

ROLLNO:00421518

HACKATHON 03

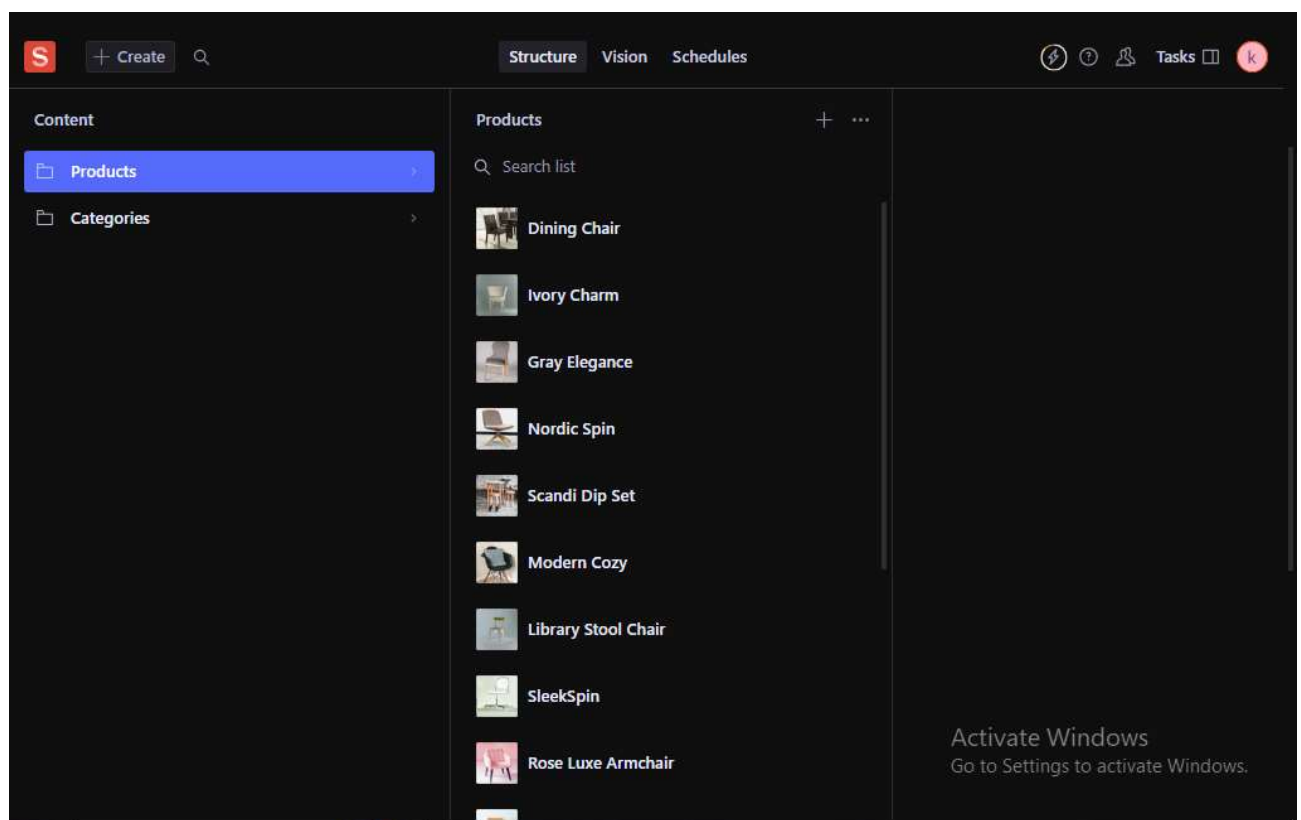
DAY 03:

Document Title:

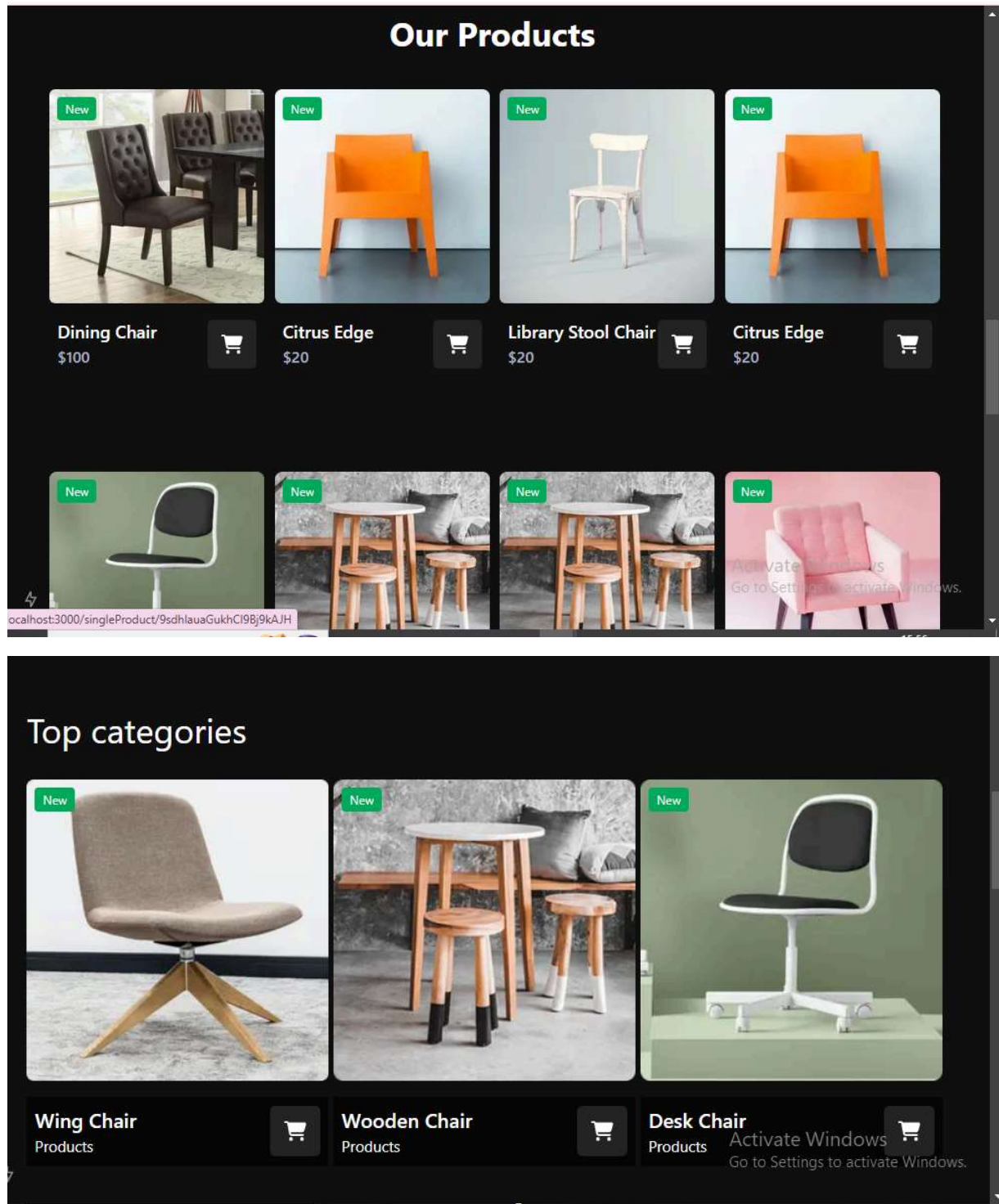
Day3-API Integration Report
Comforty-E-Commerce

SCREEN SHOTS:

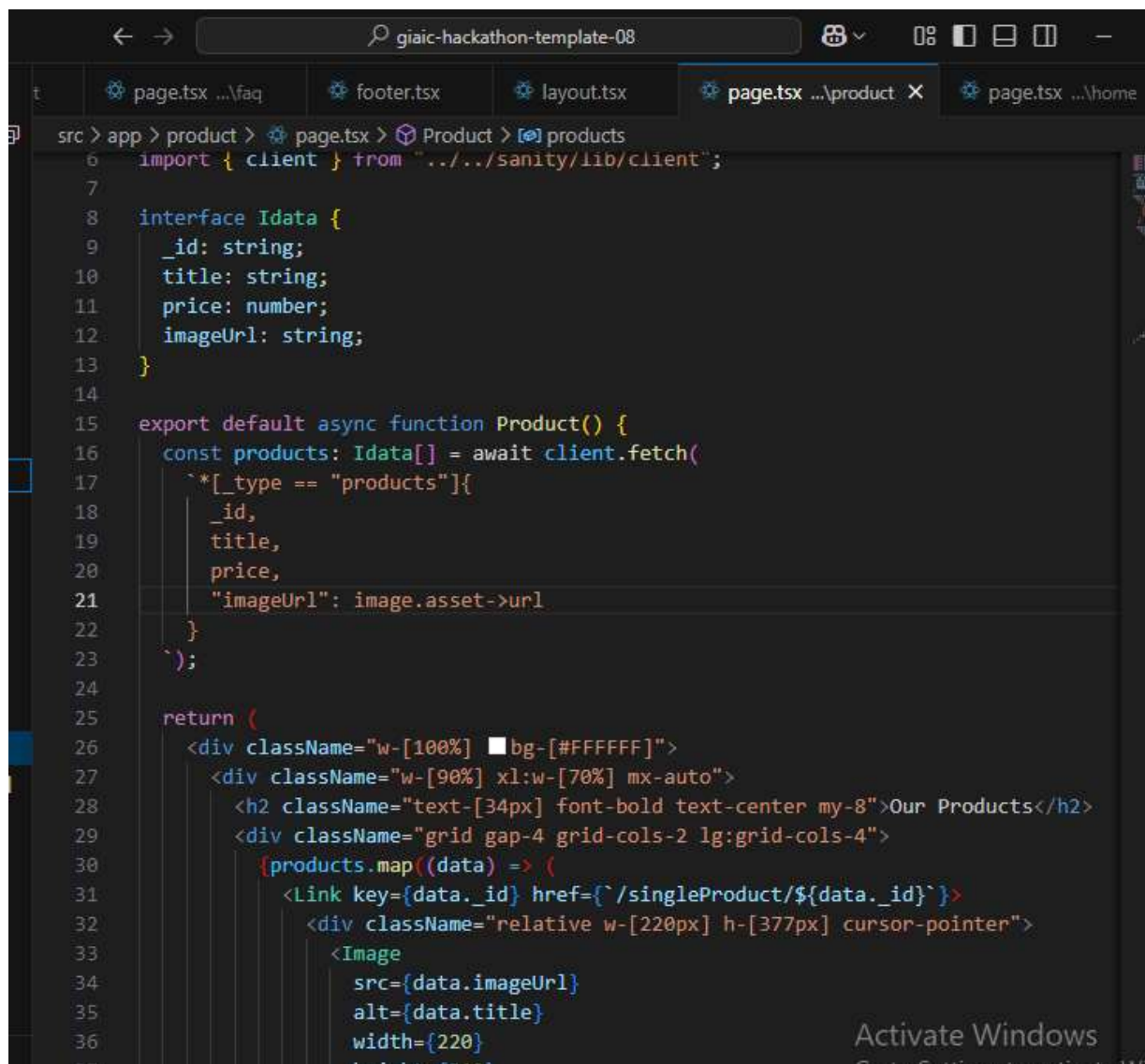
POPULATED SANITY CMS FIELD



DATA SUCCESSFULLY DISPLAYED ON THE FRONTED



API CALLS



```
src > app > product > page.tsx > Product > products
6  import { client } from "../../sanity/lib/client";
7
8  interface Idata {
9    _id: string;
10   title: string;
11   price: number;
12   imageUrl: string;
13 }
14
15 export default async function Product() {
16   const products: Idata[] = await client.fetch(
17     `*[_type == "products"]{
18       _id,
19       title,
20       price,
21       "imageUrl": image.asset->url
22     }
23   `);
24
25   return (
26     <div className="w-[100%] bg-[#FFFFFF]">
27       <div className="w-[90%] xl:w-[70%] mx-auto">
28         <h2 className="text-[34px] font-bold text-center my-8">Our Products</h2>
29         <div className="grid gap-4 grid-cols-2 lg:grid-cols-4">
30           {products.map((data) => (
31             <Link key={data._id} href={` /singleProduct/${data._id}`}>
32               <div className="relative w-[220px] h-[377px] cursor-pointer">
33                 <Image
34                   src={data.imageUrl}
35                   alt={data.title}
36                   width={220}
37                   height={377}
38                 />
39               <div className="absolute bottom-0 right-0 p-2">
40                 {data.title}
41               </div>
42             </div>
43           ))}
44         </div>
45       </div>
46     </div>
47   );
48 }
```

```

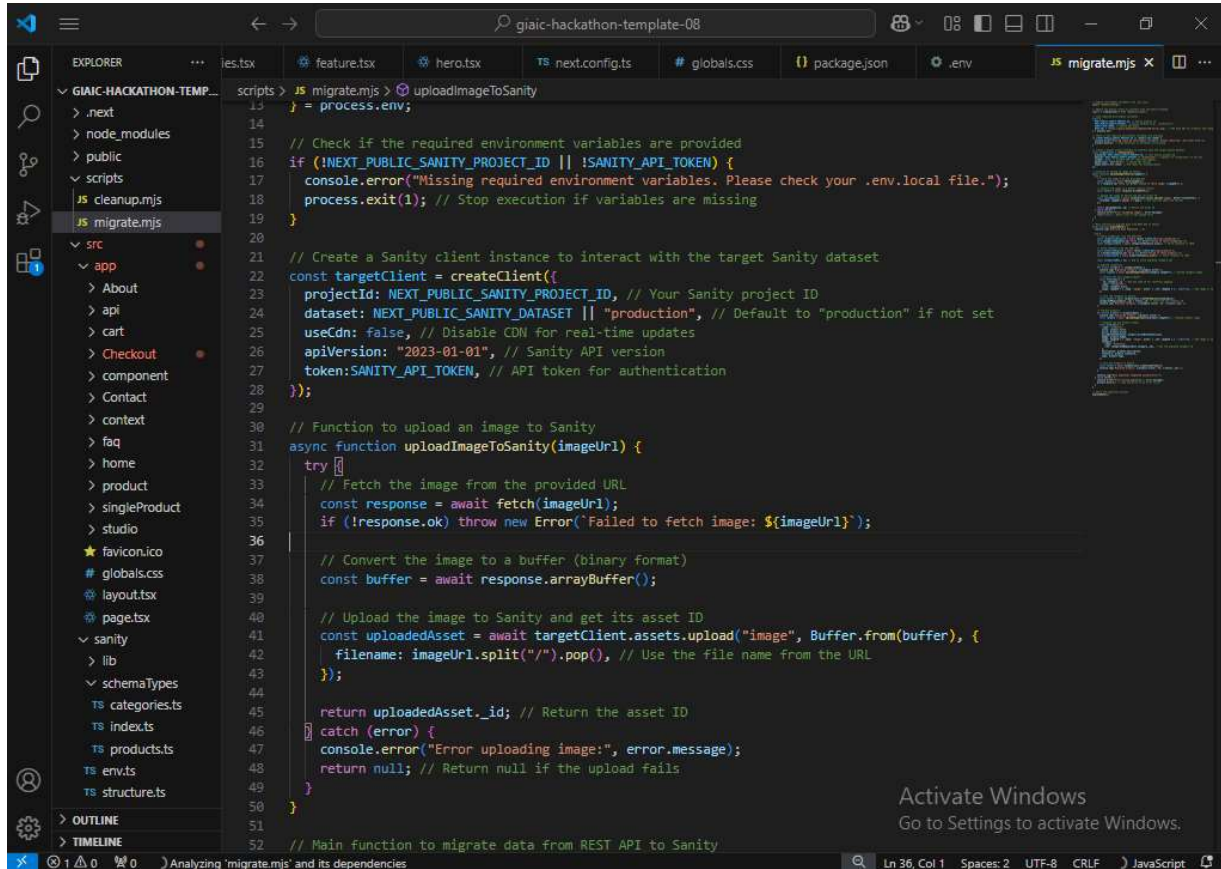
src > app > component > topCategories.tsx > topCategories > categories
6  interface IData {
7      title: string;
8      numberOfProducts: number;
9      imageUrl: string;
10 }
11
12 export default async function TopCategories() {
13     const categories: IData[] = await client.fetch<
14     >{
15         *_[_type == "categories"]{
16             title,
17             numberOfProducts,
18             "imageUrl": image.asset->url
19         }
20     };
21
22     return (
23         <div className="w-[90%] xl:w-[70%] mx-auto">
24             <h2 className="text-[34px] mb-6 mt-4">Top categories</h2>
25             <div className="grid gap-2 grid-cols-3 mb-6">
26                 {categories.map((data, index) => (
27                     <div key={index} className="relative w-[200px] lg:w-[300px] h-
28                     <Image
29                         src={data.imageUrl}
30                         alt={data.title}
31                         width={250}
32                         height={312}
33                         className="rounded-[6px] w-[250px] lg:w-[300px]"
34                         priority
35                     />
36                     <button className="absolute top-2 left-2 bg-[#01AD5A]
37                         New
38                     </button>
39                     <div className="flex justify-between bg-[#000000]
40                     <div className="text-[#FFFFFF]"
41                 )
42             )}
43         </div>
44     );
45 }

```

Ln 14, Col 9 Spaces: 4 UTF-8 CRLF {} TypeScript JSX

19°C 10:04 29/01/2025

Code snippets for API Integration and Migration Script

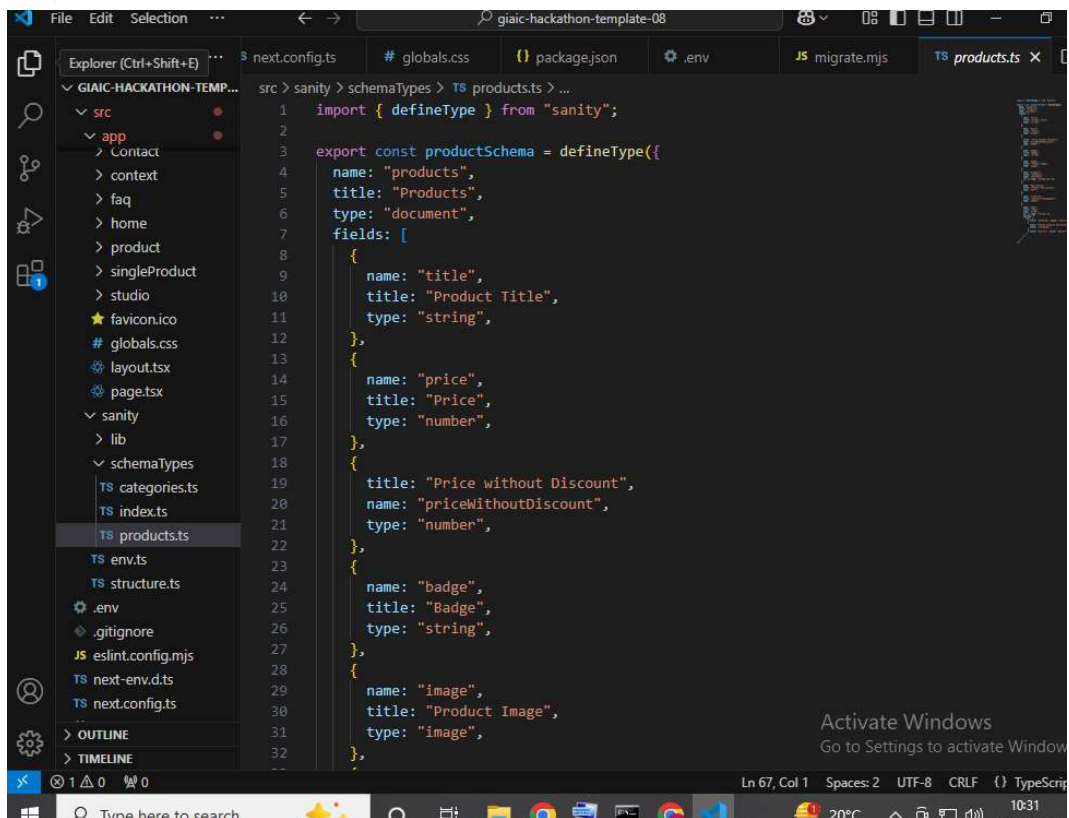


The screenshot shows a VS Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a project structure for 'GIAIC-HACKATHON-TEMP...'. The code editor displays a JavaScript file named 'migrate.mjs' with the following content:

```
13 scripts > JS migrate.mjs > uploadImageToSanity
14   = process.env;
15
16 // Check if the required environment variables are provided
17 if (!NEXT_PUBLIC_SANITY_PROJECT_ID || !SANITY_API_TOKEN) {
18   console.error("Missing required environment variables. Please check your .env.local file.");
19   process.exit(1); // Stop execution if variables are missing
20 }
21
22 // Create a Sanity client instance to interact with the target Sanity dataset
23 const targetClient = createClient({
24   projectId: NEXT_PUBLIC_SANITY_PROJECT_ID, // Your Sanity project ID
25   dataset: NEXT_PUBLIC_SANITY_DATASET || "production", // Default to "production" if not set
26   useCdn: false, // Disable CDN for real-time updates
27   apiVersion: "2023-01-01", // Sanity API version
28   token: SANITY_API_TOKEN, // API token for authentication
29 });
30
31 // Function to upload an image to Sanity
32 async function uploadImageToSanity(imageUrl) {
33   try {
34     // Fetch the image from the provided URL
35     const response = await fetch(imageUrl);
36     if (!response.ok) throw new Error(`Failed to fetch image: ${imageUrl}`);
37
38     // Convert the image to a buffer (binary format)
39     const buffer = await response.arrayBuffer();
40
41     // Upload the image to Sanity and get its asset ID
42     const uploadedAsset = await targetClient.assets.upload("image", Buffer.from(buffer), {
43       filename: imageUrl.split("/").pop(), // Use the file name from the URL
44     });
45
46     return uploadedAsset._id; // Return the asset ID
47   } catch (error) {
48     console.error("Error uploading image:", error.message);
49     return null; // Return null if the upload fails
50   }
51 }
52
53 // Main function to migrate data from REST API to Sanity
```

