MARKET PLACE
E-COMMERCE FURNITURE

18-01-2025

Ayan Sheikh

DAY-3 API INTEGERATION

1. Overview

API integration is one of the most crucial aspects of any website, and it's not just about integrating an API. The real challenge lies in **handling the API**, which is often one of the toughest tasks. Interestingly, this is also my **favorite part** of development. API integration is not as simple as just dealing with data; it's about ensuring smooth communication between the backend and frontend, managing data flow, and making sure everything works seamlessly.

Today, I worked on integrating the **API for the furniture website**, focusing on **product** and **category** data. This step is essential to provide dynamic content on the platform and ensure that products and categories are displayed accurately for the users.

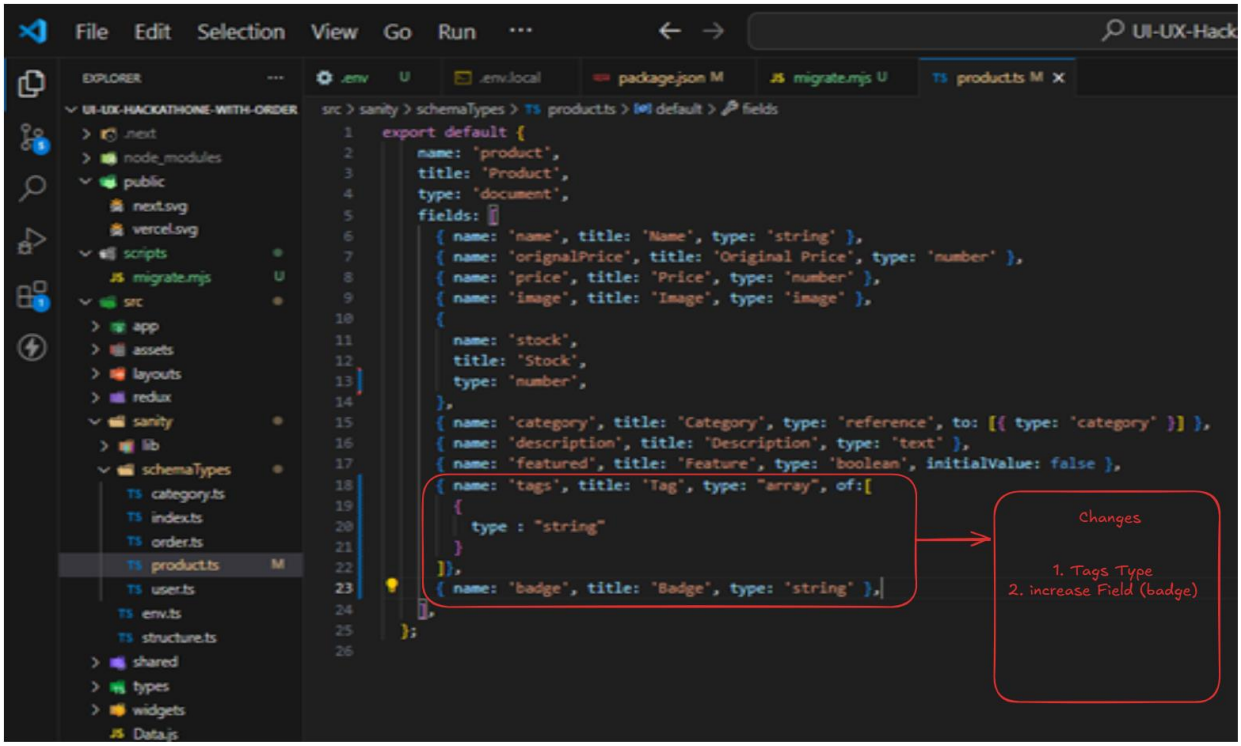
2. Api Understanding

- We gained a solid understanding of API functionality and how to integrate it within our system
- Identified key endpoints for managing **product** and **category** data.
- Planned integration points with the frontend to fetch and display product details dynamically.

