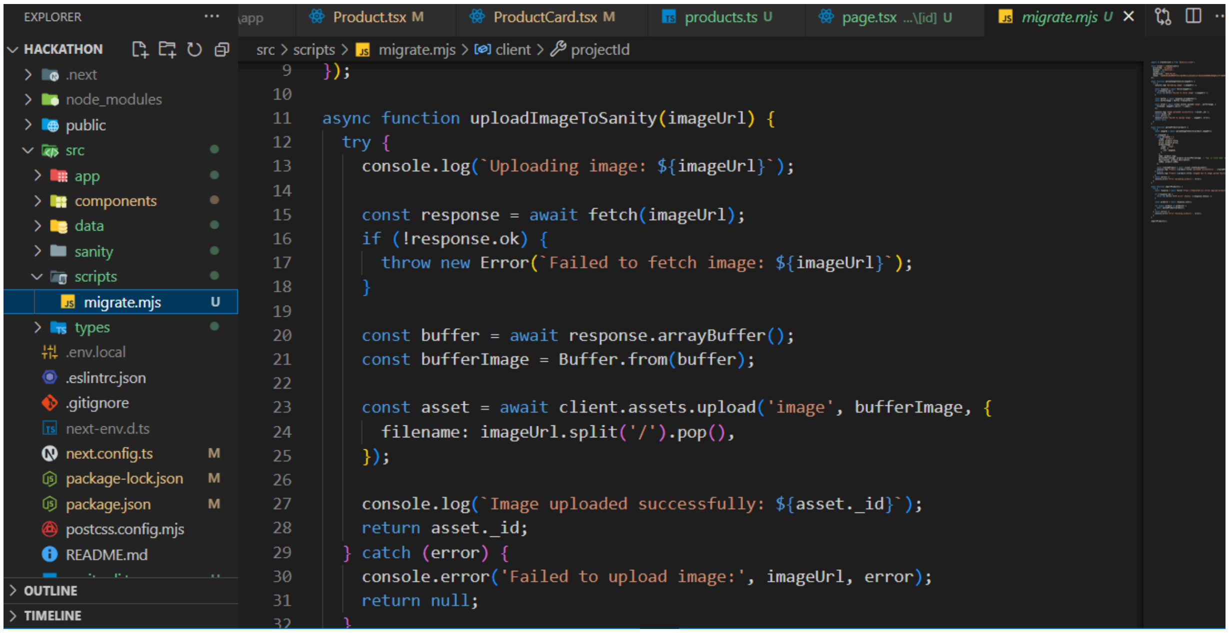
# ***API INTEGRATION AND DATA MIGRATION***

## 1.Products API Integration:

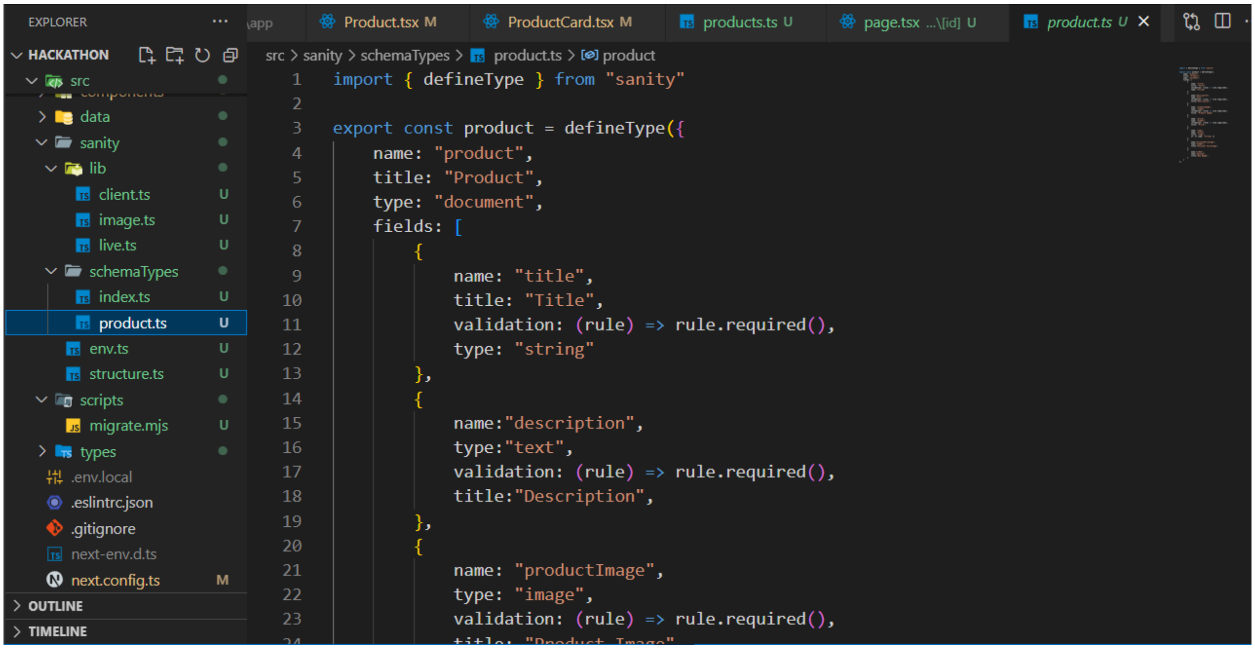
I have made an imigration file named migration.mjs and then fetch data from api and then stored it to sanity cms.



