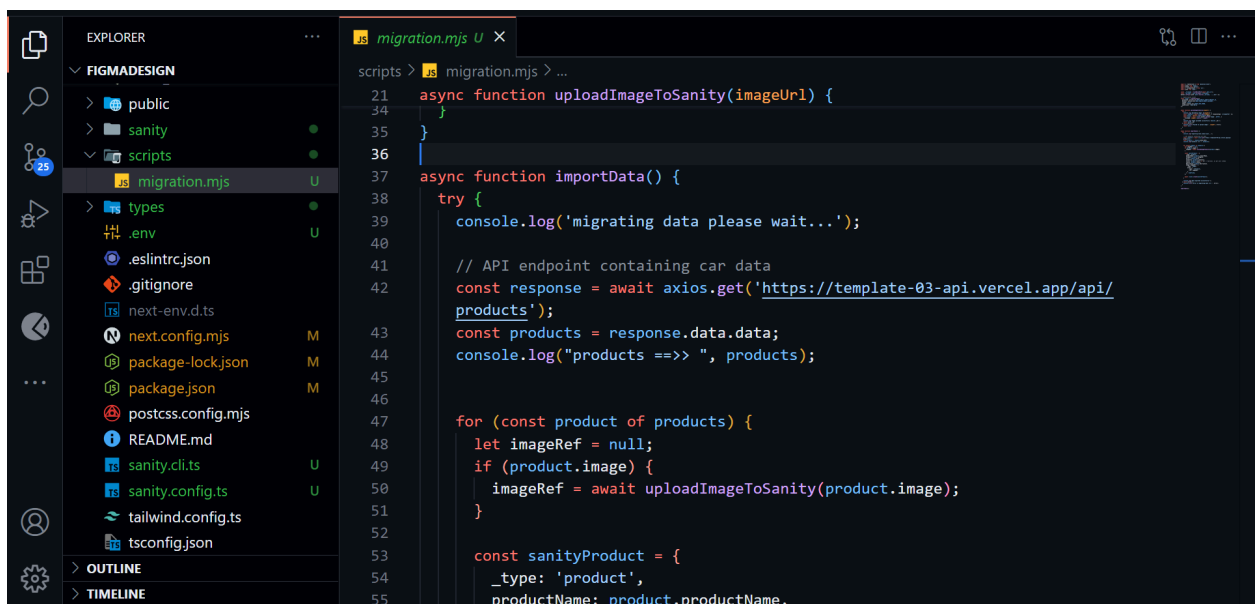


# API Integration & Data Migration

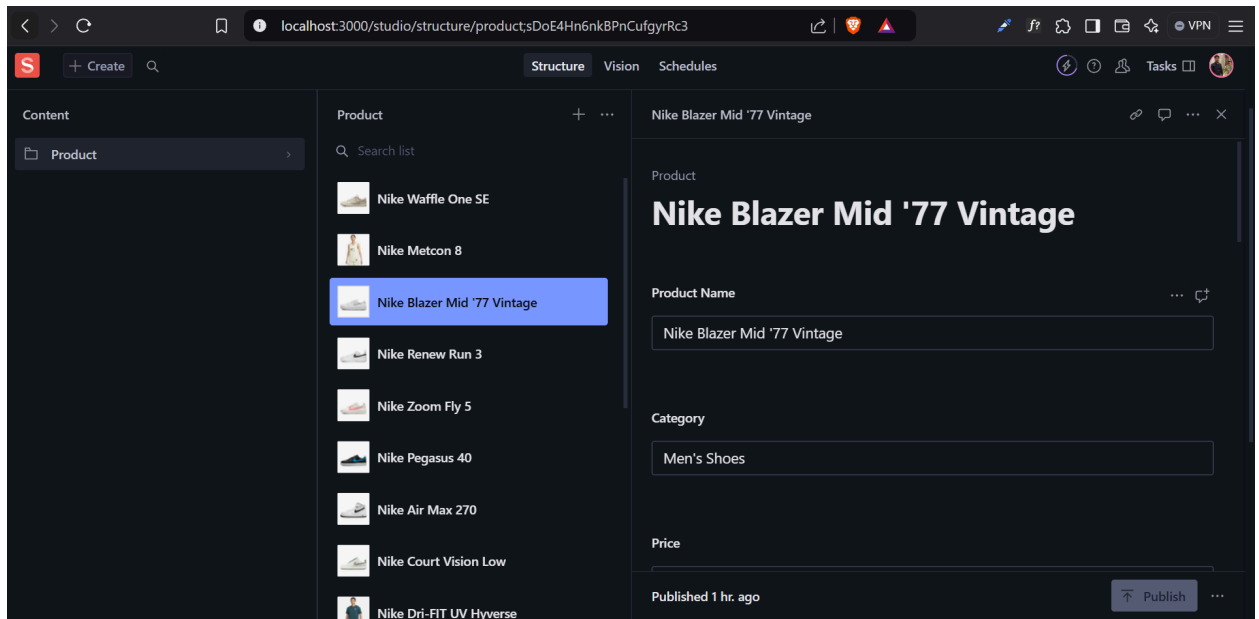
## Products API Integration:

- I reviewed APIs given in the document and also explored the whole document to know the process.
- Then I made a migration file to fetch data from API and insert that data into Sanity CMS.



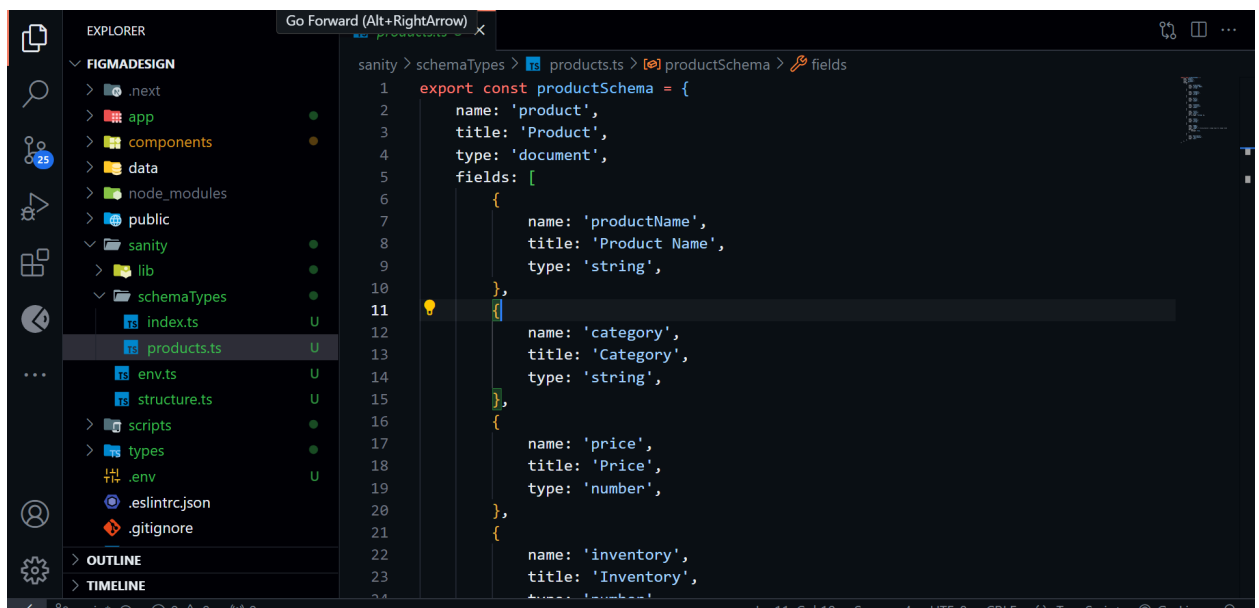
The screenshot shows a Visual Studio Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'public', 'sanity', and 'scripts'. The 'scripts' folder contains a file named 'migration.mjs'. The code editor shows the content of 'migration.mjs', which is a JavaScript file with the following code:

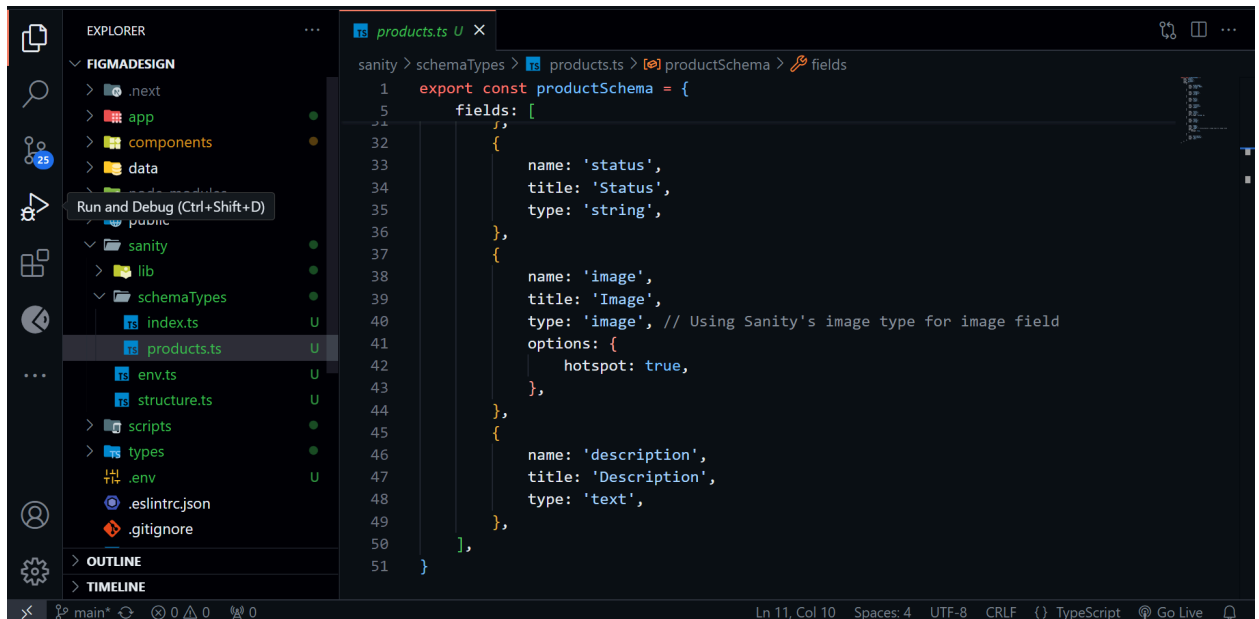
```
21 async function uploadImageToSanity(imageUrl) {
34 }
35 }
36
37 async function importData() {
38   try {
39     console.log('migrating data please wait...');
40
41     // API endpoint containing car data
42     const response = await axios.get('https://template-03-api.vercel.app/api/products');
43     const products = response.data.data;
44     console.log("products ==>> ", products);
45
46     for (const product of products) {
47       let imageRef = null;
48       if (product.image) {
49         imageRef = await uploadImageToSanity(product.image);
50       }
51
52       const sanityProduct = {
53         _type: 'product',
54         productName: product.productName,
```



## Schema Adjustments:

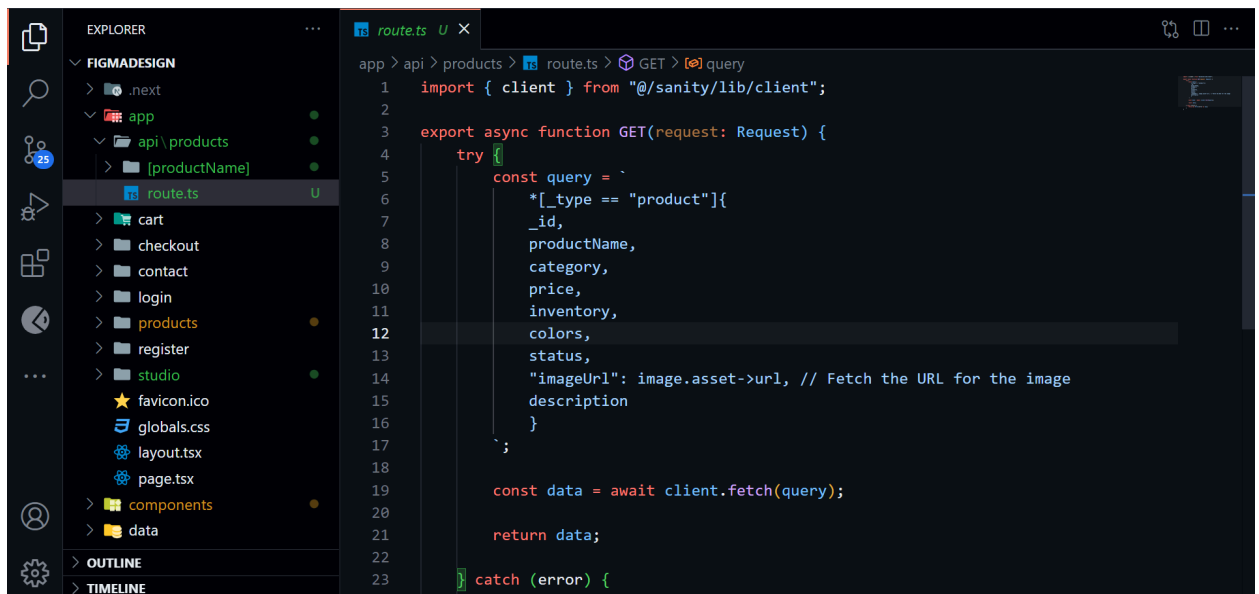
- I adjusted my sanity schema because the products that are provided in API are different from my products schema, so i made some changes.