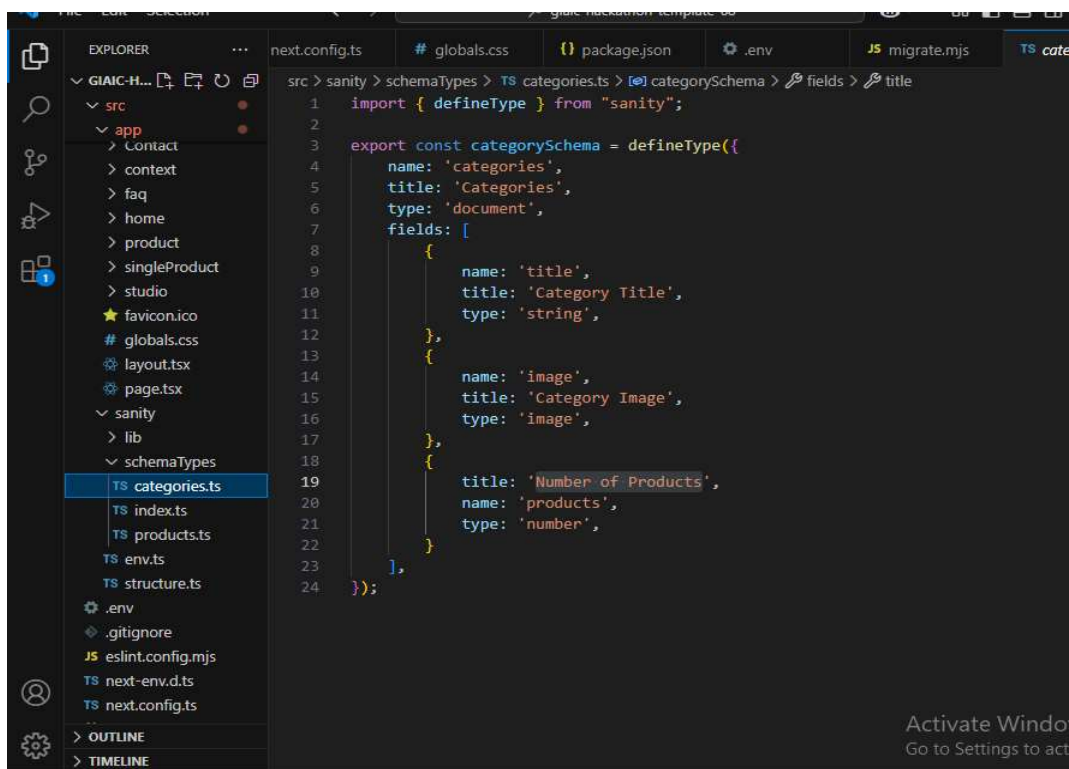
An 'Activate Windows' watermark is visible in the bottom right corner of the code editor.

