3 - API Integration and Data Migration

1. Introduction In Day 3 of this project, the focus was on integrating APIs and migrating data into Sanity CMS. This process allowed for the creation of a functional backend for the marketplace. The main objective was to use external APIs to populate the Sanity CMS, migrate data, and integrate APIs into the Next.js frontend.

Step 1: Understand the Provided API

I thoroughly reviewed the API documentation provided for various templates.

Key API endpoints such as Product Listings, Categories, and Order History were identified and understood. For example:

Template 1 API: Used the documentation to understand product and category endpoints. I focused on understanding the request format, response structure, and parameters needed for successful data fetching. Step 2: Validate and Adjust Schema

I compared the existing Sanity CMS schema from Day 2 with the API data structure.

I identified the differences in field names, types, and relationships between the Sanity schema and the provided API data. Adjustments were made to the schema to ensure compatibility with the API data. For example:

API Field: product_title -> Sanity Field: name Mapped other fields such as descriptions, prices, images, etc., to match API response attributes. Step 3: Data Migration

I used three different methods to migrate the data: Using Provided APIs: I wrote scripts to fetch and transform data from the API into a format compatible with Sanity CMS. Manual Import: For smaller datasets, I exported data as JSON and used Sanity’s import tools to upload it directly. External Data Sources: I explored Shopify and WooCommerce APIs as potential external data sources for importing product and order data. These data sources were integrated using custom migration scripts.

I tested the migration thoroughly to ensure data integrity and consistency in Sanity CMS. Step 4: API Integration in Next.js Task Completed: I created utility functions to fetch data from the API and display it on the Next.js frontend. Key steps involved: Created a function to fetch data from the API (product list, categories).

Data was passed to the frontend and rendered in components, ensuring the frontend displayed the correct product data. I tested the API integration by using Postman and browser developer tools to validate that data was being fetched correctly. I also implemented error handling to display user-friendly messages in case of issues with the API calls or data fetching. Step 5: Screenshots and Code Snippets.

I captured screenshots of:

API responses: Validating successful API calls using tools like Postman. Sanity CMS: Showing the data successfully populated in the CMS (like products and categories). Frontend: Displaying product listings on the frontend as rendered from the API data. I also included code snippets for the API integration and migration s Step 6: Testing and Validation

I thoroughly tested the integration by checking:

Data consistency in Sanity CMS after migration. Display of correct data on the frontend (Next.js app). Validated API calls using Postman and browser dev tools. Step 7: Error Handling

Implemented error handling strategies to manage API errors and data inconsistencies. For instance:

Logging errors: Errors from API calls were logged in a centralized file for debugging. Fallback data: If an API call failed, fallback data was displayed temporarily.

User-friendly messages: Errors were communicated to the user in a clear and non-technical way. 3. Conclusion

I successfully integrated APIs into the Next.js frontend, populated the Sanity CMS with data through various methods (API calls, manual import, and external platforms), and ensured data consistency and compatibility.

I also implemented error handling to manage any failures in API calls, ensuring smooth user experience. 4. Next Steps

The integration and migration process is now complete for Day 3. I will move on to testing edge cases and refining the project further. I will also gather feedback from peers and instructors to improve the integration process. 5. Screenshots I have included relevant screenshots showing: Successful API calls and responses. Data displayed on the frontend. Data populated in the Sanity CMS. 6. Code Snippets Code snippets for API integration and migration scripts are provided above.