

Document Title: "Day 3 API Integration Report"

What to submit 👍

A report documenting:

- API integration process.
- Adjustments made to schemas.
- Migration steps and tools used.

Screenshots of:

- API calls.
- Data successfully displayed in the frontend.
- Populated Sanity CMS fields.
- Code snippets for API integration and migration scripts.

C:\Windows\System32\cmd.exe

Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.

C:\Users\BINARY COMPUTERS\Desktop\hackathon>code .

C:\Users\BINARY COMPUTERS\Desktop\hackathon>npm create sanity@latest

■ You are logged in as syedobaidullahsharif@gmail.com using Google

■ Fetching existing projects

? Create a new project or select an existing one [Create new project](#)

? Your project name: [day3api](#)

Your content will be stored in a dataset that can be public or private, depending on whether you want to query your content with or without authentication.

The default dataset configuration has a public dataset named "production".

? Use the default dataset configuration? [Yes](#)

■ Creating dataset

? Project output path: C:\Users\BINARY COMPUTERS\Desktop\hackathon

? Select project template [Clean project with no predefined schema types](#)

? Do you want to use TypeScript? [Yes](#)

■ Bootstrapping files from template

■ Resolving latest module versions

■ Creating default project files

? Package manager to use for installing dependencies? [npm](#)

Running 'npm install --legacy-peer-deps'

npm WARN deprecated @sanity/block-tools@3.70.0: Renamed to use `@pocketlint/block-tools` instead. `@sanity/block-tools`

```
244 packages are looking for funding
  run 'npm fund' for details
```

```
1 moderate severity vulnerability
```

```
To address all issues, run:
npm audit fix --force
```

```
Run 'npm audit' for details.
```

```
Success! Your Sanity configuration files has been added to this project
```



All Bookm



Syedobaid ullahsharif

template-0Ubaid

29 days left in trial



Syedobaid ullahsharif

template-0Ubaid

PLAN

Growth Trial

STATUS

Active

PROJECT ID

uz8braz2



Getting started



Overview



Members



Studios



Datasets



Access



Activity



Usage



Plan



API



Settings

Next steps

Initialize your project with the CLI

Run this command in your Terminal to continue setting up your project.

```
npm create sanity@latest -- --project uz8braz2 --dataset produ
```

Copy

Having issues with the CLI?

What's new

Sanity Create Content Manning. Visual Editing.

Project members

View all →



Invite your first team member

+ Invite project members

Getting startedOverviewMembersStudiosDatasetsAccessActivityUsagePlanAPISettings

Webhooks

CORS origins

Tokens

CORS origins

Hosts that can connect to the project API.

+ Add CORS origin

ORIGIN	CREDENTIALS	CREATED	
http://localhost:3000	Allowed	yesterday	
http://localhost:3333	Allowed	yesterday	

Webhooks

CORS origins

Tokens

Tokens

Tokens are used to authenticate apps and scripts to access project data.

+ Add API token

NAME	PERMISSIONS	CREATED	
template-0	Editor	yesterday	

MY-APP

next

node_modules

public

scripts

src

app

components

lib

sanity

sanityClient.js

.env.local

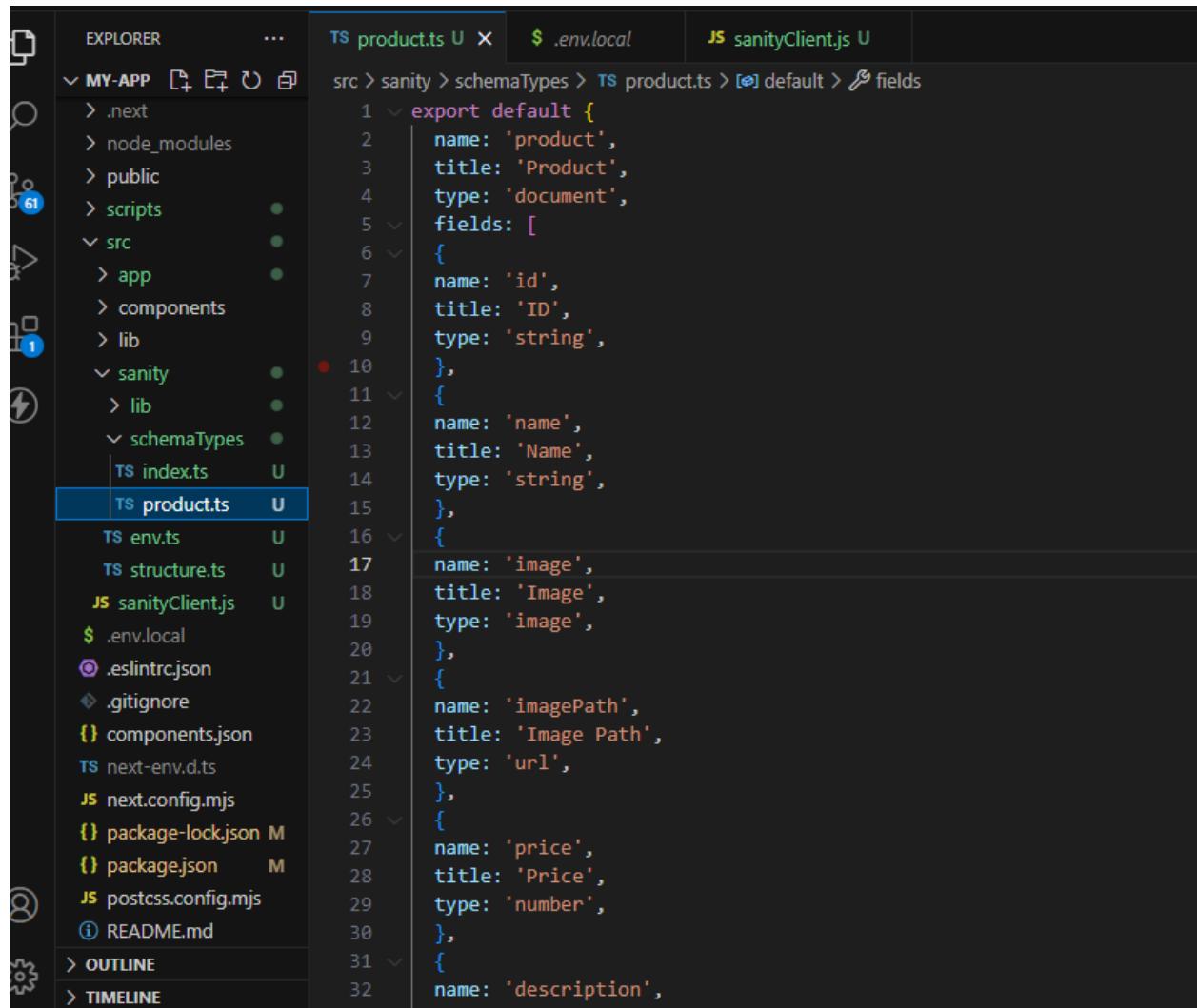
eslintrc.json

\$.env.local

1 NEXT_PUBLIC_SANITY_PROJECT_ID="uz8braz2"

2 NEXT_PUBLIC_SANITY_DATASET="production"

3 SANITY_API_TOKEN="skNe5onwKjJewgPFj321khhtFD8S3MPtttZiWtcmDutB8Wr5f9qqPwDTVk6MppeU1pIjC1UHsBg85XD1TX82W3wIVDv50k3bD7VwdoBAZ98VP2Ubc8mpRK2Mz9G"



MY-APP

- .next
- node_modules
- public
- scripts
- src
 - app
 - components
 - lib
 - sanity
 - lib
 - schemaTypes
 - TS index.ts
 - TS product.ts**
 - TS env.ts
 - TS structure.ts
 - JS sanityClient.js
- .env.local
- .eslintrc.json
- .gitignore
- components.json
- next-env.d.ts
- next.config.mjs
- package-lock.json
- package.json
- postcss.config.mjs
- README.md

