

Made By: Shahzain Ali

Institute: GIAIC

Slot: Saturday

1. API Integration Process

- **API Used:** Product API (<https://next-ecommerce-template-4.vercel.app/api/product>)
 - **Steps:**
 1. Used Axios to fetch product data from the API.
 2. Divided the products into batches (batch size: 10) for stability and efficiency.
 3. Processed each product according to the Sanity CMS schema and uploaded it using `client.create()`.
 4. Uploaded images to Sanity separately using `uploadImageToSanity` and associated the `_id` with the product.
-

2. Schema Adjustments

- Made the following changes to the Sanity CMS schema for better functionality:
 - Added `stockLevel` to manage product stock.
 - Added `category` to define product categories like "Chair" or "Sofa".
-

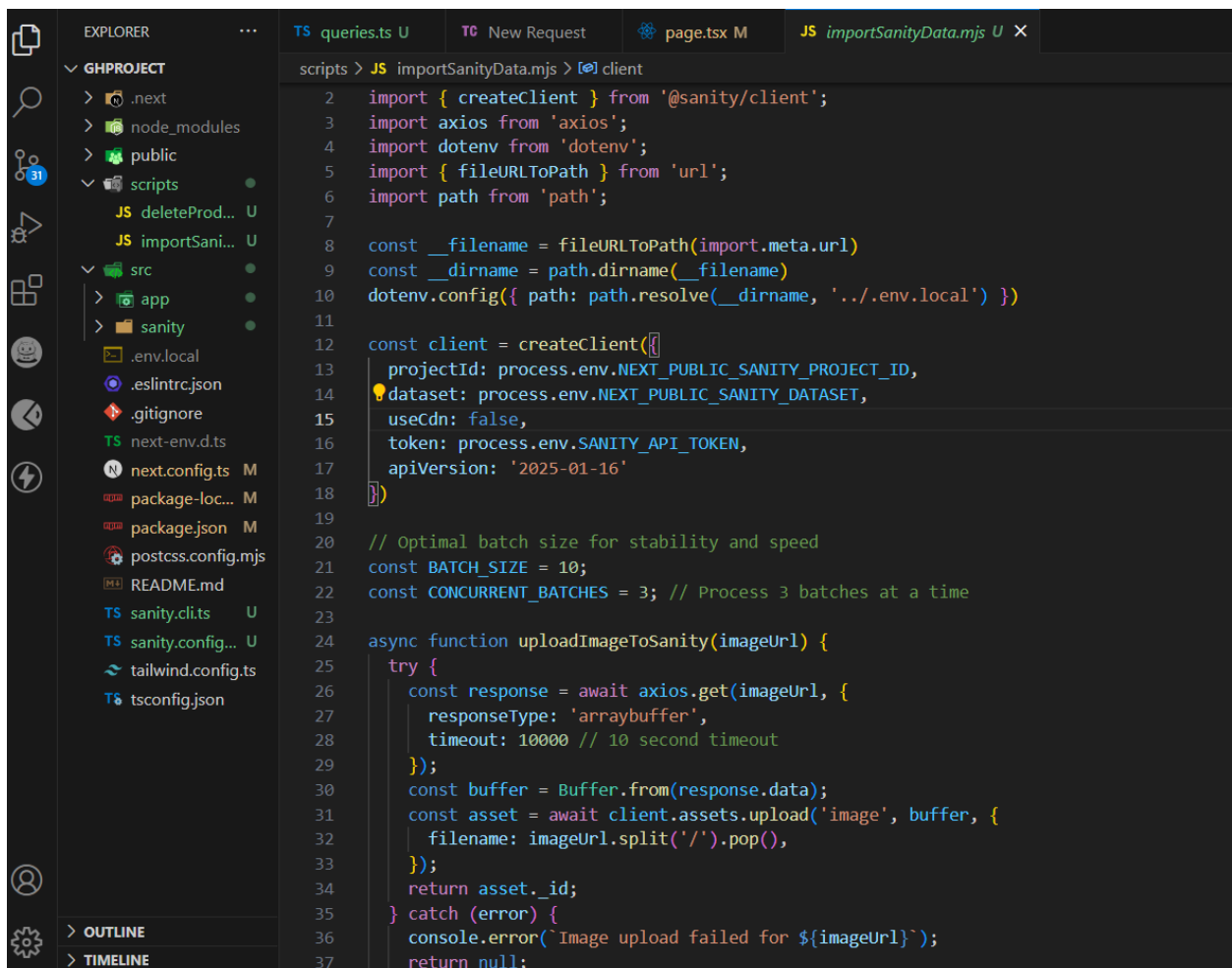
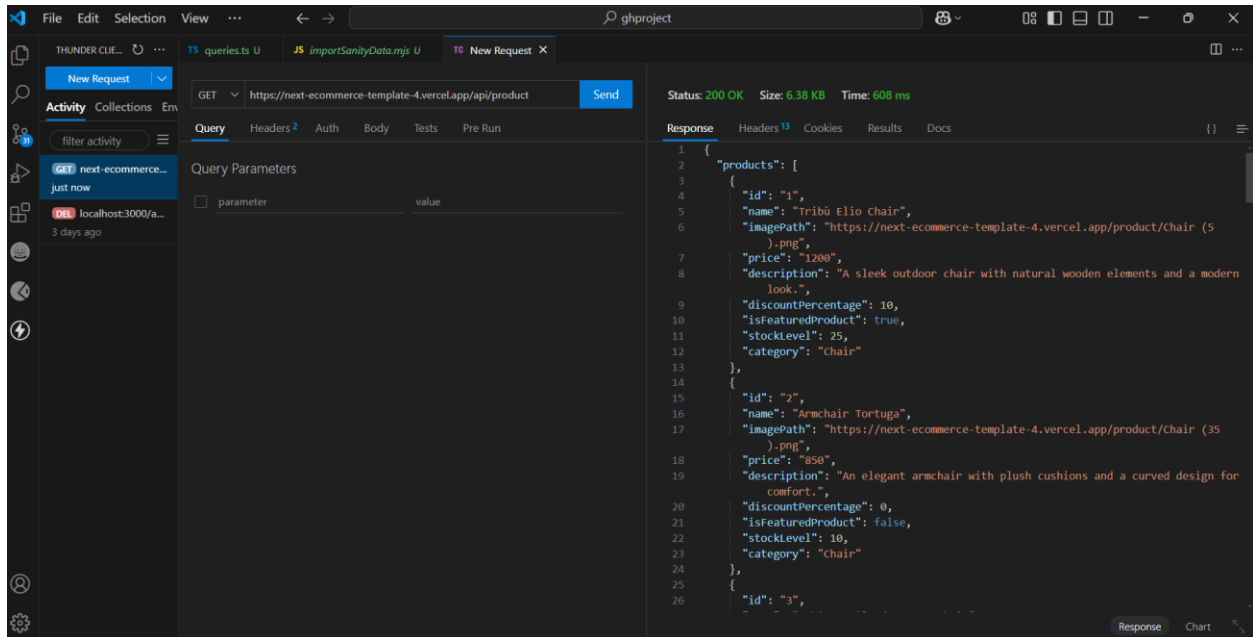
3. Migration Steps and Tools

- **Steps:**
 1. Fetched data from the old system (API).
 2. Uploaded products and their images in batches to Sanity CMS.
 3. Deleted old products in bulk using a Sanity query and `transaction.delete()`.
- **Tools Used:** Axios (for API requests), Sanity Client (for CMS integration).

