

DAY 3 - API INTEGRATION REPORT-[General E-commerce]

1. Introduction

This report summarizes the API integration process, covering key steps, challenges, and solutions.

2. Integration Process

2.1 Key Steps Identify API requirements, endpoints, and authentication methods.

- Obtain and securely configure API access.
- Implement API requests using proper HTTP methods.
- Handle errors, rate limits, and logging.
- Optimize performance with caching and pagination.
- Secure API usage with HTTPS and authentication mechanisms.
- Test using Postman/curl and deploy with monitoring.

2.2 Migration & Tools

- Assess and map old and new schemas.
- Develop and test migration scripts.
- Use tools like Postman, Swagger, AWS DMS, and database migration tools.

3. Challenges & Solutions

Challenge	Solution
-----------	----------

DAY 3 - API INTEGRATION REPORT-[General E-commerce]

Authentication Issues	Use correct API keys and guidelines.
Rate Limiting	Implement request throttling and caching.
Data Format Mismatch	Validate and adjust schemas as needed.
Downtime & Failures	Use retries, fallbacks, and monitoring.
Migration Complexity	Automate with scripts and thorough testing.

4. Conclusion

Effective API integration requires planning, security, and optimization. Schema adjustments and migration strategies ensure smooth transitions and data integrity.

b) Screenshots of:

- . Api calls.
- . Data successfully displayed in Frontend
- . Populated Sanity CMS fields.

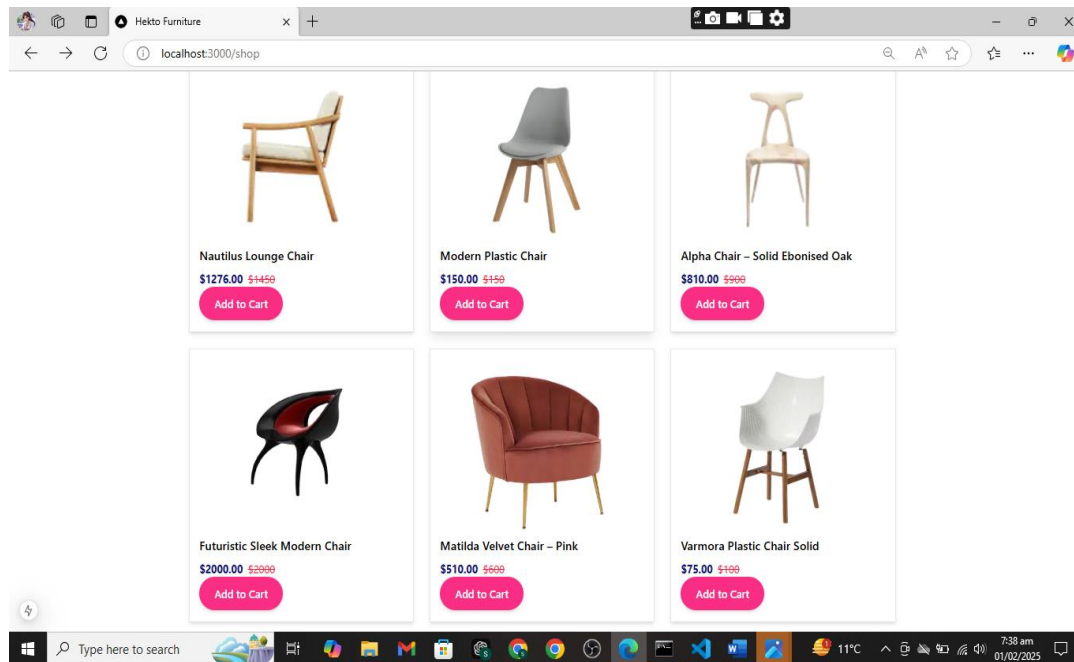
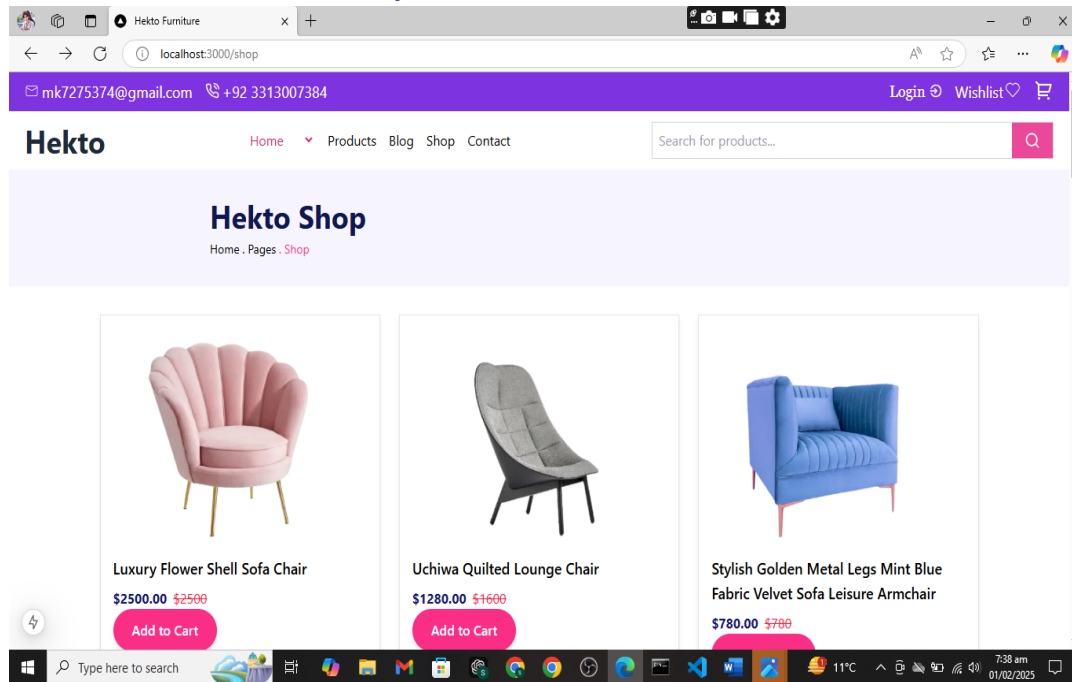
Screenshots of:

Screenshots of:

DAY 3 - API INTEGRATION REPORT-[General E-commerce]

Data successfully Displayed in the frontend.

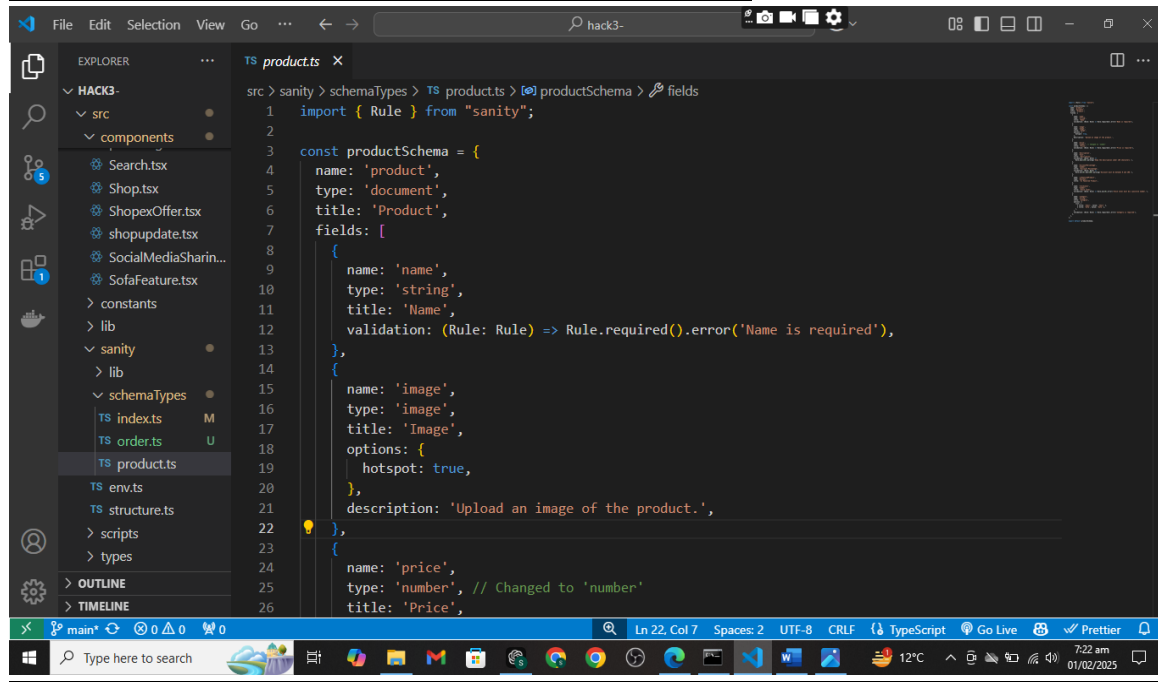
Shop



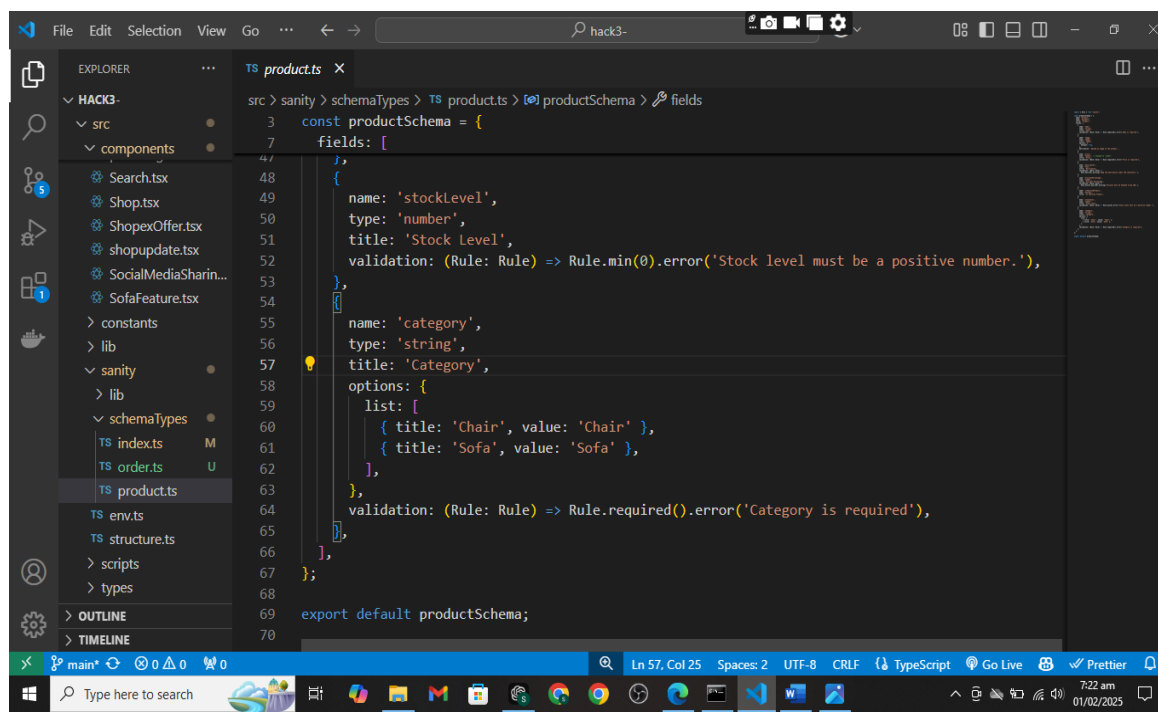
Screenshots of;

DAY 3 - API INTEGRATION REPORT-[General E-commerce]

Populated Sanity CMS fields.



```
src > sanity > schemaTypes > TS product.ts > productSchema > fields
1  import { Rule } from "sanity";
2
3  const productSchema = {
4    name: 'product',
5    type: 'document',
6    title: 'Product',
7    fields: [
8      {
9        name: 'name',
10       type: 'string',
11       title: 'Name',
12       validation: (Rule: Rule) => Rule.required().error('Name is required'),
13     },
14     {
15       name: 'image',
16       type: 'image',
17       title: 'Image',
18       options: {
19         hotspot: true,
20       },
21       description: 'Upload an image of the product.',
22     },
23     {
24       name: 'price',
25       type: 'number', // Changed to 'number'
26       title: 'Price',
```



```
src > sanity > schemaTypes > TS product.ts > productSchema > fields
3  const productSchema = {
7    fields: [
47
48     {
49       name: 'stockLevel',
50       type: 'number',
51       title: 'Stock Level',
52       validation: (Rule: Rule) => Rule.min(0).error('Stock level must be a positive number.'),
53     },
54     {
55       name: 'category',
56       type: 'string',
57       title: 'Category',
58       options: {
59         list: [
60           { title: 'Chair', value: 'Chair' },
61           { title: 'Sofa', value: 'Sofa' },
62         ],
63       },
64       validation: (Rule: Rule) => Rule.required().error('Category is required'),
65     },
66   ],
67 };
68
69 export default productSchema;
```

Data migration.

DAY 3 - API INTEGRATION REPORT-[General E-commerce]

Data Successfully migrated to Sanity Studio

