DAY 3 - API INTEGRATION

AND DATA MIGRATION

In Day3 of our Marketplace builder hackathon series, we were assigned a basic task of integrating sanity with any custom API. Means we will upload data from API to sanity and then this API data will be displayed in our frontend. Earlier, it was a heard core data and we just map and displayed in frontend. Now as mentioned, it was hardcore data, and now it is an API data, one of the biggest issues was structure mismatch. As my website was purely built on previous data structure, so there was a need to do some changes throughout my code in different parts. Here I'l discuss this along with many other issues I faced and successfully resolved.

Understand the Provided API:

The API provided was **https://template1-neon-nu.vercel.app/api/products**. On accessing it endpoint /product it returns the list of products.

Product Data Structure Comparison:

The product data structure was:

```
{
    "name": "Casual Green Bomber Jacket",
    "description": "This stylish green bomber jacket offers a sleek
and modern twist on a classic design. Made from soft and comfortable
fabric, it features snap buttons and ribbed cuffs, giving it a sporty
yet refined look. The minimalist style makes it perfect for layering
over casual t-shirts or hoodies. Whether you're out with friends or
just lounging, this jacket provides a laid-back yet fashionable vibe.
Its muted green color adds a subtle, earthy tone that pairs well with
a variety of outfits, making it a versatile addition to your casual
wardrobe.",
    "price": 300,
    "imageUrl":
"https://cdn.sanity.io/images/7xt4qcah/production/4e2ed6a9eaa6e1413843
e53f3113ccfd2104c301-278x296.png",
    "colors": [
      "Blue",
      "Red",
      "Black",
      "Yellow"
    ],
    "sizes": [
      "S",
      "XXL",
      "XL",
      "L"
    ],
    "_id": "0dc7c847-8599-45d0-b02c-34429f7a639e",
    "category": "hoodie",
    "discountPercent": 20,
    "isNew": true
  },
```

The data was consist of properties like name, description, price, imageUrl,colors, sizes, _id,category, discountPercent and isNew. When I was working on previous hardcore data its structure was purely different. Here is its structure.

```
{
    id: "1",
    itemName: "One Life Graphic T-shirt",
    image1: "/item/graphictshirt1.png",
    image2: "/item/graphictshirt2.png",
    image3: "/item/graphictshirt3.png",
    image4: "/item/graphictshirt4.png",
    discountedPrice: 260.0,
    actualPrice: 300.0,
    rating: 4.5,
    discountpercent: 40,
  },
```

So, obviously in the first step, I sort out the difference in structure, and aligned my project with new data.

Validate and Adjust Your Schema:

As mentioned earlier, I need to sort out my data with new structure, so I had to change my schema. Earlier my schema was:

```
import { defineField, defineType } from "sanity";

export const ProductType = defineType({
    title: "Product",
    type: "document",
    name: "product",
    fields: [
        defineField({
            title: "Product Name",
            type: "string",
            name: "itemName"
        }),
        defineField({
            title: "Product Price",
            type: "number",
            name: "actualPrice"
        }),
        defineField({
            title: "Discounted Product Price",
            type: "number",
            name: "discountedPrice"
        }),
        defineField({
            title: "Discount Percentage",
            type: "number",
            name: "discountpercent"
        }),
        defineField({
            title: "Product Ratings",
            type: "number",
            name: "rating"
        }),
        defineField({
            name: 'section',
            title: 'Product Section',
            type: 'string',
            options: {
                list: [
                    { title: 'New Arrivals', value: 'newArrivals' },
                    { title: 'Top Sellings', value: 'topSellings' },
                    { title: 'Recommended', value: 'recommended' },
                ]
            }
        }),
        defineField({
            name: "category",
            type: "array",
            title: "Product Category",
            of: [{ type: 'reference', to: { type: 'category' } }]
        }),
        defineField({
            title: "Featured Image 1",
            name: "image1",
            type: "image",
            options: {
                hotspot: true
            },

        })})}
```

