PREPARED BY: HUBAB IKRAM
MONDAY (2 TO 5 PM)

HACKATHON_DAY_3

"API INTEGRATION REPORT – [COMFY]"

Overview of Marketplace

COMFY is an online furniture marketplace that offers a range of interior products such as sofas, chairs, and tables. The platform is built using **Next.js, Tailwind CSS, and Sanity CMS**, ensuring a seamless shopping experience with an intuitive user interface, efficient API integration, and a robust backend.

Objective

The objective is to develop a fully functional e-commerce platform that delivers a smooth user experience, dynamic product management, and easy integration with third-party APIs for product data. The website aims to enhance accessibility, responsiveness, and scalability for future growth.

Approach

- **Frontend:** Utilize Next.js and Tailwind CSS for a fast and responsive user interface.
- **Backend:** Implement Sanity CMS for structured content management.
- **API Integration:** Fetch product data from external sources and sync it with the Sanity backend.
- **Data Migration:** Automate the import of structured product data into Sanity.
- **Optimization:** Ensure smooth navigation, fast loading times, and a user-friendly interface.

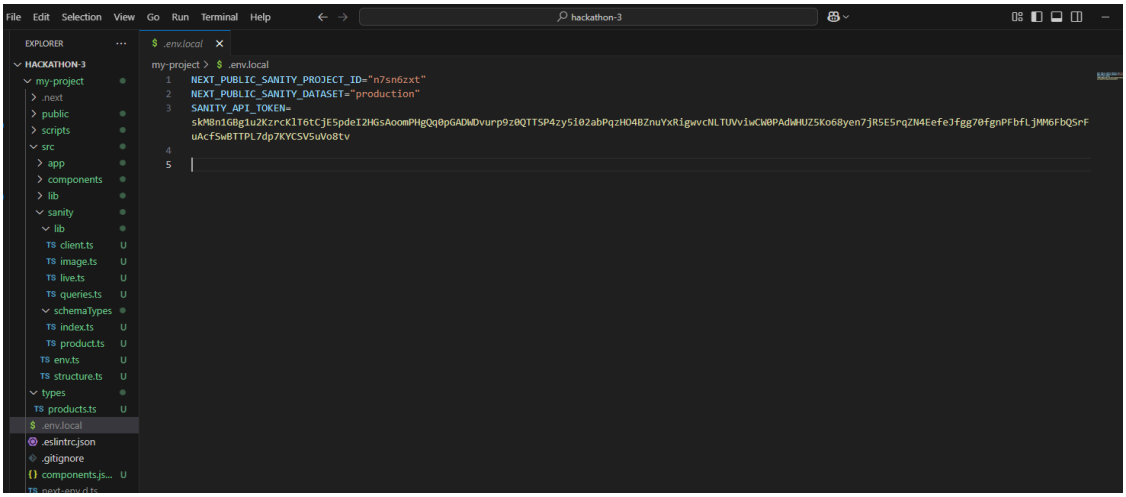
Goals to Achieve

- **Implement API Integration:** Connect the frontend with Sanity CMS and third-party APIs for product data.
- **Schema Adjustments:** Modify Sanity schemas to accommodate product details like name, price, image, and category.
- **Data Migration Process:** Extract, transform, and import product data into Sanity CMS.
- **Use of Migration Tools:** Utilize scripts and Sanity tools for automated data migration.
- **Verify API Calls:** Ensure successful API responses and data retrieval.
- **Frontend Data Display:** Confirm that fetched data is correctly rendered on the website.
- **Sanity CMS Population:** Validate that migrated data is properly stored and structured in Sanity CMS.
- **Code Implementation:** Provide API integration and migration script snippets for future reference.

API Integration Process

- **Schema Definition:** Created product.ts in the Sanity schema folder to structure product data.
- **Type Definitions:** Defined types in a separate types folder for better TypeScript support.
- **Data Fetching:** Implemented API calls in a dedicated file to keep logic separate and organized.
- **Scripts for Migration:** Developed a script to fetch product data and import it into Sanity.