Adjustment of schemas



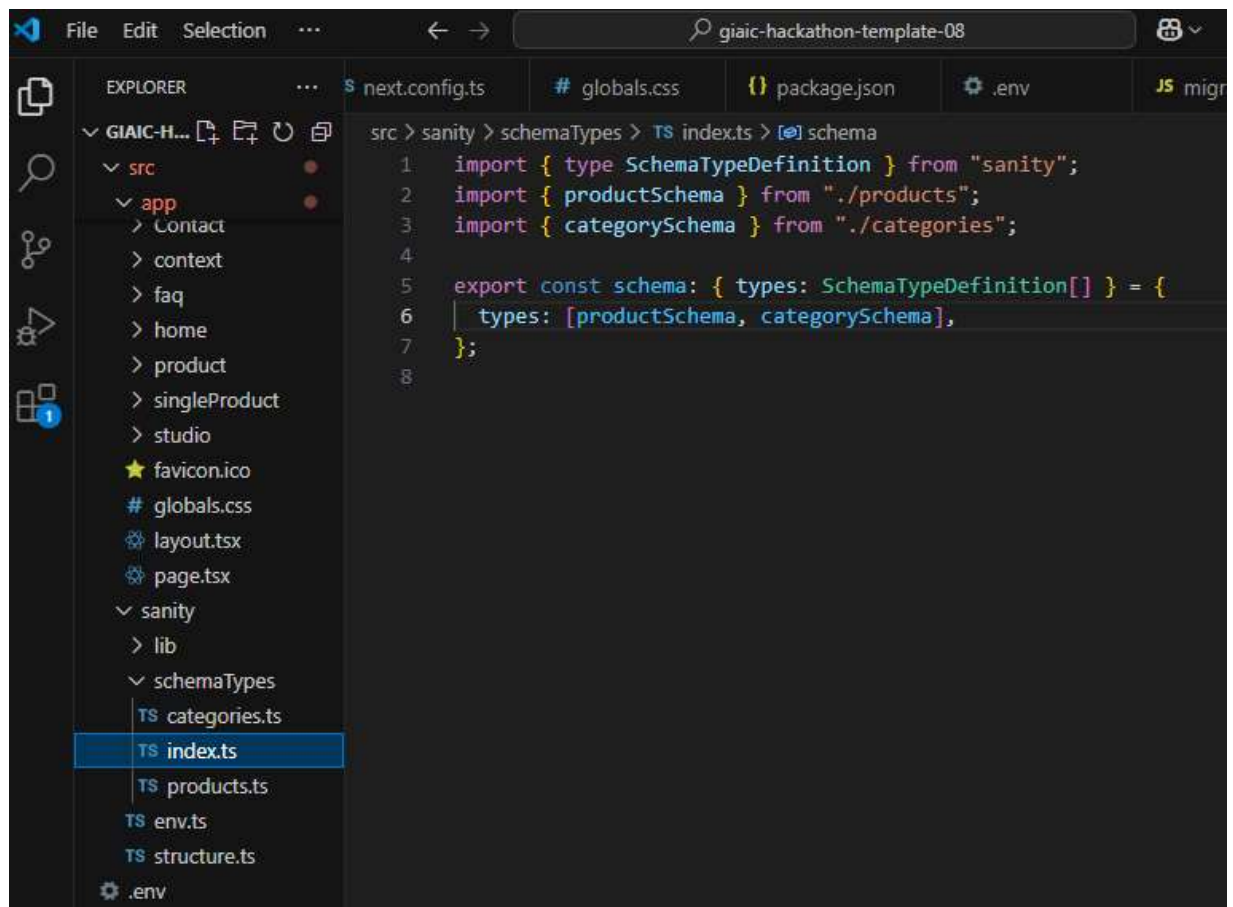
The screenshot shows the VS Code editor with the file explorer on the left. The project structure includes a 'sanity' directory with 'lib' and 'schemaTypes'. The 'products.ts' file is selected in the explorer and open in the editor. The code defines a 'productSchema' using 'defineType' from 'sanity'. The schema has a name 'products', title 'Products', and type 'document'. It contains three fields: 'title' (string), 'price' (number), and 'image' (image). The 'price' field has a title 'Price' and a description 'Price without Discount'. The 'image' field has a title 'Product Image'.

```
1 import { defineType } from "sanity";
2
3 export const productSchema = defineType({
4   name: "products",
5   title: "Products",
6   type: "document",
7   fields: [
8     {
9       name: "title",
10      title: "Product Title",
11      type: "string",
12    },
13    {
14      name: "price",
15      title: "Price",
16      type: "number",
17    },
18    {
19      title: "Price without Discount",
20      name: "priceWithoutDiscount",
21      type: "number",
22    },
23    {
24      name: "badge",
25      title: "Badge",
26      type: "string",
27    },
28    {
29      name: "image",
30      title: "Product Image",
31      type: "image",
32    },
33  ],
34 });
```