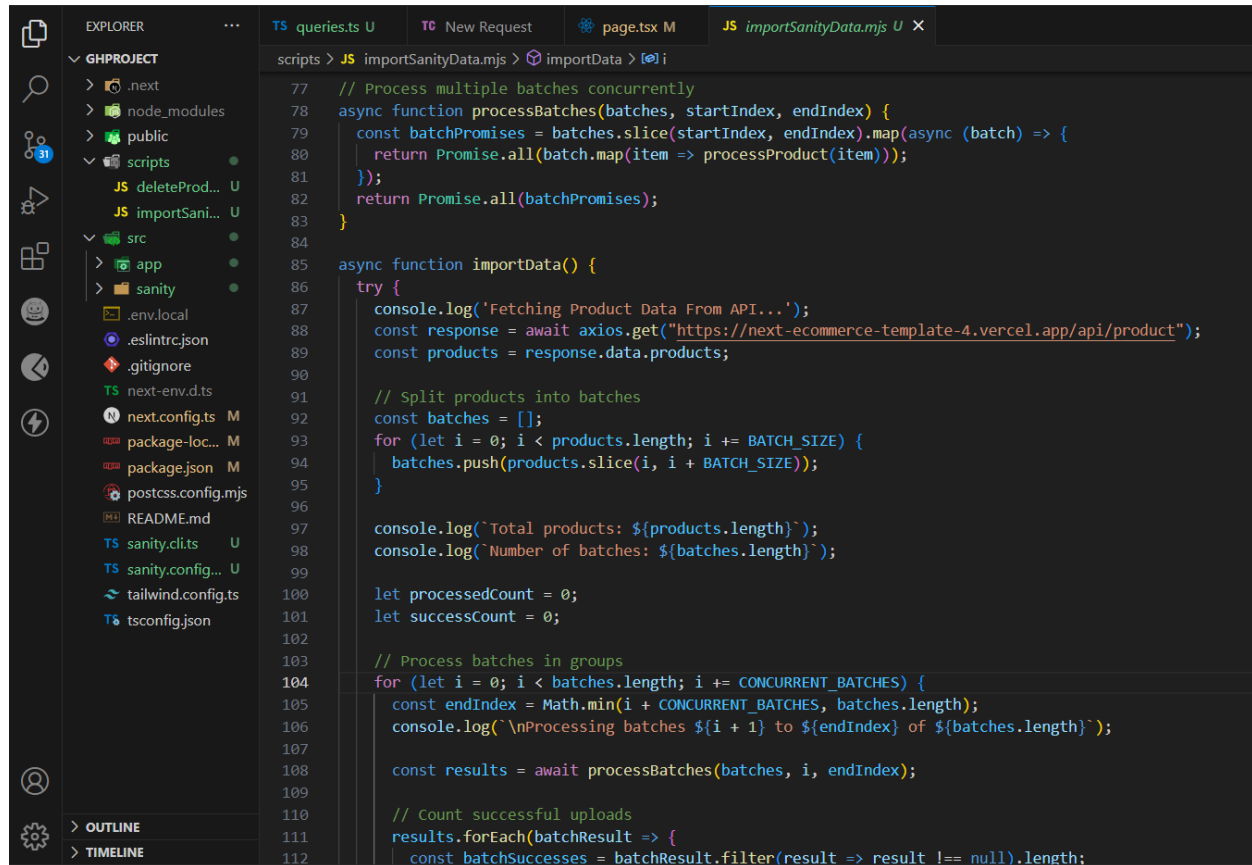
Outcome:

- Successfully migrated the data, populated Sanity CMS, and optimized the process with batch uploads and a cleanup setup.