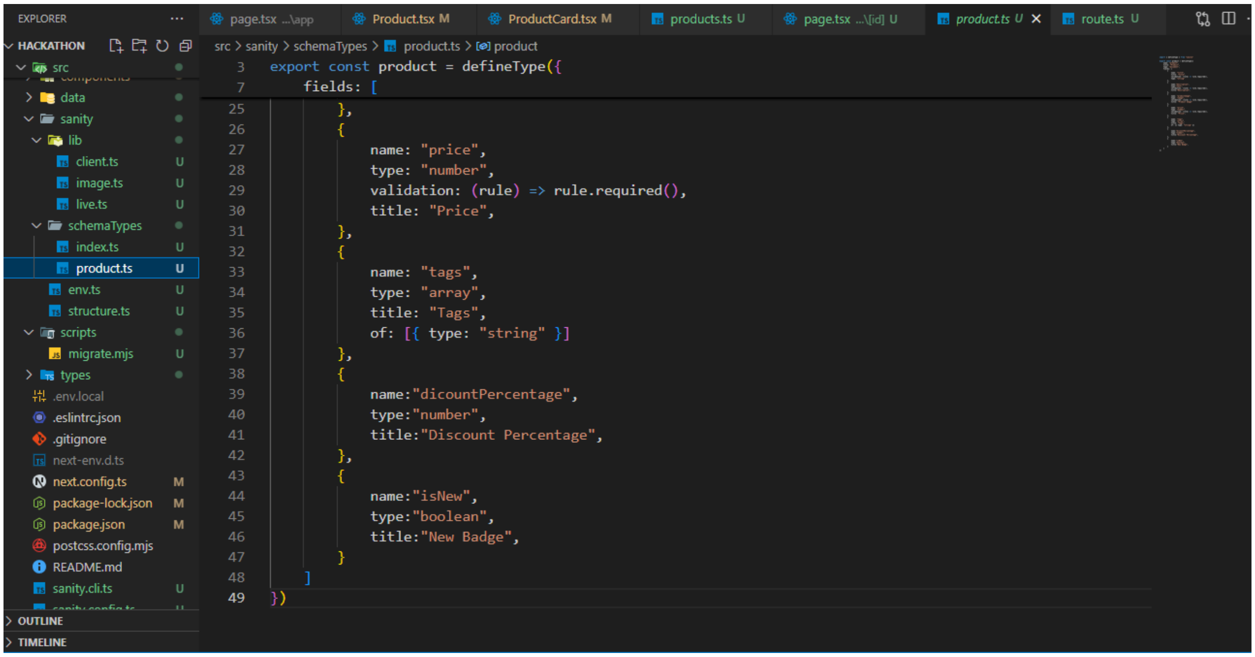
```
});

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);

    const response = await fetch(imageUrl);
    if (!response.ok) {
      throw new Error(`Failed to fetch image: ${imageUrl}`);
    }

    const buffer = await response.arrayBuffer();
    const bufferImage = Buffer.from(buffer);

    const asset = await client.assets.upload('image', bufferImage, {
      filename: imageUrl.split('/').pop(),
    });

    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
```

# 2. Schema:

I have adjusted the new schema, and have made some changes in my products schema.