And the category was actually refrence typed, like

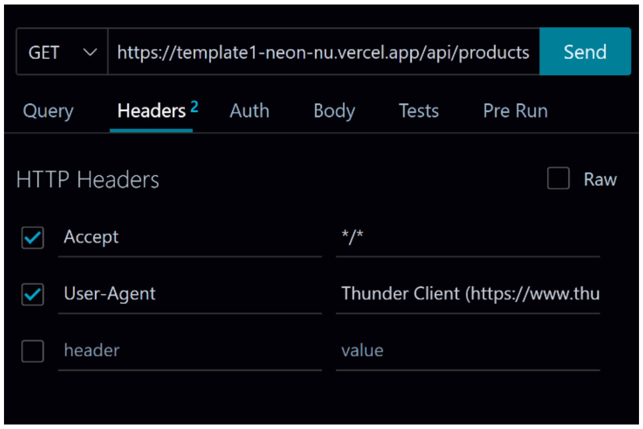
```
import { defineField, defineType } from "sanity";

export const CategoryType = defineType({
    name: "category",
    type: "document",
    title: "Product Category",
    fields: [
        defineField({
            name: "category",
            type: "string",
            title: "Product Category",
        })
    ]
})
```

In my new sata, it was a simple structure with some so my new schema was:

```
import { defineField, defineType } from "sanity";

export const ProductCatalog = defineType({
    title: "Product Catalog",
    name: "catalog",
    type: "document",
    fields: [
        defineField({
            name: "name",
            title: "Product Name",
            type: "string"
        }),
        defineField({
            name: "description",
            title: "Product Description",
            type: "string"
        }),
        defineField({
            name: "price",
            title: "Product Price",
            type: "number"
        }),
        defineField({
            name: "category",
            title: "Product 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' },
                ]
            }
        }),
        defineField({
            name: "discountPercent",
            title: "Product Discount Percentage",
            type: "number"
        }),
        defineField({
            name: "isNew",
            title: "New",
            type: "boolean"
        }),
        defineField({
            name: "colors",
            title: "Product Colors",
            type: "array",
            of: [{ type: "string" }]
        }),
        defineField({
            name: "sizes",
            title: "Product Sizes",
            type: "array",
            of: [{ type: "string" }]
        }),
        defineField({
            name: "imageUrl",
            title: "Product Image",
            type: "image",
            options: {
                hotspot: true
            }
        }),
    ]
})
```

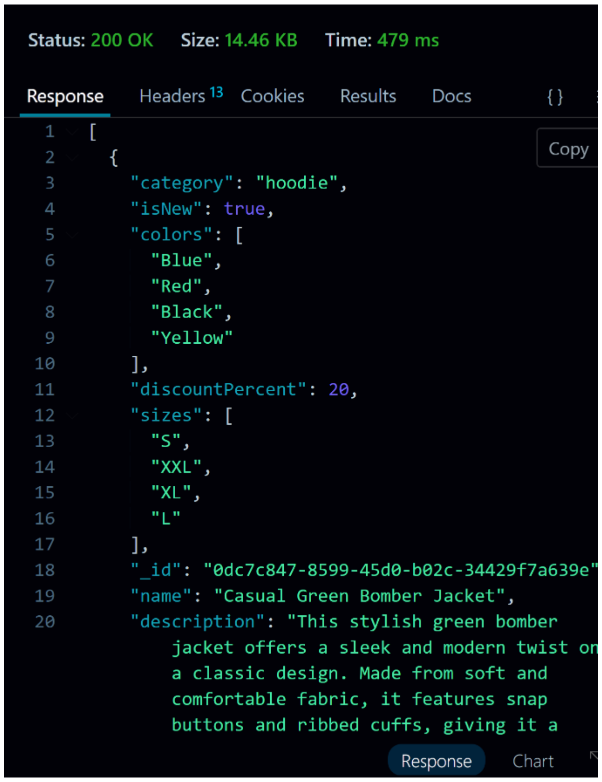
API Calls:

Api was tested through thunder client extension in vscode. The result was successful



```
GET  https://template1-neon-nu.vercel.app/api/products  Send
```

✓	Accept	*/*
✓	User-Agent	Thunder Client (https://www.thu
	header	value



Status: 200 OK   Size: 14.46 KB   Time: 479 ms

```
[
  {
    "category": "hoodie",
    "isNew": true,
    "colors": [
      "Blue",
      "Red",
      "Black",
      "Yellow"
    ],
    "discountPercent": 20,
    "sizes": [
      "S",
      "XXL",
      "XL",
      "L"
    ],
    "_id": "0dc7c847-8599-45d0-b02c-34429f7a639e"
    "name": "Casual Green Bomber Jacket",
    "description": "This stylish green bomber
        jacket offers a sleek and modern twist on
        a classic design. Made from soft and
        comfortable fabric, it features snap
        buttons and ribbed cuffs, giving it a
```

Data Migration Options:

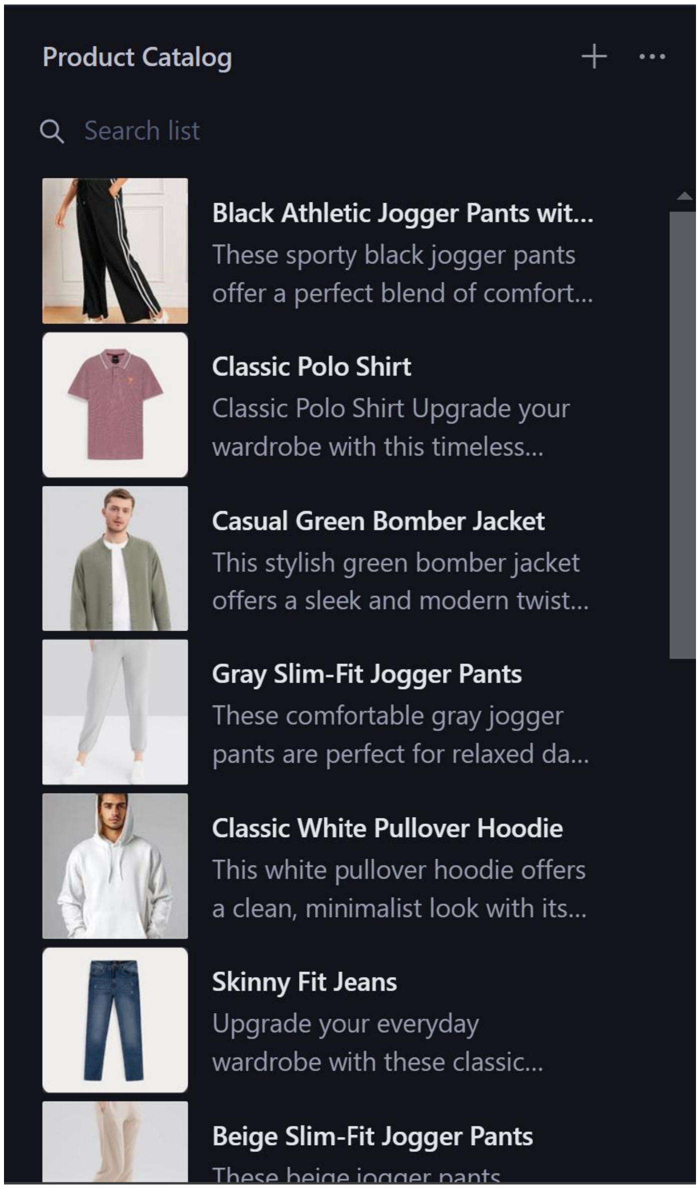
For Data Migration I built a helper function SanityCreate.ts. Although it was already created but due to change in data structure, I had to made some changes here as well. And after some changes it was now applicable to handle both structure the new one and old as well.
To handle my new data, I added else if block in my helper function. Where route is an argument which will be provided while running command through terminal. If route is api it will apply code related to new data otherewise if route matches with requirements of previous data it will go through this.

```
else if (route === "api") {
        const response = await fetch("https://template1-neon-
nu.vercel.app/api/products")
        data = await response.json()
        data = await Promise.all(data.map(async (productData:
ProductType) => {
            try {
                const imageResponse = await
fetch(productData.imageUrl);

                if (!imageResponse.ok) {
                    throw new Error(`Failed to fetch image:
${productData.imageUrl}`);
                }
                const arrayBuffer = await imageResponse.arrayBuffer();
                const imageBuffer = Buffer.from(arrayBuffer);
                const imageAsset = await client.assets.upload("image",
imageBuffer);
                return {
                    ...productData,
                    imageUrl: imageAsset._id
                };
            } catch (error) {
                console.error(`Error processing product
${productData.name}:`, error);
                return productData;
            }
        }))
    }
```