Made By: Shahzain Ali
Institute: GIAIC
Slot: Saturday

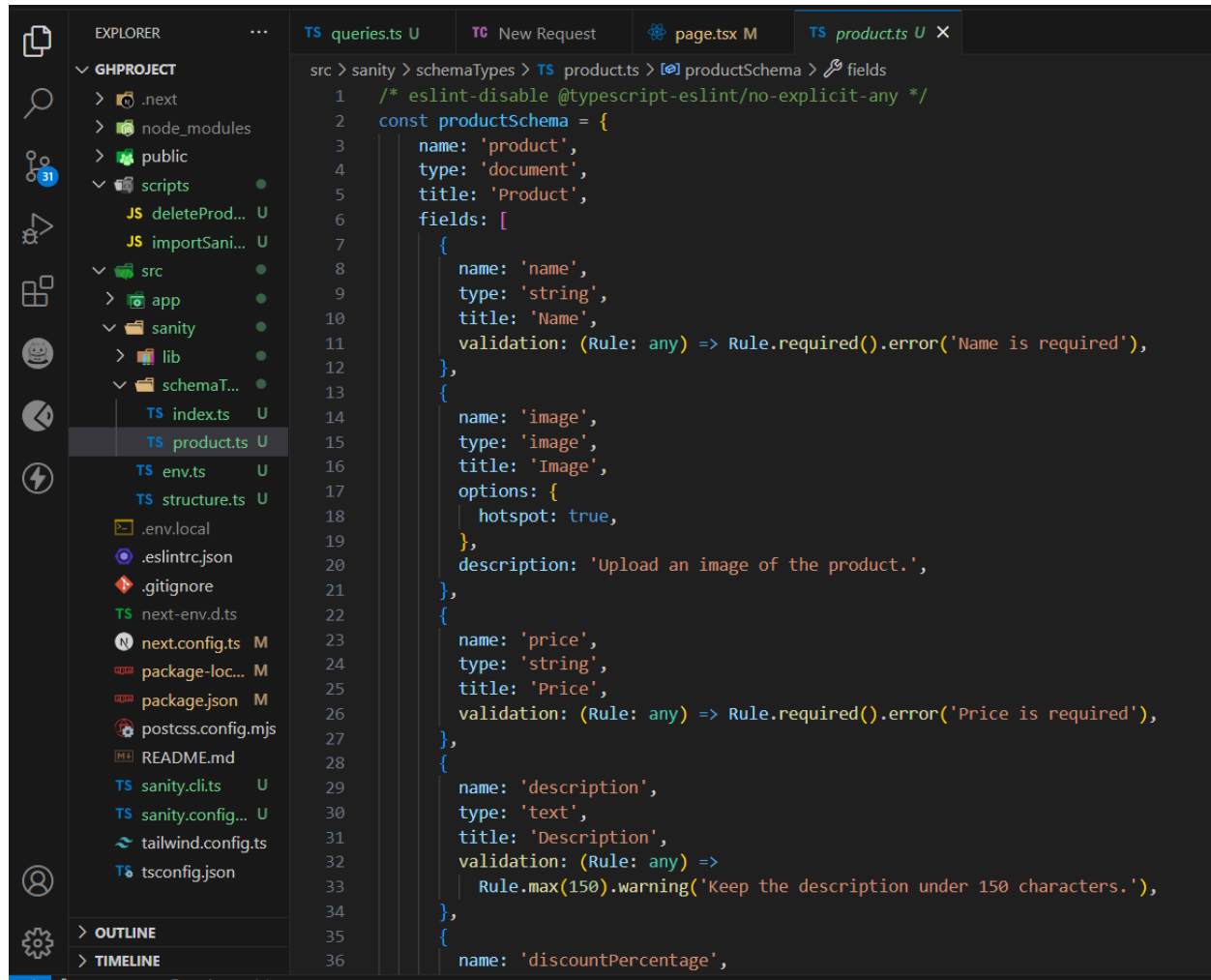


Made By: Shahzain Ali
Institute: GIAIC
Slot: Saturday



```
77 // Process multiple batches concurrently
78 async function processBatches(batches, startIndex, endIndex) {
79   const batchPromises = batches.slice(startIndex, endIndex).map(async (batch) => {
80     return Promise.all(batch.map(item => processProduct(item)));
81   });
82   return Promise.all(batchPromises);
83 }
84
85 async function importData() {
86   try {
87     console.log('Fetching Product Data From API...');
88     const response = await axios.get("https://next-ecommerce-template-4.vercel.app/api/product");
89     const products = response.data.products;
90
91     // Split products into batches
92     const batches = [];
93     for (let i = 0; i < products.length; i += BATCH_SIZE) {
94       batches.push(products.slice(i, i + BATCH_SIZE));
95     }
96
97     console.log(`Total products: ${products.length}`);
98     console.log(`Number of batches: ${batches.length}`);
99
100     let processedCount = 0;
101     let successCount = 0;
102
103     // Process batches in groups
104     for (let i = 0; i < batches.length; i += CONCURRENT_BATCHES) {
105       const endIndex = Math.min(i + CONCURRENT_BATCHES, batches.length);
106       console.log(`\nProcessing batches ${i + 1} to ${endIndex} of ${batches.length}`);
107
108       const results = await processBatches(batches, i, endIndex);
109
110       // Count successful uploads
111       results.forEach(batchResult => {
112         const batchSuccesses = batchResult.filter(result => result !== null).length;
```

Made By: Shahzain Ali
Institute: GIAIC
Slot: Saturday



