“Day 3 Integration Report”

Api Integration and Data migration

Table of Contents

1. Setting Up Environment Variables

First, let’s set up our environment variables. Create a `.env.local` file in your project root if it doesn't already exist. Add the following variables:

```
NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id
NEXT_PUBLIC_SANITY_DATASET=production
SANITY_API_TOKEN=your_sanity_token
```

Remember, variables prefixed with `NEXT_PUBLIC_` will be exposed to the browser, so be cautious about what you prefix.

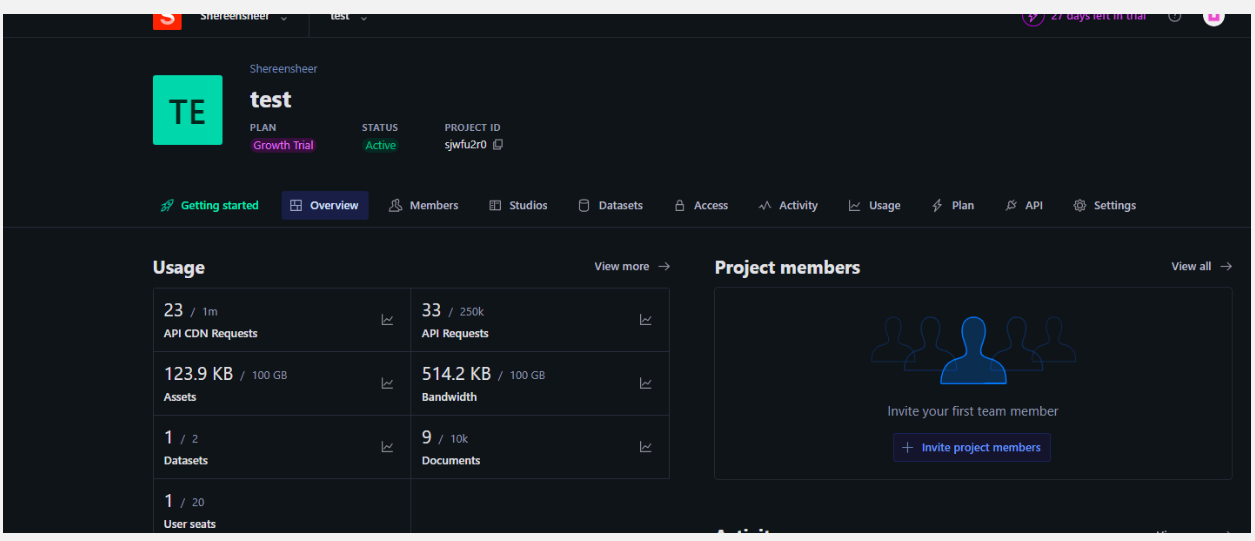
2. Obtaining Sanity Project ID and API Token

Project ID

To find your Sanity project ID:

1. Log in to your Sanity account at https://www.sanity.io/manage
2. Select your project
3. In the project dashboard, you’ll see the project ID listed

Use this ID for the `NEXT_PUBLIC_SANITY_PROJECT_ID` in your `.env.local` file.



Shereensheer

test

PLAN Growth Trial

STATUS Active

PROJECT ID sjwfu2r0

Usage

23 / 1m API CDN Requests	33 / 250k API Requests
123.9 KB / 100 GB Assets	514.2 KB / 100 GB Bandwidth
1 / 2 Datasets	9 / 10k Documents
1 / 20 User seats	

Project members

API Token

To generate a Sanity API token:

