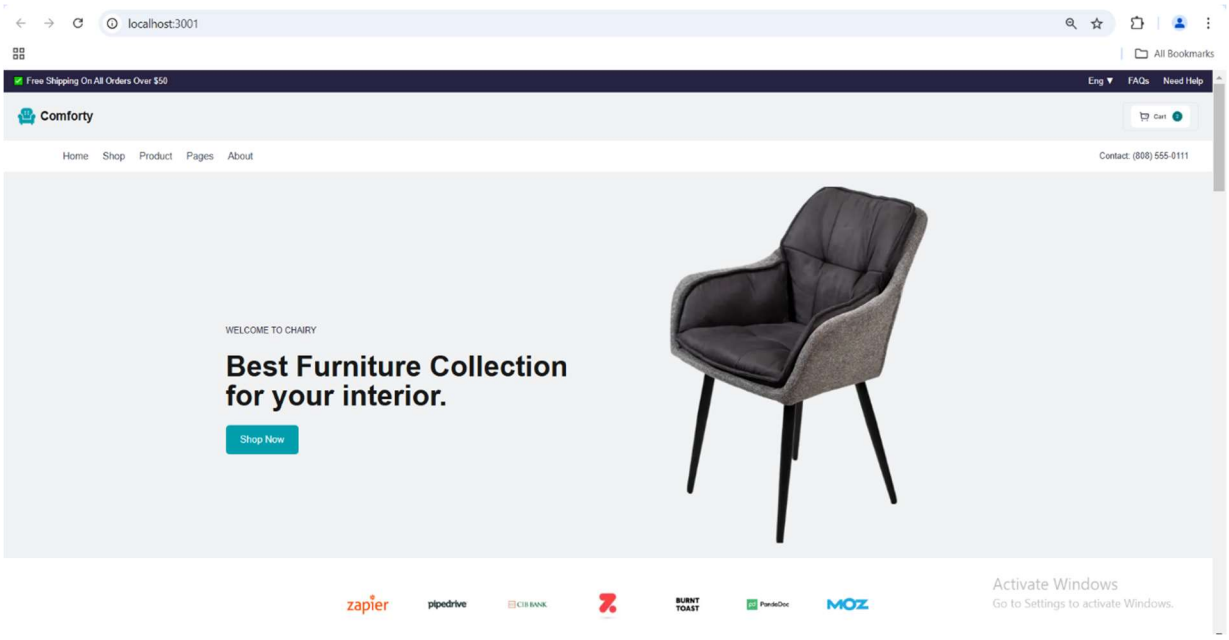
Hackathon Day 3

API Integration and Data Migration



Understanding the Requirements:

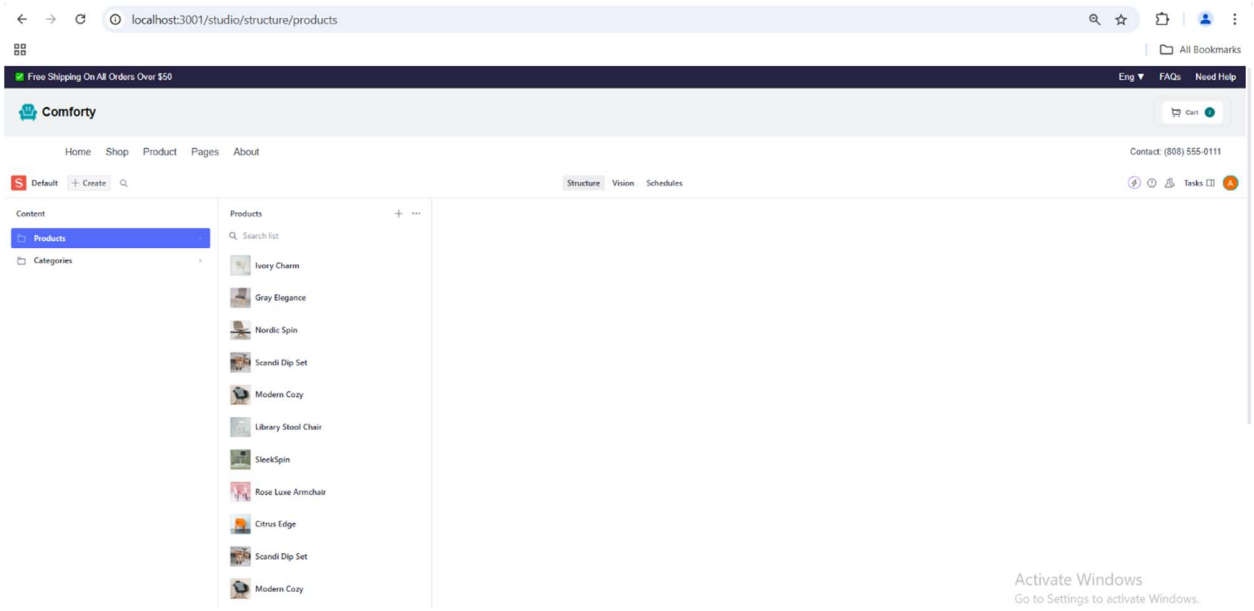
The goal was to integrate a REST API and migrate its data, specifically products and categories, into a Sanity CMS using Node.js. This process also involved setting up the environment to facilitate smooth integration.

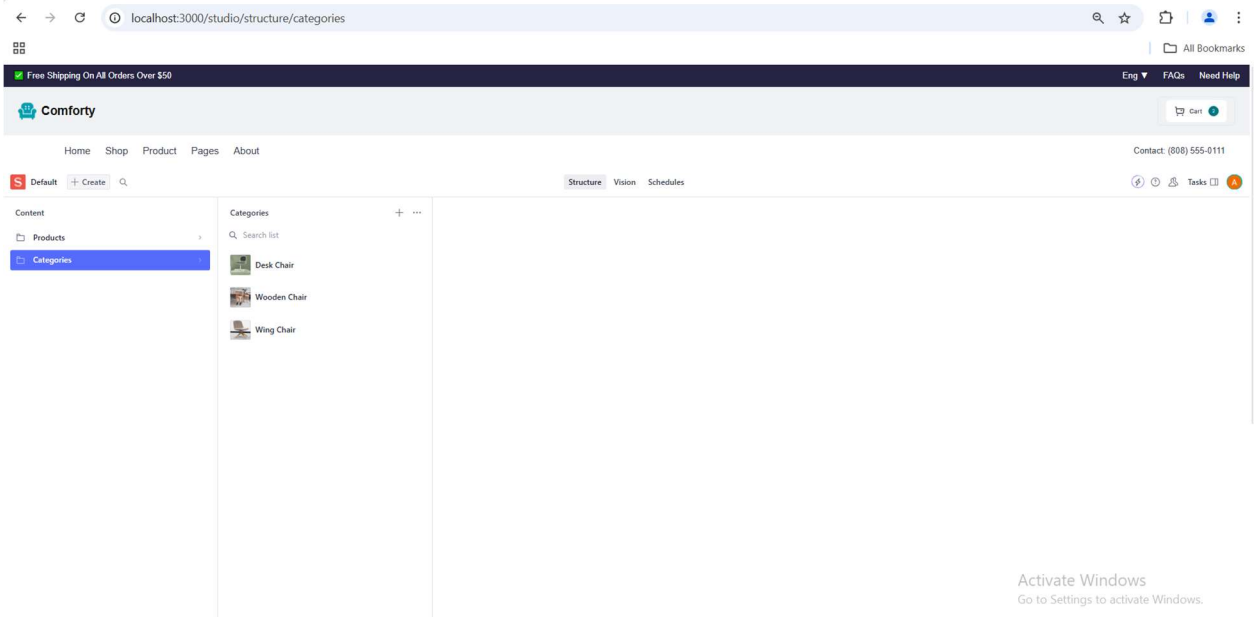
Setting Up the Environment:

- Configured `.env.local` file with the necessary Sanity credentials:
  - `SANITY_PROJECT_ID`
  - `SANITY_DATASET`
  - `SANITY_AUTH_TOKEN`
- Ensured that the correct environment variables were loaded in the script for seamless execution.

Data Migration Script:

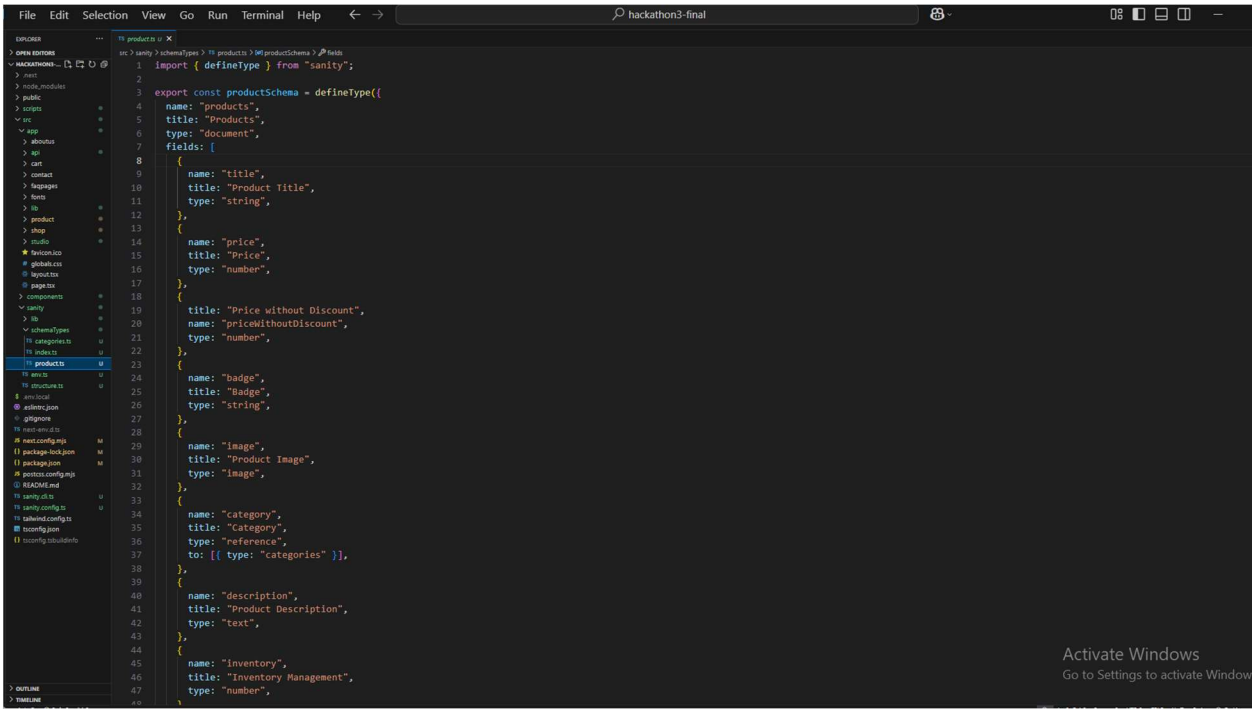
- Developed a migration script (`migrate.mjs`) to perform the following tasks:
  - Fetch product and category data from the REST API.
  - Utilize the Sanity client to:
    - Upload images.
    - Create new records for categories and products in the Sanity CMS.



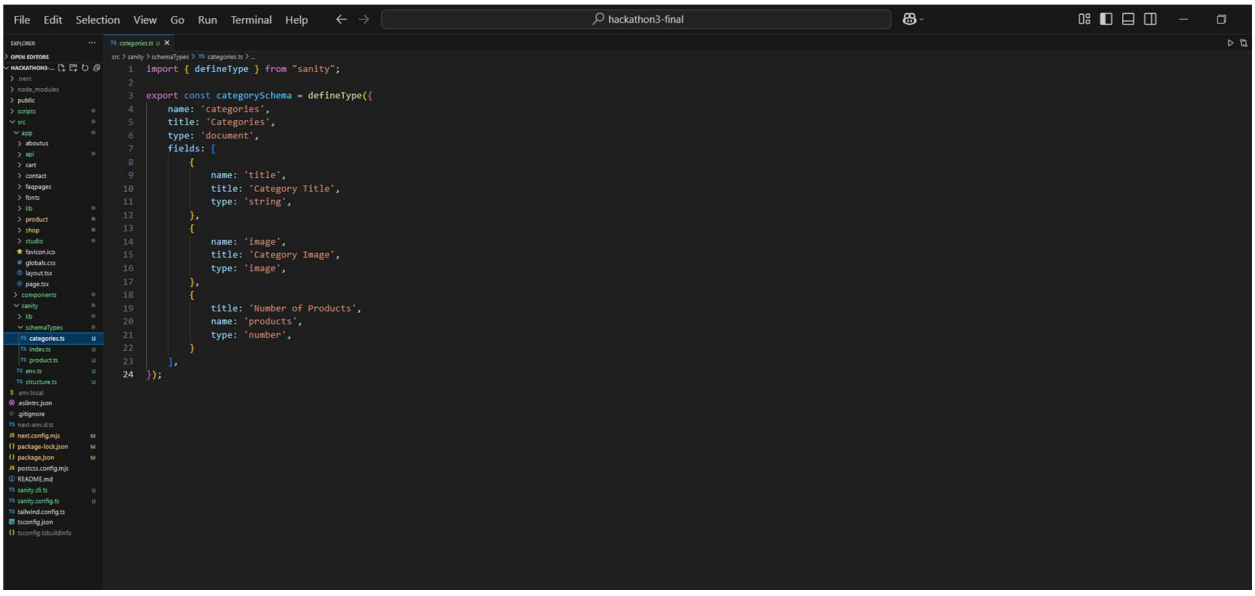


Sanity Schema:

- **Category Schema:**
  - Fields for category name and description.
  - Reference field for linking products to categories.
- **Product Schema:**
  - Fields for product name, description, price & badge.
  - Image field to store product images.
  - Reference field to associate products with categories.



