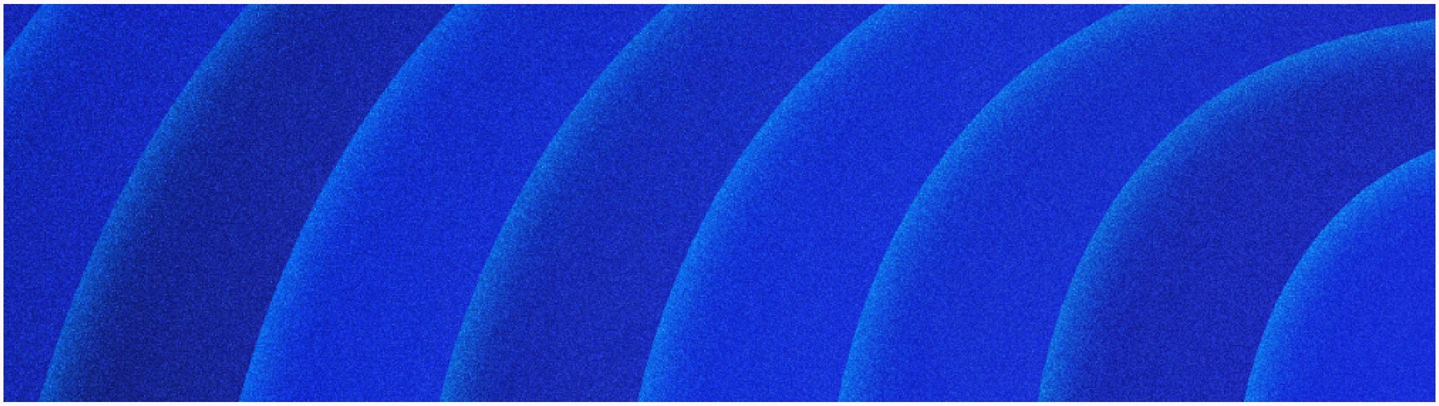


API INTEGRATION AND DATA MIGRATION

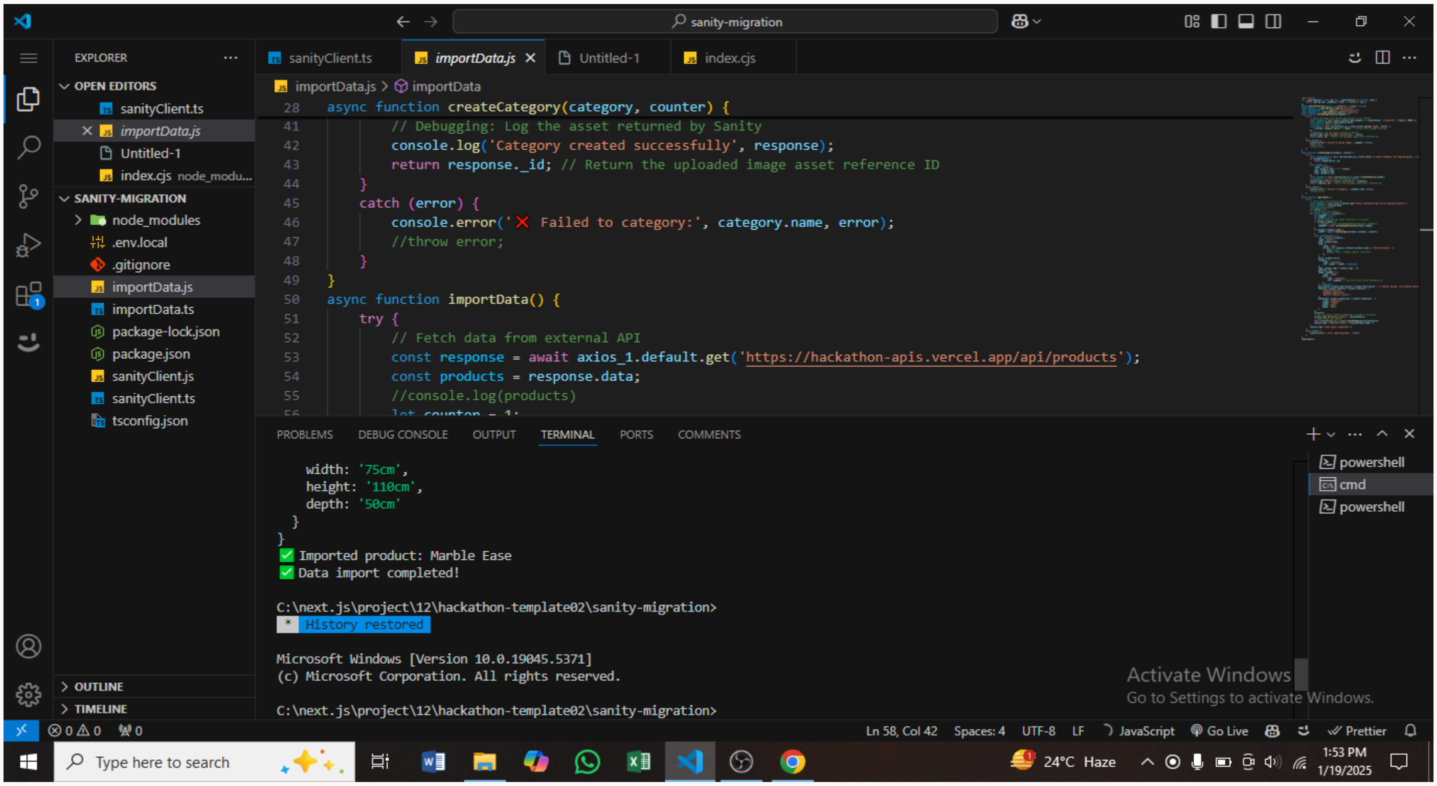
Date

Step 1 clone sanity MIGRATION

I opened the Ap1 project provided by Sir and accessed its Git repository. I then cloned the repository and migrated it into a new empty folder. Next, I ran the `npm install` command to install all the necessary packages.

After that, I created an `.env` file where I added the project ID and API token from Sanity. Then, I compiled the project using the `tsc` command.

To fetch data from the API, I ran a Node.js file named `importData.js` (or it could also be a file named something like `example.js`, depending on the file name). Once the data was successfully imported, the process was complete, and the data was imported successfully.



```javascript
async function createCategory(category, counter) {
            // Debugging: Log the asset returned by Sanity
            console.log('Category created successfully', response);
            return response._id; // Return the uploaded image asset reference ID
        }