OUTLINE

TIMELINE

src > sanity > schemaTypes > TS product.ts > default > fields

```
1 export default {
5   fields: [
27     {
28       name: 'price',
29       title: 'Price',
30       type: 'number',
31     },
32     {
33       name: 'description',
34       title: 'Description',
35       type: 'text',
36     },
37     {
38       name: 'discountPercentage',
39       title: 'Discount Percentage',
40       type: 'number',
41     },
42     {
43       name: 'isFeaturedProduct',
44       title: 'Is Featured Product',
45       type: 'boolean',
46     },
47     {
48       name: 'stockLevel',
49       title: 'Stock Level',
50       type: 'number',
51     },
52     {
53       name: 'category',
54       title: 'Category',
55       type: 'string',
56     },
57   ],
58 }
```

TypeScript Importer: Symbols: 88

EXPLORER

- MY-APP
- .next
- node_modules
- public
- scripts
- src
 - app
 - components
 - lib
 - sanity
 - lib
 - schemaTypes
 - TS index.ts
 - TS product.ts
 - TS env.ts

TS products U

TS index.ts U X

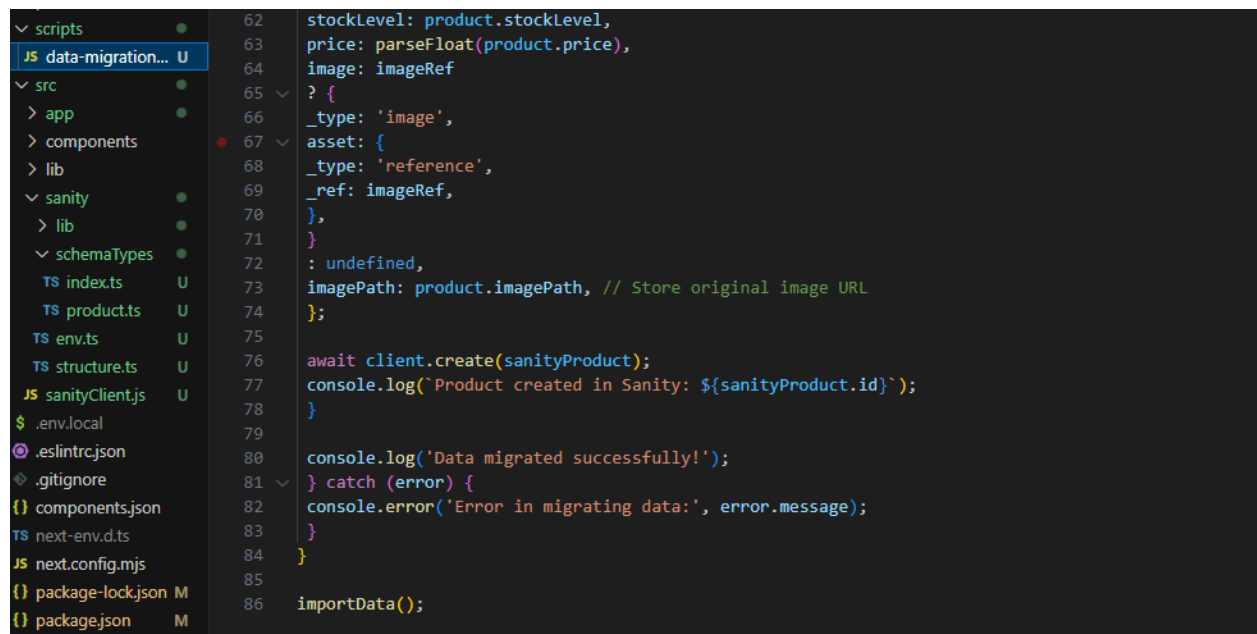
JS sanityClient.js U

src > sanity > schemaTypes > TS index.ts > schema > types

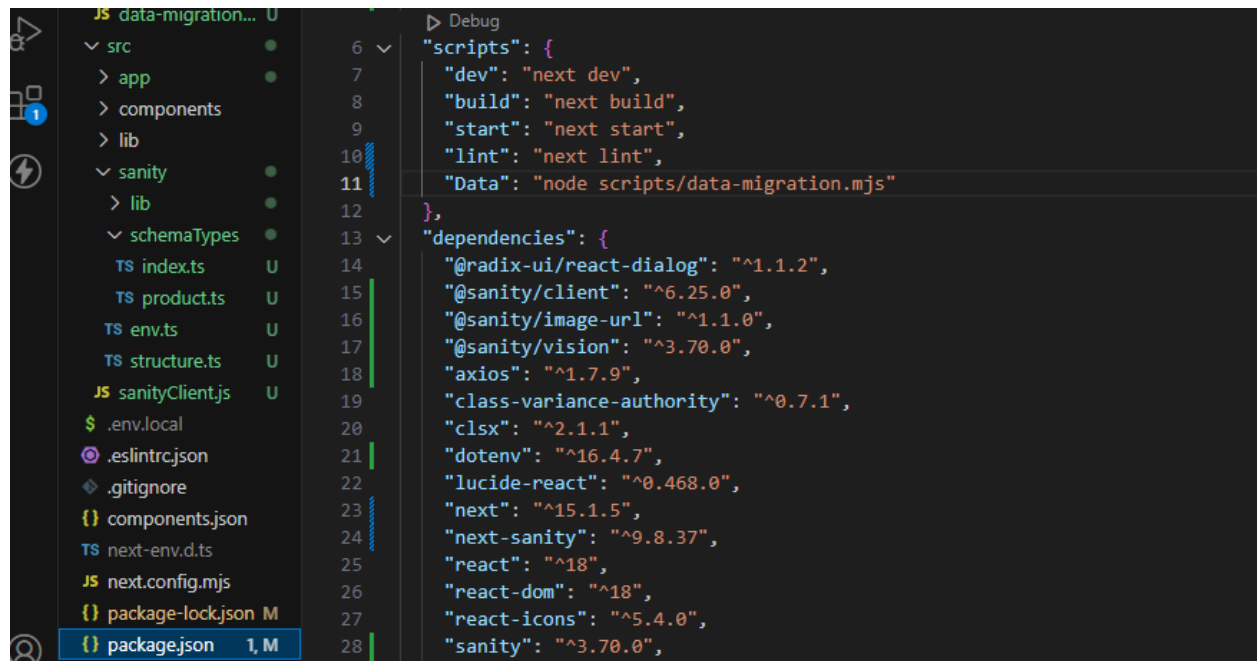
```
1 import { type SchemaTypeDefinition } from 'sanity'
2 import product from './product'
3
4 export const schema: { types: SchemaTypeDefinition[] } = {
5   types: [product],
6 }
7
```

```
scripts > JS data-migration.mjs > uploadImageToSanity
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  async function uploadImageToSanity(imageUrl) {
22    try {
23      console.log(`Uploading image: ${imageUrl}`);
24      const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
25      const buffer = Buffer.from(response.data);
26      const asset = await client.assets.upload('image', buffer, {
27        filename: imageUrl.split('/').pop(),
28      });
29      console.log(`Image uploaded successfully: ${asset._id}`);
30      return asset._id;
31    } catch (error) {
