

Furniture E-Commerce Marketplace

Technical Documentation

Introduction:

This document outlines the technical plan for a furniture e-commerce marketplace. The goal of this project is to create a seamless platform where users can browse furniture, place orders, and track shipments in real-time. Built using modern tools like Next.js, Sanity CMS, and third-party APIs, the marketplace emphasizes scalability, user-friendliness, and secure transactions.

Tech Stack:

1. Frontend:

- Next.js: Enables responsive and fast user interfaces.
- Tailwind CSS: Simplifies styling with utility-first CSS.
- Shadcn UI: Provides customizable and reusable components for a consistent design.

2. Backend:

- Sanity CMS: Handles product data, categories, orders, and customer details efficiently with structured content schemas.

3. Third-Party APIs:

- Payment Gateways:
 - 2Checkout
 - PayFast
 - QuickPay
- Shipment Tracking:
 - ShipEngine API for real-time tracking and updates.

4. Development Tools:

- GitHub: Version control for code collaboration.
- Postman: Testing and validating APIs.
- Lucidchart/Excalidraw: For creating architecture and workflow diagrams.

Essential Pages:

1. Home Page:

- Highlights featured products, deals, and popular categories.

2. Product Listing Page:

- Displays products filtered by categories or all available products.

3. Product Details Page:

- Provides detailed product descriptions, pricing, stock availability, and images.

4. Cart Page:

- Allows users to review their selected items and modify quantities.

5. Checkout Page:

- Collects user details, shipping address, and payment information.

6. Order Confirmation Page:

- Displays a summary of the completed order along with a success message.

7. User Login/Registration Page:

- Allows new users to sign up and existing users to log in.

8. Order History Page:

- Displays the user's past orders and shipment statuses.
-

System Architecture (Workflow Overview):

1. User Registration:

- User data is sent to Sanity CMS, and a confirmation is sent back to the user.

2. Viewing Categories and Products:

- Categories are fetched from Sanity CMS Category API, and products are retrieved based on the selected category or displayed as a complete list.

3. Order Placement:

- Items are added to the cart, and user details with order data are saved in Sanity CMS.

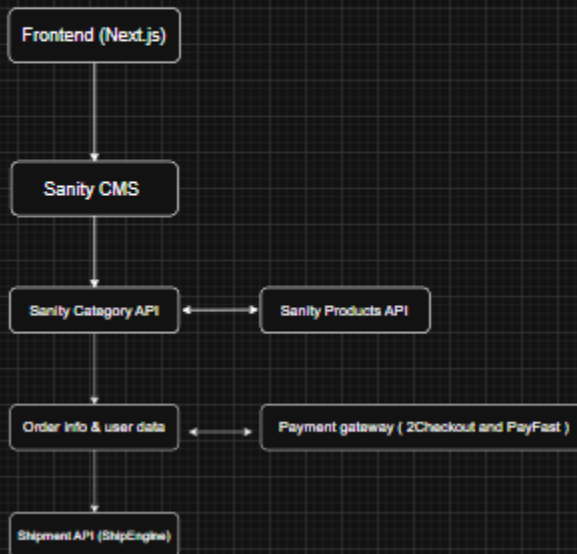
4. Shipment Tracking:

- Order status is fetched via ShipEngine API and displayed to the user.

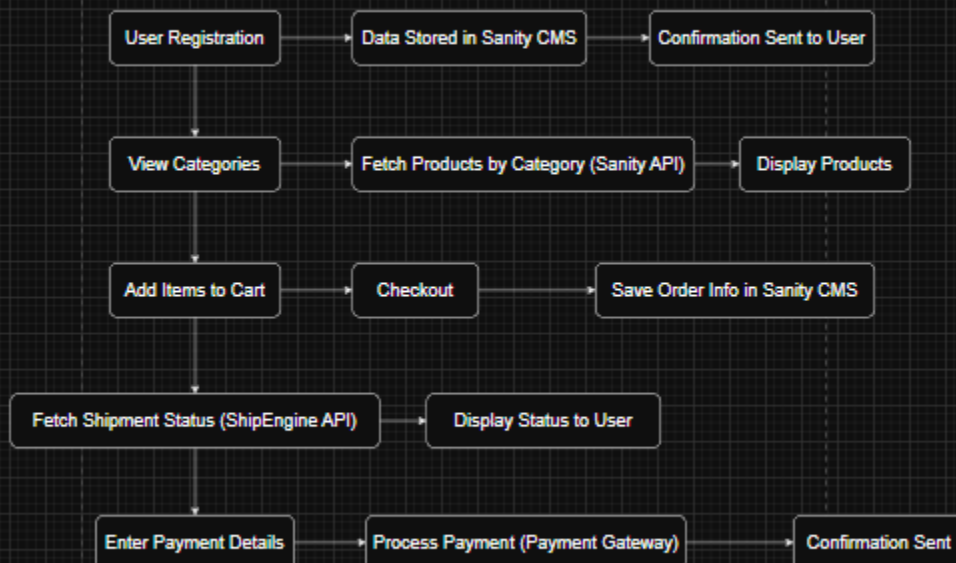
5. Payment Processing:

- Payments are securely processed via 2Checkout, PayFast, or QuickPay, and a confirmation is sent back to the user while being recorded in Sanity CMS.

