Day 3 - API Integration and Data Migration

1. Introduction

This report outlines the process of integrating product and category data from an external API into the backend system of Comforty, a furniture e-commerce platform specializing in chairs. The integration utilizes Sanity CMS for efficient content management and Next.js for seamless frontend rendering, ensuring a dynamic and user-friendly shopping experience.

Sanity Installation and Schema Setup for Comforty

To manage product and category data effectively, we integrated Sanity CMS into Comforty. Here's how the process was executed:

1. Sanity Installation

Sanity was installed as the content management system for Comforty. For step-by-step instructions, we referred to the **Sanity Setup Guide**. Since the project was built on an existing Next.js application, the initial step of creating a new project was skipped.

2. Schema Setup
After completing the installation, schemas for products and categories were created to structure the data:

- **Navigate to the Schema Folder**:

  - If using a `src` folder: `/src/sanity/schemaTypes/`
  - Otherwise: `/sanity/schemaTypes/`
- **Create Schema Files**:
  Two schema files were added:

- **products.ts**: Located at
    `/src/sanity/schemaTypes/products.ts`
  - **categories.ts**: Located at
    `/src/sanity/schemaTypes/categories.ts`

2. Fetching Data from API

To integrate product and category data into Comforty, data was dynamically fetched from the following external API:

API URL:
https://giaic-hackathon-template-08.vercel.app/api/products

Using this API, product information such as names, prices, descriptions, categories, and images were retrieved. The data was then structured and migrated into Sanity CMS, ensuring compatibility with the platform's templates and frontend design. This dynamic fetching ensures the product listings are always up-to-date with the latest information.

```js
scripts > JS migrate.mjs > ...
  1    // Import environment variables from .env.local
  2    import "dotenv/config";  7.5k (gzipped: 3.2k)
  3
  4    // Import the Sanity client to interact with the Sanity backend
  5    import { createClient } from "@sanity/client";  324.2k (gzipped: 85.4k)
  6
  7    // Load required environment variables
  8    const {
  9      NEXT_PUBLIC_SANITY_PROJECT_ID, // Sanity project ID
 10      NEXT_PUBLIC_SANITY_DATASET, // Sanity dataset (e.g., "production")
 11      NEXT_PUBLIC_SANITY_AUTH_TOKEN, // Sanity API token
 12      BASE_URL = "https://giaic-hackathon-template-08.vercel.app", // API base URL for products and categories
 13    } = process.env;
 14
```