        catch (error) {
            console.error('❌ Failed to category:', category.name, error);
            //throw error;
        }
}
async function importData() {
    try {
        // Fetch data from external API
        const response = await axios_1.default.get('https://hackathon-apis.vercel.app/api/products');
        const products = response.data;
        //console.log(products)
```

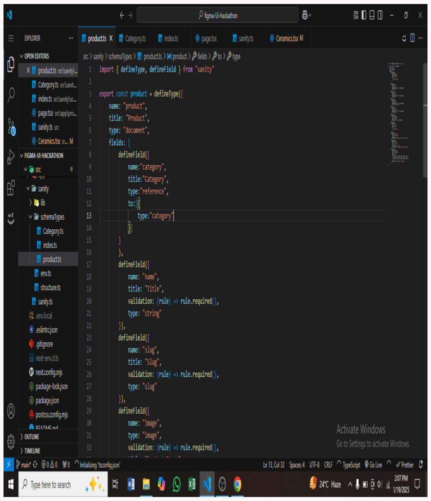
```
    width: '75cm',
    height: '110cm',
    depth: '50cm'
  }
}
✅ Imported product: Marble Ease
✅ Data import completed!

C:\next.js\project\12\hackathon-template02\sanity-migration>
 * History restored

Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.

C:\next.js\project\12\hackathon-template02\sanity-migration>
```

Step 2

After that, I opened my main project and went to the Sanity dashboard. From there, I navigated to the "Overview" section and copied the install command. I ran that command in my project, and once Sanity was installed, I defined the schema for the product data in Sanity (like shown in the example image below).