32      console.error(`Failed to upload image:`, imageUrl, error.message);
33    }
34  }
```

```
35  }
36
37  async function importData() {
38    try {
39      console.log('Migrating data, please wait...');
40
41      // Fetch products from the API
42      const response = await axios.get('https://template-0-beta.vercel.app/api/product');
43      const products = response.data;
44
45      console.log('Products fetched:', products);
46
47      for (const product of products) {
48        let imageRef = null;
49
50        if (product.imagePath) {
51          imageRef = await uploadImageToSanity(product.imagePath);
52        }
53
54        const sanityProduct = {
55          _type: 'product',
56          id: product.id,
57          name: product.name,
58          category: product.category,
59          description: product.description,
60          discountPercentage: product.discountPercentage,
61          isFeaturedProduct: product.isFeaturedProduct,
62          stockLevel: product.stockLevel,
63          price: parseFloat(product.price),
64          image: imageRef
65        };
66
67        if (imageRef) {
68          sanityProduct.image = imageRef;
69        }
70
71        // Save product to Sanity
72        const createdProduct = await client.create(sanityProduct);
73        console.log(`Product created:`, createdProduct);
74      }
75    } catch (error) {
76      console.error('Error migrating data:', error);
77    }
78  }
```



```
62 stockLevel: product.stockLevel,
63 price: parseFloat(product.price),
64 image: imageRef
65 ? {
66   _type: 'image',
67   asset: {
68     _type: 'reference',
69     _ref: imageRef,
70   },
71 }
72 : undefined,
73 imagePath: product.imagePath, // Store original image URL
74 };
75
76 await client.create(sanityProduct);
77 console.log(`Product created in Sanity: ${sanityProduct.id}`);
78 }
79
80 console.log('Data migrated successfully!');
81 } catch (error) {
82   console.error('Error in migrating data:', error.message);
83 }
84 }
85
86 importData();
```



