

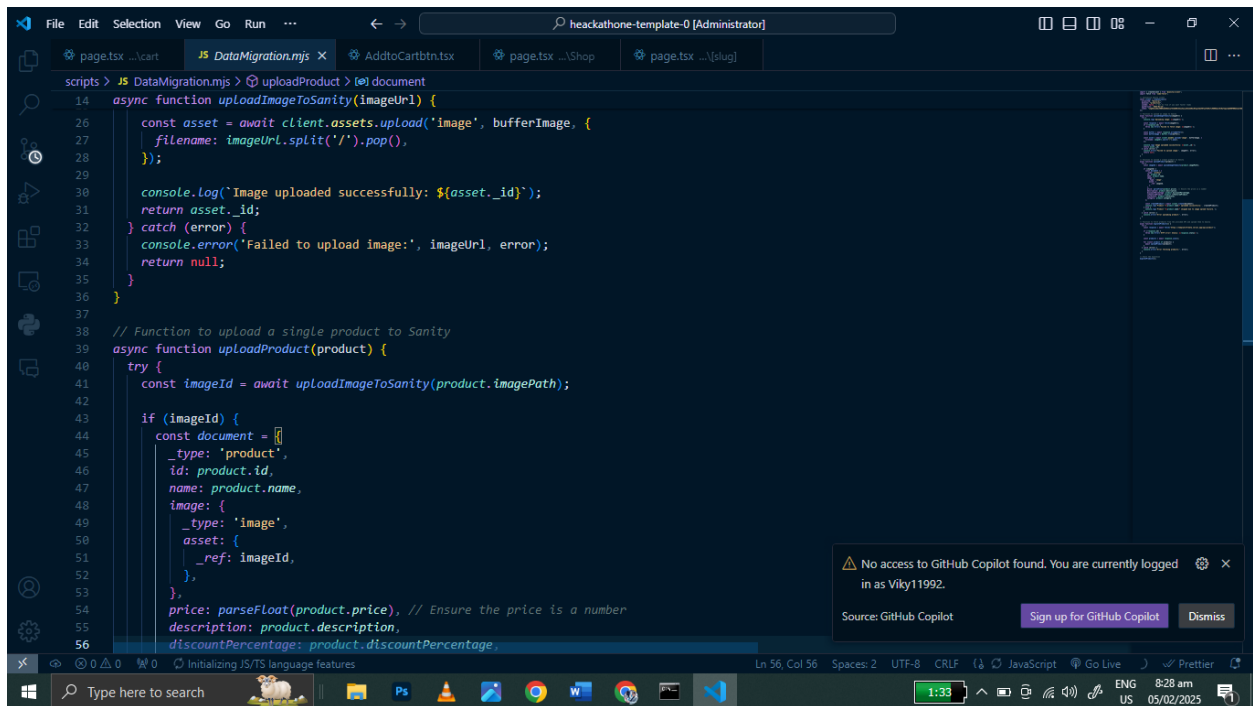
# Furniture Palace

Day 03

## API Integration and Data Migration:

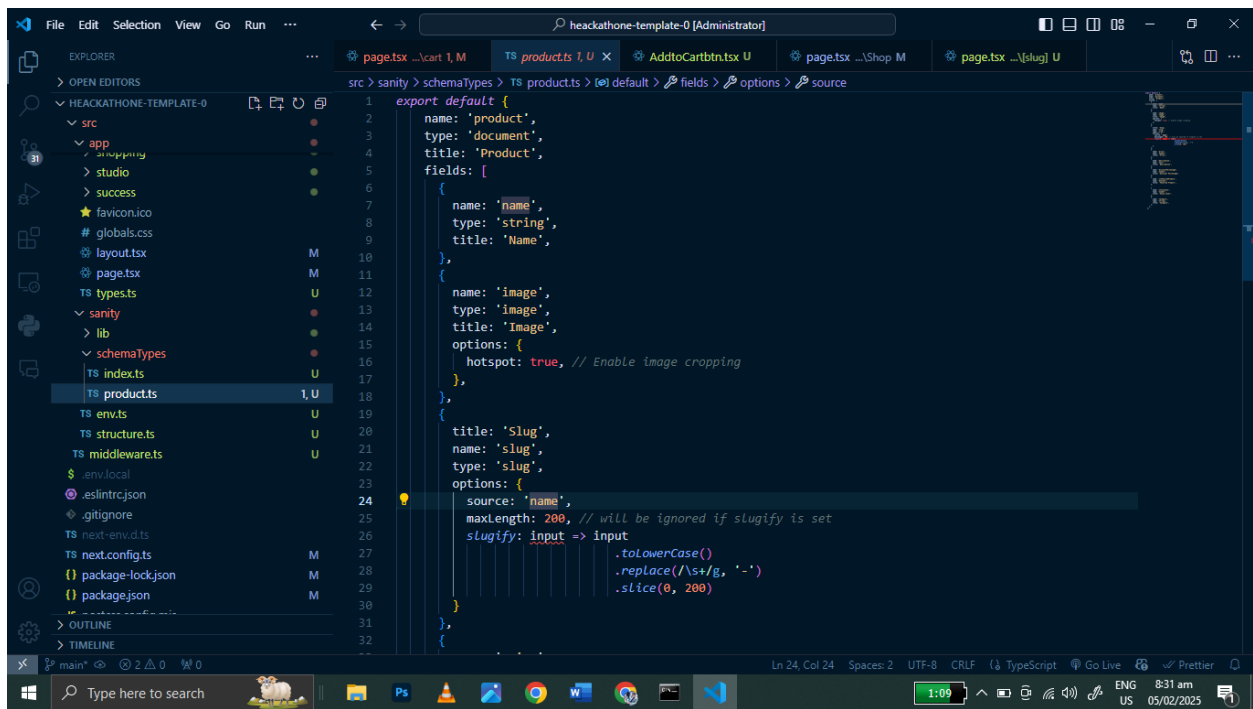
Integrating APIs into your Next.js project involves making requests to external or internal APIs and handling the data efficiently. Since you're already working with **Sanity API**, you can apply similar concepts to other APIs as well. Here's a structured approach:

### Create client for Product API:

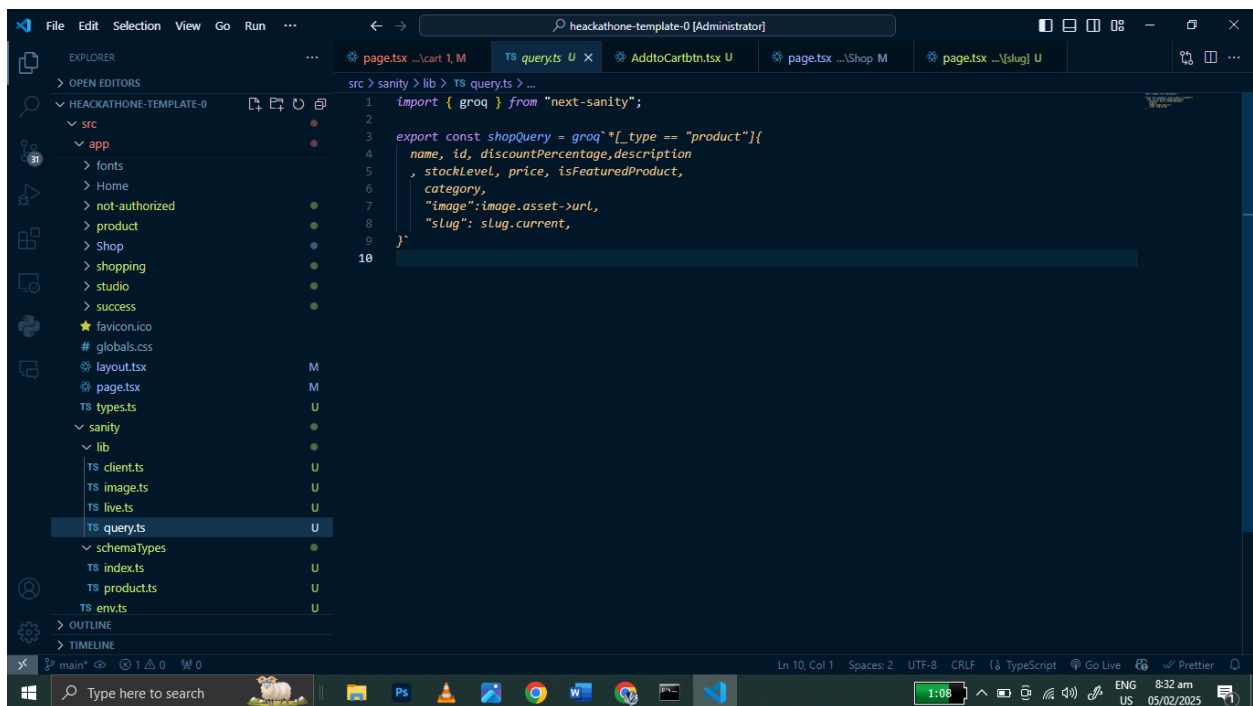


```
File Edit Selection View Go Run ... heackathone-template-0 [Administrator]
page.tsx ...cart JS DataMigration.mjs x AddtoCartbtn.tsx page.tsx ...Shop page.tsx ...[slug]
scripts > JS DataMigration.mjs > uploadProduct > document
14 async function uploadImageToSanity(imageUrl) {
26   const asset = await client.assets.upload('image', bufferImage, {
27     filename: imageUrl.split('/').pop(),
28   });
29
30   console.log('Image uploaded successfully: ${asset.id}');
31   return asset.id;
32 } catch (error) {
33   console.error('Failed to upload image:', imageUrl, error);
34   return null;
35 }
36
37
38 // Function to upload a single product to Sanity
39 async function uploadProduct(product) {
40   try {
41     const imageId = await uploadImageToSanity(product.imagePath);
42
43     if (imageId) {
44       const document = {
45         _type: 'product',
46         id: product.id,
47         name: product.name,
48         image: {
49           _type: 'image',
50           asset: {
51             _ref: imageId,
52           },
53         },
54         price: parseFloat(product.price), // Ensure the price is a number
55         description: product.description,
56         discountPercentage: product.discountPercentage,
```