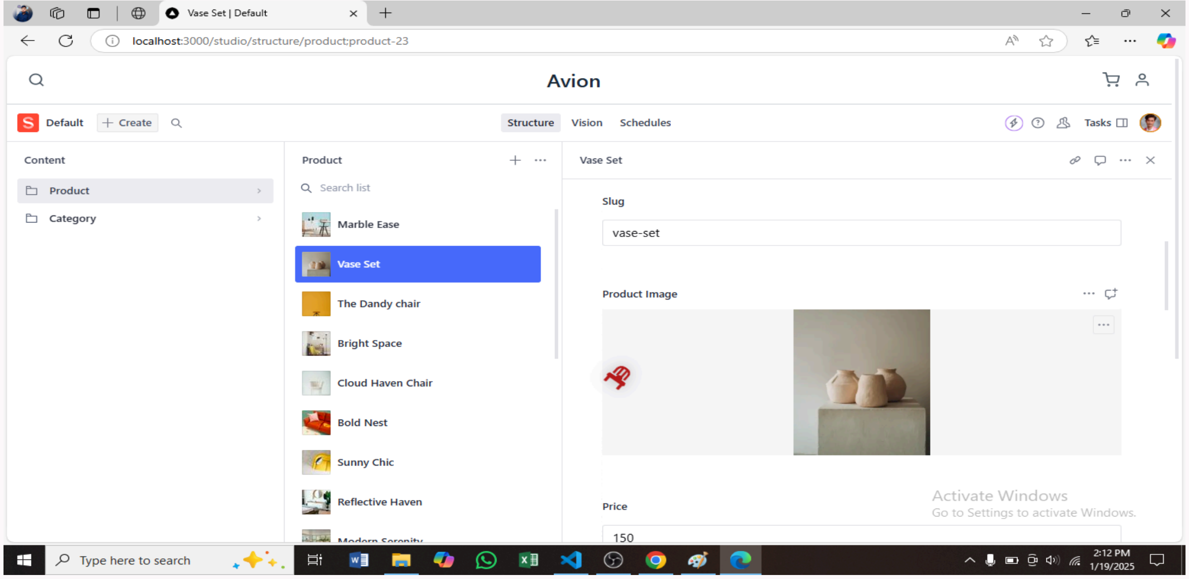
To open Sanity Studio, I ran my project and accessed it to verify that the API data was being fetched correctly. I confirmed that the data was successfully imported and visible through the API.

Sanityschema



```
import { defineType, defineField } from "sanity"

export const product = defineType({
    name: "product",
    title: "Product",
    type: "document",
    fields: [
        defineField({
            name:"category",
            title:"Category",
            type:"reference",
            to:[{
                type:"category"
            }]
        }
        ),
        defineField({
            name: "name",
            title: "Title",
            validation: (rule) => rule.required(),
            type: "string"
        }),
        defineField({
            name: "slug",
            title: "Slug",
            validation: (rule) => rule.required(),
            type: "slug"
        }),
        defineField({
            name: "image",
            type: "image",
            validation: (rule) => rule.required(),
```

