**Day 3 - API Integration Report - Cartify**

**API Integration Process**

**1. Overview**

Cartify integrates with multiple APIs, including:

* **Sanity CMS** (for product and category management)
* **Stripe API** (for payments and checkout process)
* **Custom Backend APIs** (for order processing and user authentication)

**2. Steps for API Integration**

1. **Configured API Endpoints**
   * Connected Sanity API using @sanity/client.
   * Integrated Stripe API for handling checkout sessions.
   * Implemented authentication API for user login and order history retrieval.
2. **Environment Setup**
   * Added .env.local file with required API keys.
   * Ensured secure storage of sensitive credentials.
3. **Fetching and Displaying Data**
   * Utilized getServerSideProps and useEffect to fetch data from Sanity.
   * Implemented API calls using fetch and axios.
4. **Handling API Responses**
   * Implemented error handling using try/catch blocks.
   * Used Zod for response validation.

**Adjustments Made to Schemas**

**1. Product Schema**

* **Added stripeProductId field** to link Sanity products with Stripe.
* **Updated price field** to support multiple currency formats.

**2. Order Schema**

* **Added paymentStatus field** to track completed and pending payments.
* **Modified userId field** to store authenticated user reference.

**3. Category Schema**

* **Added parentCategory field** to support nested categories.

**Migration Steps and Tools Used**

**1. Tools Used**

* **Sanity CLI** (sanity dataset export/import)
* **Custom Node.js Scripts** for migrating legacy data.
* **Postman** for API testing.

.

**Screenshots**

1. **API Calls in Postman** (Attached)
2. **Data Displayed on Frontend** (Attached)
3. **Populated Sanity CMS Fields** (Attached)

**Conclusion**

* Successfully integrated **Sanity, Stripe, and authentication APIs**.
* Adjusted schemas for **better data consistency**.
* Verified API functionality with **Postman and frontend implementation**.