3. Schema Adjustment

Define the Schema:

- After making the necessary adjustments based on the provided API, we updated the schema for the furniture products. The schema includes important fields such as **Product Name**, **Price**, **Size**, **Availability**, **Category**, and **Image**.
- These fields were chosen carefully to ensure all critical information about the products is captured



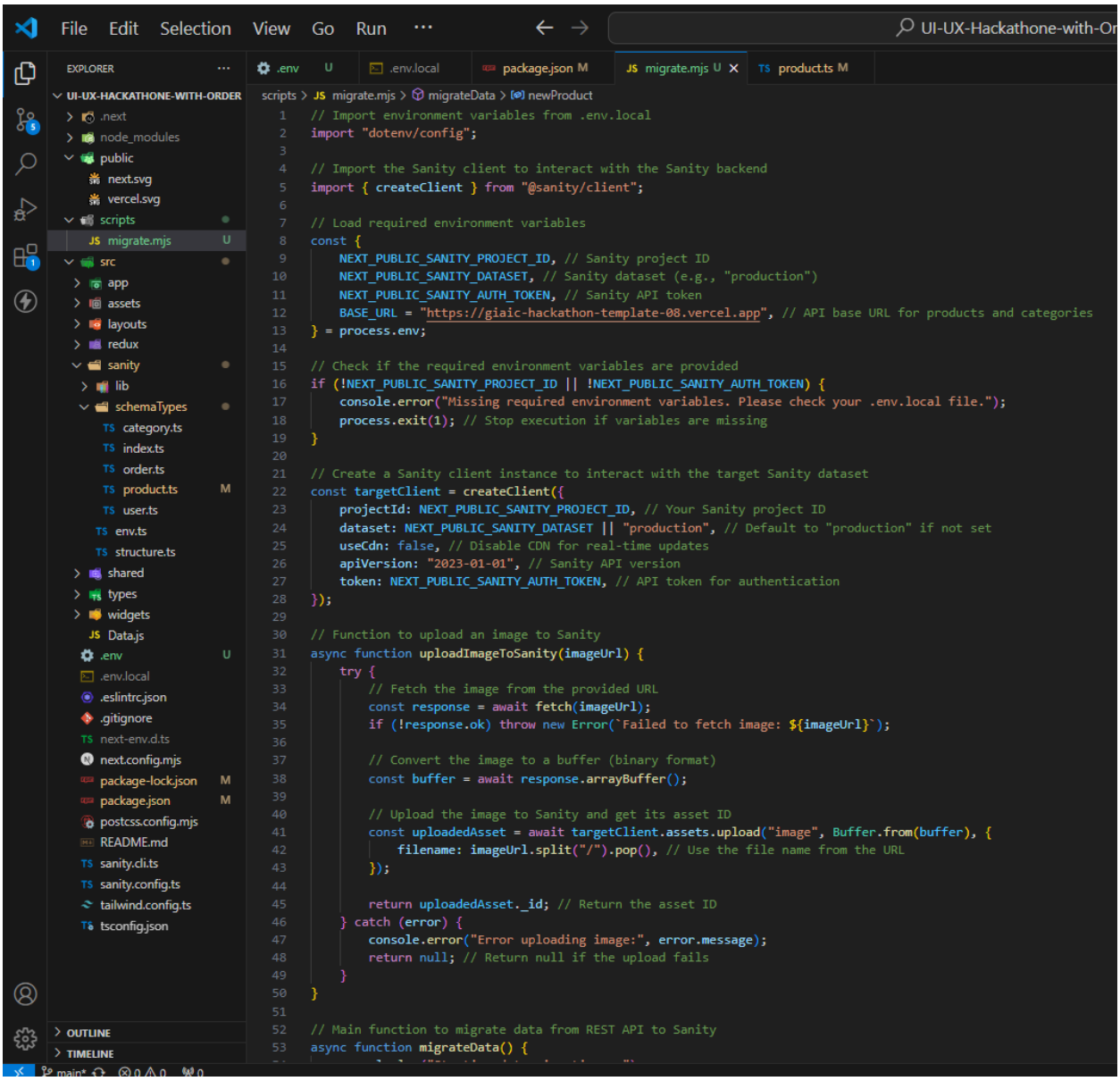
Check List:

Task Category	Task Description	Status
API Setup	Set up API to fetch furniture product data from the backend and ensure proper endpoints are created.	✓
API Integration	Integrate API with frontend using Axios/Fetch. Ensure correct response and error handling.	✓
Data Migration	Migrate product and category data to the given api and verify data in sanity.	✓
Fetching Data	Set up data fetching using Axios/Fetch API. Handle loading states and errors.	✓
Dynamic Data Display	Map over fetched data to display product info (name, price, category, image, etc.).	✓
Styling & Responsiveness	Style the page using Tailwind CSS. Make the layout responsive.	✓
Error Handling	Display loading spinners and error messages when necessary.	✓

4. Miigration

- Successfully migrated product and category data into **Sanity CMS**
- Extracted Data from given api and store in Sanity through migration

- Transformed the data to align with the updated schema structure.