PASTING OF API TOKEN IN .ENV.LOCAL FILE

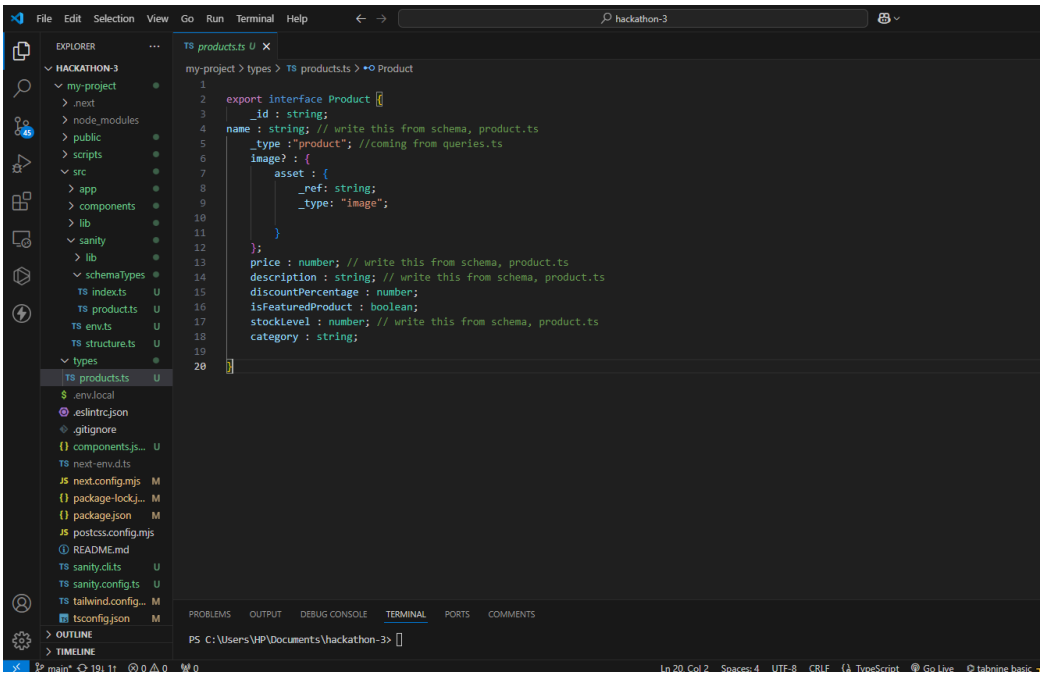


2. Schema Adjustments

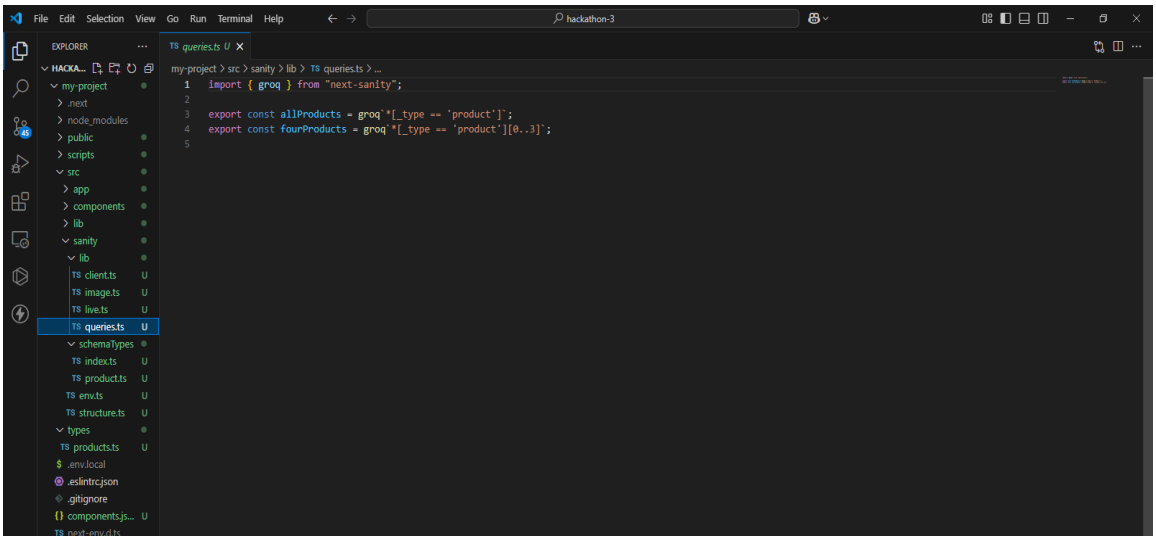
- **Created product.ts schema** with fields:
    - name (string)
    - id (number)
    - price (number)
    - image (image)
    - description (text)
    - category (string)
    - discountpercentage (number)
    - stocklevel (number)
    - trending/featured (boolean)

```ts
// product.ts U
// my-project > src > sanity > schemaTypes > product.ts
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Name',
      validation: (Rule: any) => Rule.required().error('Name is required'),
    },
    {
      name: 'image',
      type: 'image',
      title: 'Image',
      options: {
        hotspot: true,
      },
      description: 'Upload an image of the product.',
    },
    {
      name: 'price',
      type: 'string',
      title: 'Price',
      validation: (Rule: any) => Rule.required().error('Price is required'),
    },
    {
      name: 'description',
      type: 'text',
      title: 'Description',
      validation: (Rule: any) =>
        Rule.max(150).warning('Keep the description under 150 characters.'),
    },
    {
```

