

Day 3 - API Integration Report - E-commerce Furniture

1. Introduction

This report documents the API integration process, adjustments made to schemas, migration steps, and tools used. It also includes screenshots, code snippets, and best practices followed during the development.

2. API Integration Process

1 API Selection and Setup

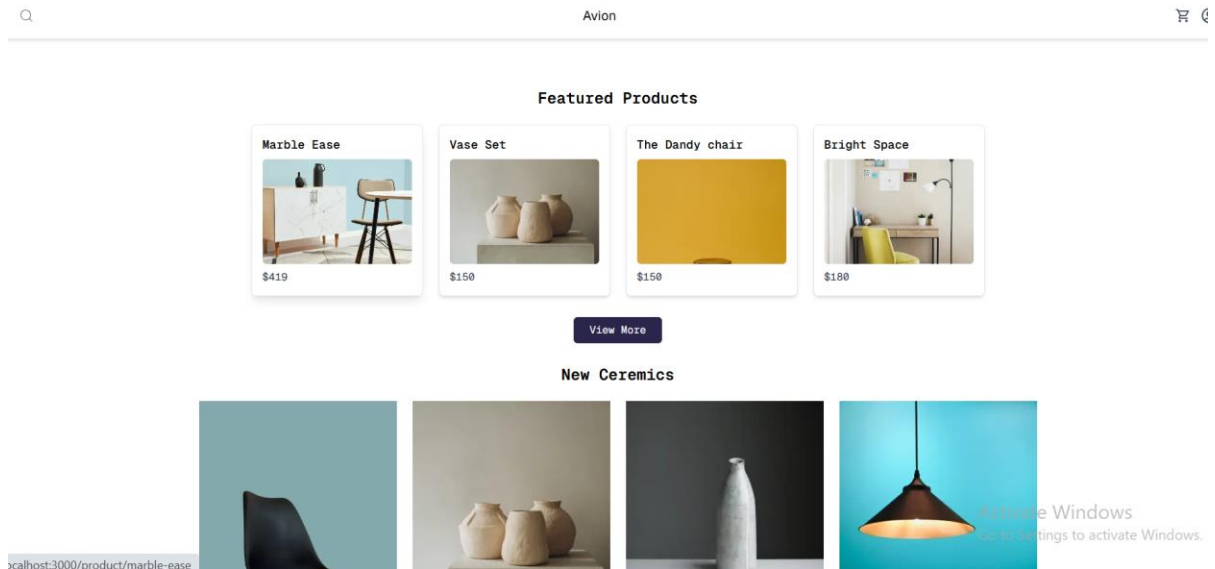
- Selected API: [Sanity CMS](#) for content management and product catalog.
- Setup API credentials and stored them securely in `.env.local`.

2 API Call Implementation in Next.js

- Used `fetch` and `axios` to interact with API endpoints.
- Implemented API calls inside `services/api.js` for modular access.

3 Frontend Integration

- Created a `ProductList` component to display fetched data.
- Used `useEffect` in React to fetch API data on page load.



3. Adjustments Made to Schemas

1 Schema Changes in Sanity CMS

- Added category and stock fields to the product schema.
- Modified price field to support floating values.

4. Data Migration Steps and Tools Used

1 Migration Steps

1. Exported existing product data from the old system.
2. Transformed data to match the new schema.
3. Used `sanity import` to migrate the data.

2 Tools Used

- **Sanity CLI** for schema management.
- **Postman** for API testing.
- **Lodash.js** for data transformation.

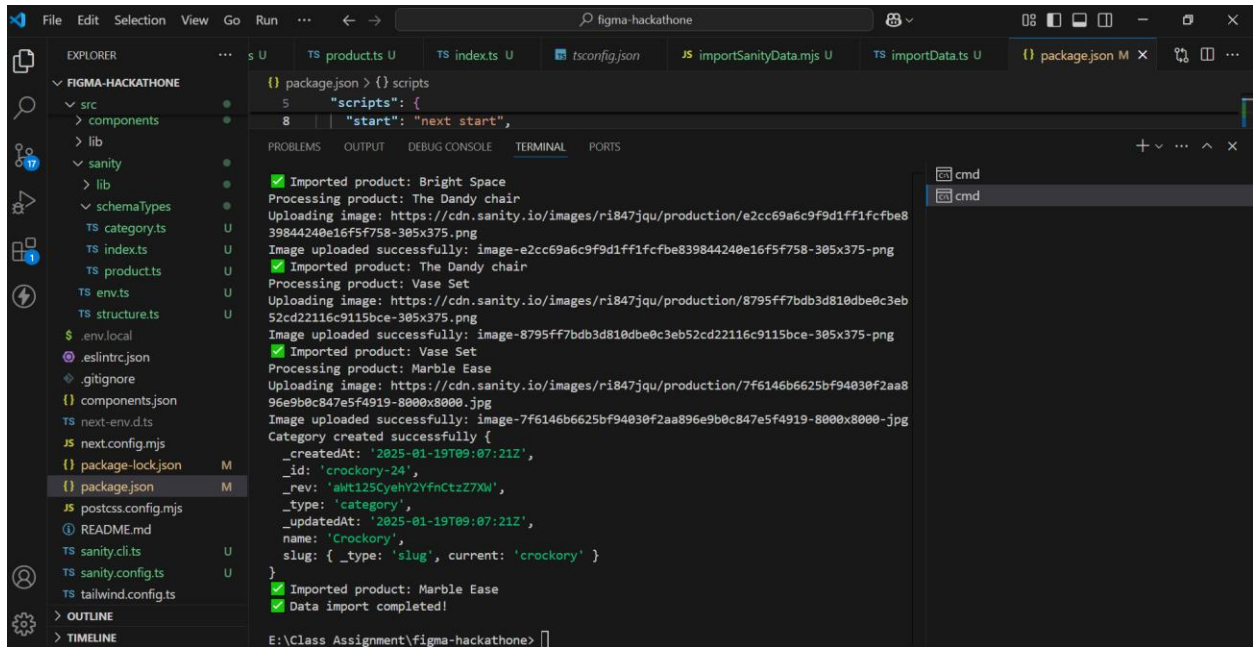
5. Best Practices Followed

1. **Environment Variables**

- a. Used `.env.local` to store API keys securely.
- 2. Clean Coding Practices**
 - a. Used descriptive variable names.
 - b. Modularized functions for reusability.
 - c. Added comments for complex logic.
- 3. Data Validation**
 - a. Implemented schema validation to ensure data integrity.
 - b. Logged any discrepancies for debugging.
- 4. Documentation and Version Control**
 - a. Maintained a changelog for schema updates.
 - b. Used Git for version control, committing frequently.
- 5. Thorough API Testing**
 - a. Used Postman to test API endpoints.
 - b. Handled edge cases like empty responses.
- 6. Peer Review**
 - a. Shared code with team members for feedback.

6. Screenshots and Attachments

1. Screenshot of API request and response in **Postman**.
2. Screenshot of products successfully displayed in the frontend.
3. Screenshot of populated **Sanity CMS** product fields.



7. Checklist for Submission

Task	Status
API Understanding	✓
Schema Validation	✓
Data Migration	✓
API Integration in Next.js	✓
Submission Preparation	✓

8. Conclusion

This report provides a structured overview of the API integration process, schema adjustments, data migration steps, and best practices. By following these steps, we ensured a smooth integration and successful display of product data in the frontend.

BY: AQSA MEMON