1. Go to https://www.sanity.io/manage and select your project

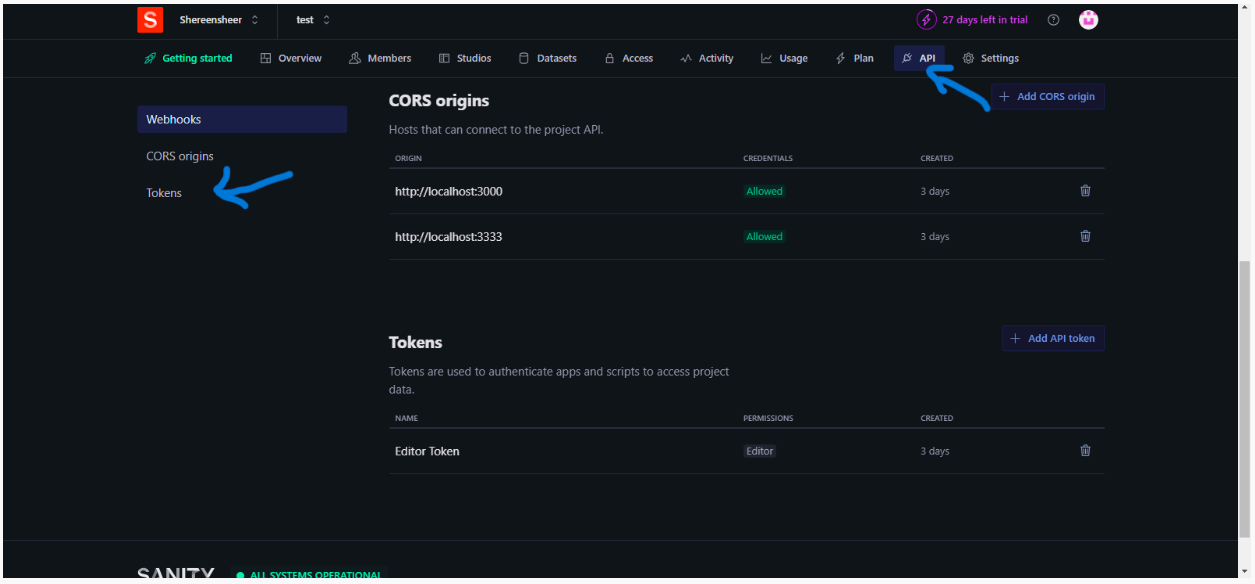
2. Navigate to the “`API`" tab

3. Under "`Tokens`," click "`Add API token`"

4. Give your token a name and select the appropriate permissions

(usually "`Editor`" for full read/write access)

5. Copy the generated token



CORS origins

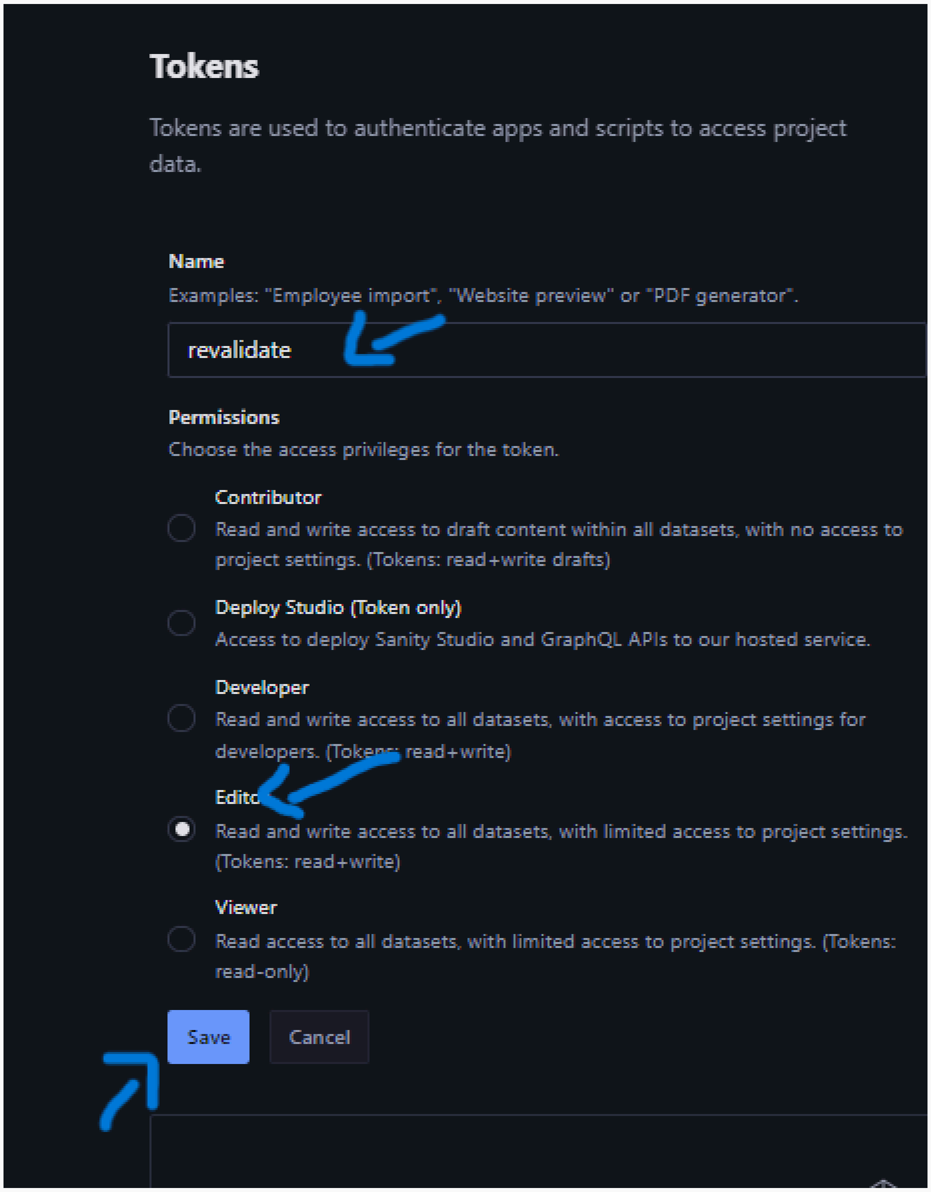
Hosts that can connect to the project API.