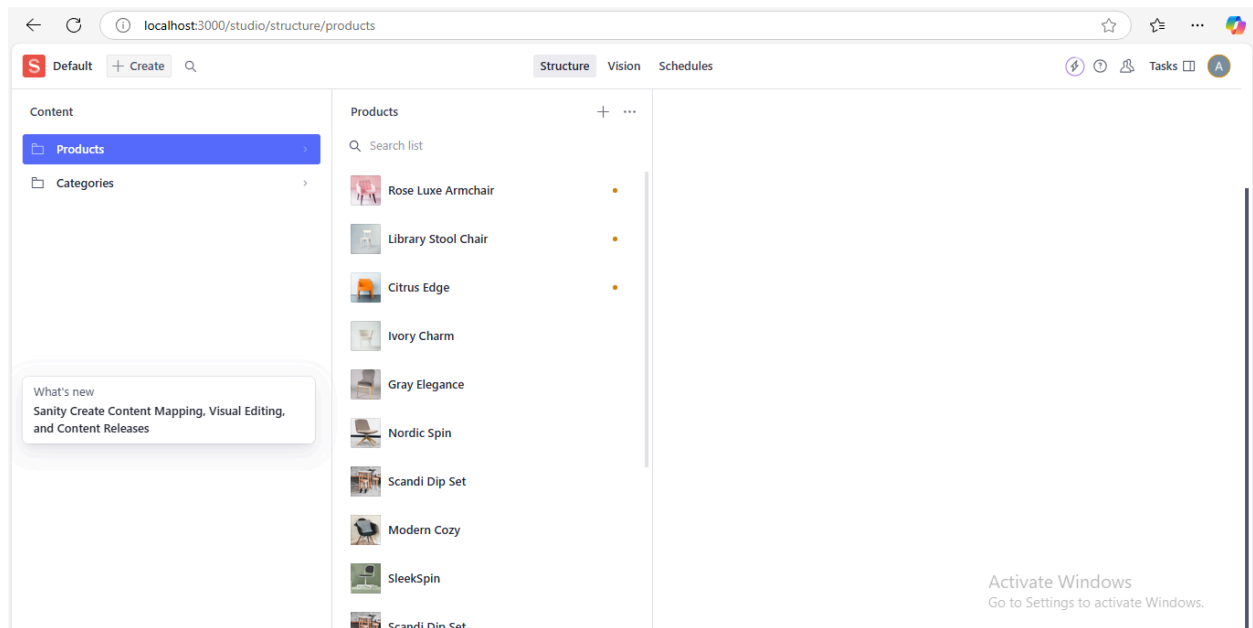## 3. Migrating Data to Sanity CMS

For Comforty, the dynamically fetched chair data was structured and migrated into Sanity CMS using its JavaScript client library. The product schema was enhanced to include additional fields for detailed data management tailored to Comforty's requirements.
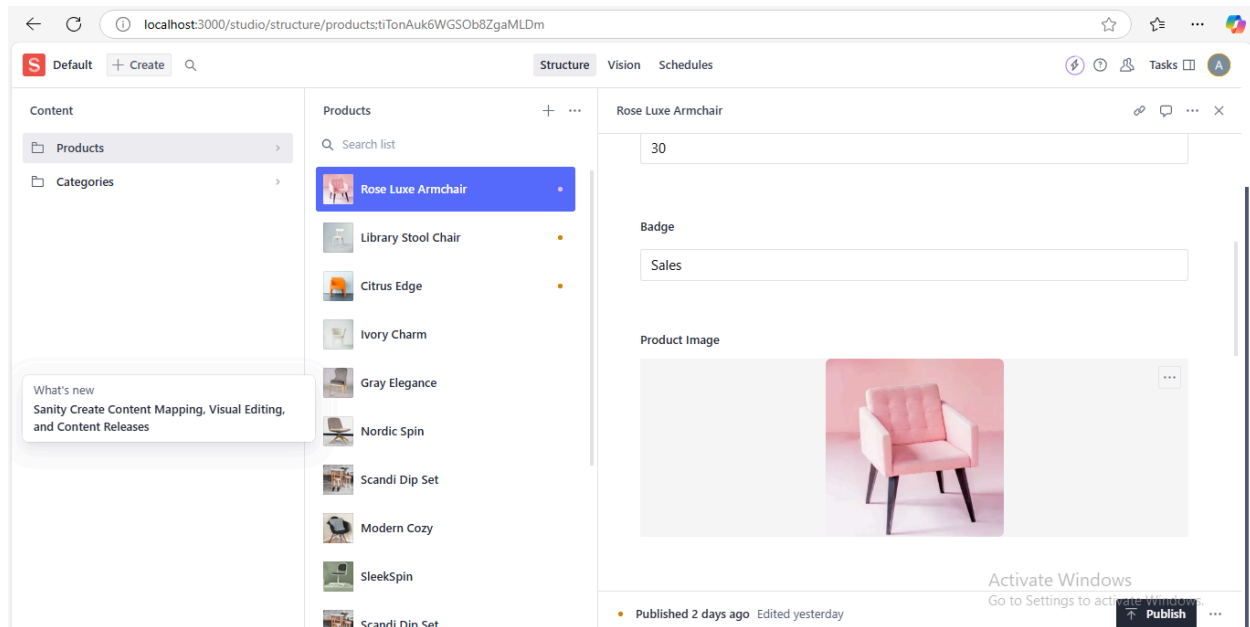
Key Schema Fields:

- **title**: Name of the chair.
- **priceWithoutDiscount**: Original price of the chair before any discounts.
- **badge**: Promotional labels like "Sale" or "New Arrival."
- **category**: References the chair's category (e.g., office chairs, dining chairs).
- **image**: URL for the chair's image.
- **inventory**: Stock level of the chair.
- **tags**: Tags such as "Featured" or "Trending" to improve discoverability.

These schema fields ensure comprehensive data management and seamless integration of chair products with Comforty's templates and frontend display.

Sanity Studio

4. Product Display

The product listing page for Comforty dynamically renders chair products fetched from the API. Each product is displayed within a responsive grid layout, showcasing essential details such as the product name, price, and image. This ensures a visually appealing and user-friendly browsing experience.

Key features of the product listing page include:

- **Dynamic Rendering**: Products are fetched in real-time and displayed instantly on the frontend.
- **Responsive Design**: The grid layout adjusts seamlessly to different screen sizes for better accessibility.
- **Essential Details**: Each chair product card includes the name, price, and an image.

