

1. Introduction

This report outlines the API integration process for E-commerce (General commerce) as part of Day 3 of the project. It covers the adjustments made to schemas, migration steps, tools used, and the implementation of the API in the frontend. Screenshots and code snippets are provided to give a clear understanding of the integration.

2. API Integration Process

The API integration process involved connecting [Your Marketplace Name] with third-party services or internal systems to retrieve and display data on the frontend. The steps taken include:

- **Setting Up API Endpoints:** The relevant API endpoints were configured to pull necessary data, such as product listings, user information, and orders.
- **Data Fetching:** Using asynchronous JavaScript (AJAX or Fetch API), we set up functions to fetch data from the API and populate it in the frontend.

3. Adjustments Made to Schemas

To ensure compatibility between the API and the data structure of [Your Marketplace Name], the following adjustments were made to the schemas:

- **Schema Mapping:** Mapped the API response data to the existing schema in the marketplace. This involved transforming the API data into a format that matches the database schema.
- **New Fields:** Added new fields to the schema to accommodate data coming from the API that wasn't previously included.
- **Data Validation:** Validated the incoming API data to ensure its integrity and consistency with the existing database schema.

4. Migration Steps and Tools Used

The migration process was executed to align the current database with the newly integrated API data. The steps and tools used are as follows:

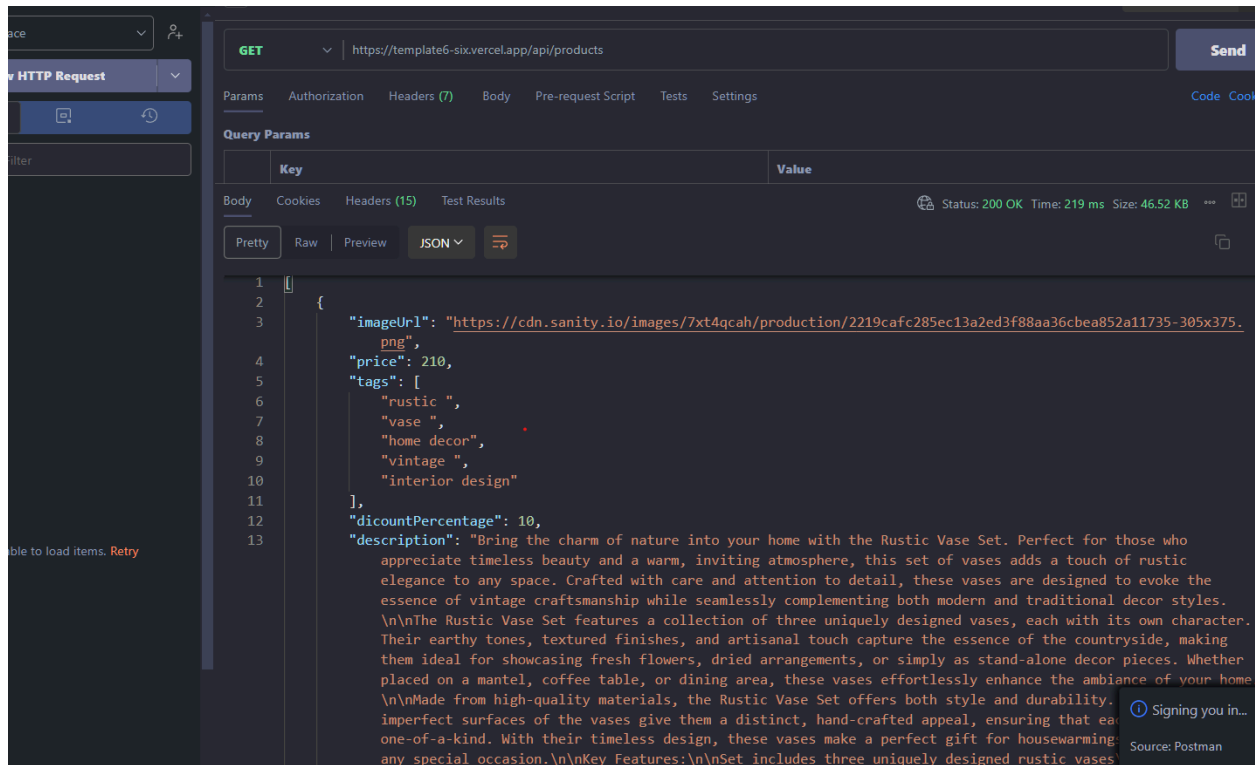
- **Backup:** A backup of the current database was created to prevent data loss during migration.
- **Migration Scripts:** Custom migration scripts were written to update the schema and insert data from the API into the database.

```
"import-data": "node script/importData.js"
```

