Day 3 Api Intergrations Reports

Welcome to Comforty

A one-stop furniture marketplace that offers ergonomic chairs, trendy stools, and luxurious sofas designed for comfort and style.

Project Overview:

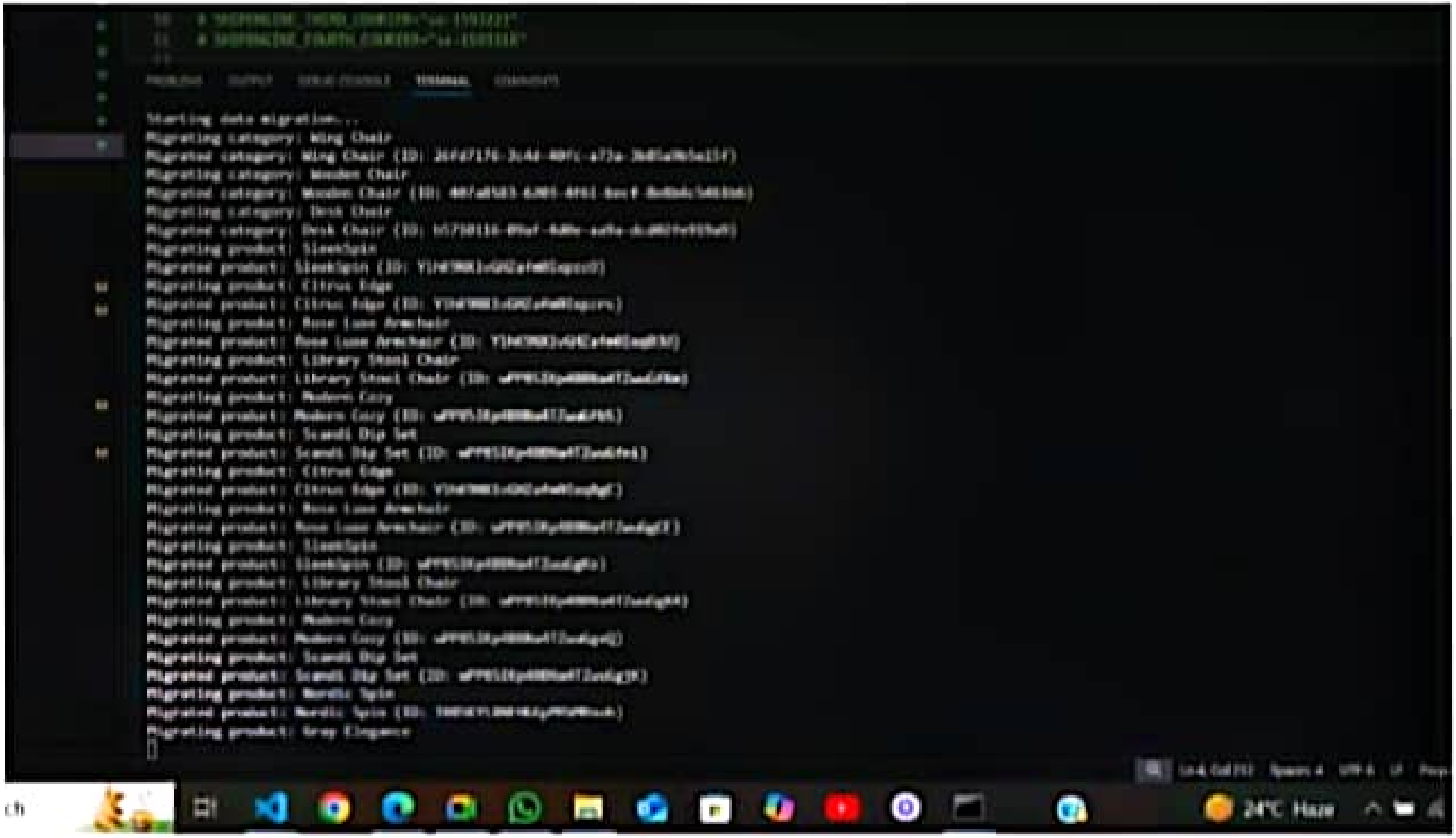
What We Aim to Achieve:

1. Comforty is built using **Next.js** for the frontend and **Sanity CMS** for managing product data.
2. The focus of this hackathon project is to integrate Sanity CMS for managing product listings effectively.
3. We aim to offer a smooth user experience with interactive filters for browsing different furniture types: chairs, stools, and sofas.

