🚀 Hackathon Day 3:
API Integration Report - [Furniture website]

Objective:

To streamline the process of fetching data from an external API, storing it in Sanity CMS, and displaying it dynamically on the Furniture website.

---

1. API Integration Process:

Step 1: Understanding API Endpoints

- Identify the external API providing furniture data (e.g., products, categories, prices, and images).

- Example endpoint: https://next-ecommerce-template-4.vercel.app/api/product
- API authentication handled using tokens or API keys.

Step 2: Fetching Data from API

- Use tools like Axios or Fetch API in Next.js to make GET requests to the API.
- Implement error handling to account for potential API response issues, such as timeouts or missing fields.
-

2. Storing Data in Sanity CMS:

Step 1: Setting Up Sanity Schema

- Define schemas for furniture data in Sanity (e.g., `product`, `category`).
- Example schema for a product:

```
export default {

  name: 'product',

  title: 'Product',

  type: 'document',

  fields: [

    { name: 'name', title: 'Name', type: 'string' },

    { name: 'price', title: 'Price', type: 'number' },

    { name: 'image', title: 'Image', type: 'image' },

    { name: 'description', title: 'Description', type: 'text' },

    { name: 'category', title: 'Category', type: 'reference', to: [{ type: 'category' }] },

  ],

};
```

Step 2: Ingesting API Data into Sanity

- Use Sanity's client library to insert data programmatically.
- Code example for adding products:

```
import sanityClient from '@sanity/client';


const client = sanityClient({

  projectId: 'yourProjectId',

  dataset: 'production',

  useCdn: false,

  apiVersion: '2023-01-01',

  token: 'YOUR_SANITY_TOKEN',

});
```

3. Displaying Data on Furniture Website:

Step 1: Dynamic Routes for Product Pages

- Use Next.js dynamic routing to create pages for individual products.
- Folder structure: `/app/products/[slug]/page.tsx`

Step 2: Product Grid Display

- Fetch all products and display them on the homepage or a dedicated catalog page.

4. Conclusion:

This process ensures efficient data handling by integrating external APIs with Sanity CMS and dynamically displaying data on the Furniture website. It leverages modern frameworks like Next.js and the flexibility of Sanity to deliver a robust user experience.