eCommerce Website System Architecture

Overview

This document outlines the technical foundation and workflows for your eCommerce website, focusing on the sale of furniture. It includes system architecture, workflows, API endpoints, and schema definitions.

System Architecture

The architecture is designed to handle frontend interactions, backend content management, and third-party service integrations efficiently. Below is the breakdown of components:

Components:

1. **Frontend (Next.js)**
   - User interface for browsing and purchasing furniture.
   - Responsive design for desktop and mobile devices.
2. **Backend (Sanity CMS)**
   - Manages product data, customer details, and order records.
   - Custom schemas tailored to business needs.
3. **Third-Party Services**
   - Payment processing for secure transactions.
   - Shipping APIs for real-time order tracking.
   - Analytics for monitoring sales and user behavior.

Key Workflows

1. **User Registration and Authentication**
   - Users register or log in.
   - Data stored in Sanity CMS.
2. **Product Browsing**
   - Users view product categories and details.
   - Data fetched from Sanity CMS via API.
3. **Order Placement**
   - Users add items to the cart and proceed to checkout.
   - Order details are saved in Sanity CMS.
4. **Payment Processing**
   - Transactions handled through third-party payment gateways.
5. **Shipment Tracking**
   - Shipping updates fetched via third-party APIs and displayed to users.

API Endpoints

A table of possible API endpoints and their purposes:

Endpoint	Method	Description
`/api/products`	GET	Fetch list of products.
`/api/products/:id`	GET	Fetch product details by ID.
`/api/cart`	POST	Add items to the cart.
`/api/orders`	POST	Create a new order.
`/api/orders/:id`	GET	Retrieve order details by ID.
`/api/shipping/:order`	GET	Fetch shipping updates for an order.

Sanity CMS Schema Examples

Product Schema

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'description', type: 'text', title: 'Description' },
    { name: 'stock', type: 'number', title: 'Stock Level' },
    { name: 'category', type: 'string', title: 'Category' },
    { name: 'image', type: 'image', title: 'Product Image' }
  ]
};
```

Order Schema

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'orderId', type: 'string', title: 'Order ID' },
    { name: 'customer', type: 'reference', to: [{ type: 'customer' }], title:
'Customer' },
    { name: 'items', type: 'array', of: [{ type: 'reference', to: [{ type:
'product' }] }], title: 'Items' },
    { name: 'totalAmount', type: 'number', title: 'Total Amount' },
    { name: 'status', type: 'string', title: 'Order Status' }
  ]
};
```

Technical Roadmap

1. **Day 1: Initial Setup**
    - Set up Next.js project.
    - Install and configure Sanity CMS.
    - Define schemas for products and orders.
2. **Day 2: API Development**
    - Develop core API endpoints for products, cart, and orders.
    - Test endpoints using Postman.
3. **Day 3: Frontend Integration**
    - Build and design product listing and detail pages.
    - Integrate APIs with frontend pages.
    - Implement cart functionality.
4. **Day 4: Payment and Shipping Integration**
    - Integrate payment processing with third-party APIs.
    - Add shipment tracking functionality.
5. **Day 5: Testing and Deployment**
    - Perform thorough end-to-end testing.
    - Deploy the application to production.

Conclusion

This document provides a comprehensive guide for building and deploying your furniture eCommerce platform. By following the outlined system architecture, workflows, and roadmap, you can create a robust and scalable application tailored to your business needs.

Thank you for taking the time to explore this blueprint. Let’s bring your vision to life and deliver an exceptional shopping experience to your users!