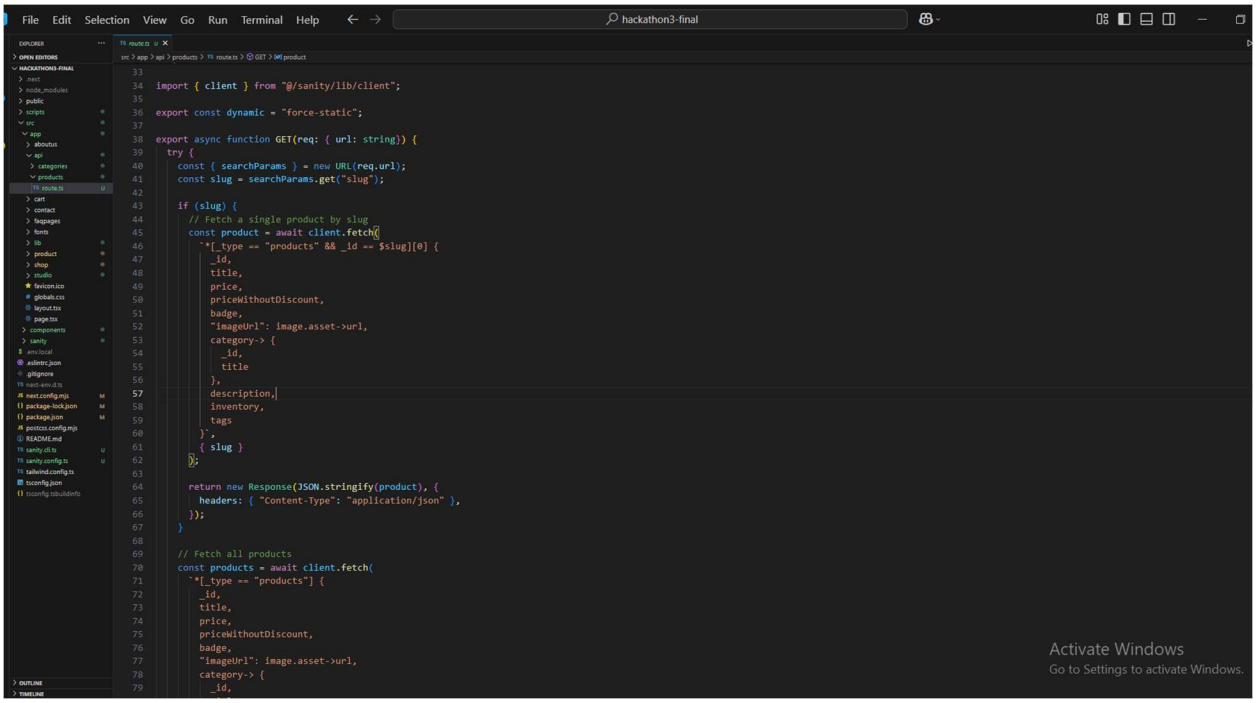
```
import { defineType } from "sanity";

export const productSchema = defineType({
  name: "products",
  title: "Products",
  type: "document",
  fields: [
    {
      name: "title",
      title: "Product Title",
      type: "string",
    },
    {
      name: "price",
      title: "Price",
      type: "number",
    },
    {
      title: "Price Without Discount",
      name: "priceWithoutDiscount",
      type: "number",
    },
    {
      name: "badge",
      title: "Badge",
      type: "string",
    },
    {
      name: "image",
      title: "Product Image",
      type: "image",
    },
    {
      name: "category",
      title: "Category",
      type: "reference",
      to: [{ type: "categories" }],
    },
    {
      name: "description",
      title: "Product Description",
      type: "text",
    },
    {
      name: "inventory",
      title: "Inventory Management",
      type: "number",
```