ORIGIN	CREDENTIALS	CREATED
http://localhost:3000	Allowed	3 days
http://localhost:3333	Allowed	3 days

Tokens

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

NAME	PERMISSIONS	CREATED
Editor Token	Editor	3 days



Tokens

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

Name

Examples: "Employee import", "Website preview" or "PDF generator".

revalidate

Permissions

Choose the access privileges for the token.

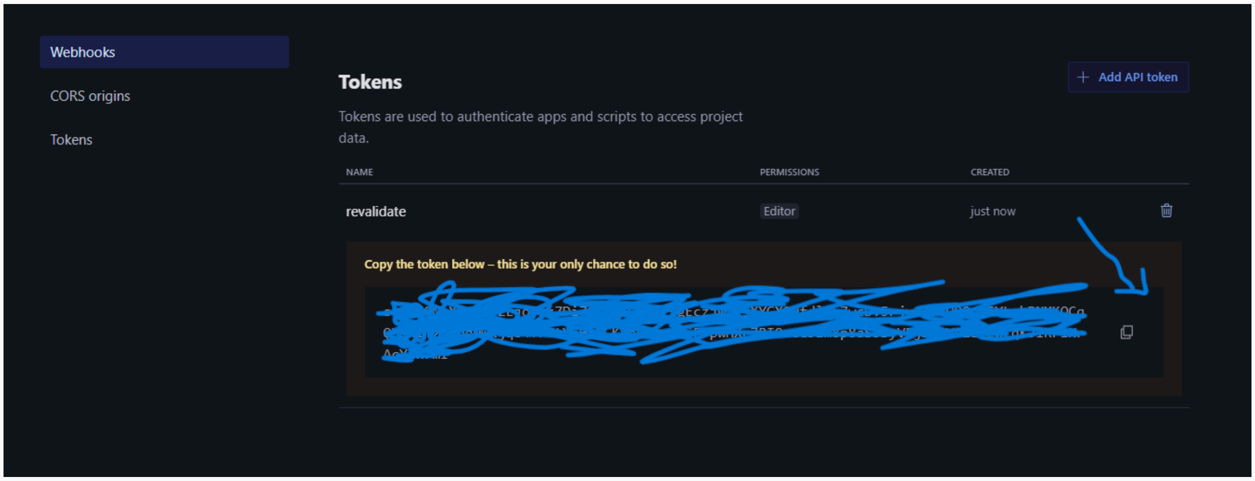
○ **Contributor**
Read and write access to draft content within all datasets, with no access to project settings. (Tokens: read+write drafts)

○ **Deploy Studio (Token only)**
Access to deploy Sanity Studio and GraphQL APIs to our hosted service.

○ **Developer**
Read and write access to all datasets, with access to project settings for developers. (Tokens: read+write)

● **Editor**
Read and write access to all datasets, with limited access to project settings. (Tokens: read+write)

○ **Viewer**
Read access to all datasets, with limited access to project settings. (Tokens: read-only)



Tokens

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

NAME	PERMISSIONS	CREATED
revalidate	Editor	just now

Copy the token below – this is your only chance to do so!