- Added Types in **types/product.ts** for better TypeScript support.



- Make a queries.ts file inside sanity > lib folder.

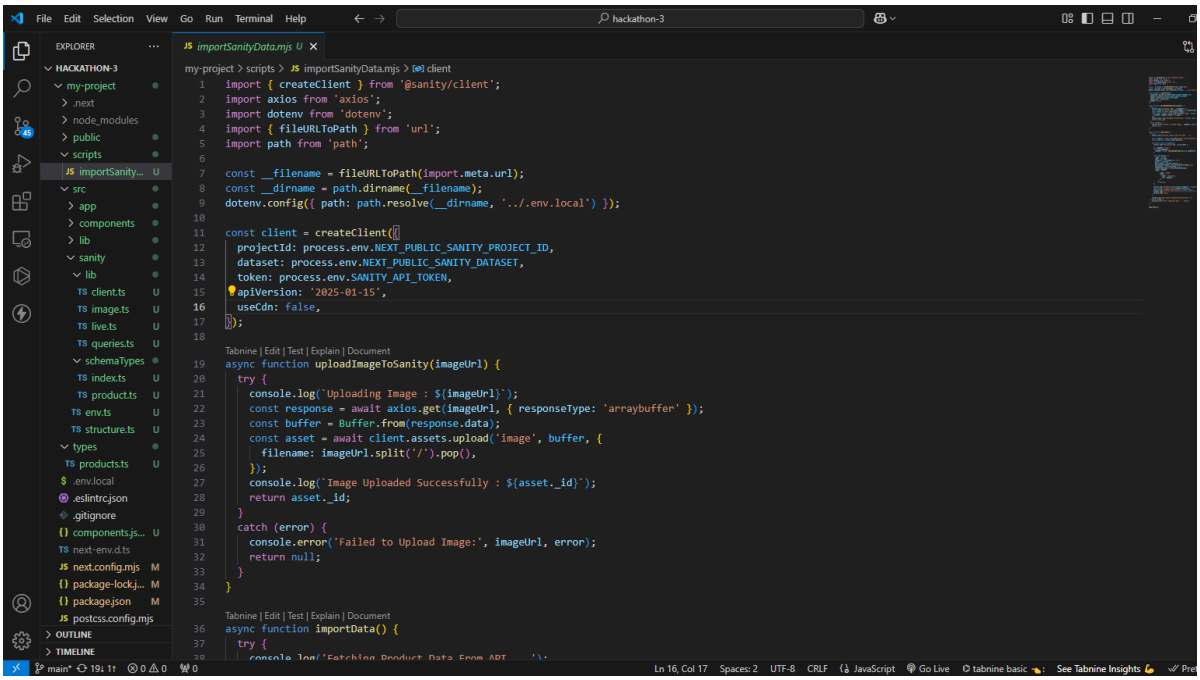


3. Data Migration Steps

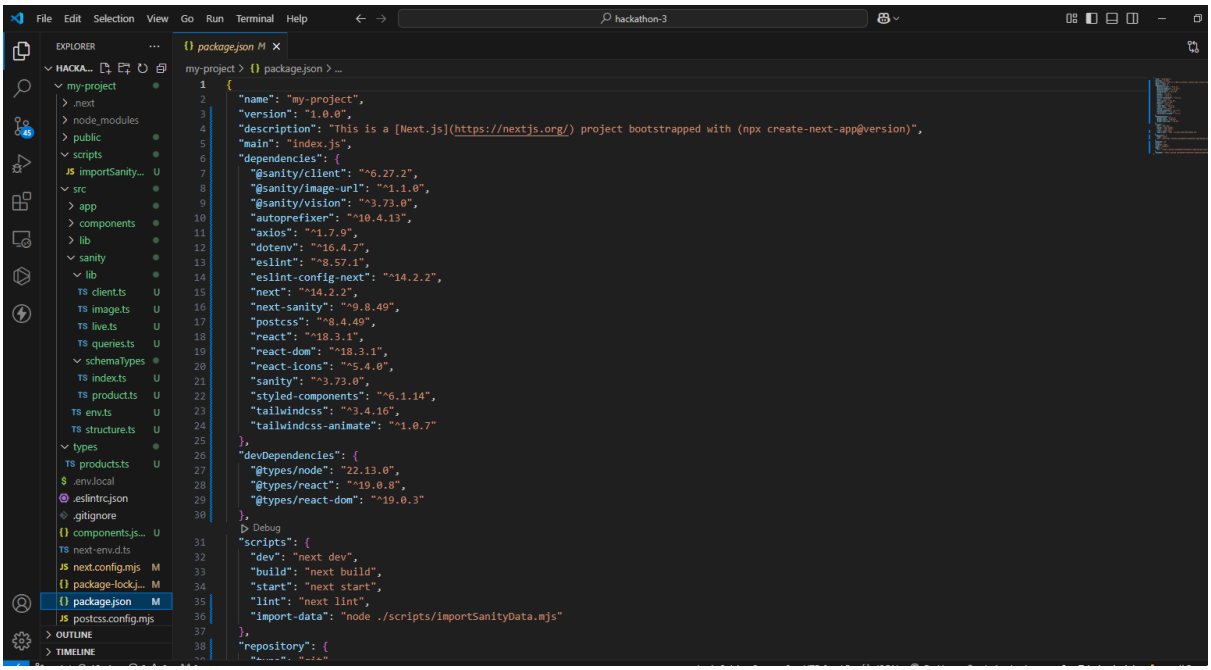
- **Fetch Data from API:** Called the third-party API to retrieve product details.
- **Transform Data:** Converted API response into the Sanity-compatible format.
- **Import into Sanity:** Used the Sanity client to add the data programmatically.

STEPS FOR MIGRATION:-

1. Created a script named **importSanityData.js** to migrate data into Sanity.



