

## **API INTEGRATION PROCESS AND SCHEMA ADJUSTMENT REPORT**

### **1. API Integration Process**

#### **• Setting Up API Access**

- ✓ Configured API keys and access credentials for secure communication between the Next.js application and Sanity.
- ✓ Added CORS policy in Sanity to allow requests from Vercel and localhost.

#### **• Fetching and Displaying Data**

- ✓ Integrated Sanity's client library (@sanity/client) in the Next.js project.
- ✓ Implemented data fetching in shop and product detail pages using Sanity's GROQ queries.
- ✓ Used getServerSideProps and useEffect to fetch real-time data dynamically.

#### **• Handling Authentication and Security**

- ✓ Enabled credentials in API requests to allow secure data fetching.
- ✓ Adjusted API permissions in Sanity to allow read and write operations as required.

### **2. Schema Adjustments**

#### **• Product Schema Updates**

- ✓ Added fields for description, price, images, category, and stock availability.
- ✓ Included references to relate products with categories dynamically.

#### **• Product Detail Schema**

- ✓ Created a separate productDetail schema to store additional information like dimensions, materials, and customer reviews.
- ✓ Linked productDetail schema with product schema using references.

#### **• Product and Order Schema Enhancements**

- ✓ Introduced Product schema to manage user-selected products.
- ✓ Modified order schema to include payment status and delivery tracking.

### **3. Migration Steps and Tools Used**

#### **• Data Export and Backup**

- ✓ Used Sanity's CLI (Sanity Dataset export) to export existing data before making modifications.
- ✓ Created a backup of schema definitions and dataset configurations.

#### **• Schema Migration**

- ✓ Updated schema definitions in schemas folder of the Sanity project.
- ✓ Used sanity deploy to push new schema changes.

#### **• Data Migration**

- ✓ Utilized Sanity's GROQ queries to transform and migrate data to new schemas.
- ✓ Manually adjusted entries where necessary to ensure data integrity.

#### **• Testing and Validation**

- ✓ Verified API responses using Postman and Sanity Studio.
- ✓ Checked front-end integration with updated data structures.

#### **• Tools Used**

- ✓ **Sanity CLI** (Sanity Dataset, Import/Export, Sanity Deploy)
- ✓ **Postman** for API testing.
- ✓ **Next.js & React DevTools** for debugging.
- ✓ **GitHub** for version control and rollback options.

These integrations and adjustments ensure seamless data management and a smooth user experience in the furniture website.