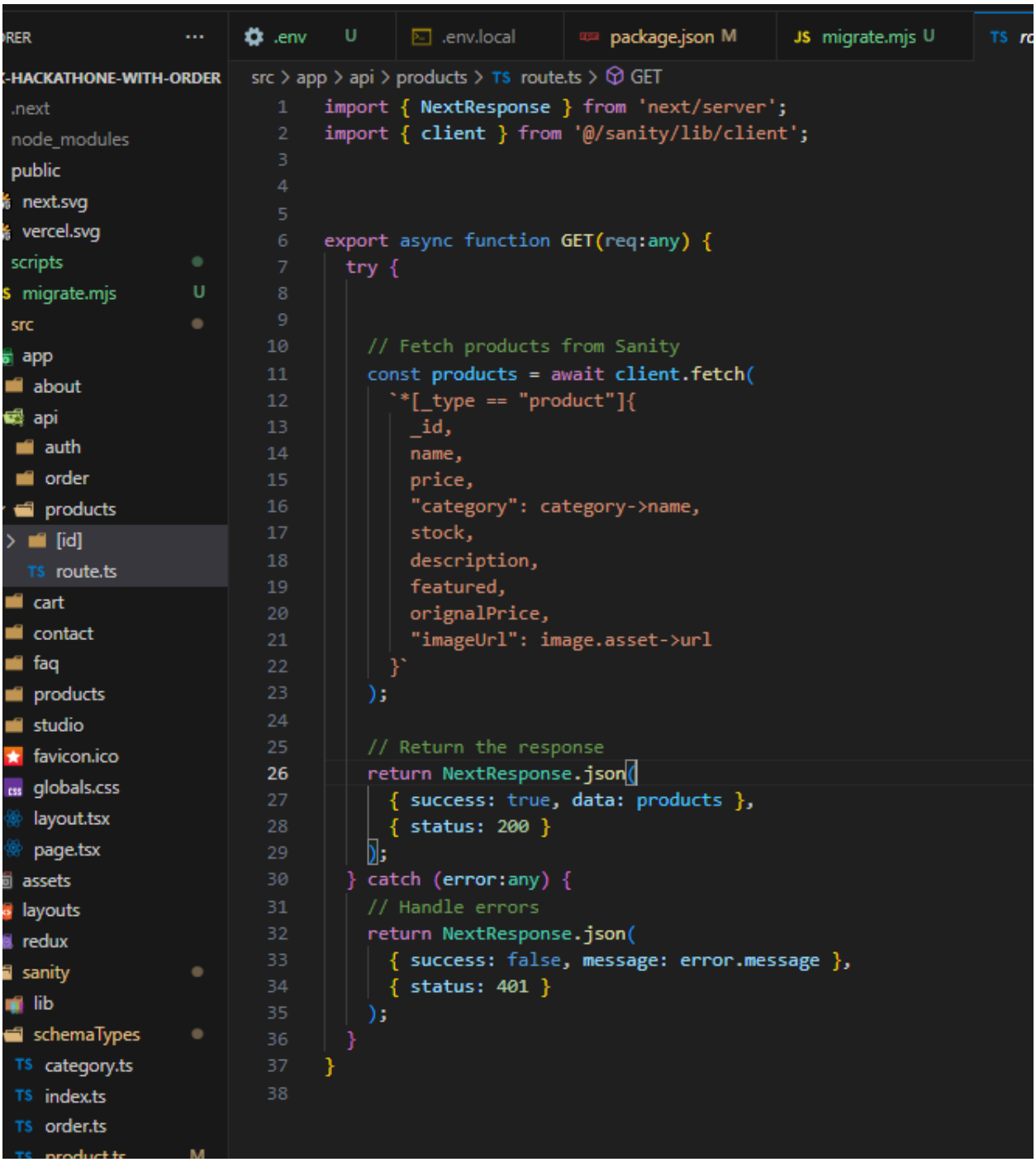


5. Fetch Data in Next js

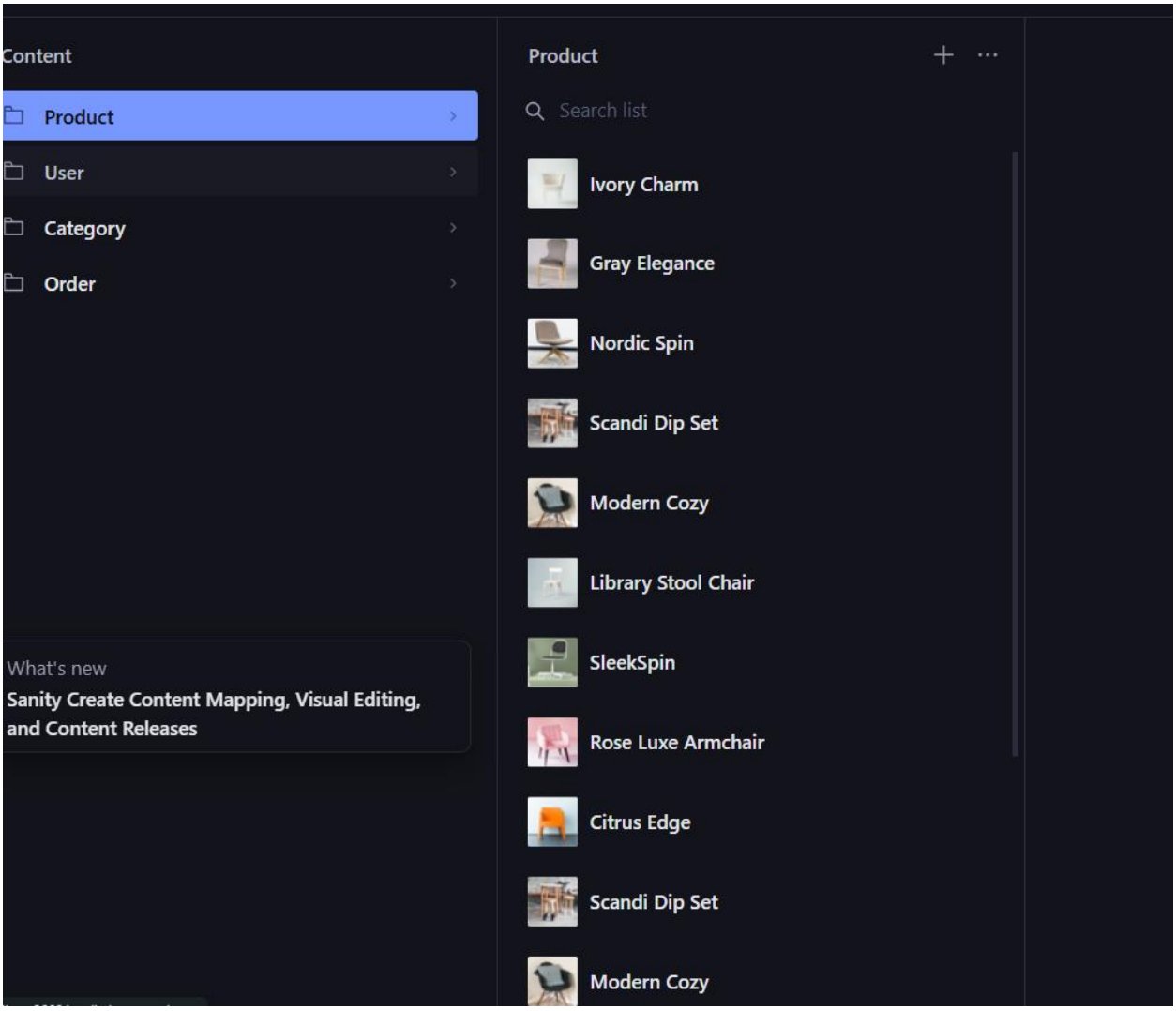
- Implemented API calls to fetch **furniture product** and **category** data dynamically.
- Set up proper state management to store the fetched data and enhance the user experience.

- Conducted tests to ensure the API integration is efficient and data is accurately displayed

Code Screenshot:



6. Verify Data in Sanity



7. Show On Frontent