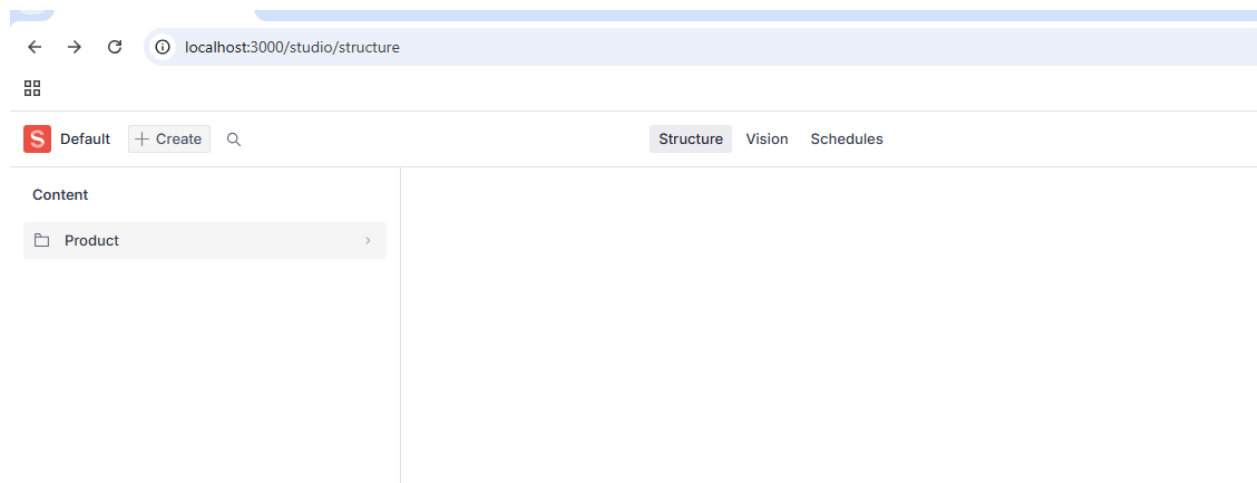
```
6  "scripts": {
7    "dev": "next dev",
8    "build": "next build",
9    "start": "next start",
10   "lint": "next lint",
11   "Data": "node scripts/data-migration.mjs"
12 },
13 "dependencies": {
14   "@radix-ui/react-dialog": "^1.1.2",
15   "@sanity/client": "^6.25.0",
16   "@sanity/image-url": "^1.1.0",
17   "@sanity/vision": "^3.70.0",
18   "axios": "^1.7.9",
19   "class-variance-authority": "^0.7.1",
20   "clsx": "^2.1.1",
21   "dotenv": "^16.4.7",
22   "lucide-react": "^0.468.0",
23   "next": "^15.1.5",
24   "next-sanity": "^9.8.37",
25   "react": "^18",
26   "react-dom": "^18",
27   "react-icons": "^5.4.0",
28   "sanity": "^3.70.0",
```

Command run:

npm install @sanity axios dotenv

Command Run for import Data:

Npm run Data



S

Default

+ Create

Q

StructureVisionSchedules


Content


Product


Product

+ ...