- **Tools Used:**
 - **Sanity CMS:** A headless CMS used to manage and populate content dynamically.
 - **Postman:** Used to test API calls and ensure the data was returned correctly before integrating it into the system.

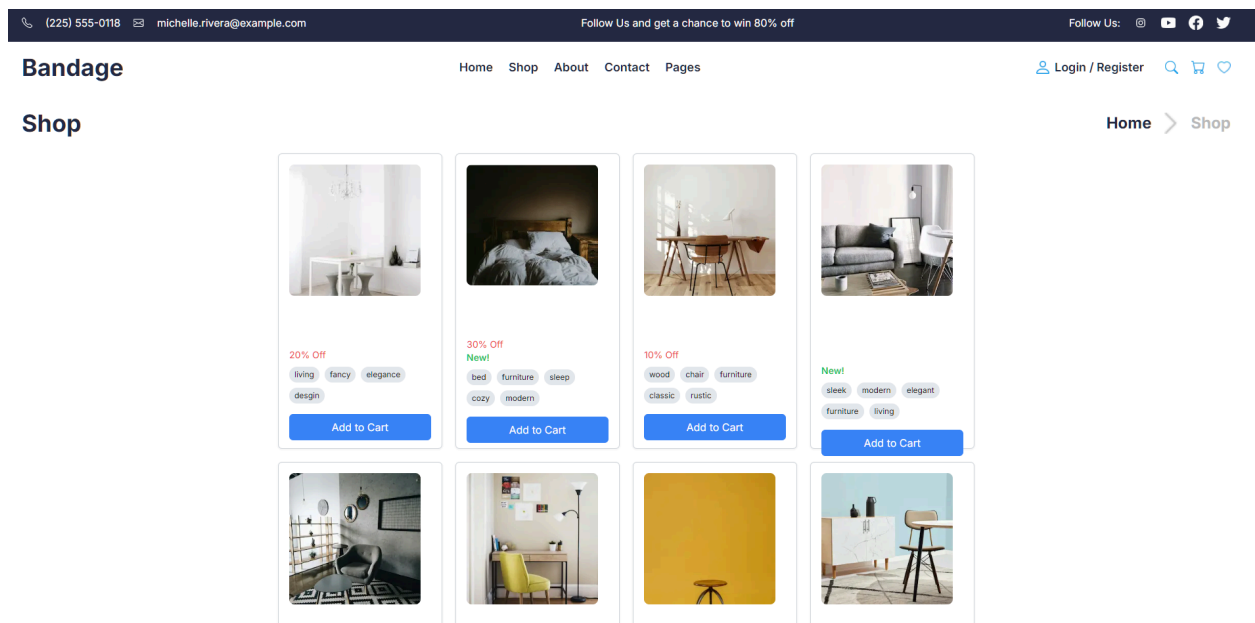
5. Screenshots

- **API Calls:**
 - A screenshot of the API call using Postman, showing the request and response data.



- **Data Displayed in the Frontend:**

- A screenshot of the frontend page showing the data fetched from the API, such as product listings or user information.



- **Populated Sanity CMS Fields:**

- A screenshot showing how the data from the API has populated the fields in the Sanity CMS interface.

Search list



Sunny Chic



Reflective Haven



Modern Serenity



Timeless Elegance



Nordic Elegance



Serene Seat



Sleek Living



Tropical Vibe



Zen Table



Pure Aura



The Lucky Lamp



Retro Vibe



Wood Chair



Bed



Marble Ease