The image shows a screenshot of the Visual Studio Code editor. On the left, the Explorer sidebar displays the file structure of a project named 'GHPROJECT'. The file 'product.ts' is selected under the 'src' directory. The main editor area shows the content of 'product.ts', which defines a schema for a product. The code includes a TypeScript comment to disable a specific ESLint rule, followed by a constant declaration for 'productSchema'. This schema defines fields for 'name', 'image', 'price', 'description', and 'discountPercentage', each with its type, title, and validation rules. The 'name' field is required. The 'image' field is of type 'image' and has a 'hotspot' option set to true, with a description 'Upload an image of the product.'. The 'price' field is required. The 'description' field has a maximum length warning of 150 characters. The 'discountPercentage' field is also defined.

```
src > sanity > schemaTypes > TS product.ts > [E] productSchema > fields
1  /* eslint-disable @typescript-eslint/no-explicit-any */
2  const productSchema = {
3      name: 'product',
4      type: 'document',
5      title: 'Product',
6      fields: [
7          {
8              name: 'name',
9              type: 'string',
10             title: 'Name',
11             validation: (Rule: any) => Rule.required().error('Name is required'),
12         },
13         {
14             name: 'image',
15             type: 'image',
16             title: 'Image',
17             options: {
18                 hotspot: true,
19             },
20             description: 'Upload an image of the product.',
21         },
22         {
23             name: 'price',
24             type: 'string',
25             title: 'Price',
26             validation: (Rule: any) => Rule.required().error('Price is required'),
27         },
28         {
29             name: 'description',
30             type: 'text',
31             title: 'Description',
32             validation: (Rule: any) =>
33                 Rule.max(150).warning('Keep the description under 150 characters.'),
34         },
35         {
36             name: 'discountPercentage',
```