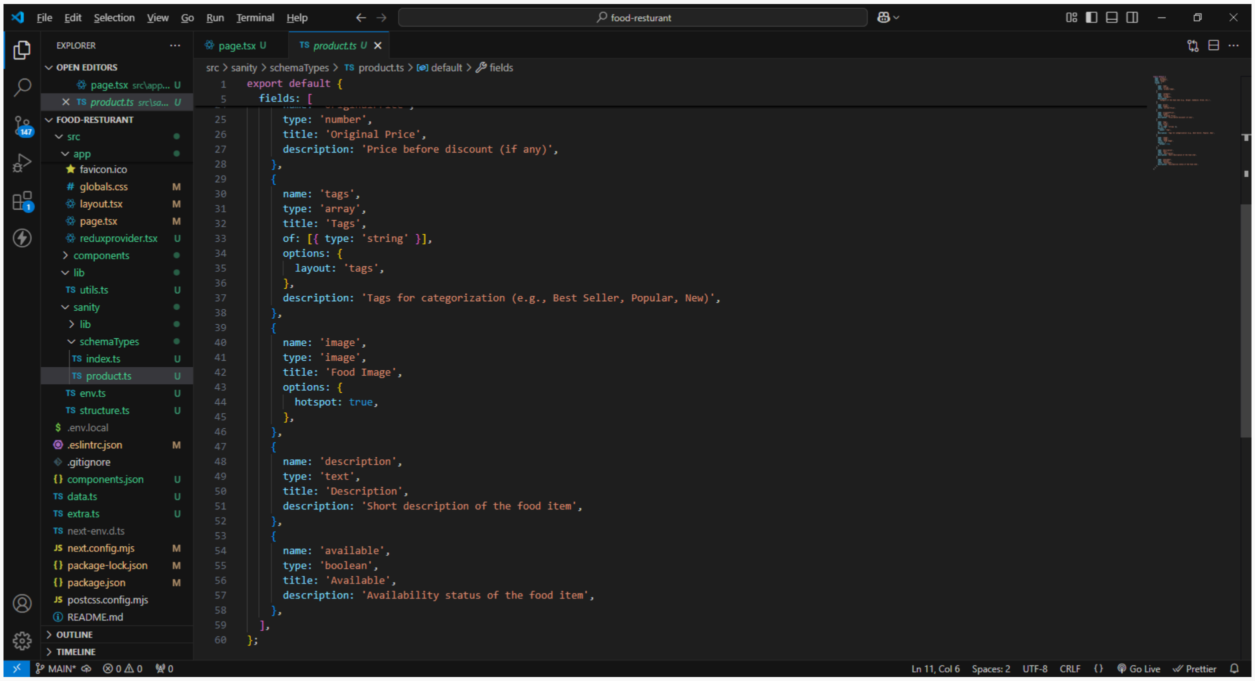
[redacted]

Use this token for the `` `SANITY_API_TOKEN` `` in your `` `.env.local` `` file.**3.**

Creating the Sanity Schema

Now, let's create a schema for our products. In your Sanity schema folder (usually `` `sanity/schemaTypes` ``), create a new file called `` `product.ts` ``:

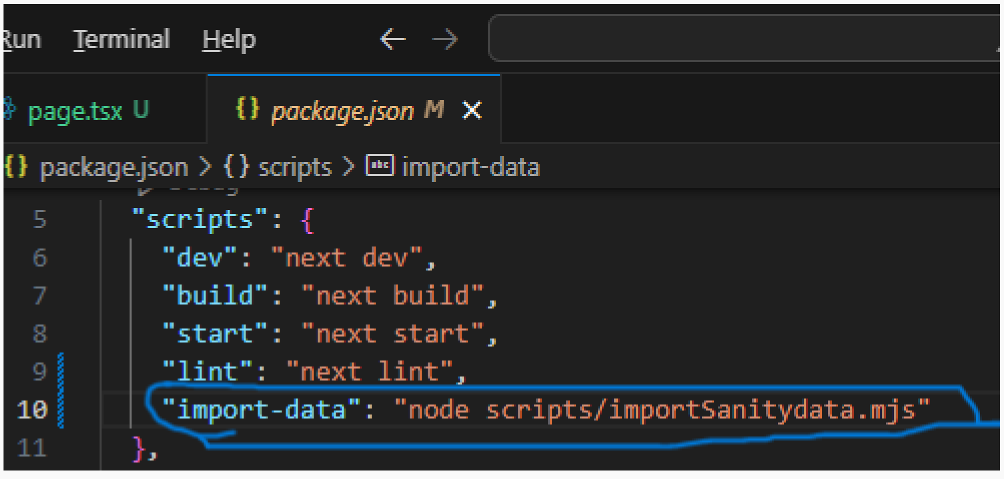
Then, update your `` `sanity/schemaTypes/index.ts` `` file to include the new product schema:



```ts
export default {
  fields: [
        type: 'number',
        title: 'Original Price',
        description: 'Price before discount (if any)',
      },
      {
        name: 'tags',
        type: 'array',
        title: 'Tags',
        of: [{ type: 'string' }],
        options: {
          layout: 'tags',
        },
        description: 'Tags for categorization (e.g., Best Seller, Popular, New)',
      },
      {
        name: 'image',
        type: 'image',
        title: 'Food Image',
        options: {
          hotspot: true,
        },
      },
      {
        name: 'description',
        type: 'text',
        title: 'Description',
        description: 'Short description of the food item',
      },
      {
        name: 'available',
        type: 'boolean',
        title: 'Available',
        description: 'Availability status of the food item',
      },
    ],
};
```

