**Peerzada Store  
System Architecture - Day 2**

**Overview:**

The system architecture of the Peerzada Store marketplace integrates **frontend**, **backend**, and **third-party services** to ensure a seamless and scalable user experience. Each component is designed to interact efficiently, supporting key workflows like product browsing, order placement, and shipment tracking.

**Components:**

1. **Frontend (Next.js):**
   * **Purpose**: The frontend serves as the user interface where customers browse, customise, and place orders.
   * **Responsibilities**:
     + Display product listings dynamically.
     + Handle user interactions (e.g., customisation previews, form submissions).
     + Provide a responsive and mobile-friendly experience.
   * **Key Pages**:
     + Home Page
     + Product Listing Page
     + Product Details Page
     + Cart Page
     + Checkout Page
     + Order Confirmation Page
2. **Sanity CMS (Backend):**
   * **Purpose**: Acts as the central database for managing products, customers, and orders.
   * **Responsibilities**:
     + Store product information (e.g., names, categories, prices, customisation options).
     + Track customer details and order history.
     + Record order data and status updates.
   * **Schema**:
     + Products
     + Customers
     + Orders
     + Packaging Options
3. **Third-Party APIs:**
   * **Purpose**: Enhance functionality by integrating external services for payments and logistics.
   * **APIs Used**:
     + **Shipment Tracking API**: Provides real-time delivery status.
     + **Payment Gateway API**: Handles secure transaction processing.

**Workflows:**

1. **Product Browsing Workflow**:
   * The user navigates to the marketplace.
   * The frontend sends a request to Sanity CMS to fetch product data.
   * The data is displayed dynamically on the website.
2. **Order Placement Workflow**:
   * The user adds items to the cart and proceeds to checkout.
   * Order details (e.g., customer info, products, total amount) are sent to Sanity CMS.
   * The order is saved, and the user receives a confirmation message.
3. **Shipment Tracking Workflow**:
   * Shipment details are retrieved via the third-party API.
   * The delivery status is displayed on the user’s dashboard.

**Architecture Diagram:**

[Frontend (Next.js)]

|

[Sanity CMS] ---> [Third-Party APIs (Shipment, Payment)]

**Benefits of the Architecture:**

1. **Scalability**: The modular design allows for easy integration of additional features or APIs.
2. **Flexibility**: Sanity CMS provides a dynamic backend, enabling quick updates to product and order data.
3. **User Experience**: Real-time interactions and responsive design ensure a seamless customer journey.
4. **Security**: Secure APIs and data storage practices protect customer information.