
Day 2 Tasks: Planning the Technical Foundation for Your E-Commerce Marketplace (EVERY WEAR HERE)

---

1. Define Technical Requirements

Translate your business goals into actionable technical features.

- **Frontend Requirements**:
    - Build a responsive and user-friendly interface using **Next.js**.
    - Design essential pages:
        - Home
        - Product Listing
        - Product Details
        - Cart
        - Checkout
        - Order Confirmation
- **Sanity CMS as Backend**:
    - Create schemas for:
        - Products (ID, Name, Price, Stock, Category, Tags)
        - Customers (ID, Name, Contact Info, Address, Preferences, Order History)
        - Orders (ID, Product Info, Customer Info, Total Price, Status)
    - Use **Sanity CMS** to manage real-time data updates.
- **Third-Party APIs**:
    - Plan integrations for:
        - **Payment Gateways** (e.g., Stripe, PayPal,Easypaisa,JazzCash,Online Banking)
        - **Shipment Tracking** (real-time order status updates).
        - **Pre-made Fake API** (Provided by Hackathon Administrations)

---

2. Design System Architecture

Create a diagram to visualize how the components interact. Key architecture components include:

1️⃣**Frontend (Next.js and Tailwind)**:

- User interface for browsing products and placing orders.

2️⃣**Sanity CMS**:

- Manages content for products, orders, and customers.

3️⃣**Third-Party APIs**:

- Payment gateway for secure transactions.
- Shipment tracking API for real-time delivery updates.

Example Workflow:

1. User browses products -> **Next.js** fetches data from **Sanity CMS**.
2. User places an order -> Data is stored in **Sanity CMS**.
3. Payment processed via **Payment Gateway** -> Order status updated in **Sanity CMS**.
4. Shipment details fetched from **Shipment API** -> Displayed to the user.

---

3. Plan API Requirements

Define endpoints needed for your marketplace:

1️⃣**Products API**

- **Endpoint**: `/products`
- **Method**: GET
- **Purpose**: Fetch product details from Sanity CMS.
- **Response**: `{ "id": 1, "name": "T-Shirt", "price": 100, "stock": 50 }`

2️⃣**Orders API**

- **Endpoint**: `/orders`
- **Method**: POST
- **Purpose**: Record a new order in Sanity CMS.

- **Payload**: `{ "customerId": 1, "products": [ { "id": 1, "quantity": 2 } ] }`

3️⃣ **Shipment API**

- **Endpoint**: `/shipment`
- **Method**: GET
- **Purpose**: Fetch real-time shipment tracking info.
- **Response**: `{ "shipmentId": 123, "status": "In Transit", "eta": "2 Days" }`

---

4. Write Technical Documentation

Prepare professional documentation for each technical component:

- **System Architecture**:
    - Diagram showing how **Next.js**, **Sanity CMS**, and APIs interact.
- **API Specifications**:
    - Clearly outline endpoints, methods, payloads, and responses.
- **Sanity Schemas**:
    - Include schemas for:
        - Products
        - Orders
        - Customers
        - Orders Details
        - Shipments
        - Delivery Zones