The screenshot shows the VS Code editor with the file explorer on the left. The project structure is the same as the previous screenshot. The 'categories.ts' file is selected in the explorer and open in the editor. The code defines a 'categorySchema' using 'defineType' from 'sanity'. The schema has a name 'categories', title 'Categories', and type 'document'. It contains three fields: 'title' (string), 'image' (image), and 'products' (number). The 'title' field has a title 'Category Title'. The 'image' field has a title 'Category Image'. The 'products' field has a title 'Number of Products'.

```
1 import { defineType } from "sanity";
2
3 export const categorySchema = defineType({
4   name: 'categories',
5   title: 'Categories',
6   type: 'document',
7   fields: [
8     {
9       name: 'title',
10      title: 'Category Title',
11      type: 'string',
12    },
13    {
14      name: 'image',
15      title: 'Category Image',
16      type: 'image',
17    },
18    {
19      title: 'Number of Products',
20      name: 'products',
21      type: 'number',
22    },
23  ],
24 });
```



```
src > sanity > schemaTypes > TS index.ts > [e] schema
1  import { type SchemaTypeDefinition } from "sanity";
2  import { productSchema } from "../products";
3  import { categorySchema } from "../categories";
4
5  export const schema: { types: SchemaTypeDefinition[] } = {
6    types: [productSchema, categorySchema],
7  };
8
```