Made By: Shahzain Ali

Institute: GIAIC

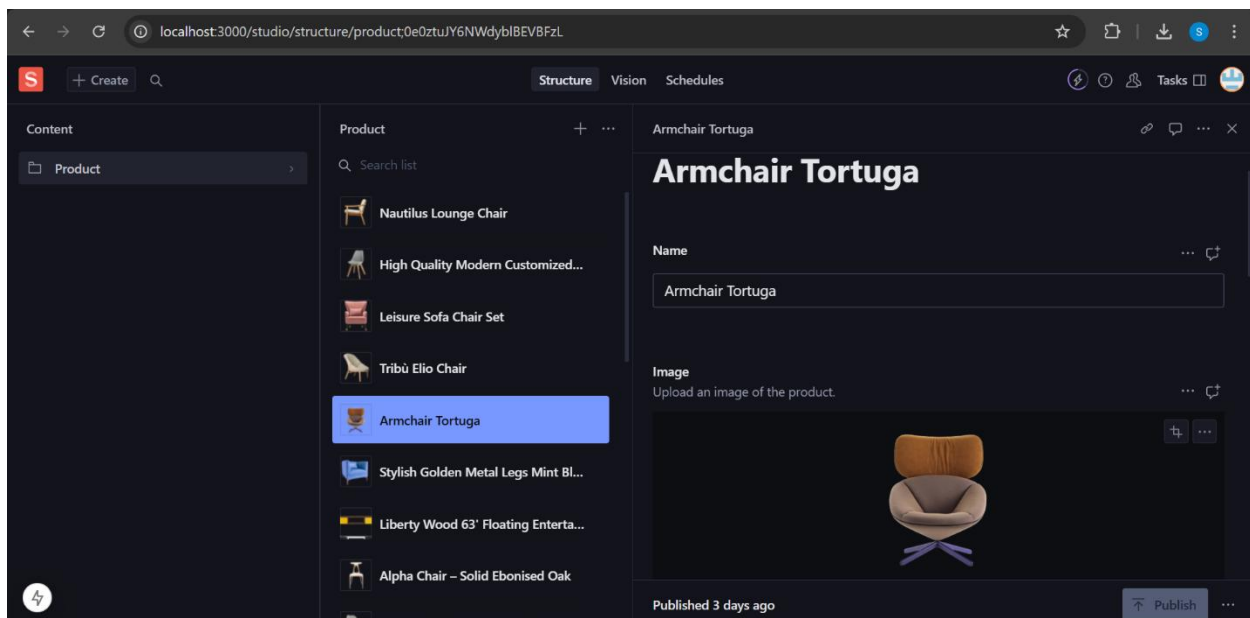
Slot: Saturday

```
D:\Governer House Projects\E-commerce Project\ghproject>npm run import-data

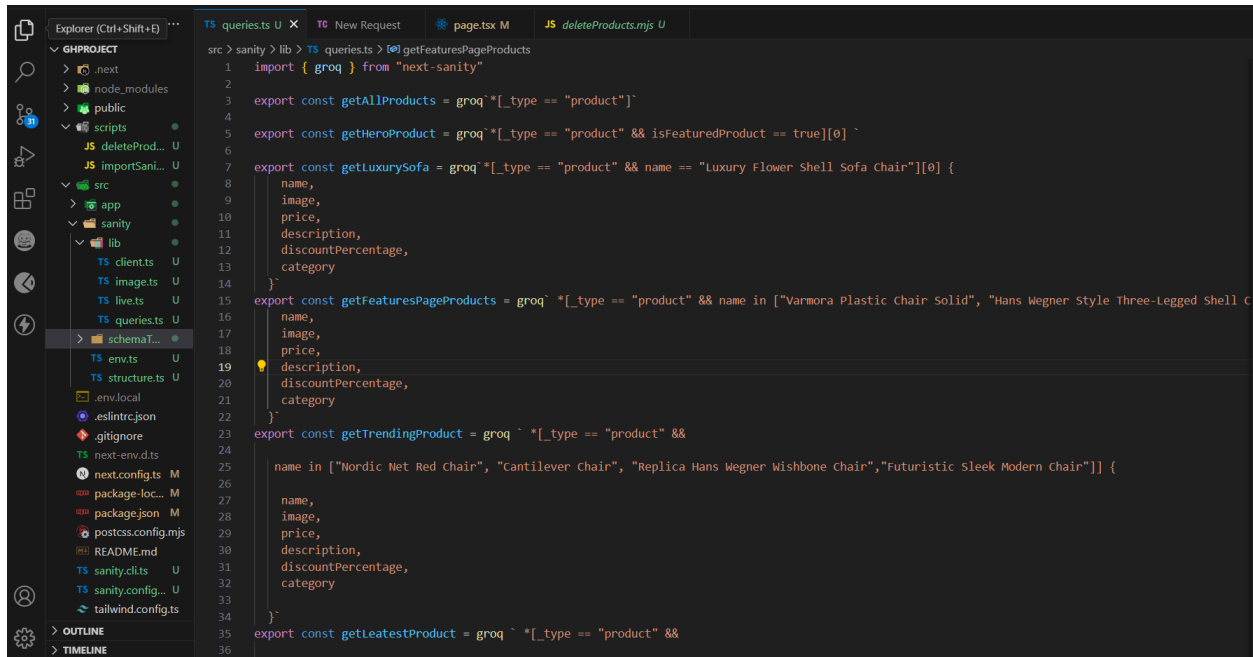
> ghproject@0.1.0 import-data
> ts-node scripts/importSanityData.mjs

Fetching Product Data From API...
Total products: 21
Number of batches: 3

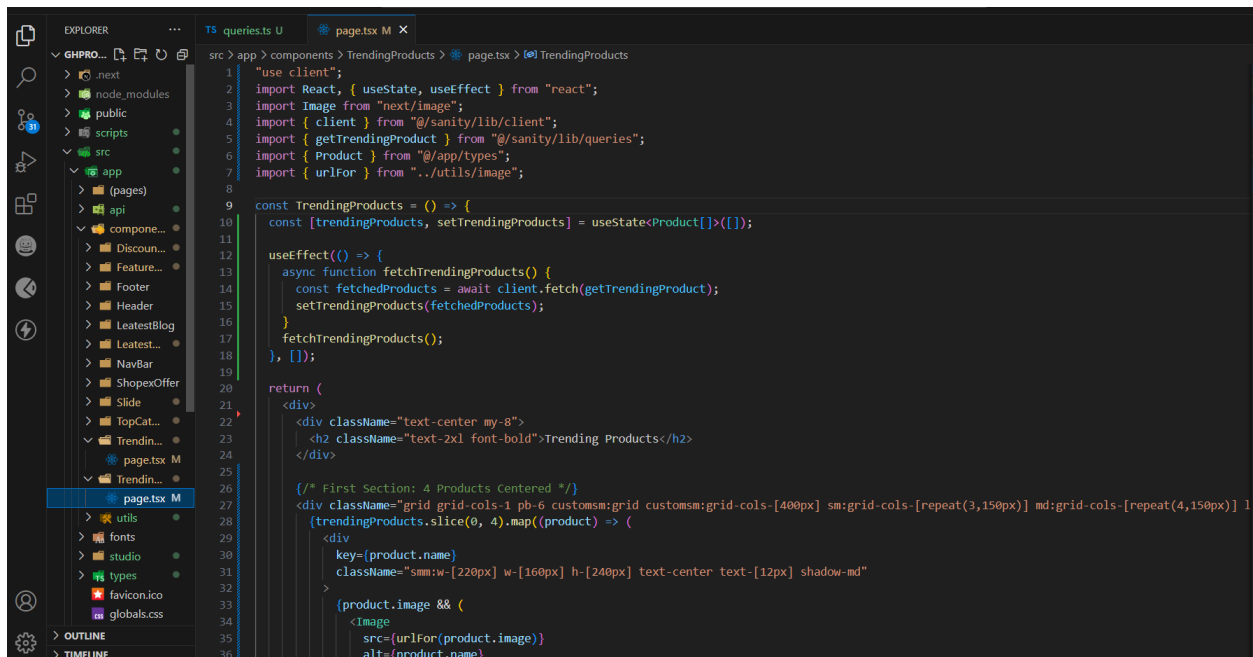
Processing batches 1 to 3 of 3
✓ Uploaded: Replica Hans Wegner Wishbone Chair
✓ Uploaded: Nautilus Lounge Chair
✓ Uploaded: Alpha Chair – Solid Ebonised Oak
✓ Uploaded: Liberty Wood 63' Floating Entertainment Center
✓ Uploaded: Hans Wegner Style Three-Legged Shell Chair
✓ Uploaded: Cantilever Chair
✓ Uploaded: Uchiwa Quilted Lounge Chair
✓ Uploaded: High Quality Modern Customized Plastic Chair
✓ Uploaded: Sobuy Blue Folding Chair Wooden Padded
✓ Uploaded: Matilda Velvet Chair – Pink
✓ Uploaded: Futuristic Sleek Modern Chair
✓ Uploaded: Leisure Sofa Chair Set
✓ Uploaded: Nordic Net Red Chair
✓ Uploaded: Tribù Elio Chair
✓ Uploaded: Rapson Thirty-Nine Guest Chair
✓ Uploaded: Cozy Armchair
✓ Uploaded: Diondre Chair – Tuft Button – Acrylic Legs
✓ Uploaded: Varmora Plastic Chair Solid
✓ Uploaded: Stylish Golden Metal Legs Mint Blue Fabric Velvet Sofa Leisure Armchair
✓ Uploaded: Armchair Tortuga
✓ Uploaded: Luxury Flower Shell Sofa Chair
Progress: 21/21 (21 successful)
```



