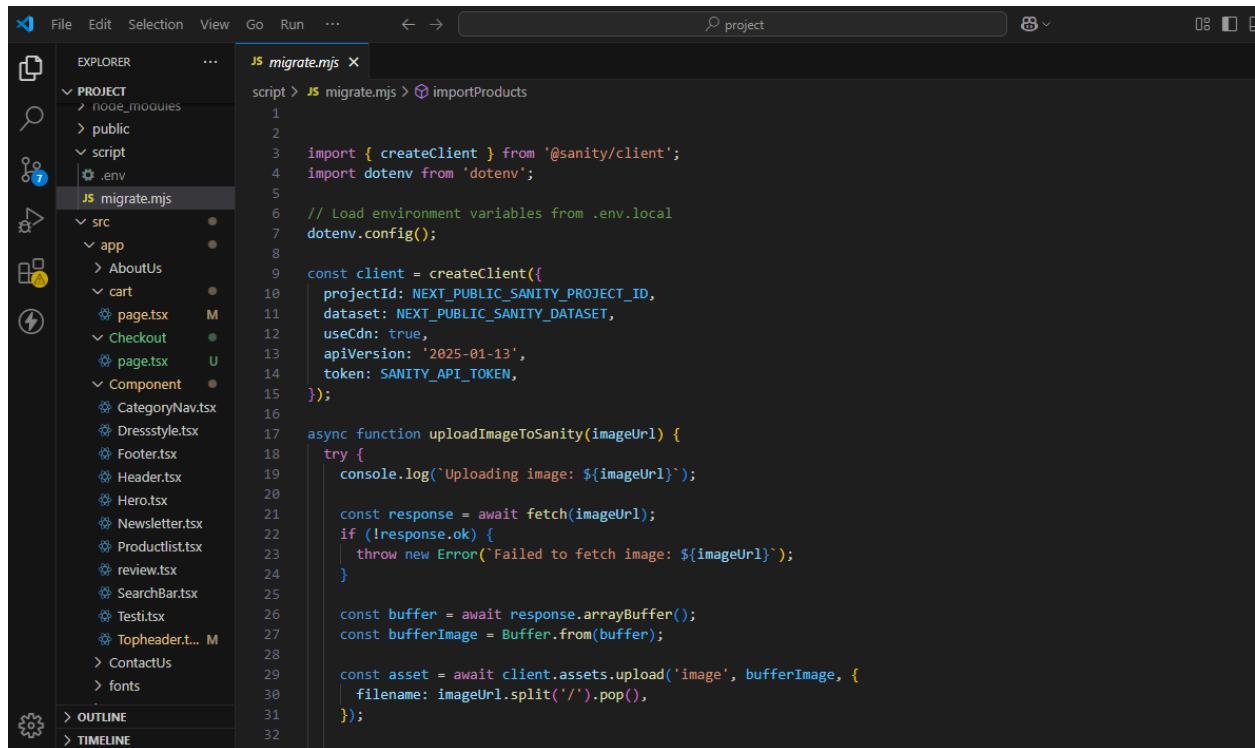
# Data Successfully Displayed in the Frontend:



# Populated Sanity CMS Fields:

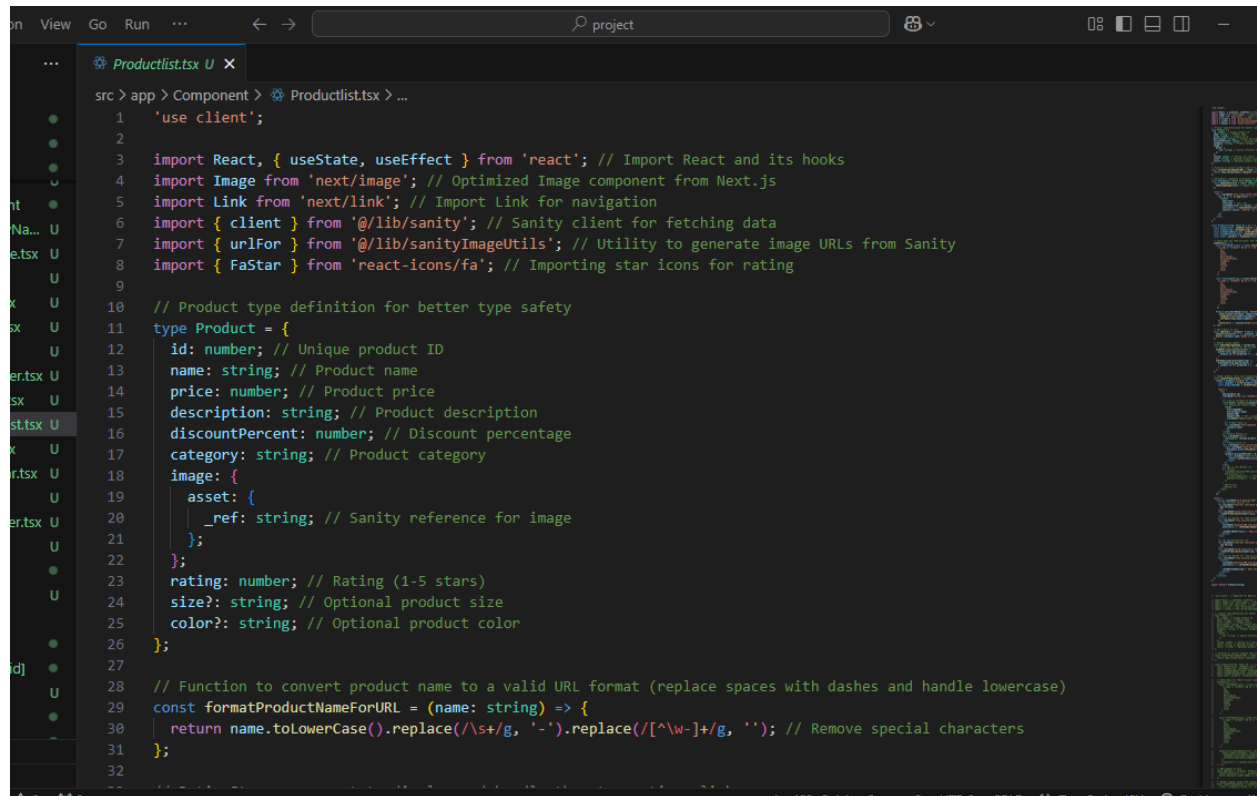


## Code snippets for API integration and migration scripts.



The screenshot shows a code editor with a dark theme. On the left is the Explorer sidebar showing a project structure with folders like 'public', 'script', and 'src'. The 'script' folder contains a file named 'migrate.mjs'. The main editor area displays the content of 'migrate.mjs', which is a JavaScript file for migrating data to Sanity.io. The script imports 'createClient' from '@sanity/client' and 'dotenv' from 'dotenv'. It loads environment variables from '.env.local' and configures a client with project ID, dataset, API version, and token. An async function 'uploadImageToSanity' is defined, which fetches an image, converts it to a buffer, and uploads it to Sanity's assets.

```
script > JS migrate.mjs > importProducts
1
2
3 import { createClient } from '@sanity/client';
4 import dotenv from 'dotenv';
5
6 // Load environment variables from .env.local
7 dotenv.config();
8
9 const client = createClient({
10   projectId: NEXT_PUBLIC_SANITY_PROJECT_ID,
11   dataset: NEXT_PUBLIC_SANITY_DATASET,
12   useCdn: true,
13   apiVersion: '2025-01-13',
14   token: SANITY_API_TOKEN,
15 });
16
17 async function uploadImageToSanity(imageUrl) {
18   try {
19     console.log(`Uploading image: ${imageUrl}`);
20
21     const response = await fetch(imageUrl);
22     if (!response.ok) {
23       throw new Error(`Failed to fetch image: ${imageUrl}`);
24     }
25
26     const buffer = await response.arrayBuffer();
27     const bufferImage = Buffer.from(buffer);
28
29     const asset = await client.assets.upload('image', bufferImage, {
30       filename: imageUrl.split('/').pop(),
31     });
32 }
```



```
1 'use client';
2
3 import React, { useState, useEffect } from 'react'; // Import React and its hooks
4 import Image from 'next/image'; // Optimized Image component from Next.js
5 import Link from 'next/link'; // Import Link for navigation
6 import { client } from '@lib/sanity'; // Sanity client for fetching data
7 import { urlFor } from '@lib/sanityImageUtils'; // Utility to generate image URLs from Sanity
8 import { FaStar } from 'react-icons/fa'; // Importing star icons for rating
9
10 // Product type definition for better type safety
11 type Product = {
12   id: number; // Unique product ID
13   name: string; // Product name
14   price: number; // Product price
15   description: string; // Product description
16   discountPercent: number; // Discount percentage
17   category: string; // Product category
18   image: {
19     asset: {
20       _ref: string; // Sanity reference for image
21     };
22   };
23   rating: number; // Rating (1-5 stars)
24   size?: string; // Optional product size
25   color?: string; // Optional product color
26 };
27
28 // Function to convert product name to a valid URL format (replace spaces with dashes and handle lowercase)
29 const formatProductNameForURL = (name: string) => {
30   return name.toLowerCase().replace(/\s+/g, '-').replace(/[^\w-]+/g, ''); // Remove special characters
31 };
32
```

## Day 3 Checklist:

Self-Validation Checklist: API Understanding: ✓

Schema Validation: ✓

Data Migration: ✓

API Integration in Next.js: ✓

Submission Preparation: ✓

