

PRESENT BY : Hooriya M.Fareed

MENTOR : SIR HAMZAH SYED

Technical Documentation for Furniture

E-Commerce Website:

- *This document outlines the system architecture, workflows, API requirements, and technical details for your furniture e-commerce website. The goal is to create a seamless user experience, from browsing products to tracking shipments, while ensuring secure payments and efficient content management.*

1. System Architecture:

- *This document outlines the system architecture, workflows, API requirements, and technical details for your furniture e-commerce website. The goal is to create a seamless user experience, from browsing products to tracking shipments, while ensuring secure payments and efficient content management.*

[Frontend (Next.js)]



[Sanity CMS] [Stripe] [ShipEngine] [Clerk]

| |
| |
[Product Data API] [Shipment Tracking API]

Key Components:

- *Frontend (Next.js): The user interface where customers browse products, add items to the cart, and complete purchases.*
- *Sanity CMS: Manages product data, customer orders, and inventory.*
- *Stripe: Handles secure payment processing.*
- *ShipEngine: Provides shipment tracking and delivery updates.*
- *Clerk: Manages user authentication (sign-up, login, and profile management).*

2. Workflows:

1. User Registration:

- *Step 1: User signs up using Clerk.*
- *Step 2: Clerk stores user data securely and sends a confirmation email.*
- *Step 3: User logs in to access their account.*

2. Product Browsing:

- *Step 1: User visits the website and browses product categories.*
- *Step 2: The frontend fetches product data from Sanity CMS via the Product Data API.*
- *Step 3: Products are displayed dynamically on the website.*

3. Order Placement:

- *Step 1: User adds items to the cart.*
- *Step 2: User proceeds to checkout and enters payment details.*
- *Step 3: Stripe processes the payment securely.*
- *Step 4: Order details (customer info, product details, payment status) are saved in Sanity CMS.*

4. Shipment Tracking:

- *Step 1: After the order is placed, ShipEngine generates a shipment ID.*
- *Step 2: The frontend fetches shipment tracking details via the Shipment Tracking API.*
- *Step 3: Real-time shipment status (e.g., "Shipped," "Out for Delivery") is displayed to the user.*

3. API REQUIREMENTS:

1. Product Data API:

Endpoint Name: /products

Method: GET

Description: Fetches all available products from Sanity CMS.

Response Example:

Json

```
{  
  "id": 1,  
  "name": "Modern Sofa",  
  "price": 500,  
  "stock": 10,  
  "image": "https://example.com/sofa.jpg"
```

```
}
```

2. Order API:

Endpoint Name: /orders

Method: POST

Description: Creates a new order in Sanity CMS.

Payload Example:

```
{  
  "customer_id": 123,  
  "products": [  
    { "id": 1, "quantity": 2 }  
  ],  
  "total_amount": 1000,  
  "payment_status": "paid"  
}
```

3. Shipment Tracking API:

Endpoint Name: /shipment

Method: GET

Description: Fetches shipment tracking details from ShipEngine.

Response Example:

```
{  
  "shipment_id": "SH12345",  
  "order_id": 456,  
  "status": "In Transit",  
  "expected_delivery": "2023-10-30"  
}
```

4. Sanity Schema Example:

Sanity CMS uses a schema to define the structure of your data.
Below is an example schema for your furniture e-commerce website:

1. Product Schema:

// schemas/product.ts

```
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Product Name',
    },
    {
      name: 'price',
      type: 'number',
      title: 'Price',
    },
    {
      name: 'stock',
      type: 'number',
      title: 'Stock Quantity',
    },
    {
      name: 'image',
      type: 'image',
      title: 'Product Image',
      options: {
        hotspot: true, // Enables image cropping
      },
    },
    {
      name: 'description',
      type: 'text',
      title: 'Product Description',
    },
  ],
}
```

```
    name: 'category',
    type: 'reference',
    title: 'Category',
    to: [{ type: 'category' }], // Links to the Category schema
  },
],
};
```

2. Category Schema:

```
// schemas/category.ts
export default {
  name: 'category',
  type: 'document',
  title: 'Category',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Category Name',
    },
    {
      name: 'slug',
      type: 'slug',
      title: 'Slug',
      options: {
        source: 'name', // Automatically generates a slug from the category name
      },
    },
  ],
};
```

3. Order Schema:

```
// schemas/order.ts
export default {
  name: 'order',
```

```
type: 'document',
title: 'Order',
fields: [
  {
    name: 'customer_id',
    type: 'string',
    title: 'Customer ID',
  },
  {
    name: 'products',
    type: 'array',
    title: 'Products',
    of: [
      {
        type: 'object',
        fields: [
          {
            name: 'product_id',
            type: 'reference',
            title: 'Product',
            to: [{ type: 'product' }], // Links to the Product schema
          },
          {
            name: 'quantity',
            type: 'number',
            title: 'Quantity',
          },
        ],
      },
    ],
  },
  {
    name: 'total_amount',
    type: 'number',
    title: 'Total Amount',
  },
  {
    name: 'payment_status',
    type: 'string',
    title: 'Payment Status',
  },
]
```

```
options: {  
  list: ['pending', 'paid', 'failed'], // Predefined statuses  
},  
{  
  name: 'shipment_id',  
  type: 'string',  
  title: 'Shipment ID',  
},  
],  
};
```

5. Pages and Functionality:

1. Home Page:

Purpose: Display featured products, categories, and promotions.

Features:

- Hero banner with promotions.
- Grid or carousel of featured products.
- Links to product categories.

2. Product Listing Page:

Purpose: Show products filtered by categories.

Features:

- Filters for price, category, and ratings.
- Pagination for large product lists.
- Dynamic product cards with images, names, and prices.

3. Product Details Page

Purpose: Provide detailed information about selected products.

Features:

- *High-quality product images.*
- *Product description, price, and availability.*
- *"Add to Cart" button.*

4. Cart Page:

Purpose: *Allow users to review and modify selected items.*

Features:

- *List of items in the cart.*
- *Quantity adjustment and item removal options.*
- *Subtotal and "Proceed to Checkout" button.*

Workflow:

- *User adds items to the cart and clicks "Proceed to Checkout."*
- *If the user is not logged in, they are redirected to the Clerk Sign-Up/Login Page.*
- *After successful authentication, the user is redirected back to the Checkout Page.*
- *"Proceed to Payment" button to redirect to Stripe's payment page.*
- *After clicking on "Proceed to Payment." The frontend sends the order details (products, total amount, customer info) to Sanity CMS to save the order.*
- *After successful payment, it moves to the success page*

3. Stripe Payment Page

6. Shipment Tracking Page

Purpose: *Allow users to track their order status after shipment initiation.*

Features:

- *Real-time shipment status fetched from ShipEngine.*
- *Expected delivery date and tracking history.*
- *Link to return to the homepage or view order history.*

6. Tools and Technologies Used:

Frontend: Next.js

CMS: Sanity

Authentication: Clerk

Payment Gateway: Stripe

Shipment Tracking: ShipEngine

7. Conclusion:

This e-commerce marketplace is your ultimate destination for a truly exceptional online shopping experience. Combining fast delivery, a sleek and user-friendly interface, and effortless functionality, we provide a seamless and enjoyable journey. Discover a curated selection of elegant and classy furniture, perfect for transforming homes, offices, and beyond. Elevate your shopping experience with unmatched convenience and sophistication, all in one place.