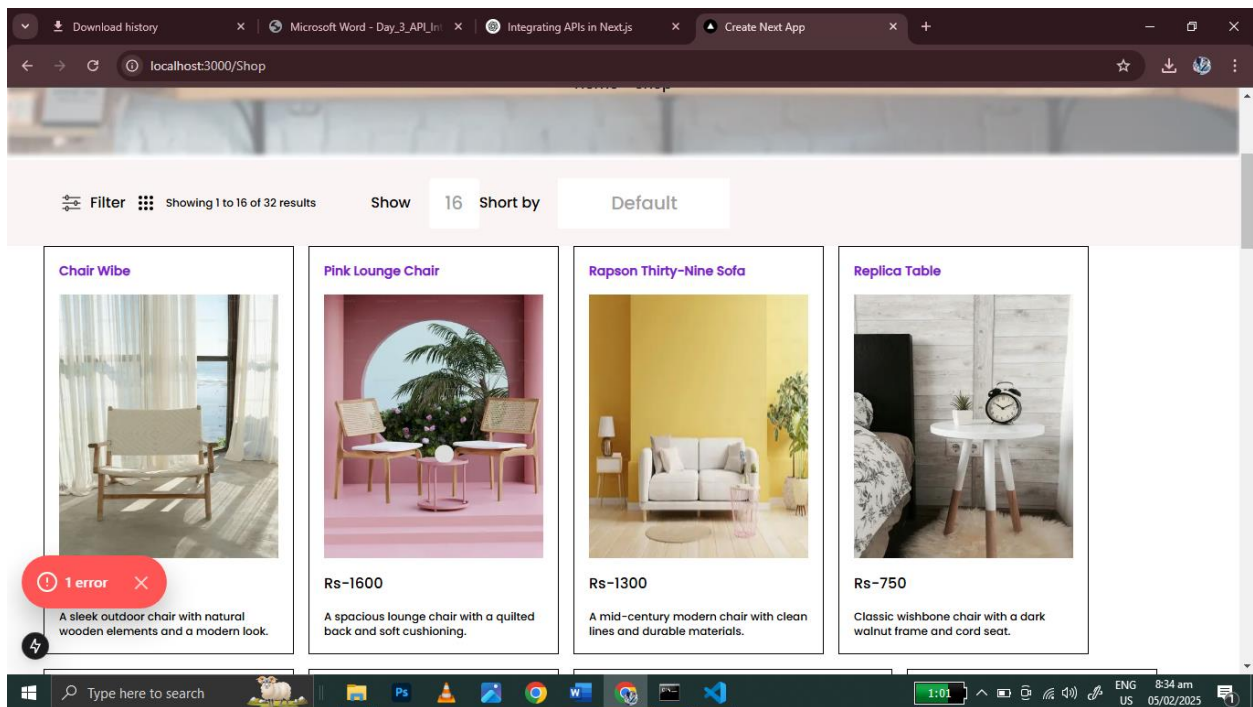
### Sanity Product schema:



## Fetch Product data from sanity:



## Display products:



## Brief overview:

API migration in Sanity allows transferring structured data into the Sanity CMS using the Sanity client. After migration, Next.js fetches this data using Sanity's API client with `getServerSideProps` or `getStaticProps`. The fetched data is then rendered dynamically in a Next.js component, enabling seamless integration of Sanity-managed content into the frontend.