Now this will basically post my data to /api/catalog. And then catalog route will handle uploading to sanity.
Now when in terminal, I'll run command as

```
npm run create - api
```

It will fetch data from API and then populate Sanity CMS with this new data. And it did as per expectation Here is my sanity after it was populated with new data.


Product Catalog

Search list

Black Athletic Jogger Pants wit...
These sporty black jogger pants offer a perfect blend of comfort...

Classic Polo Shirt
Classic Polo Shirt Upgrade your wardrobe with this timeless...

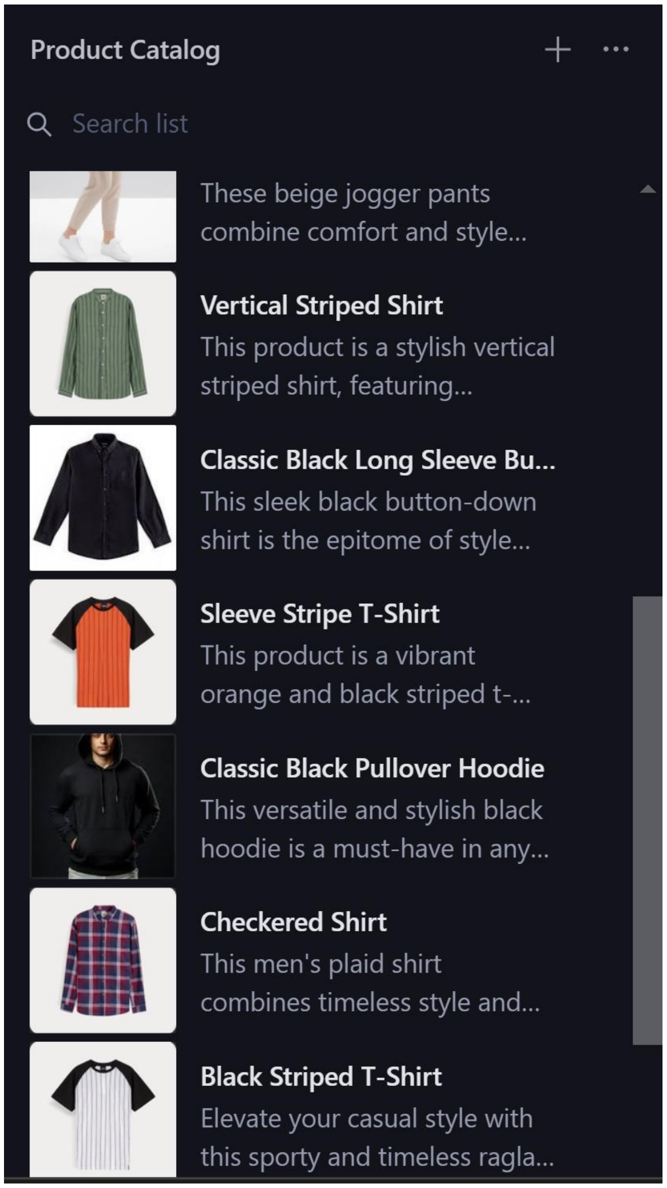
Casual Green Bomber Jacket
This stylish green bomber jacket offers a sleek and modern twist...

