Hackathon 3 DAY 2

Documentation for General E-Commerce Website

**Presented by Muhammad Rohan Mirza**

● This document outlines the system architecture, workflows, API requirements, and technical details for your General 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 Overview**

● 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]**

**Components:**

• **Frontend (Next.js + Tailwind CSS):** Manages the user interface, browsing, and interaction.

• **Sanity CMS (Backend):** Manages product, order, customer, and supplier data.

• **Third-Party APIs:**

* Stripe API: Manages secure payment processing.
* Shippo API: Provides shipment tracking and logistics information.

**System Interaction Flow:**

1. User Interaction: The user accesses the marketplace via the frontend.

2. Product Data Fetching: The frontend fetches product data from Sanity CMS.

3. Order Management: Order details are stored in Sanity CMS.

4. Payment Processing: Payments are securely processed via Stripe API.

5. Shipment Tracking: Real-time shipping details are fetched using Shippo API.

**2. Workflow:**

**1. User Registration:**

● Step 1: User signs up using Clerk.

● Step 2: The clerk stores user data securely and sends a confirmation email.

● Step 3: The 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. Shipmen 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 Specification Document**

**Product Endpoints:**

• Endpoint: /api/products

• Method: GET

• Description: Fetch all products.

• Response Example:

{

"id": "p123",

"name": "Smart Scale X",

"price": 120

}

**Order Endpoints:**

• Endpoint: /api/orders

• Method: POST

• Description: Create a new order.

• Payload:

{

"customerId": "c001",

"orderItems": [{ "productId": "p123", "quantity": 2 }],

"totalAmount": 240

}

**Summary of Key API Requirements:**

1. Product API: Fetch all products and individual product details.

2. Customer API: Create and fetch customer records.

3. Order API: Create and retrieve order details.

4. Payment API: Secure payment processing using Stripe.

5. Shipment API: Real-time shipment tracking using Shippo.

**4. Data Schema Design**

**Product Schema:**

• name: String

• description: Text

• price: Number

• stockQuantity: Number

• category: String

• supplier: Reference to Supplier

**Customer Schema:**

• firstName: String

• lastName: String

• email: String

• phone: String

**Order Schema:**

• customer: Reference to Customer

• orderItems: Array (Product references)

• totalAmount: Number

• paymentStatus: String

**Review Schema:**

• product: Reference to Product

• customer: Reference to Customer

• rating: Number

• review Text: Text

**5. Technical Roadmap**

**Milestone 1: Setup Project Environment**

• Initialize Next.js and Sanity CMS.

• Configure version control with GitHub.

**Milestone 2: Develop Core Pages**

• Build Home, Product Listing, Product Details, Cart, Checkout, About, Contact, Blog pages.

**Milestone 3: Backend Configuration**

• Set up Sanity CMS schemas for products, orders, customers, suppliers, and reviews.

**Milestone 4: API Integration**

• Integrate Stripe API for payment processing.

• Integrate Shippo API for real-time shipment tracking.

**Milestone 5: Testing & Deployment**

• Perform unit and integration testing.

• Optimize performance for scalability.

• Deploy to Vercel for production.

**6. Security & Best Practices**

• Data Protection: Implement SSL encryption for all transactions.

• User Privacy: Ensure compliance with GDPR standards.

• Version Control: Use GitHub for collaborative development.

**This comprehensive technical foundation ensures a professional, market-ready product for the Marketplace.**