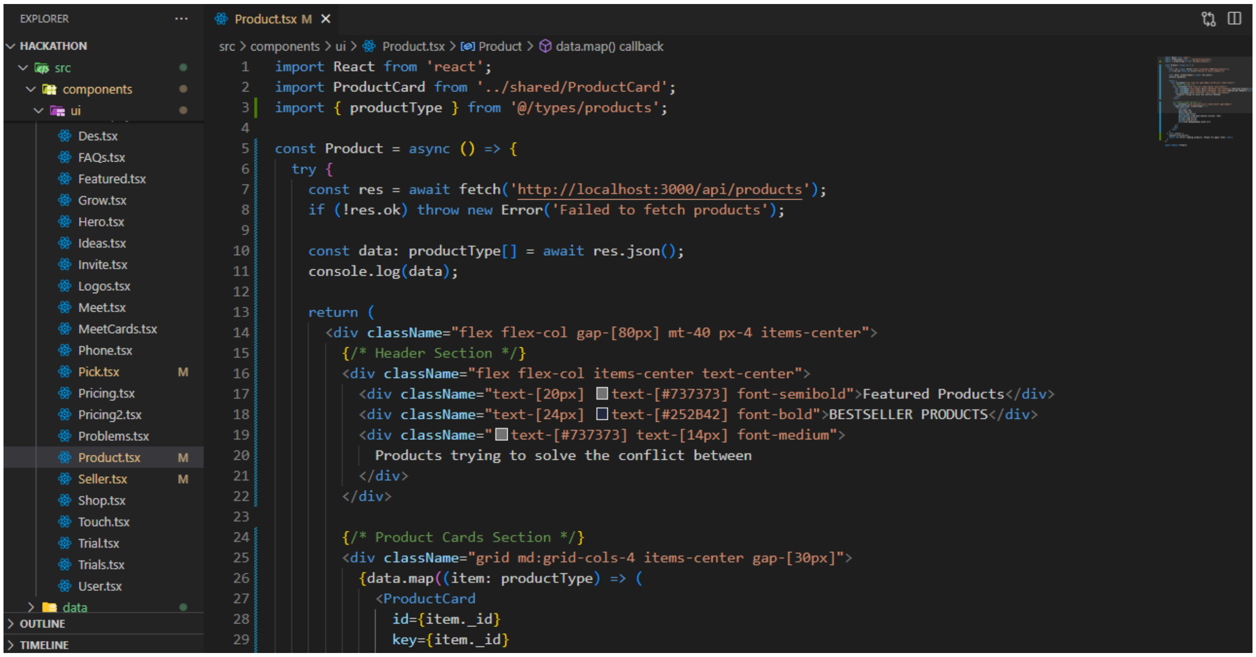
```
import { defineType } from "sanity"

export const product = defineType({
    name: "product",
    title: "Product",
    type: "document",
    fields: [
        {
            name: "title",
            title: "Title",
            validation: (rule) => rule.required(),
            type: "string"
        },
        {
            name:"description",
            type:"text",
            validation: (rule) => rule.required(),
            title:"Description",
        },
        {
            name: "productImage",
            type: "image",
            validation: (rule) => rule.required(),
            title: "Product Image",
```



```
export const product = defineType({
    fields: [
        },
        {
            name: "price",
            type: "number",
            validation: (rule) => rule.required(),
            title: "Price",
        },
        {
            name: "tags",
            type: "array",
            title: "Tags",
            of: [{ type: "string" }]
        },
        {
            name:"dicountPercentage",
            type:"number",
            title:"Discount Percentage",
        },
        {
            name:"isNew",
            type:"boolean",
            title:"New Badge",
        }
    ]
})
```