Gray Slim-Fit Jogger Pants
These comfortable gray jogger pants are perfect for relaxed da...

Classic White Pullover Hoodie
This white pullover hoodie offers a clean, minimalist look with its...

Skinny Fit Jeans
Upgrade your everyday wardrobe with these classic...

Beige Slim-Fit Jogger Pants
These beige jogger pants


Product Catalog

Search list

These beige jogger pants combine comfort and style...

Vertical Striped Shirt
This product is a stylish vertical striped shirt, featuring...

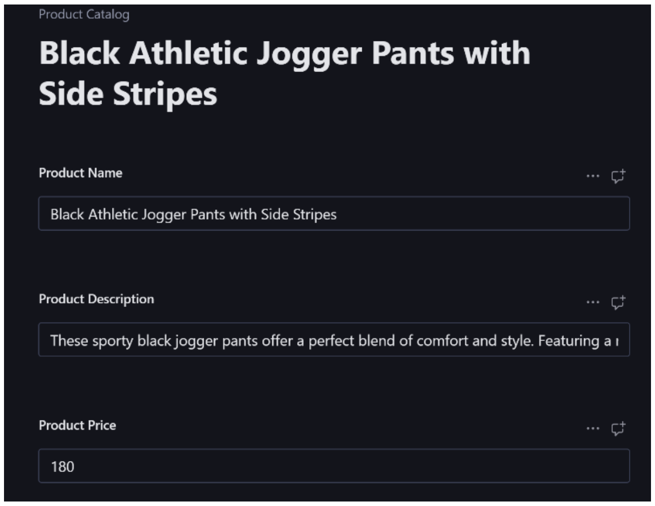
Classic Black Long Sleeve Bu...
This sleek black button-down shirt is the epitome of style...

Sleeve Stripe T-Shirt
This product is a vibrant orange and black striped t-...

Classic Black Pullover Hoodie
This versatile and stylish black hoodie is a must-have in any...

Checkered Shirt
This men's plaid shirt combines timeless style and...

Black Striped T-Shirt
Elevate your casual style with this sporty and timeless ragla...