Key Features:

- **Real-time data integration** using APIs for product and category details.
- **Seamless migration** of existing data into the Sanity CMS.
- **Attractive, responsive UI** that adapts to all devices.

Import Data in Sanity



Sanity Schema Overview

1 Product Schema Overview

```
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" },
    { name: "priceWithoutDiscount", title: "Price without Discount", 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" },
    {
      name: "tags",
      title: "Tags",
      type: "array",
      of: [{ type: "string" }],
      options: {
        list: [
          { title: "Featured", value: "featured" },
          { title: "Follow products and discounts on Instagram", value: "instagram" },
          { title: "Gallery", value: "gallery" }
        ]
      }
    }
  ]
});
```

2:- Categories Schema Overview

```
const productSchema = {
  name: "product",
  type: "document",
  title: "Product",
  fields: [
    {
      name: "badge",
      title: "Badge",
      type: "string",
    },
    { name: "title", type: "string", title: "Title" },
    { name: "price", type: "number", title: "Price" },
    { name: "image", type: "image", title: "Image" },
    { name: "stock", type: "number", title: "Stock" },
    {
      name: "slug",
      type: "slug",
      title: "Slug",
      options: {
        source: "title",
        maxLength: 200,
      },
    },
  ],
};

export default productSchema;
```

Data Migration Process:-

- NEXT_PUBLIC_SANITY_PROJECT_ID="<Project ID>"
- NEXT_PUBLIC_SANITY_DATASET="production"
- NEXT_PUBLIC_SANITY_AUTH_TOKEN="<Auth Token>"

Migration Code:

```
import "dotenv/config";

// Import the Sanity client to interact with the Sanity backend
import { createClient } from "@sanity/client";

// Load required environment variables
const {
  NEXT_PUBLIC_SANITY_PROJECT_ID, // Sanity project ID
  NEXT_PUBLIC_SANITY_DATASET, // Sanity dataset (e.g., "production")
  NEXT_PUBLIC_SANITY_AUTH_TOKEN, // Sanity API token
  BASE_URL = "https://giaic-hackathon-template-08.vercel.app", // API base URL for products and categor
} = process.env;

// Check if the required environment variables are provided
if (!NEXT_PUBLIC_SANITY_PROJECT_ID || !NEXT_PUBLIC_SANITY_AUTH_TOKEN) {
  console.error("Missing required environment variables. Please check your .env.local file.");
  process.exit(1); // Stop execution if variables are missing
}

// Create a Sanity client instance to interact with the target Sanity dataset
const targetClient = createClient({
  projectId: NEXT_PUBLIC_SANITY_PROJECT_ID, // Your Sanity project ID
  dataset: NEXT_PUBLIC_SANITY_DATASET || "production", // Default to "production" if not set
  useCdn: false, // Disable CDN for real-time updates
  apiVersion: "2023-01-01", // Sanity API version
  token: NEXT_PUBLIC_SANITY_AUTH_TOKEN, // API token for authentication
});

// Function to upload an image to Sanity
async function uploadImageToSanity(imageUrl) {
  try {
```

How Data is Displayed in Sanity Studio

Diagram Breakdown:

- **Product Data Fields:**
  - **Product Title**: Displayed as a text field.
  - **Price**: Displayed as a number field.
  - **Product Image**: Shown as an image field, where you can upload images for each product.
  - **Category**: A reference to the categories schema. Clicking on this allows you to select a category for the product.
  - **Inventory Management**: Displayed as a number field to track product stock.
  -

Category Data Fields:

- **Category Title**: Displayed as a text field for naming the category.
- **Slug**: Shows the URL-friendly version of the category name.
- **Category Image**: A field to upload an image representing the category.

Frontend Display of Product and Category Data

Featured Products

Library Stool Chair
$30

Cesca Chair
$35

Slipper Chair
$69

Deco Chair
$22

Top Categories

Slipper Chair

Conclusion

Comforty *is now fully equipped with an efficient data management system using Sanity for handling products and categories.*

Future developments can include integrating dynamic categories and more advanced filtering options.