Day 2: Planning the Technical Foundation

Marketplace Name: Reliable Furniture Hub

Technical Requirements

1. Frontend Requirements:

- Develop a user-friendly interface using **Next.js**.
- Ensure a **responsive design** suitable for all screen sizes.
- Implement **essential pages**:
  - **Home:** Showcases featured furniture collections.
  - **Product Listing:** Displays categorized furniture items.
  - **Product Details:** Provides detailed descriptions, images, and pricing.
  - **Cart:** Summarizes selected furniture items.
  - **Checkout:** Handles customer details and payment processing.
  - **Order Confirmation:** Displays order details and shipment tracking.

2. Backend Requirements:

- Use **Sanity CMS** to manage:
  - Product inventory (name, price, stock, category, material, customization options).
  - Customer orders and shipment tracking.
- Integrate **third-party APIs** for:
  - **Payment Gateway:** Secure transactions.
  - **Shipment Tracking:** Real-time order updates.
  - **Email Notifications:** Automated order confirmations.

3. Security Requirements:

- Enforce **HTTPS** for secure communication.
- Implement **authentication** for CMS access.
- Validate **user input** to prevent malicious attacks.

System Architecture

High-Level Architecture Overview:

```
[Frontend (Next.js)]

   |
   v

[Sanity CMS] <-----> [Product Data API]

   |
   v

[Third-Party APIs] ---> [Payment Gateway API]

                        ---> [Shipment Tracking API]
```

Workflow Example:

1. A user visits the website and browses furniture items.
2. The frontend fetches **product data** from Sanity CMS.
3. Upon order placement:
   - Order details are stored in **Sanity CMS**.
   - Payment is processed via the **Payment Gateway API**.
4. The **Shipment Tracking API** updates the order status.
5. An **order confirmation email** is sent to the customer.

API Requirements

1. Product APIs:

- **Endpoint:** /products
  - **Method:** GET
  - **Description:** Fetch all furniture items.
  - **Response Example:**
  - [
  -   { "id": 1, "name": "Oakwood Dining Table", "price": 1500 },
  -   { "id": 2, "name": "Leather Sofa Set", "price": 3000 }

]

2. Order APIs:

- **Endpoint:** /orders
  - **Method:** POST
  - **Description:** Create a new order.
  - **Payload Example:**
  - {
  - "customerName": "Owais Khan",
  - "products": [
  - { "id": 1, "quantity": 1 },
  - { "id": 2, "quantity": 1 }
  -],
  - "paymentStatus": "Completed"

}

  - **Response Example:**

{ "orderId": 6789, "status": "Order Confirmed" }

3. Shipment APIs:

- **Endpoint:** /shipment
  - **Method:** GET
  - **Description:** Fetch order delivery status.
  - **Response Example:**
  - {
  - "shipmentId": 12345,
  - "status": "Out for Delivery",
  - "estimatedDelivery": "2025-01-20"

}

Key Workflows

1. User Registration and Login:

- User creates an account and logs in securely.

2. Product Browsing & Filtering:

- Users can filter products by category, material, and price.

3. Order Placement & Payment:

- Users add items to the cart and complete checkout.
- Payments are processed securely through third-party gateways.

4. Shipment Tracking:

- Order statuses update dynamically via API integration.
- Customers can track delivery progress.

Sanity CMS Schemas

1. Product Schema:

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'category', type: 'string', title: 'Category' },
  ]
};
```

2. Order Schema:

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'customerName', type: 'string', title: 'Customer Name' },
    { name: 'products', type: 'array', of: [{
      type: 'object',
      fields: [
        { name: 'productId', type: 'reference', to: [{ type: 'product' }] },
        { name: 'quantity', type: 'number', title: 'Quantity' }
      ]
    }] },
    { name: 'paymentStatus', type: 'string', title: 'Payment Status' },
    { name: 'timestamp', type: 'datetime', title: 'Order Date & Time' }
  ]
};
```