Q Search list


Luxury Flower Bed


Blue Bed


Red Bed


Published 13 hr. ago


White Bed

Solid Bed

Matilda Velvet Bed

Diondre Chair

Leisure Sofa Chair Set

Liberty Center

localhost:3000/studio/structure/product

S

Default

+ Create

Q

StructureVisionSchedules

Tasks

Content


Product


Product


+ ...


Luxury Flower Bed


Q Search list


Luxury Flower Bed


Blue Bed


Red Bed


White Bed


Solid Bed


Matilda Velvet Bed


Diondre Chair


Leisure Sofa Chair Set


Liberty Center

Sleek Modern Table

Replica Table

Alpha Table

Cozy Sofa

High Quality Modern Sofa

Product

Luxury Flower Bed


ID

21

Name

Luxury Flower Bed

Image



Sanity Studio interface showing a query for products.

Dataset: production
API Version: Other
Custom API Version: v2025-01-17
Perspective: raw
Query URL: https://uz8braz2.api.sanity.io/v2025-01-17/data/quer...

QUERY

```
1 *[_type == product]
```

PARAMS

```
1 {  
2  
3 }
```

RESULT

```
[...] 353 items  
0: {...} 7 properties  
  members: [...] 1 item  
    0: p0mKF662Z  
    _rev: qQpkQmFIZ2DSQkhvXLCNXf  
    _type: system.group  
    _id: __groups.administrator  
    _updatedAt: 2025-01-17T19:41:11Z  
  grants: [...] 2 items  
    0: {...} 2 properties  
      filter: _id in path("****")  
      permissions: [...] 4 items  
    1: {...} 2 properties  
      filter: (_id in path("****")) && _id in path("drafts.****")  
      permissions: [...] 4 items
```

Execution: 77ms End-to-end: 467ms

Save result as JSON CSV

VS Code editor showing a Next.js application with a Sanity client configuration.

EXPLORER

- MY-APP
 - src
 - app
 - page
 - arrivals
 - asgardsofa
 - blog
 - blogdetail
 - blogdetail...
 - carndetail
 - checkout
 - contact
 - contactinfo
 - footer
 - hero
 - instagram
 - navbar
 - navbar2
 - products
 - products
 - shop
 - shopbar
 - shopservices
 - sidebarr
 - table
 - toppicks
 - studio
 - favicon.ico
 - globals.css
 - layout.tsx
 - page.tsx
 - components