## API Creation for products fetching:

- Now, I created one endpoint called /products where I fetched all the products that are stored in my Sanity CMS.



## Fetching Products to Frontend:

- Then, I fetched that API I created on the frontend to display all the products. Below is the image showcasing how I did.

```

1  import Image from 'next/image'
2  import React from 'react'
3  import Filter from '@public/images/Filters.png'
4  import Categories from '../ui/Categories'
5  import FilterController from '../ui/FilterController'
6  import { IoIosArrowDown } from 'react-icons/io'
7  import Product from '../ui/Product'
8  import { ProductType } from '@types/product'
9
10
11 const Products = async () => {
12   const res = await fetch('http://localhost:3000/api/products');
13
14   const data: ProductType[] = await res.json();
15
16   return (
17     <div className='w-full flex flex-col gap-4 md:px-10 px-4 py-10 over'>
18       <div className='flex items-center w-full justify-between text-primary'>
19         <div className='text-[24px]'>New (500)</div>
20         <div className='md:flex hidden items-center gap-6 text-[16px]'>
21           <div className='flex items-center gap-1'>
22             <div>Hide Filters</div>
23             <div>

```

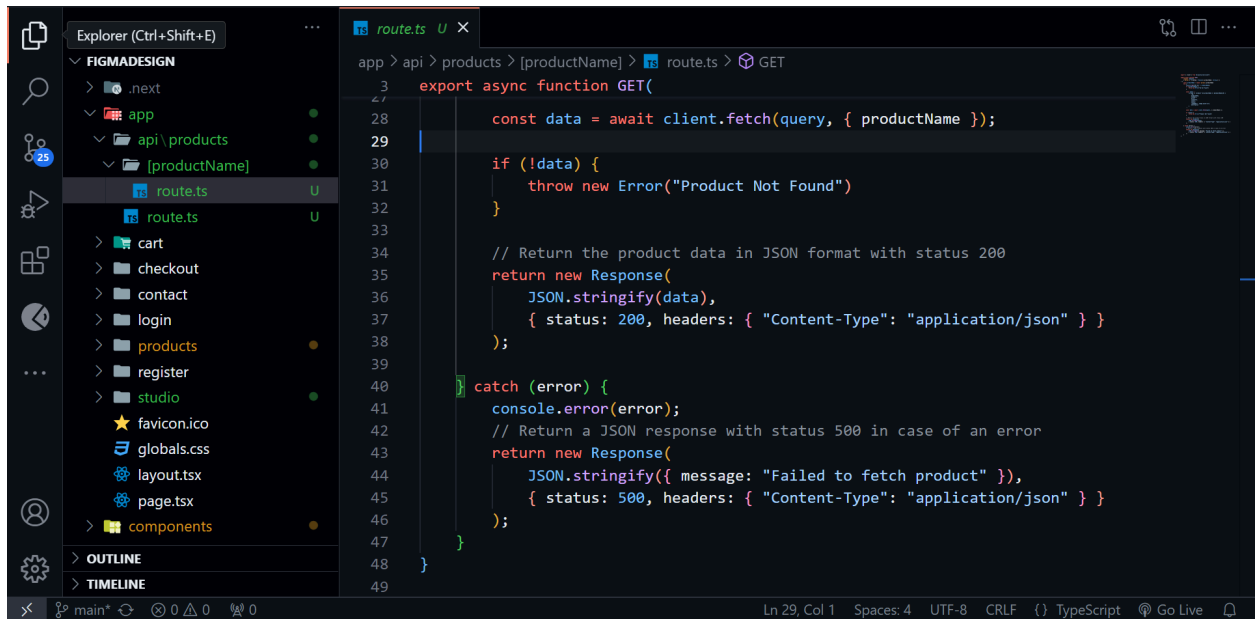
## Individually Fetching Products With Dynamic API Route:

- I made a dynamic API route to fetch a product that matches the name of the product requested by the user.

```

1  import { client } from '@sanity/lib/client';
2
3  export async function GET(
4    request: Request,
5    { params }: { params: Promise<{ productName: string }> }
6  ) {
7    const productName = (await params).productName
8    try {
9      console.log("API Id: ", productName);
10     if (!productName) {
11       throw new Error("Id Not Found")
12     }
13
14     const query = `
15       *[_type == "product" && productName == $productName][0] {
16         _id,
17         productName,
18         category,
19         price,
20         inventory,
21         colors,
22         status,
23         "imageUrl": image.asset->url,

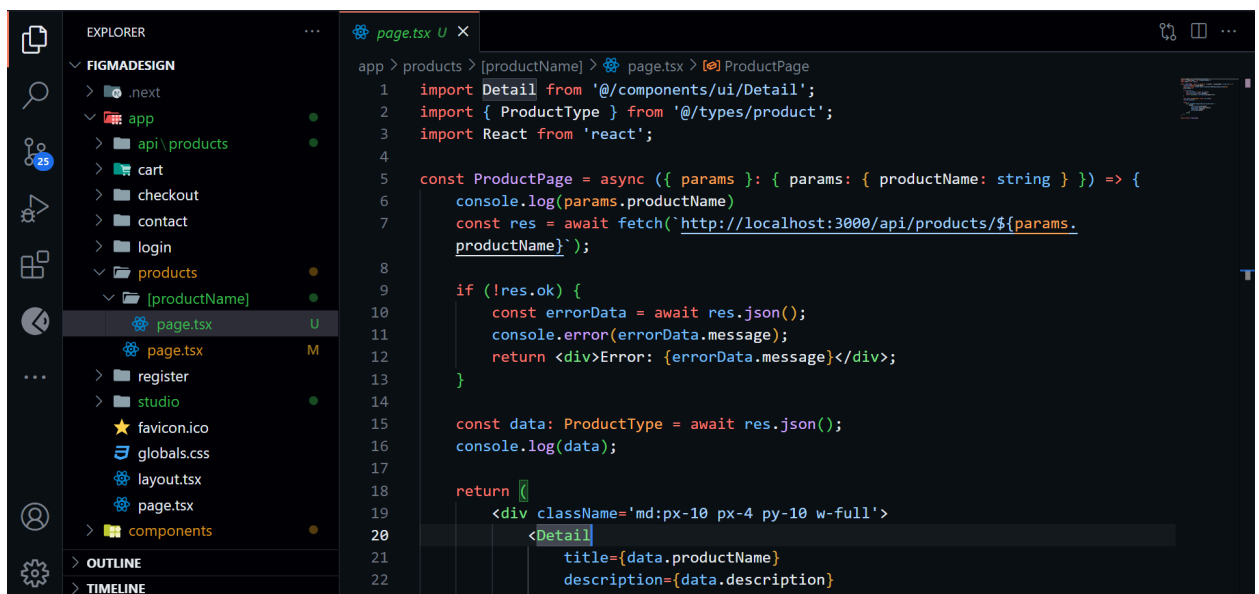
```



```
app > api > products > [productName] > route.ts > GET
3 export async function GET(
28
29
30
31   if (!data) {
32     throw new Error("Product Not Found")
33   }
34
35   // Return the product data in JSON format with status 200
36   return new Response(
37     JSON.stringify(data),
38     { status: 200, headers: { "Content-Type": "application/json" } }
39   );
40 } catch (error) {
41   console.error(error);
42   // Return a JSON response with status 500 in case of an error
43   return new Response(
44     JSON.stringify({ message: "Failed to fetch product" }),
45     { status: 500, headers: { "Content-Type": "application/json" } }
46   );
47 }
48 }
49
```

## Individually Displaying Products on Dynamic Page:

- After creating an API for an individual product I made an route for a dynamic page and fetched that API there to display desired product.



```
app > products > [productName] > page.tsx > ProductPage
1 import Detail from '@components/ui/Detail';
2 import { ProductType } from '@types/product';
3 import React from 'react';
4
5 const ProductPage = async ({ params }: { params: { productName: string } }) => {
6   console.log(params.productName)
7   const res = await fetch(`http://localhost:3000/api/products/${params.productName}`);
8
9   if (!res.ok) {
10     const errorData = await res.json();
11     console.error(errorData.message);
12     return <div>Error: {errorData.message}</div>;
13   }
14
15   const data: ProductType = await res.json();
16   console.log(data);
17
18   return (
19     <div className='md:px-10 px-4 py-10 w-full'>
20       <Detail
21         title={data.productName}
22         description={data.description}
23         image={data.imageUrl}
24       />
25     </div>
26   );
27 }
```