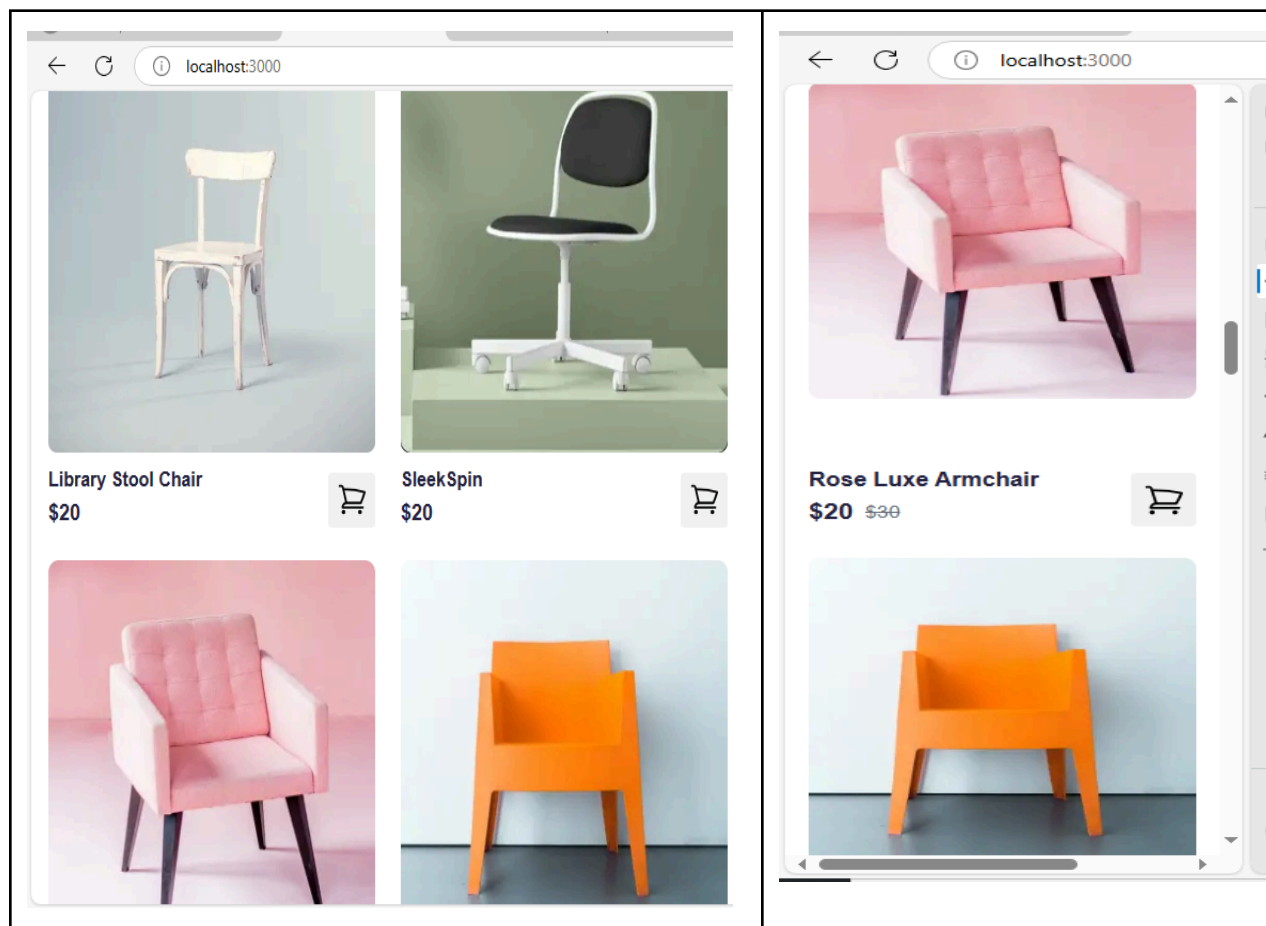## Featured Products

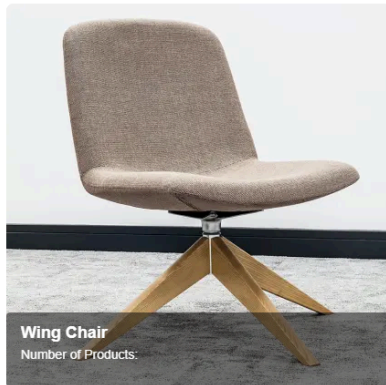**Library Stool Chair**
$20

SleekSpin
$20

Rose Luxe Armchair
$20 $30

Citrus Edge
$20

Responsive Design

localhost:3000

Library Stool Chair
$20

SleekSpin
$20

localhost:3000

Rose Luxe Armchair
$20 $30

Day 3 Summary: API Integration and Data Migration

Day 3 focused on integrating APIs and migrating data into Sanity CMS to build a scalable backend for Comforty. Product and category data were dynamically fetched from an external API, structured, and stored in Sanity CMS with enhanced schema fields like title, price, badge, and inventory. A dynamic product listing page was created in Next.js with a responsive grid layout, displaying chair products with essential details. This ensured a robust backend and an engaging frontend experience.

Author: Shumaila Waheed
Slot: Saturday 2 to 5