Now, let's install the necessary packages. Run the following command in your terminal:

```
npm install @sanity/client
```



```
"scripts": {
  "dev": "next dev",
  "build": "next build",
  "start": "next start",
  "lint": "next lint",
  "import-data": "node scripts/importSanitydata.mjs"
},
```

5. Running the Import Script

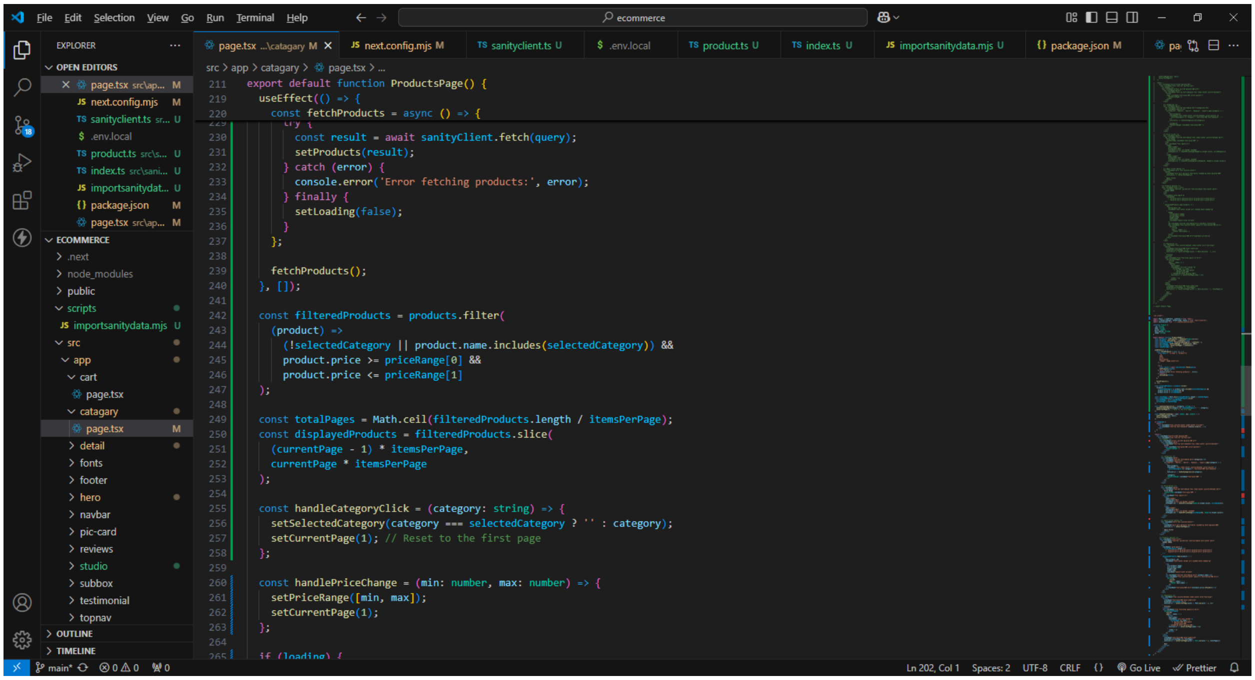
To run the import script, we need to add a new script to our `` `package.json` `` file. Open your `` `package.json` `` and add the following to the `` `"scripts"` `` section:

```
"scripts": {
 "dev": "next dev --turbopack",

 "build": "next build",

 "start": "next start",

 "lint": "next lint",

 "import-data": "node scripts/importSanityData.mjs"

}
```