# 3.Api creation and data fetching:

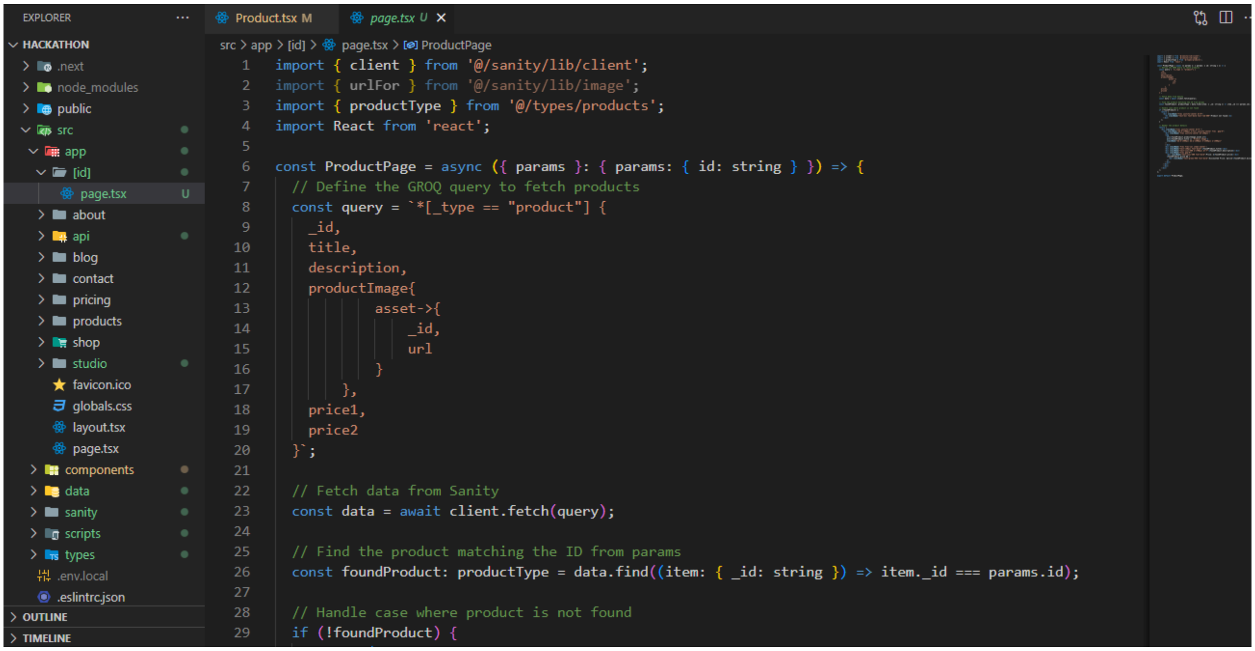
Then , I have fetched all the products from my Sanity cms.