System Architecture



Workflow



Key Features of the Marketplace:

1. Category-Based Browsing:
 - Users can filter products by categories (e.g., Living Room, Bedroom).
 2. Real-Time Shipment Tracking:
 - Users can check their order status, fetched through ShipEngine API.
 3. Secure Payment Gateways:
 - Multiple payment gateways ensure secure transactions (2Checkout, PayFast, QuickPay).
 4. Customizable Furniture Options:
 - Certain products allow users to choose colors, materials, and sizes.
 5. Mobile-Optimized Design:
 - Responsive layouts ensure a seamless experience on both desktop and mobile devices.
 6. Order History:
 - Users can view and track their past orders along with statuses.
-

API Specifications:

1. User Registration:

- **Endpoint:** `/register`
- **Method:** `POST`
- **Description:** Saves user data in Sanity CMS and sends a confirmation response.

Payload Example:

```
{  
  "name": "Ali Khan",  
  "email": "ali.khan@example.com",  
  "password": "securepassword"  
}
```

Response Example:

```
{  
  "message": "Registration successful",  
  "userId": "12345"  
}
```

2. Fetch Categories:

- **Endpoint:** `/categories`
- **Method:** `GET`
- **Description:** Retrieves all product categories stored in Sanity CMS.

Response Example:

```
[  
  { "id": "1", "name": "Living Room" },  
  { "id": "2", "name": "Bedroom" }  
]
```

3. Fetch Products by Category:

- **Endpoint:** `/products?category={categoryId}`
- **Method:** `GET`
- **Description:** Fetches all products under a specific category. If no category is specified, all products are returned.

Response Example:

```
[  
  { "id": "101", "name": "Sofa", "price": 500, "stock": 10 },  
  { "id": "102", "name": "Dining Table", "price": 800, "stock": 5 }  
]
```

4. Place Order:

- **Endpoint:** `/orders`
- **Method:** `POST`
- **Description:** Saves the user's order details and items in Sanity CMS.

Payload Example:

```
{  
  "userId": "12345",  
  "orderItems": [  
    { "productId": "101", "quantity": 1 },  
  ]  
}
```

```
{ "productId": "102", "quantity": 2 }  
  
],  
  
"totalAmount": 2100  
  
}
```

Response Example:

```
{  
  
  "message": "Order placed successfully",  
  
  "orderId": "78901"  
  
}
```

5. Track Shipment:

- **Endpoint:** `/shipment/{orderId}`
- **Method:** `GET`
- **Description:** Retrieves shipment tracking details via the ShipEngine API.

Response Example:

```
{  
  
  "shipmentId": "SE123",  
  
  "status": "In Transit",  
  
  "estimatedDelivery": "2025-01-18"  
  
}
```


6. Process Payment:

- **Endpoint:** `/payment`
- **Method:** `POST`
- **Description:** Processes payment securely via a payment gateway.

Payload Example:

```
{  
  "orderId": "78901",  
  "amount": 2100,  
  "paymentMethod": "PayFast"  
}
```

Response Example:

```
{  
  "message": "Payment successful",  
  "transactionId": "TX456789"  
}
```

Sanity CMS Schemas

1. Products Schema:

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock' },
    { name: 'category', type: 'reference', to: [{ type: 'category' }] },
  ],
  { name: 'description', type: 'text', title: 'Description' },
  { name: 'images', type: 'array', of: [{ type: 'image' }], title:
'Product Images' }
  ]
};
```

2. Orders Schema:

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'userId', type: 'reference', to: [{ type: 'user' }],
title: 'User' },
    { name: 'orderItems', type: 'array', of: [{ type: 'orderItem' }],
title: 'Order Items' },
    { name: 'totalAmount', type: 'number', title: 'Total Amount' },
    { name: 'status', type: 'string', title: 'Order Status', options:
{ list: ['Pending', 'Completed', 'Shipped'] } }
  ]
};
```

3. Users Schema:

```
export default {
  name: 'user',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Full Name' },
    { name: 'email', type: 'string', title: 'Email Address' },
    { name: 'password', type: 'string', title: 'Password' }
  ]
};
```