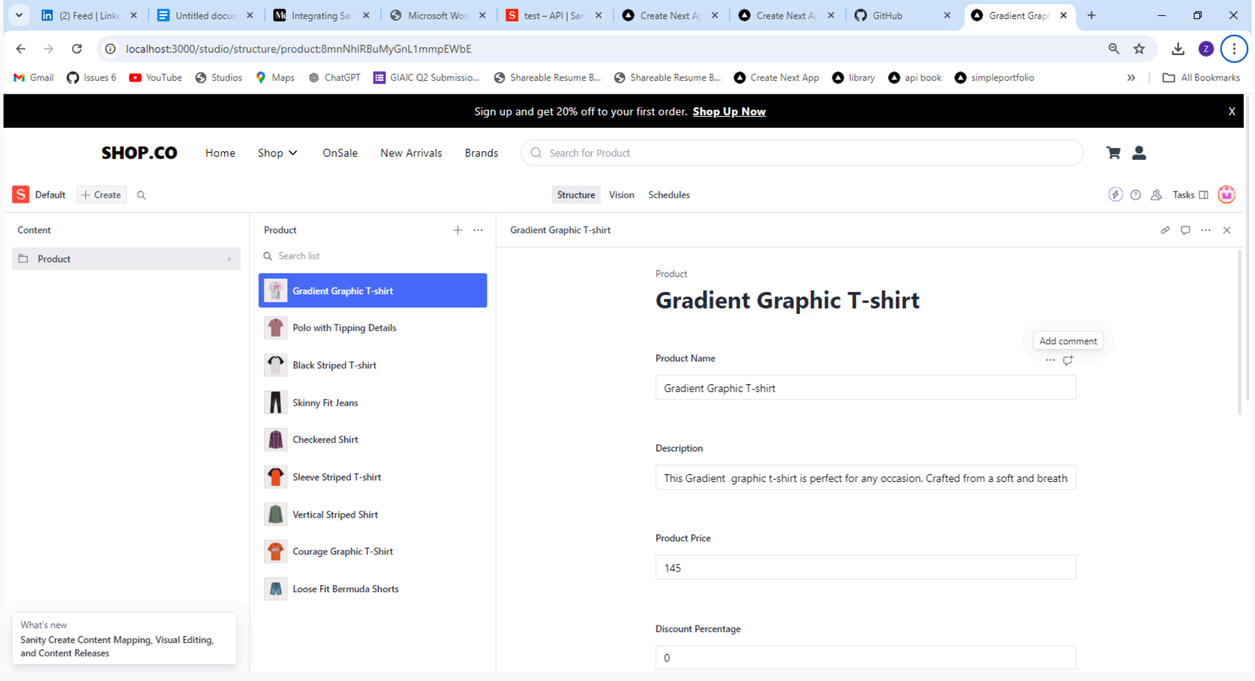
Now you can run the import script using:

```
npm run import-data
```

This script will fetch products from the FakeStoreAPI, upload any associated images to Sanity’s asset store, and then create new product documents in your dataset sanity.



```
export default function ProductsPage() {
  useEffect(() => {
    const fetchProducts = async () => {
      try {
        const result = await sanityClient.fetch(query);
        setProducts(result);
      } catch (error) {
        console.error('Error fetching products:', error);
      } finally {
        setLoading(false);
      }
    };

    fetchProducts();
  }, []);

  const filteredProducts = products.filter(
    (product) =>
      (!selectedCategory || product.name.includes(selectedCategory)) &&
      product.price >= priceRange[0] &&
      product.price <= priceRange[1]
  );

  const totalPages = Math.ceil(filteredProducts.length / itemsPerPage);
  const displayedProducts = filteredProducts.slice(
    (currentPage - 1) * itemsPerPage,
    currentPage * itemsPerPage
  );

  const handleCategoryClick = (category: string) => {
    setSelectedCategory(category === selectedCategory ? '' : category);
    setCurrentPage(1); // Reset to the first page
  };

  const handlePriceChange = (min: number, max: number) => {
    setPriceRange([min, max]);
    setCurrentPage(1);
  };

  if (loading) {
```