page.tsx

```
src > app > page > products > page.tsx > ProductCards > products.map() callback  
1 "use client";  
2 import React, { useEffect, useState } from "react";  
3 import { createClient } from "@sanity/client";  
4 import Image from "next/image";  
5  
6 // Sanity client configuration  
7 const sanity = createClient({  
8   projectId: "uz8braz2",  
9   dataset: "production",  
10  apiVersion: "2021-08-31",  
11  useCdn: true,  
12 });  
13  
14 interface Product {  
15   _id: string;  
16   title: string;  
17   price: number;  
18   description: string;  
19   discountPercentage: number;  
20   imageUrl: string;  
21 }  
22  
23 const ProductCards: React.FC = () => {  
24   const [products, setProducts] = useState<Product[]>([]);  
25   const [cart, setCart] = useState<Product[]>([]);  
26  
27   const fetchProducts = async () => {  
28     try {  
29       const query = `  
30         *[_type == "product"] {  
31           _id,  
32           title,  
33           price,  
34           description,  
35           discountPercentage,  
36           "imageUrl": image.asset->url  
37         }  
38       `;  
39  
40       const data: Product[] = await sanity.fetch(query);  
41     } catch (error) {  
42       console.error("Error fetching products:", error);  
43     }  
44   }  
45 }  
46  
47 export default ProductCards;
```

```
src > app > page > products > page.jsx > ProductCards > products.map() callback
23 const ProductCards: React.FC = () => {
27   const fetchProducts = async () => {
38   };
39   const data: Product[] = await sanity.fetch(query);
40   setProducts(data);
41   } catch (error) {
42     console.error("Error fetching products:", error);
43   }
44 };
45
46 const addToCart = (product: Product) => {
47   setCart(prevCart => [...prevCart, product]);
48   alert(`${product.title} has been added to your cart!`);
49 };
50
51 // const truncateDescription = (description: string) => {
52 //   return description.length > 100 ? `${description.substring(0, 100)}...` : description;
53 // };
54 const truncateDescription = (description: string | undefined) => {
55   if (!description) {
56     return ""; // Return an empty string if description is undefined or null
57   }
58   return description.length > 100
59     ? `${description.substring(0, 100)}...`
60     : description;
61 };
62
63 useEffect(() => {
64   fetchProducts();
65 }, []);
66
67 return (
68   <div className="p-4">
69     <h2 className="text-center">Products From API's Data:</h2>
70     <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
```

```
72   </h2>
73   <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
74     {products.map((product) => (
75       <div
76         key={product._id}
77         className="bg-white shadow-md rounded-lg p-4 hover:shadow-lg transition-shadow duration-300"
78       >
79         <Image
80           src={product.imageUrl || "/placeholder.png"} // Fallback image if no URL
81           alt={product.title}
82           width={300}
83           height={300}
84           className="w-full h-48 object-cover rounded-md"
85         />
86         <div className="mt-4">
87           <h2 className="text-lg font-semibold">{product.title}</h2>
88           <p className="text-slate-800 mt-2 text-sm">
89             {truncateDescription(product.description)}
90           </p>
91           <div className="flex justify-between items-center mt-4">
92             <div
93               <p className="text-slate font-bold">${product.price}</p>
94               (product.discountPercentage > 0 && {
95                 <p className="text-sm text-green-600">
96                   (product.discountPercentage)% OFF
97                 </p>
98               })
99             </div>
100           </div>
101           <button
102             className="mt-4 bg-blue-600 text-white py-2 rounded-md hover:bg-blue-700"
103             onClick={() => addToCart(product)}
104           >
105             Add To Cart
106           </button>
107         </div>
108       </div>
109     )
110   )}
111 </div>
```

```
File Edit Selection View Go ... my-app
EXPLORER
  MY-APP
    src
      app
        page
          arrivals
          asgardsofa
          blog
          biogdetail
          biogdetailh...
          carddetail
          cart
          checkout
          contact
          contactinfo
          footer
          hero
          instagram
          navbar
          navbar2
          products
            page.jsx U
          shop
          shopbar
          shoppeservices
          sidebarrr
          table
          toppicks
          studio
          faviconico
          globalcss
          layout.jsx
          page.jsx M
          components
        OUTLINE
        TIMELINE
  next.config.js M
  data-migration.mjs U
  page.jsx _products U X
  page.jsx _app M

src > app > page > products > page.jsx > ProductCards > products.map() callback
23 const ProductCards: React.FC = () => {
113
114   /* Cart Summary */
115   <div className="mt-4 bg-slate-100 p-6 rounded-lg shadow-md">
116     <h2 className="text-lg font-black text-red-800">Cart Summary</h2>
117     {cart.length > 0 ? (
118       <ul className="space-y-4">
119         {cart.map((item, index) => (
120           <li
121             key={index}
122             className="flex justify-between items-center bg-white shadow-sm p-4 rounded-md"
123           >
124             <div>
125               <p className="font-medium text-slate-900">{item.title}</p>
126               <p className="text-sm text-blue-600">
127                 ${item.price.toFixed(2)}
128               </p>
129             </div>
130             <img
131               src={item.imageUrl}
132               alt={item.title}
133               width={50}
134               height={50}
135               className="rounded-md"
136             />
137           </li>
138         )
139       )
140     ) : (
141       <p className="text-black text-center">
142         Your Cart Is Empty. Please Add Products.
143       </p>
144     )
145   }</div>
146 }
147
148 export default ProductCards;
149
150
```

