Day 3 - API Integration Report

Reports:

API Integration Process

For Day 3, we integrated the **DummyJSON API** (`https://dummyjson.com/products/furniture`) to fetch furniture products and tested them to use in my website.

Steps Taken:

Searched for a Detailed API

- I want very detailed level data for my products. That's why I searched for an API which contains everything for a product like stock, dimensions, ratings, colors, shipping information, warranty etc.

Checked API Responses

- Tested the API responses using Postman to make sure it had all the required data.

Adjustments made to schema

Steps Taken:

Made a Custom Schema

- Because I'm using my custom API so I created a new schema in Sanity to match my API data.

Migration steps and tools

Steps Taken:

Created a Migration Script

- Used GPT to generate a script that transfers product data to Sanity.

Made an env file

- Because I'm inserting data in sanity so I need environment variables like projectId, dataset, token etc so I made an env file for to keep my secrets secure.

Matched Schema and API Data

- Ensured the API response and schema fields were correctly aligned.

Added Migration Script to package.json

- Included a migration command in `package.json`.

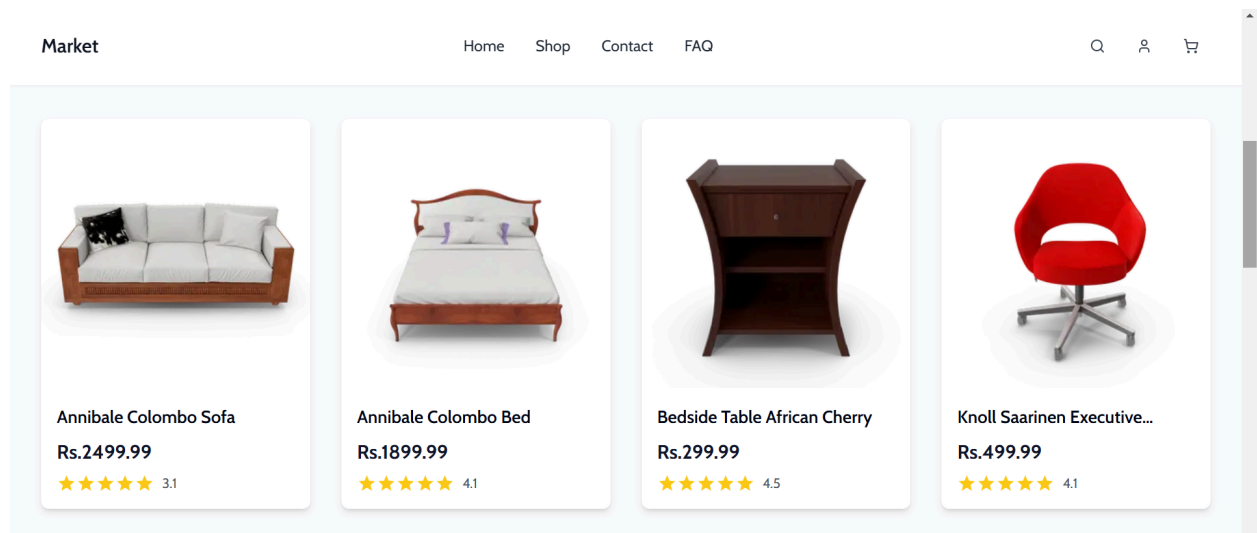### Ran the Migration Script

- Used `npm run migrate` to transfer data from API to Sanity.

Checked GROQ Queries

- Verified that our GROQ queries correctly retrieved data from Sanity.

Tested API with Postman

- Used Postman to confirm the API calls were working properly.

Screenshots:

API Calls in Postman:

Data Displayed in the Frontend:

Populated Sanity CMS Fields: