**Peerzada Store –**

**Day 2 Technical Foundation**

# 1. Define Technical Requirements

**Frontend Requirements:**

* A **user-friendly interface** for seamless browsing of products.
* **Responsive design** for both desktop and mobile users.
* Key pages:
  + Home Page
  + Product Listing Page
  + Product Details Page
  + Cart Page
  + Checkout Page
  + Order Confirmation Page

**Sanity CMS Backend:**

* Use Sanity CMS to manage:
  + Product data.
  + Customer details.
  + Order records.
* Design Sanity schemas aligned with the business goals.

**Third-Party APIs:**

* Integrate APIs for:
  + **Shipment tracking**.
  + **Payment gateways**.
  + **Notifications** (e.g., SMS/email updates).

# 2. Design System Architecture

**Architecture Overview:**

1. **Frontend (Next.js):**
   * Displays product listings, customisation previews, and order details.
   * Handles user interactions and collects data for backend processing.
2. **Sanity CMS:**
   * Acts as the central database for managing products, orders, and customer details.
3. **Third-Party APIs:**
   * Shipment Tracking API for real-time delivery updates.
   * Payment Gateway for secure transactions.

**System Workflow:**

1. **Product Browsing Workflow:**
   * User browses the website -> Data fetched via Sanity CMS API -> Products displayed dynamically.
2. **Order Placement Workflow:**
   * User adds products to cart -> Proceeds to checkout -> Order details saved in Sanity CMS.
3. **Shipment Tracking Workflow:**
   * Shipment details retrieved via third-party API -> Displayed on user’s dashboard.

# 3. Plan API Requirements

**Core API Endpoints:**

1. **Fetch Products:**
   * **Endpoint:** /products
   * **Method:** GET
   * **Description:** Fetch all available products.
   * **Response Example:** { "id": 1, "name": "Wallet", "price": 2000 }
2. **Create Order:**
   * **Endpoint:** /orders
   * **Method:** POST
   * **Description:** Create a new order in Sanity CMS.
   * **Payload Example:** { "customerId": 123, "products": [1, 2], "total": 5000 }
   * **Response Example:** { "orderId": 456, "status": "Success" }
3. **Track Shipment:**
   * **Endpoint:** /shipment
   * **Method:** GET
   * **Description:** Fetch shipment details for an order.
   * **Response Example:** { "orderId": 456, "status": "In Transit", "ETA": "2 days" }

# 4. Write Technical Documentation

**System Architecture Overview:**

* Include a diagram (Frontend → Sanity CMS → Third-Party APIs).
* Brief description of how components interact.

**Key Workflows:**

* **Product Browsing:**
  + User visits homepage -> Sanity CMS API fetches product data -> Products displayed dynamically.
* **Order Placement:**
  + User adds products to cart -> Proceeds to checkout -> Order data saved in Sanity CMS.
* **Shipment Tracking:**
  + Third-party API provides updates -> Displayed in user’s dashboard.

**API Documentation:**

* Each endpoint with method, description, payload, and response examples.

# 5. Collaborate and Refine

**Steps for Collaboration:**

1. **Brainstorm with Peers:**
   * Use tools like Slack or Google Meet for discussions.
   * Share innovative ideas for architecture and APIs.
2. **Peer Reviews:**
   * Share drafts with peers/mentors for constructive feedback.
   * Incorporate suggestions to enhance scalability and performance.
3. **Version Control:**
   * Use GitHub to track changes and ensure team collaboration.

**Key Outcomes:**

1. **Technical Plan Aligned with Business Goals:**
   * Comprehensive plan tailored to the marketplace’s unique requirements.
2. **System Architecture Visualized:**
   * Detailed diagram showcasing interactions between components.
3. **API Requirements Documented:**
   * List of endpoints, methods, payloads, and responses.
4. **Sanity Schemas Drafted:**
   * Designed to handle key data entities like products, orders, and customers.