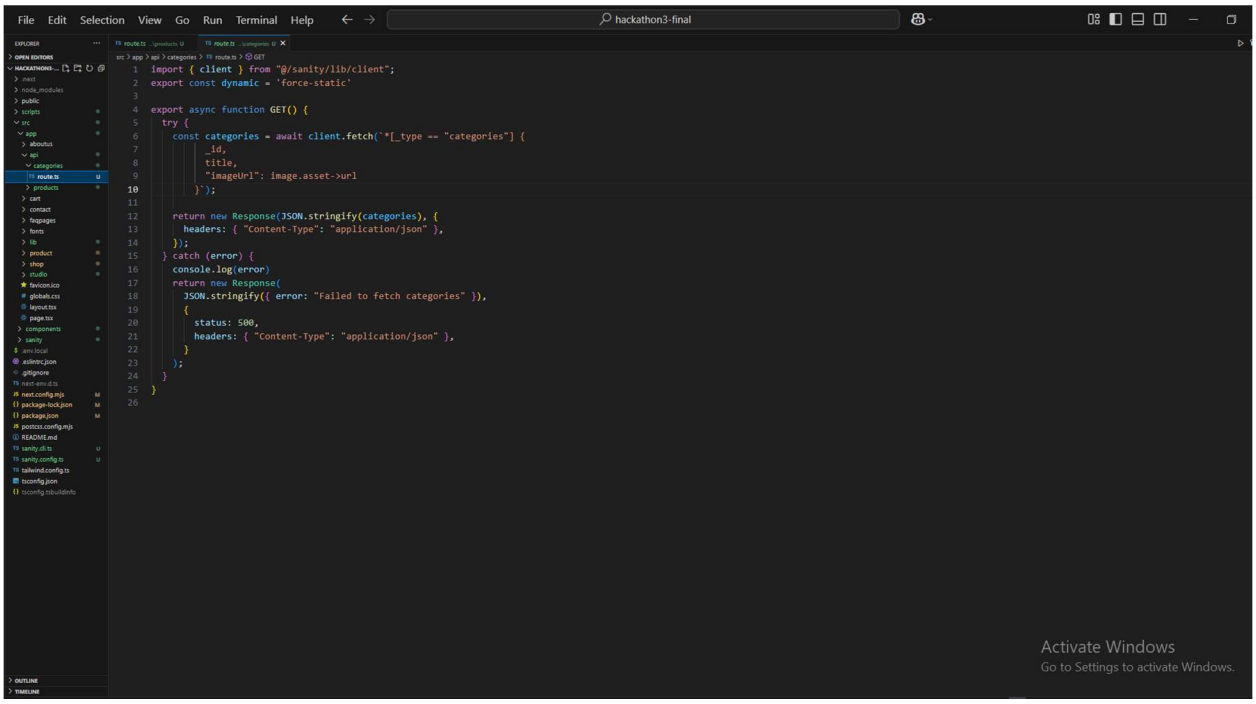
```
import { defineType } from "sanity";

export const categorySchema = defineType({
    name: 'categories',
    title: 'Categories',
    type: 'document',
    fields: [
        {
            name: 'title',
            title: 'Category Title',
            type: 'string',
        },
        {
            name: 'image',
            title: 'Category Image',
            type: 'image',
        },
        {
            title: 'Number of Products',
            name: 'products',
            type: 'number',
        }
    ],
});
```

Data Migration Process:

1. Migrated categories and products to Sanity, ensuring proper data mapping.
2. Managed category mapping effectively to maintain data consistency.



```
import { client } from "@/sanity/lib/client";

export const dynamic = "force-static";

export async function GET(req: { url: string}) {
  try {
    const { searchParams } = new URL(req.url);
    const slug = searchParams.get("slug");

    if (slug) {
      // Fetch a single product by slug
      const product = await client.fetch(
        `*[_type == "products" && _id == $slug][0] {
          _id,
          title,
          price,
          priceWithoutDiscount,
          badge,
          "imageUrl": image.asset->url,
          category-> {
            _id,
            title
          },
          description,
          inventory,
          tags
        }`,
        { slug }
      );

      return new Response(JSON.stringify(product), {
        headers: { "Content-Type": "application/json" },
      });
    }

    // Fetch all products
    const products = await client.fetch(
      `*[_type == "products"] {
        _id,
        title,
        price,
        priceWithoutDiscount,
        badge,
        "imageUrl": image.asset->url,
        category-> {
          _id,
```

```
import { client } from "@/sanity/lib/client";
export const dynamic = 'force-static'

export async function GET() {
  try {
    const categories = await client.fetch(`*[_type == "categories"] {
          _id,
          title,
          "imageUrl": image.asset->url
        }`);

    return new Response(JSON.stringify(categories), {
      headers: { "Content-Type": "application/json" },
    });
  } catch (error) {
    console.log(error)
    return new Response(
      JSON.stringify({ error: "Failed to fetch categories" }),
      {
        status: 500,
        headers: { "Content-Type": "application/json" },
      }
    );
  }
}
```

Finalization:

- Successfully completed the data migration process with no errors.
- Confirmed that the project is ready for front-end integration.