**Marketplace Builder Hackathon 2025 Day 2:**

**1. System Architecture Overview**

#### **Frontend Requirements**

* User-friendly interface for browsing natural products (honey, herbs, shilajit).
* Responsive design for both mobile and desktop users.
* Essential pages:
  + Home
  + Product Listing
  + Product Details
  + Cart
  + Checkout
  + Order Confirmation

#### **Backend Requirements (Sanity CMS & APIs)**

* **Sanity CMS:**
  + Manage product data, customer details, and order records.
  + Schema aligns with business goals.
* **Third-Party APIs:**
  + Payment gateway for secure transactions.
  + Shipment tracking API for real-time updates.

**2. System Architecture Diagram**

[Frontend (Next.js)]

| |

[Sanity CMS] [Third-Party APIs]

#### **Data Flow Explanation**

1. A user visits the marketplace frontend to browse products.
2. The frontend requests product data from Sanity CMS via API.
3. User places an order, and details are stored in Sanity CMS.
4. Order tracking is updated via a Third-Party API.
5. Payment is securely processed through the Payment Gateway.

**3. Key Workflows**

### **User Registration:**

* User signs up -> Data is stored in Sanity -> Confirmation sent to user.

### **Product Browsing:**

* User views product categories -> Sanity API fetches data -> Products displayed.

### **Order Placement:**

* User adds items to cart -> Proceeds to checkout -> Order details saved in Sanity.

### **Shipment Tracking:**

* Order status updates fetched via third-party API -> Displayed to user.

**5. Sanity Schema Example**

|  |
| --- |
| export default {  name: 'product',  type: 'document',  fields: [  { name: 'name', type: 'string', title: 'Product Name' },  { name: 'price', type: 'number', title: 'Price' },  { name: 'stock', type: 'number', title: 'Stock Level' }, |