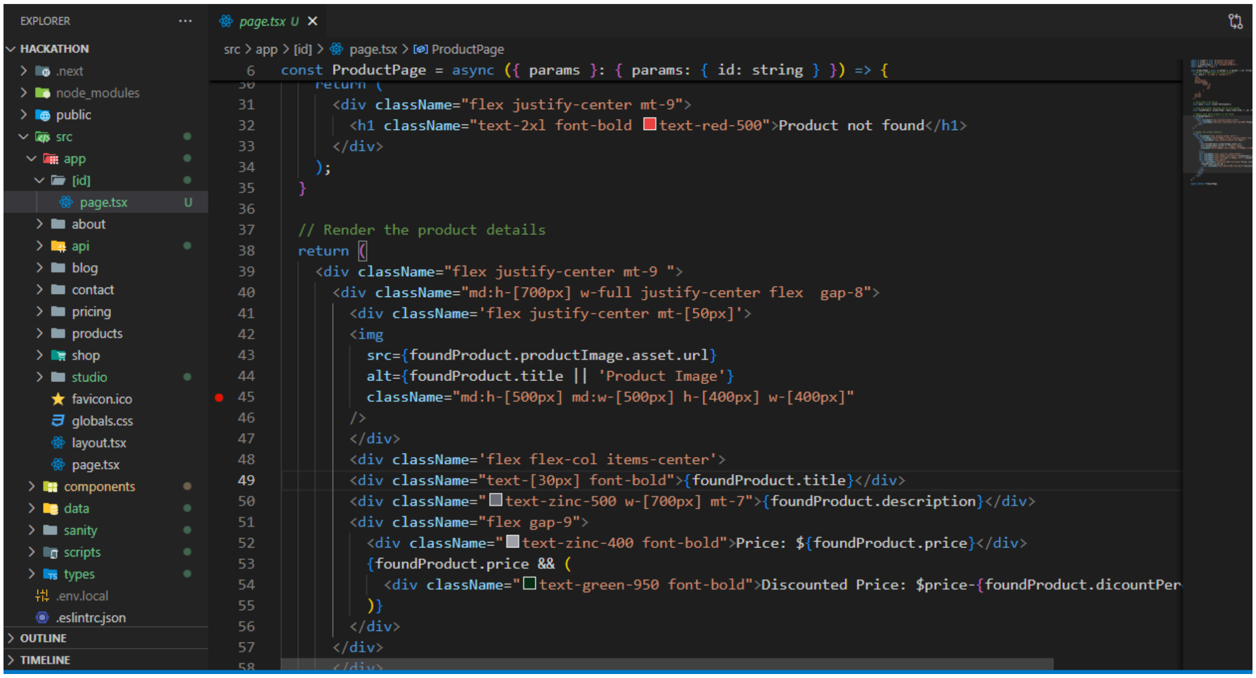
```
import React from 'react';
import ProductCard from '../shared/ProductCard';
import { productType } from '@/types/products';

const Product = async () => {
  try {
    const res = await fetch('http://localhost:3000/api/products');
    if (!res.ok) throw new Error('Failed to fetch products');

    const data: productType[] = await res.json();
    console.log(data);

    return (
      <div className="flex flex-col gap-[80px] mt-40 px-4 items-center">
        {/* Header Section */}
        <div className="flex flex-col items-center text-center">
          <div className="text-[20px] text-[#737373] font-semibold">Featured Products</div>
          <div className="text-[24px] text-[#252B42] font-bold">BESTSELLER PRODUCTS</div>
          <div className="text-[#737373] text-[14px] font-medium">
            Products trying to solve the conflict between
          </div>
        </div>

        {/* Product Cards Section */}
        <div className="grid md:grid-cols-4 items-center gap-[30px]">
          {data.map((item: productType) => (
            <ProductCard
              id={item._id}
              key={item._id}
```

## 4. Individually showing products data through Dynamic route:

Then, i have made a dynamic route to show individual product from the api.



```
import { client } from '@/sanity/lib/client';
import { urlFor } from '@/sanity/lib/image';
import { productType } from '@/types/products';
import React from 'react';

const ProductPage = async ({ params }: { params: { id: string } }) => {
  // Define the GROQ query to fetch products
  const query = `*[_type == "product"] {
    _id,
    title,
    description,
    productImage{
            asset->{
              _id,
              url
            }
        },
    price1,
    price2
  }`;

  // Fetch data from Sanity
  const data = await client.fetch(query);

  // Find the product matching the ID from params
  const foundProduct: productType = data.find((item: { _id: string }) => item._id === params.id);

  // Handle case where product is not found
  if (!foundProduct) {
```



```
const ProductPage = async ({ params }: { params: { id: string } }) => {
    return (
      <div className="flex justify-center mt-9">
        <h1 className="text-2xl font-bold text-red-500">Product not found</h1>
      </div>
    );
  }

  // Render the product details
  return (
    <div className="flex justify-center mt-9 ">
      <div className="md:h-[700px] w-full justify-center flex  gap-8">
        <div className='flex justify-center mt-[50px]'>
        <img
          src={foundProduct.productImage.asset.url}
          alt={foundProduct.title || 'Product Image'}
          className="md:h-[500px] md:w-[500px] h-[400px] w-[400px]"
        />
        </div>
        <div className='flex flex-col items-center'>
        <div className="text-[30px] font-bold">{foundProduct.title}</div>
        <div className="text-zinc-500 w-[700px] mt-7">{foundProduct.description}</div>
        <div className="flex gap-9">
          <div className="text-zinc-400 font-bold">Price: ${foundProduct.price}</div>
          {foundProduct.price && (
            <div className="text-green-950 font-bold">Discounted Price: $price-{foundProduct.dicountPer
          )}
        </div>
      </div>
      </div>
```