

System Architecture for General E-Commerce Platform

Technical Overview

The platform is designed to provide a seamless experience for purchasing, customizing, and renovating furniture and décor items. Leveraging **Next.js 14.2.15** for the frontend and **Sanity CMS** as the headless CMS, the architecture ensures scalability, performance, and a user-friendly interface.

Key Components:

1. Frontend (Next.js 14.2.15)

- **Framework:** Next.js for server-side rendering (SSR) and static site generation (SSG).
- **Features:**
 - Dynamic product pages.
 - Customization and makeover request forms.
 - User authentication and profile management.
 - Real-time updates for order status.
- **UI Framework:** Tailwind CSS for responsive and modern UI design.

2. Backend

- **Headless CMS:** Sanity CMS to manage product catalogs, orders, category, tags , services, shipment data, delivery zones and customers.
- **API Layer:**
 - Groq as a query language to fetch data from Sanity CMS.
 - REST APIs for products, customer orders, shipment tracking, and customization/makeover requests.
- **Authentication:** Next Auth authentication for secure customer login.

4. Deployment

- **Hosting:** Vercel for Next.js deployment.

5. Third-Party Integrations:

- **API containing product data:** Products and related information are stored in Sanity CMS.
 - **APIs for order and customer data:** to be stored in Sanity CMS at order submission.
 - **Payment Gateway:** Stripe for secure online payments.
 - **Shipment & Tracking:** Integrated with ShipEngine or Shippo APIs for real-time order tracking, delivery updates, and notifications.
-

Entity-Relationship Diagram (ERD)

[Product]

- ID
- Name
- Price
- Stock
- Category
- Tags
- Description

|
|

[Order/Customization/Makeover] -----> [Customer]

- | | | |
|-------------------|-------|------------------|
| - Order ID | ===== | - Customer ID |
| - Product Details | | - Name |
| - Service Details | | - Contact Number |
| - Total Price | | - Address |
| - order date | | - Customer Email |
| - Status | | |

|
|
|
|
|

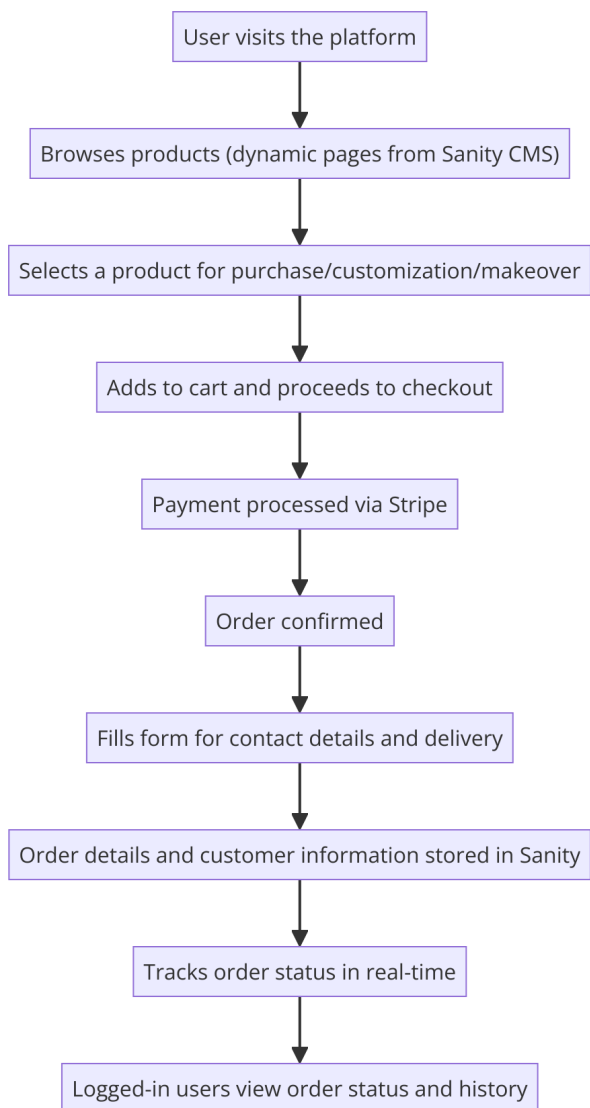
[Shipment] <-----> [Delivery Zone]

- | | |
|------------------|-----------------|
| - Shipment ID | - Zone Name |
| - Status | - Coverage Area |
| - Customer Email | |

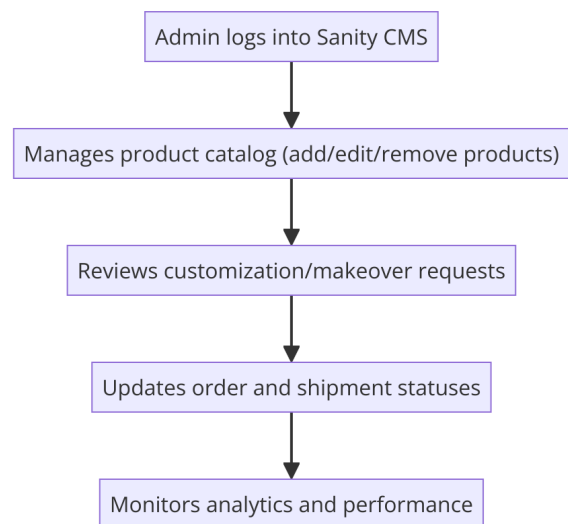
- Customer ID

Workflow Diagram

User Interaction Workflow



Admin Interaction Workflow



Smooth Workflow Implementation

1. Frontend Integration

- Utilize Next.js dynamic routing to create product-specific pages.
- Implement client-side and server-side rendering for optimal performance.
- Integrate Sanity CMS as the headless backend to fetch product data.

2. Backend Management

- Use Sanity CMS for content ,orders, cutomer data and product management.
- Develop REST APIs to handle order processing, customer data, and shipments.

3. Customization and Makeover Requests

- Allow customers to upload images and specify details during request submission.
- Automatically calculate estimated costs based on predefined rules.
- Update request statuses in real-time via APIs.

4. Delivery Zones and Shipments

- Track shipments via real-time updates integrated into the user profile.
-
-

Conclusion

This architecture ensures a scalable, secure, and user-friendly e-commerce platform with robust features for both product sales and customization/makeover services. The use of modern tools like Next.js and Sanity CMS ensures maintainability and extensibility as the business grows.