Hackathon DAY 2:

NAME : Abdul Rafay

INTRODUCTION: Welcome to our e-commerce platform where we offer high-quality products at unbeatable prices. Shop with confidence and convenience, 24/7!

**Technical Requirements**

**1: Frontend Requirements: User Interface:**

* Simple, easy-to-navigate design.
* Clear product images, descriptions, details, and pricing - Prominent buttons (e.g., "Add to Cart", "Checkout"). **Responsive Design:**
* Mobile-friendly, adjusts to different screen sizes (phones, tablets, desktops). - Touch-friendly and scalable layout. **Essential pages:**
* Home, Product Listing, Product Details, Cart, Checkout, and Order Confirmation. **Home:** Show featured products and categories.
* **Product Listing:** Filters for easy browsing.
* **Product Details:** Clear images, descriptions, and pricing.
* **Cart:** Editable product quantities and total price.  **Checkout:** Easy form for shipping and payment.  **Order Confirmation:** Display order details and tracking

**2: Sanity CMS as Backend:**

Sanity CMS for product data management, orders, and customer information. Here is a detailed schema design for Product, Order, Customer, Payment, Shipment, and Delivery Zone using Sanity CMS:

|  |
| --- |
| export interface Product {  \_id: string; name: string; description: string; price: number; images: {  \_type: 'image'; asset: {  \_ref: string;  \_type: 'reference'  }  }[]; |
| stock: number;  category: string;}    export interface Order {  \_id: string; productID: string; quantity: number; totalAmount: number; orderDate: string;  } export interface Customer {  \_id: string; name: string; email: string; phone: string; address: string;  }  export interface Payment {  \_id: string; order: Order; paymentMethod: 'Credit Card' | 'PayPal' | 'Stripe'; status: 'Pending' | 'Completed' | 'Failed'; transactionId: string; amount: number; paymentDate: string;  } export interface Shipment {  \_id: string; order: Order; trackingNumber: string; status: 'Pending' | 'In Transit' | 'Delivered' | 'Cancelled'; estimatedDelivery: string;  } export interface DeliveryZone {  \_id: string; zoneName: string; coverageAreas: string[]; shippingCost: number;} |

**3. Third-Party APIs:**

 Shipment Tracking: Integrate APIs like Shippo to provide real-time tracking updates.

 Payment Gateways: Use Stripe and Use-Shopping-Cart for payment gateway.

**2. Design System Architecture:**

**1. User Registration:**

* **Frontend (Next.js)** → User completes the registration form.
* **Sanity CMS** → Stores user information (e.g., name, email, password).
* **Confirmation** → A verification email is sent to the user.

**2. Product Browsing:**

* Frontend (Next.js) → User explores product categories.
* Product Data API (Sanity CMS) → Retrieves product details (e.g., name, images, descriptions, prices).
* Frontend (Next.js) → Dynamically displays the product list on the site.

**3. Order Placement:**

* Frontend (Next.js) → User adds products to the cart and proceeds to checkout. - Frontend (Next.js) → Sends order details (items, quantities, user info) to Sanity CMS.
* Sanity CMS → Saves the order information in the database.

Payment Gateway → Processes the payment securely and sends a confirmation.

**4. Shipment Tracking:**

* Sanity CMS → Updates the order with shipping details (e.g., tracking number, carrier).
* Third-Party API (Shipment Tracking) → Retrieves live shipment status.
* Frontend (Next.js) → Displays real-time shipping status (e.g., "In Transit", "Delivered").

**3. Plan API Requirements:**

# Endpoint: /products

* Method: GET
* Description: Fetch all available products. • Response Example

# Endpoint: /order

* Method: POST
* Description: Create a new order.
* Payload:

# Endpoint: /payment

* Method: POST
* Description: Process payment for an order.

# Endpoint: /shipment

* Method: GET
* Description: Track the shipment status for an order.
* Response Example:

# Endpoint: /delivery-zone

* Method: GET
* Description: Fetch all delivery zones and their details.
* Response Example:

**Component Descriptions:**

**Frontend (Next. js):** This is the actual layer that customers see, the layer where you (the customer) interact with the marketplace. It deals with designated UI/UX, routing, and even communicating with the backend through APIs.

**Sanity CMS:** The content creation and management hub for the marketplace, including product data, orders and other dynamic content.

**Payment API (e.g., Stripe/PayPal),** responsible for securely processing user payments and handling financial transactions.

**Shipment API (ShipEngine):** Manages shipping processes, including tracking of shipments and delivery status in real time.

**Email API (SendGrid)** Used to send email notifications, such as order confirmations, shipping updates, and promotional messages.

**Database (Sanity)**: Stores all persistent data, including product details, user orders, and inventory levels, ensuring that the marketplace operates smoothly.