2. Added import data to **package.json** file.



3. Installed requires dependencies**: npm install @sanity/client axios dotenv**
4. Run the import Script**: "npm run import-data"**
5. The product data successfully migrated into Sanity CMS and Groq queries helped to fetch the data into project.

Testing and Validation

API TESTING

- Tools used like Postman
- Verified successful API calls and responses

VALIDATION CHECKLIST

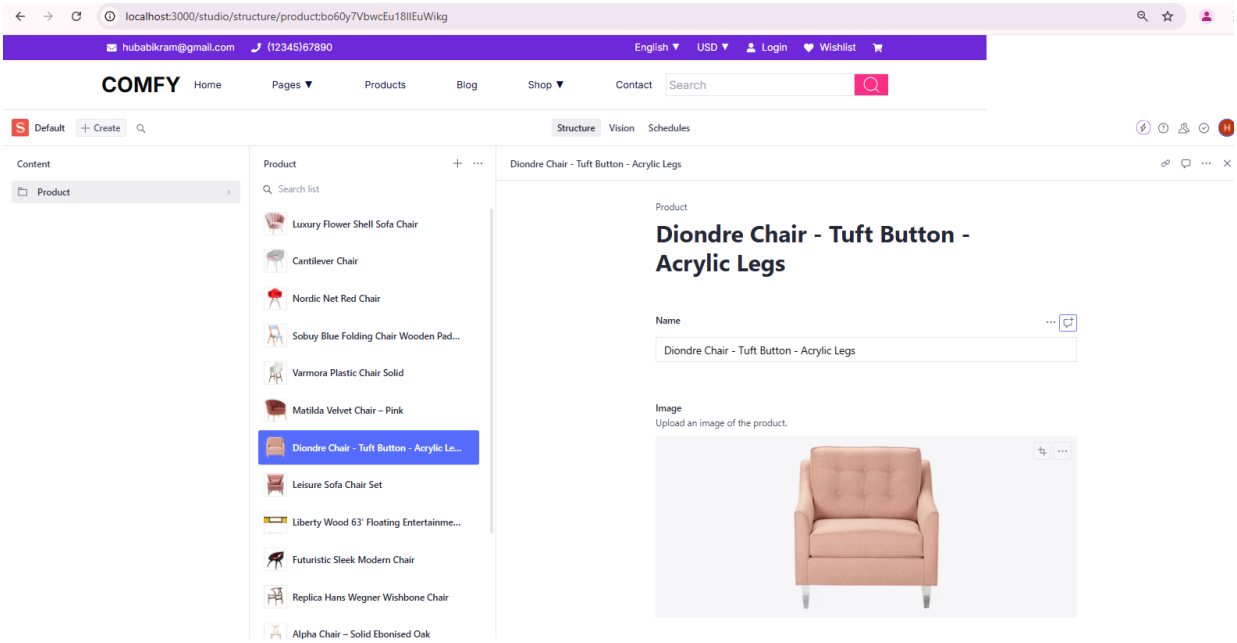
- Populated Sanity CMS ✓
- Data correctly displayed on the frontend ✓
- Errors handling ✓

