Kinza Khan

Day 3 - API Integration Report - General Ecommerce Furniture Website

1. API Integration Process

Used Mock API Tool to create a mock API for testing designed the API structure, ensuring it aligned with the required data model set field types in the schema based on API response Used Thunder client to test the API's functionality and validate responses.

2. Schema Adjustments

Defined a schema in Sanity that matched the mock API structure ensured proper data types for each field avoid inconsistencies adjusted schema fields based on API responses to maintain data integrity

3. Migration Steps & Tools Used

Created a migration file to import API data into Sanity executed migration script to transfer API data to the Sanity database used Sanity Vision write GROQ queries for data retrieval developed a utility function to fetch data from Sanity to the frontend rendered the fetched products dynamically on the fronten

Create Sanity Schema

```
type: string,
                                                   title: 'Name',
  studio \ [[...tool]]
   page.tsx
                                                   name: 'image',
  * favicon.ico
                                                   type: 'image',
  # globals.css
                                                   title: 'Image',
  layout.tsx
  not-found.tsx
                                                     hotspot: true, // Enable image cropping
  page.tsx
  components

∨ Context

                                                   name: 'price',
  cart-context.tsx
                                                   type: 'number',
  > hooks
                                                   title: 'Price',
 > lib
  sanity
                                                   name: 'description',
                                                   type: 'text',
  schemaTypes
                                                   title: 'Description',
   TS index.ts
   TS product.ts
  TS env.ts
                                                   name: 'discountPercentage',
  TS structure.ts
                                                   type: 'number',
 .env.local
                                                   title: 'Discount Percentage',
 eslintrc.json
  .gitignore
                                                   name: 'isFeaturedProduct',
 components.json
                                                   type: 'boolean',
OUTLINE
                                                   title: 'Featured Product',
 TIMELINE
o ∆ o ∆ o
                                                                                                                           Ln 51, Col 3 Spaces: 4
```

Migation file help to import API data into Sanity

```
UI-UX-HACKHATHONE-... [ → □ Scripts > Js importSanityData.mjs > 🗘 uploadImageToSanity
                                         import { createClient } from '@sanity/client';
> .next
                                         import fetch from 'node-fetch';
> node modules
> public
                                        // Initialize Sanity client

✓ scripts

                                         const client = createClient({
JS importSanityData.mjs
                                           projectId: "7h9fzpzu",
                                           dataset: "production",
∨ src
                                           useCdn: false, // Set to true if you want faster reads
 ∨ app
                                           apiVersion: '2025-01-13',
 > about
                                           token: "skicPUauQu9WRH6QKvq2JlwCeodqTwjkBxdRtFXVQWZ6CNR01EKWSgS6ZopAVPmmJIDm8mX4nRfcZ00LfkKYZqy13WCQY
  > account
                                         });
 > api
                                   12
 > cart
                                         // Function to upload an image to Sanity
                                   13
 > checkout
                                         async function uploadImageToSanity(imageUrl) {
                                   14
  > contact
                                           try {
                                             console.log(`Uploading image: ${imageUrl}`);
 > fonts
                                   17
 > heart
                                             const response = await fetch(imageUrl);
  ∨ shop
                                             if (!response.ok) {
  [productId]
                                               throw new Error(`Failed to fetch image: ${imageUrl}`);
   page.tsx
                                   21
  page.tsx
                                             const buffer = await response.arrayBuffer();
  v studio \ [[...tool]]
                                             const bufferImage = Buffer.from(buffer);
  page.tsx
                                   25
 * favicon.ico
                                             const asset = await client.assets.upload('image', bufferImage, {
 # globals.css
                                               filename: imageUrl.split('/').pop(),
 layout.tsx
                                             });
 not-found.tsx
                                   29
                                             console.log(`Image uploaded successfully: ${asset._id}`);
 page.tsx
                                             return asset._id;
OUTLINE
                                           } catch (error) {
TIMELINE
```

Sanity Structure

and Content Releases

