Hackathon DAY 2: ALI RAZA

PLANNING THE TECHNICAL FOUNDATION

Introduction: "Our e-commerce platform is dedicated to providing high-quality furniture. This document details the technical framework needed to create a scalable and user-friendly online marketplace."

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").

```
                    +---------------------+
                    | Frontend (Next.js)|
                    +-----------+---------+
                                |
                     +---------v---------+
                        | Sanity CMS |
                     +---------+---------+
                               |
          +--------------------+----------------------+
          |                    |                      |
+-------v--------+ +--------v--------+ +--------v--------+
| Product Data | | Order Data API | | Shipment API |
  |API (Sanity) | |      (Sanity)      | |   (3rd Party)     |
+----------------+ +-----------------+ +-----------------+
                               |
                      +------v------+ |
                   Payment | | Gateway |
                       +-------------+
```

3. Plan API Requirements:

1. Endpoint: /products

• Method: GET

• Description: Fetch all available products.

• Response Example

```
{
  title: 'Granite square side table',
   summary: 'The Granite Square Side Table features a sleek modern design with a durable granite top Its sturdy construction and minimalist style make it a perfect addition to any living room or outdoor space',
   discountedPrice: 10800,
   price: 12000,
   image: {
     _type: 'image',
     asset: {
       _ref: 'image-897b4d98ea9c77d10a42c8c074be8d84ec5b62ac-432x433-png',
       _type: 'reference'
     }
   },
   featured: null,
   _id: '054de453-3cc0-48ce-a11c-ecec8de602dd',
   slug: 'granite-square-side-table',
   colors: [ 'gray' ],
   sizeQuantities: { small: 13, large: 10, medium: 12 },
   totalItems: 35}
```

2. Endpoint: /order

• Method: POST

• Description: Create a new order.

• Payload:

```
{
  "productId": "054de453-3cc0-48ce-a11c-ecec8de602dd",
  "quantity": 2
}
```

Response:

```
{"status": "Success",
  "message": "Order created successfully.",
  "order": {
    "_id": "order_001",
    "title": "Granite square side table",
    "quantity": 2,
    "totalAmount": 21600,
    "orderDate": "2025-01-16"}}
```

3. Endpoint: /payment

• Method: POST

• Description: Process payment for an order.

• Payload:

```
{
  "order": {
    "_id": "order_001",
    "totalAmount": 21600
  },
  "paymentMethod": "Credit Card",
  "status": "Completed",
  "transactionId": "txn_001",
  "amount": 21600,
  "paymentDate": "2025-01-16"
}
```

Response:

```

{
  "paymentStatus": "Success",
  "transactionId": "txn_001",
  "message": "Payment has been successfully processed."
}
```

4. Endpoint: /shipment

• Method: GET

• Description: Track the shipment status for an order.

• Response Example:

```
{
  "shipmentId": "ship_001",
  "orderId": "order_001",
  "status": "In Transit",
  "expectedDeliveryDate": "2025-01-20"
}
```

5. Endpoint: /delivery-zone

• Method: GET

• Description: Fetch all delivery zones and their details.

• Response Example:

```
[
  {
    "zoneId": "zone_001",
    "zoneName": "North City",
    "deliveryCharge": 50,
    "estimatedDeliveryTime": "2-3 Days"
  },
  {
    "zoneId": "zone_002",
    "zoneName": "South City",
    "deliveryCharge": 70,
    "estimatedDeliveryTime": "3-5 Days"
  }
]
```

```
                 +--------------------+
               |   Frontend (Next.js) |
                 +--------------------+
                            |
                 +---------------------+
                  |  Sanity CMS        |
                | (Product Data, etc.)|
                 +---------------------+
                          |
           +----------------+-------------------+
             |                |                  |
     +---------+      +---------+   +---------+
   |Payment  | |Shipment  | |Email    |
        |API      | |API       | |API     |
   |(Stripe/ | |(ShipEng.)| |(SendGrid)|
       | PayPal) | +---------+ +---------+
              +---------+
                      |
              +--------------------+
             | Database (Sanity) |
               +--------------------+
```

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.