Here is the picture showing that the data is now visible in Sanity



Slug

vase-set

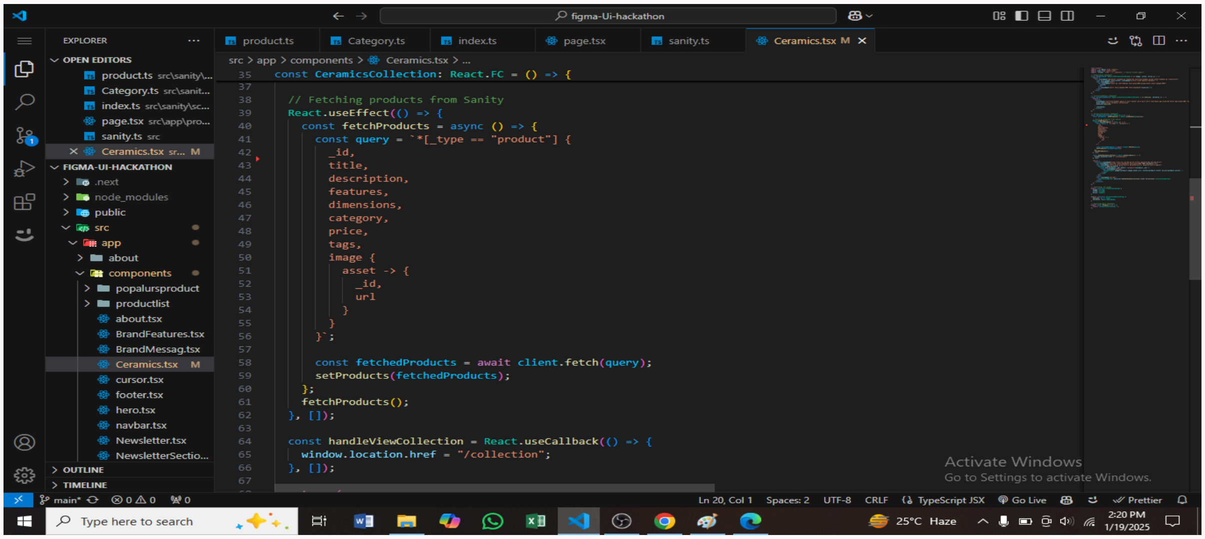
Product Image

Price

150

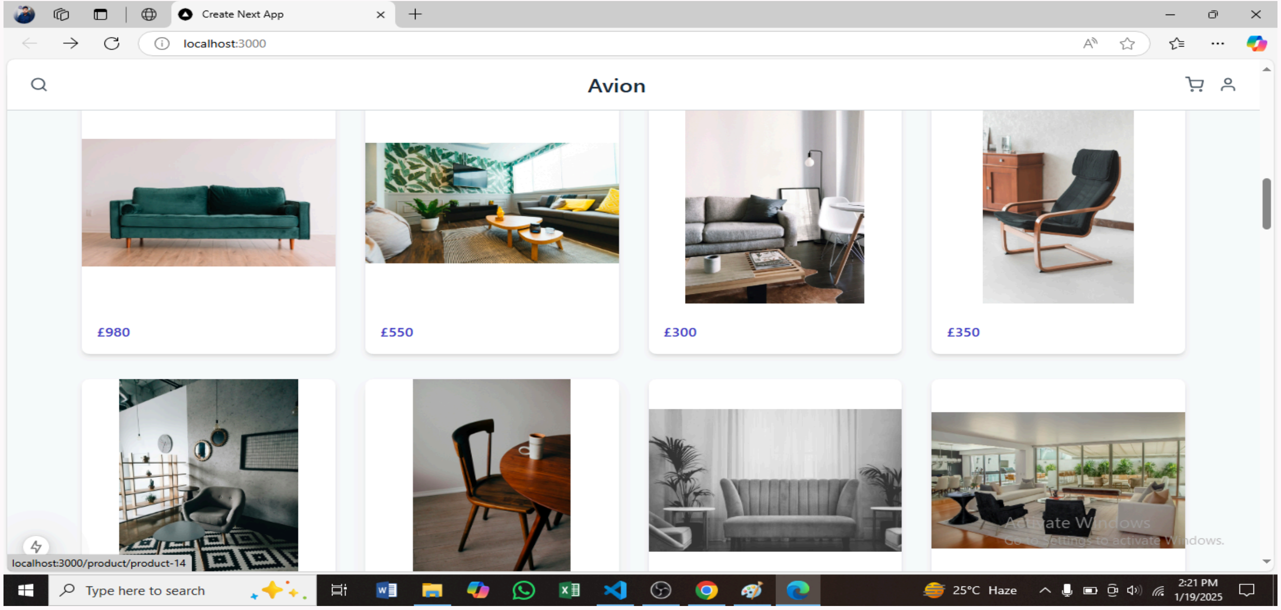
Step 3

Finally, I worked on displaying the data on the frontend. To do this, I opened the product component in the frontend where I wanted to show the data. I fetched the data from Sanity using Sanity queries and wrote the code similar to what is shown in the example image below.



```
const CeramicsCollection: React.FC = () => {

  // Fetching products from Sanity
  React.useEffect(() => {
    const fetchProducts = async () => {
      const query = `*[_type == "product"] {
        _id,
        title,
        description,
        features,
        dimensions,
        category,
        price,
        tags,
        image {
          asset -> {
            _id,
            url
          }
        }
      }`;

      const fetchedProducts = await client.fetch(query);
      setProducts(fetchedProducts);
    };
    fetchProducts();
  }, []);

  const handleViewCollection = React.useCallback(() => {
    window.location.href = "/collection";
  }, []);
```

By running these queries, all the products will be displayed on your frontend as expected.



Preferred by Shahmeer Ali.