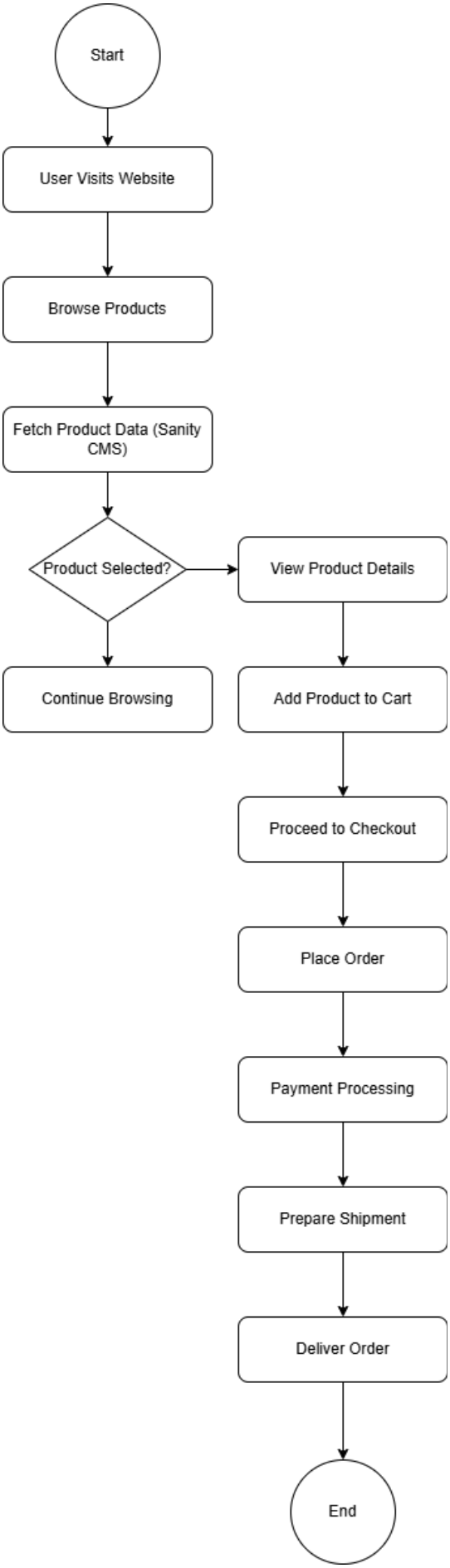
- **Workflow Details**:
    - Document user flows, such as browsing products, placing orders, and tracking shipments.

---

5. Collaborate and Refine

- Discuss ideas with peers to improve scalability and performance.
- Use GitHub for version control to track changes in documentation and designs.

ACTIVITY DIAGRAM FOR E-COMMERCE MARKETPLACE


Start
User Visits Website
Browse Products
Fetch Product Data (Sanity CMS)
Product Selected?
View Product Details
Continue Browsing
Add Product to Cart
Proceed to Checkout
Place Order
Payment Processing
Prepare Shipment
Deliver Order
End

The workflow illustrates the journey of a user interacting with the e-commerce marketplace from **browsing products** to **order delivery**. It involves the following steps:

1. **Start (User Visits Website):**
    - User opens the marketplace platform on a browser or mobile device.
2. **Browse Products:**
    - User navigates through product categories and searches for specific items.
    - The frontend fetches product details (ID, Name, Price, Stock, Tags, etc.) via the **Products API** from **Sanity CMS**.
3. **View Product Details:**
    - User clicks on a product to view details such as price, stock availability, and specifications.
4. **Add Product to Cart:**
    - User adds desired products to the shopping cart.
    - Frontend updates the cart and displays the selected items.
5. **Proceed to Checkout:**
    - User reviews the cart, provides delivery details, and selects a payment method.
6. **Place Order:**
    - The frontend sends order details (customer info, product list, total price) to the **Orders API**, which stores the order in **Sanity CMS**.
    - The order is marked as **Pending**.
7. **Payment Processing:**
    - Order details are sent to the **Payment Gateway API** for transaction processing.
    - Upon successful payment, the order status in **Sanity CMS** is updated to **Paid**.
8. **Prepare Shipment:**
    - Order information is sent to the **Shipment API** for tracking.
    - Shipment details, including tracking ID, status, and estimated delivery date, are fetched and linked to the order.
9. **Assign Delivery Zone:**
    - The shipment is assigned to a delivery zone based on the user's address.
    - The delivery zone data (zone name, assigned driver) is fetched from **Sanity CMS**.
10. **Track Shipment:**
    - User can view real-time shipment updates via the **Shipment API**.
11. **Deliver Order:**

- The delivery driver completes the delivery.
- The shipment status is updated to **Delivered** in **Sanity CMS**.

12. **End (Order Complete):**
    - The user receives the product, and the workflow ends.