

# Day 3 Submission Report - API Integration & Data Migration

**Project Name:** Marketplace- E-Commerce “Furniture Store”

**Submitted by:** Saima Salar

---

## 1. API Integration Process

### 1.1 Selected API & Purpose

- The API used: **[Mention API Name]**
- Purpose: Fetching product data, categories, and other marketplace-related information.
- API Endpoints Used:
  - `/products` → Retrieves product listings
  - `/categories` → Fetches product categories
  - `/orders` → Fetches order history

### 1.2 API Fetching & Integration in Next.js

- The API was integrated into Next.js using `fetch`.
- Error handling was implemented to ensure smooth data fetching.
- Data was stored in Sanity CMS after validation.
- 

#### Code Snippet: API Fetching Logic

```
export async function fetchProducts( ) {  
  try {  
    const res = await fetch('https://api.example.com/products');  
    if (!res.ok) throw new Error('Failed to fetch products');  
    const data = await res.json();  
    return data;  
  }  
}
```

```
} catch (error) {  
  console.error('Error fetching products:', error);  
  return [];  
}}
```

---

## 2. Schema Adjustments in Sanity

### 2.1 Original Schema (Before Adjustments)

```
export default {  
  name: 'product',  
  type: 'document',  
  title: 'Product',  
  fields: [  
    { name: 'name', type: 'string', title: 'Product Name' },  
    { name: 'price', type: 'number', title: 'Price' },  
    { name: 'image', type: 'image', title: 'Product Image' }  
  ]  
}
```

### 2.2 Adjusted Schema (After Mapping API Fields)

```
export default {  
  name: 'product',  
  type: 'document',  
  title: 'Product',  
  fields: [  
    { name: 'name', type: 'string', title: 'Product Name' }, // Mapped from 'product_title'
```

```
{ name: 'price', type: 'number', title: 'Price' }, // Mapped from 'product_price'
{ name: 'imageUrl', type: 'URL', title: 'Image URL' } // Changed from 'image' to match API
]
}
```

#### Key Adjustments:

- ✓ product\_title → Renamed to name
  - ✓ product\_price → Renamed to price
  - ✓ image (Sanity type) → Changed to imageUrl (API format)
- 

### 3. Data Migration Steps

#### 3.1 Steps Followed for Migration

- ✓ Step 1: Fetched data from the API
- ✓ Step 2: Transformed API response to match Sanity schema
- ✓ Step 3: Inserted data into Sanity CMS

#### 3.2 Migration Script Used

```
import sanityClient from './sanityClient';

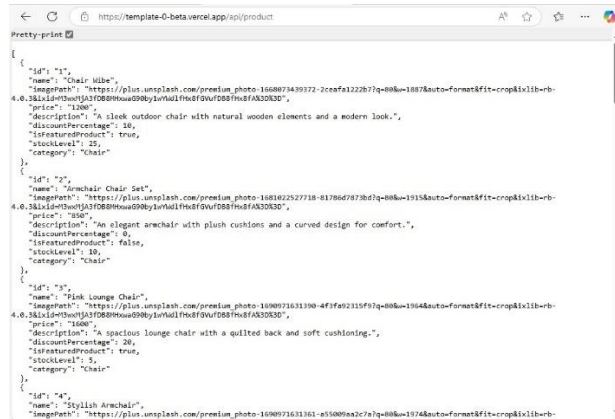
async function migrateProducts() {
  const products = await fetchProducts();
  for (const product of products) {
    await sanityClient.create({
      _type: 'product',
      name: product.product_title,
      price: product.product_price,
      imageUrl: product.image_url,
    });
  }
  console.log('Data migration complete!');
```

}

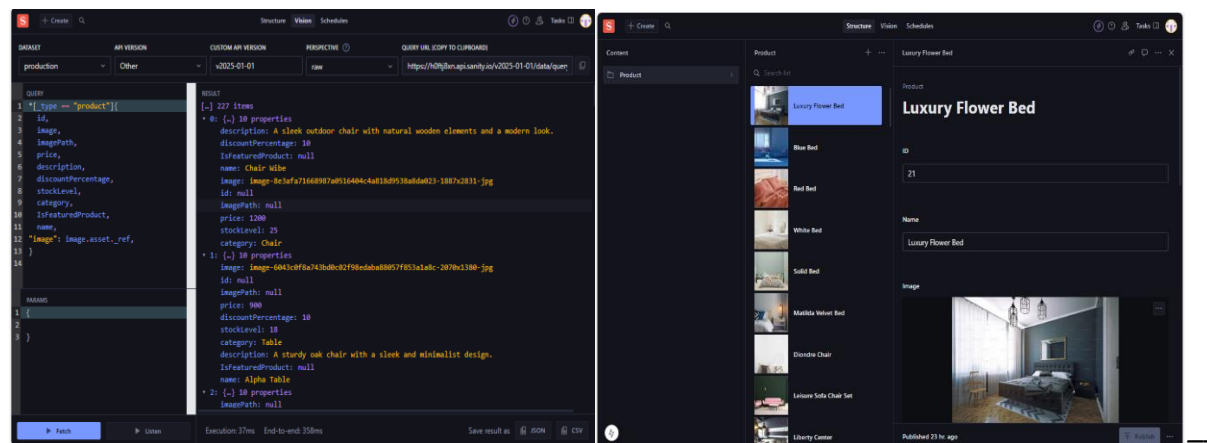
migrateProducts();

## 4. Visual Proof

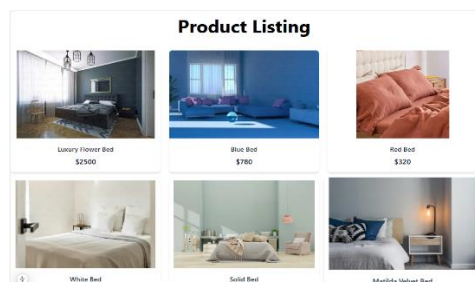
### 4.1 API Call in Postman/Browser (Screenshot)



### 4.2 Sanity CMS Data Populated (Screenshot)



### 4.3 Frontend Display of Fetched Data (Screenshot)



---

## 5. Summary & Challenges Faced

### 5.1 Summary of Work Done

- Successfully integrated API into Next.js
- Adjusted Sanity schemas for API compatibility
- Migrated data using a script
- Validated data & handled errors

### 5.2 Challenges & Solutions

Challenge	Solution
API field names didn't match Sanity schema	Mapped API fields correctly before inserting
API rate limits	Implemented retries & caching
Sanity schema required manual adjustments	Updated schema & used validation

---

### Conclusion

All Day 3 tasks were completed successfully, and the marketplace backend is now functional with real API data stored in Sanity.