Made By: Shahzain Ali
Institute: GIAIC
Slot: Saturday



```
src > sanity > lib > TS queries.ts > 00 getFeaturesPageProducts
1  import { groq } from "next-sanity"
2
3  export const getAllProducts = groq`*[_type == "product"]`
4
5  export const getHeroProduct = groq`*[_type == "product" && isFeaturedProduct == true][0]`
6
7  export const getLuxurySofa = groq`*[_type == "product" && name == "Luxury Flower Shell Sofa Chair"][0] {
8    name,
9    image,
10   price,
11   description,
12   discountPercentage,
13   category
14 }`
15
16 export const getFeaturesPageProducts = groq`*[_type == "product" && name in ["Varmora Plastic Chair Solid", "Hans Wegner Style Three-Legged Shell C
17   name,
18   image,
19   price,
20   description,
21   discountPercentage,
22   category
23 ]`
24
25 export const getTrendingProduct = groq`*[_type == "product" &&
26   name in ["Nordic Net Red Chair", "Cantilever Chair", "Replica Hans Wegner Wishbone Chair", "Futuristic Sleek Modern Chair"] {
27   name,
28   image,
29   price,
30   description,
31   discountPercentage,
32   category
33 }`
34
35 export const getLeatestProduct = groq`*[_type == "product" &&
36
```



```
src > app > components > TrendingProducts > page.tsx > 00 TrendingProducts
1  "use client";
2  import React, { useState, useEffect } from "react";
3  import Image from "next/image";
4  import { client } from "@sanity/lib/client";
5  import { getTrendingProduct } from "@sanity/lib/queries";
6  import { Product } from "@app/types";
7  import { urlFor } from "../utils/image";
8
9  const TrendingProducts = () => {
10   const [trendingProducts, setTrendingProducts] = useState<Product[]>([]);
11
12   useEffect(() => {
13     async function fetchTrendingProducts() {
14       const fetchedProducts = await client.fetch(getTrendingProduct);
15       setTrendingProducts(fetchedProducts);
16     }
17     fetchTrendingProducts();
18   }, []);
19
20   return (
21     <div>
22       <div className="text-center my-8">
23         <h2 className="text-2xl font-bold">Trending Products</h2>
24       </div>
25
26       <div className="grid grid-cols-1 pb-6 customsm:grid customsm:grid-cols-[400px] sm:grid-cols-[repeat(3,150px)] md:grid-cols-[repeat(4,150px)] lg:grid-cols-[repeat(5,150px)]">
27         <div className="grid grid-cols-1 pb-6 customsm:grid customsm:grid-cols-[400px] sm:grid-cols-[repeat(3,150px)] md:grid-cols-[repeat(4,150px)] lg:grid-cols-[repeat(5,150px)]">
28           <div className="grid grid-cols-1 pb-6 customsm:grid customsm:grid-cols-[400px] sm:grid-cols-[repeat(3,150px)] md:grid-cols-[repeat(4,150px)] lg:grid-cols-[repeat(5,150px)]">
29             <div>
30               <div>
31                 <div>
32                   <div>
33                     <div>
34                       <div>
35                         <div>
36
```

Slot: Saturday




Latest Products