**Marketplace Technical Foundation - Furniture ECommerce**

**1. Define Technical Requirements**

**Frontend Requirements:**

Essential Pages:

Home Page: Display featured furniture collections, new arrivals, and promotions.

Product Listing Page: Showcase tables and chairs by categories such as dining, office, or outdoor.

Product Details Page: Include details like dimensions, material, price, and stock availability.

Cart and Checkout Pages: Simplify the shopping and payment process.

Order Confirmation Page: Display order summary and estimated delivery date.

Responsiveness: Ensure the design adapts seamlessly to mobile, tablet, and desktop devices.

Backend Requirements:

Sanity CMS:

Store product information such as name, price, material, stock, dimensions, and images.

Manage customer details (e.g., name, email, address) and order records (e.g., product details, payment status).

Third-Party APIs:

Payment Gateway API: Securely handle online payments.

Shipment Tracking API: Provide real-time updates on delivery status.

Email Service API: Send automated emails for order confirmations and updates.

**2. Design System Architecture**

System Components:

1. Frontend (Next.js): User interface for browsing, ordering, and tracking products.

2. Sanity CMS: Backend database for managing product and order information.

3. Third-Party APIs:

Payment processing (e.g., Stripe or PayPal).

Real-time shipment tracking (e.g., FedEx API).

Architecture Diagram:

[Frontend (Next.js)]

|

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

|

[Third-Party APIs] ---> [Shipment Tracking API]

---> [Payment Gateway]

Key Workflows:

1. User Registration:

User registers on the platform → Data is stored in Sanity CMS → Email confirmation sent.

2. Product Browsing:

User browses product categories → Sanity API fetches and displays product data.

3. Order Placement:

User adds items to the cart → Proceeds to checkout → Order details stored in Sanity CMS.

4. Shipment Tracking:

Order status fetched via shipment API → Displayed in the user’s account.

---

3. Plan API Requirements

---

4. Write Technical Documentation

System Architecture Overview:

Frontend: Built with Next.js to provide a fast and interactive user experience.

Sanity CMS: Manages data like product details, customer orders, and inventory.

Third-Party APIs: Handles payment and shipping integrations.

Workflow Example:

Order Placement Workflow:

1. User adds products to the cart.

2. The frontend sends a POST request to /orders with order details.

3. Sanity CMS records the order and returns a confirmation.

4. Payment is processed through the gateway, and shipping information is fetched.

Sanity Schema Example:

export default {

name: 'product',

type: 'document',

fields: [

{ name: 'name', type: 'string', title: 'Product Name' },

{ name: 'price', type: 'number', title: 'Price' },

{ name: 'dimensions', type: 'string', title: 'Dimensions' },

{ name: 'material', type: 'string', title: 'Material' },

{ name: 'stock', type: 'number', title: 'Stock Level' },

{ name: 'image', type: 'image', title: 'Product Image' }

]

};

---

5. Collaborate and Refine

Conduct peer reviews to gather feedback on your system architecture and API design.

Use GitHub to track changes and refine the technical plan collaboratively.

---