Result & Output

DATA IN SANITY CMS

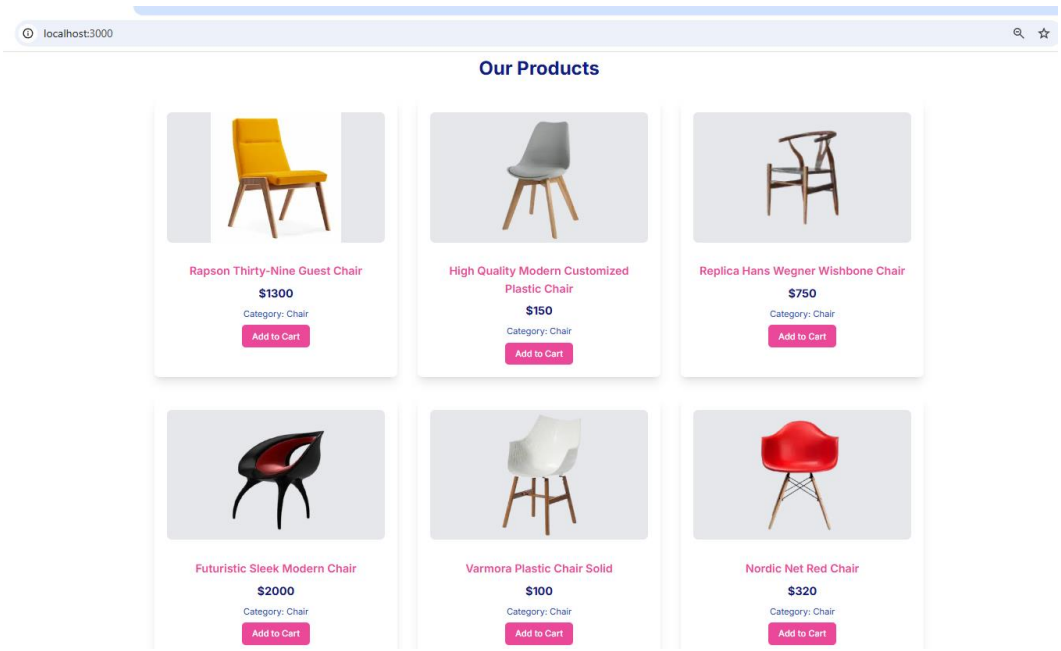
Products with fields like:

- Name
- Price
- Image
- Category



FRONTEND DISPLAY

Product details are showing up properly.



Conclusion

The API integration for COMFY was successfully executed, enabling efficient data flow between external sources and Sanity CMS. Product information, including name, price, image, and category, was migrated seamlessly and displayed correctly on the frontend. The implementation ensures a scalable, responsive platform, providing a smooth user experience while simplifying future content management and updates.

SUBMITTED BY: HUBAB IKRAM

SLOT: MONDAY (2 TO 5 PM)

TASK GIVEN BY: SIR AMEEN ALAM

CLASS TEACHER: SIR ASHARIB ALI & AHMED RAZA SHAIKH