

Amber

243034

Friday 9-12 (Morning)

Api Integration Report - [Heckto Website]

Reviewed API Documentation:

I carefully read the provided API documentation for my assigned template to understand the available endpoint `{/products}`

I identified the structure of the data returned by API ,including field names and data types.

Compared API Data with Sanity Schema

I reviewed API data structure and compared it with the existing schema in sanity CMS. I identified mismatches in feildl names and data types.

- API Field : `product_title` Sanity Fied: `name`
- API Field: `price` Sanity Field `price` (with proper data type)

I added new fields in sanity CMS to accommodate additional data from the API ,because the API is not enough to complete website product and their details.

To Migrate data from the API to Sanity CMS, I followed these steps:

I decided to use the provided API to fetch data and write a script to import it into

Sanity CMS.

- I create a script folder and then i create a migration file to fetch data from the api and transform it into formate required by Sanity CMS.
- I used the Sanity client library to upload the data to the CMS. I ran the migration script to import product data, categories and other relevant information into Sanity CMS.

In this project ,I successfully integrated the provided API into my next.js frontend and migrated data into Sanity CMS. I adjusted the schema to match the API data structure and ensured the data was accurately displayed in the frontend.This exercise helped me gain practical experience in API integration , data migration and schema validation ,which are essential skills for buildings scalable marketplace.

TASK	Status
API Understanding	<input checked="" type="checkbox"/>
Schema Validation	<input checked="" type="checkbox"/>
Data Migration	<input checked="" type="checkbox"/>
API Integration in Next.js	<input checked="" type="checkbox"/>
Submission Preparation	<input checked="" type="checkbox"/>

TS product.ts X

src > sanity > schemaTypes > TS product.ts > [0] default

```
1 export default {
2   name: 'product',
3   type: 'document',
4   title: 'Product',
5   fields: [
6     {
7       name: 'name',
8       type: 'string',
9       title: 'Name',
10      validation: (Rule: any) => Rule.required().error('Name is required'),
11    },
12    {
13      name: 'image',
14      type: 'image',
15      title: 'Image',
16      options: {
17        hotspot: true,
18      },
19      description: 'Upload an image of the product.',
20    },
21    {
22      name: 'price',
23      type: 'string',
24      title: 'Price',
25      validation: (Rule: any) => Rule.required().error('Price is required'),
26    },
27    {
28      name: 'description',
29      type: 'text',
30      title: 'Description',
31      validation: (Rule: any) =>
32        | Rule.max(150).warning('Keep the description under 150 characters.'),
33    },
34  ],
35 }
```