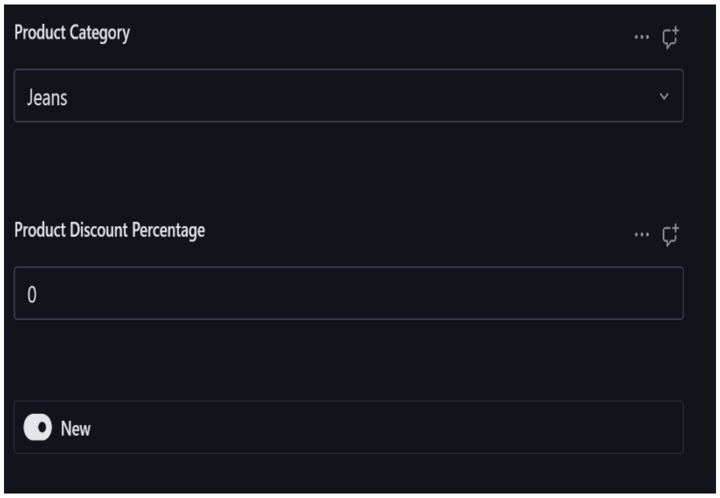

Product Catalog

Black Athletic Jogger Pants with Side Stripes

Product Name
Black Athletic Jogger Pants with Side Stripes

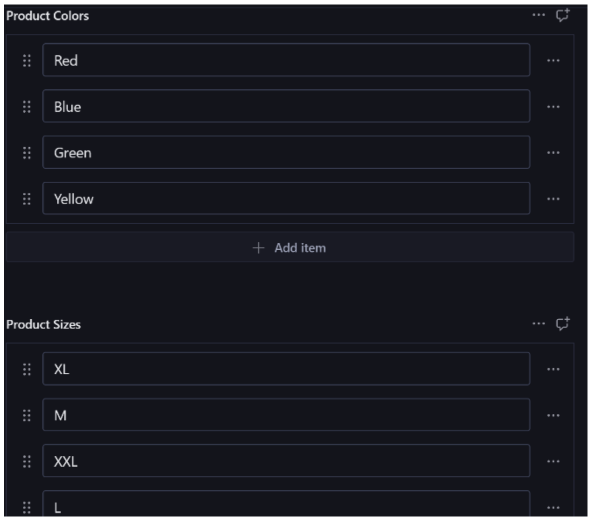
Product Description
These sporty black jogger pants offer a perfect blend of comfort and style. Featuring a

Product Price
180


Product Category
Jeans

Product Discount Percentage
0

New


Product Colors

Red

Blue

Green

Yellow

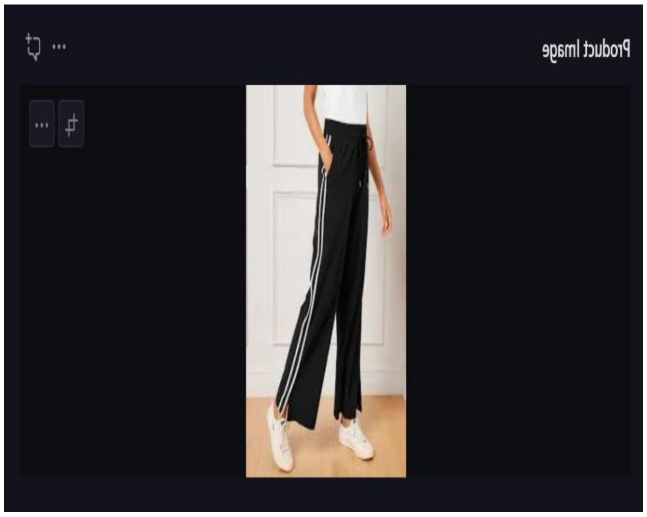
Product Sizes

XL

M

XXL

L



Now as my sanity is populated with new data, now I had to display this data to my frontend. For this I used below query.

```
export const CATALOG_QUERY = defineQuery(`*[_type=='catalog']`)
```

and used this query to fetch data like.