localhost:3000/studio/structure/product;8mnNhIRBuMyGnL1mmpEWbE

Sign up and get 20% off to your first order. **Shop Up Now**

SHOP.CO

Content

- Product

Product

- Gradient Graphic T-shirt
- Polo with Tipping Details
- Black Striped T-shirt
- Skinny Fit Jeans
- Checkered Shirt
- Sleeve Striped T-shirt
- Vertical Striped Shirt
- Courage Graphic T-Shirt
- Loose Fit Bermuda Shorts

Gradient Graphic T-shirt

Product

Gradient Graphic T-shirt

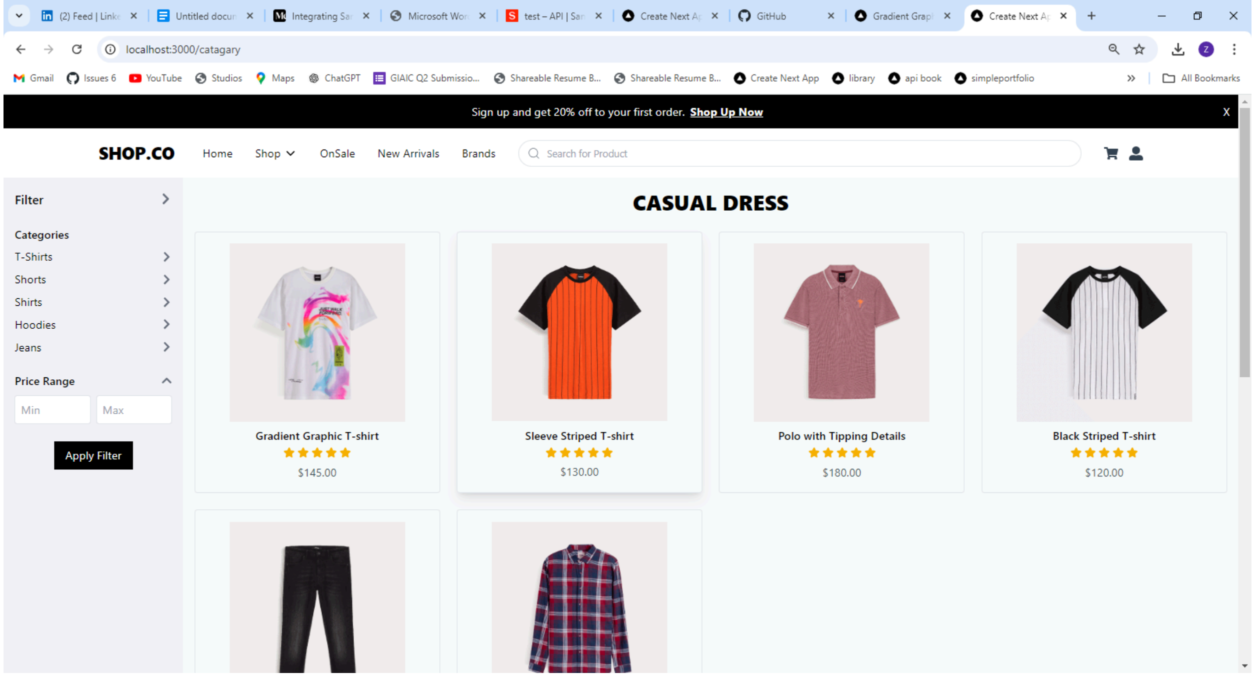
Product Name: Gradient Graphic T-shirt

Description: This Gradient graphic t-shirt is perfect for any occasion. Crafted from a soft and breath

Product Price: 145

Discount Percentage: 0

What's new
Sanity Create Content Mapping, Visual Editing, and Content Releases



localhost:3000/catagary

Sign up and get 20% off to your first order. **Shop Up Now**

SHOP.CO

Filter

Categories
- T-Shirts
- Shorts
- Shirts
- Hoodies
- Jeans

Price Range

CASUAL DRESS

Gradient Graphic T-shirt ★★★★★ $145.00	Sleeve Striped T-shirt ★★★★★ $130.00	Polo with Tipping Details ★★★★★ $180.00	Black Striped T-shirt ★★★★★ $120.00