**Marketplace-Builder-Hackathon-2025 Day 🡪 2**

**1. Define Tools & Technology Stack**

Outline the tools and technologies you'll use for various components of the marketplace:

**Frontend**

* **Framework**: Next.js (React-based for fast rendering and SEO).
* **Styling**: Tailwind CSS (for flexibility and responsiveness).
* **State Management**: Context API (for lightweight state management).

**Backend**

* **Content Management**: **Sanity CMS** to manage products, customer data, and orders.
* **Data Management**: Custom schemas in Sanity for dynamic updates.
* **APIs**: REST APIs (or GraphQL if needed for flexibility).

**Third-Party APIs and Their Roles**

**1. Payment Gateway**

**Tool**: Jazzcash / Easypaisa / PayPal)

* **Role**:
  + Securely processes payments.
  + Handles different payment methods (credit/debit cards, wallets).
  + Provides real-time payment confirmation.
* **How It Links**:
  + Integrate via APIs.
  + When a customer checks out, the frontend sends payment details to the gateway.
  + On success, the gateway notifies the backend and updates the order status in the database.
* **Example Endpoint**:  
  /processPayment (POST) – Sends order total and payment info to the API.

**2. Delivery Tracking API**

**Tool**: EasyPost or Shippo or Local shipment service

* **Role**:
  + Manages shipping logistics.
  + Tracks shipment status in real-time.
  + Calculates shipping costs dynamically based on delivery zones.
* **How It Links**:
  + When an order is placed, the backend sends the delivery address to the API.
  + The API provides tracking details, which are saved in the order schema.
* **Example Endpoint**:  
  /trackShipment (GET) – Fetches the current status of a delivery based on the tracking ID.

**2. System Architecture Plan**

**[Frontend: User Interface (Next.js)]**

**- Fetch product details from Backend.**

**- Displays product listings, cart, and customization options.**

**- Handles user actions like adding to the cart, placing orders.**

**|**

**v**

**[Backend: Sanity CMS + APIs]**

**- Manages business logic and processes user requests.**

**- Stores data like products, orders, and customer information.**

**- Fetches or updates data in Sanity CMS.**

**- Communicates with:**

**- Payment Gateway (Stripe): Secure payment processing.**

**- Delivery Tracking (EasyPost): Shipment tracking and ETAs.**

**- Notifications (Twilio): Sends email/SMS for order updates.**

**|**

**v**

**[Third-Party APIs]**

**- Payment API: Handles payments and updates order status.**

**- Delivery API: Tracks shipments and returns delivery info.**

**- Notification API: Sends alerts for confirmations and tracking.**

**3. Plan Core API Requirements**

Define key endpoints:

1. **Products**
   * /products (GET): Fetch all products with details like price, stock, and customization options.
2. **Orders**
   * /orders (POST): Submit an order with product IDs, quantities, and customization selections.
3. **Customers**
   * /customers (GET): Fetch customer details (for logged-in users).
4. **Delivery**
   * /shipment (GET): Track order delivery status.

**4. Sanity CMS Schema Design**

Draft schemas for core entities:

1. **Products Schema**
   * Fields: Name, Description, Price, Customization Options, Image URL.
2. **Orders Schema**
   * Fields: Order ID, Customer ID, Products (with customization selections), Total Price.
3. **Customers Schema**
   * Fields: Name, Email, Address, Order History.

**5. Workflow Diagram and Technical Documentation**