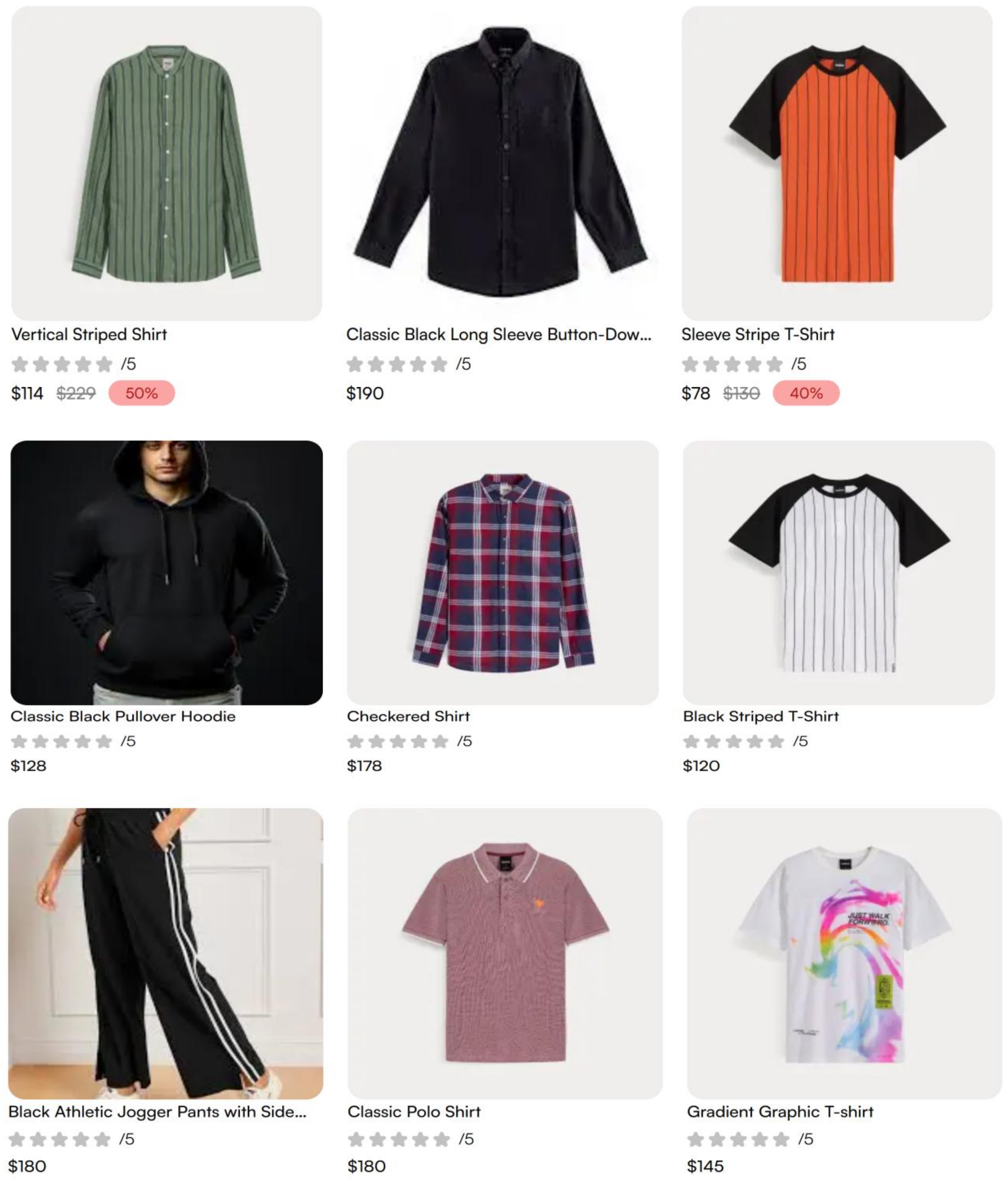
```
export async function getProductsCatalog() {
    try {
        const catalog = await sanityFetch({
            query: CATALOG_QUERY
        })
        return catalog.data || []

    } catch (error) {
        console.error('Products fetching Error:', error)
        return []
    }
}
```

Now simply, where I need to get data I will use this method to fetch all catalog from sanity and will display as per structure of data.

```
const catalog = await getProductsCatalog();
```

The displayed data on frontend:

As the new data had no option to manage reviews, this is still under consideration, and I’ll soon manage to update reviews in new data as well. Currently displayed it looks like this.



Vertical Striped Shirt
★★★★★ /5
$114 ~~$229~~ 50%

Classic Black Long Sleeve Button-Dow...
★★★★★ /5
$190

Sleeve Stripe T-Shirt
★★★★★ /5
$78 ~~$130~~ 40%

Classic Black Pullover Hoodie
★★★★★ /5
$128

Checkered Shirt
★★★★★ /5
$178

Black Striped T-Shirt
★★★★★ /5
$120

Black Athletic Jogger Pants with Side...
★★★★★ /5
$180

Classic Polo Shirt
★★★★★ /5
$180

Gradient Graphic T-shirt
★★★★★ /5
$145