Day 2 Hackathon Technical Foundation

This document outlines the technical foundation for the Furniture Marketplace project. It covers the technical requirements, system architecture, API endpoints, and other necessary details to ensure a seamless and scalable platform for both the development and deployment phases.

1. Define Technical Requirements

Based on the business goals, the following technical requirements have been defined for the Furniture Marketplace:

Frontend Requirements:

- User-friendly interface for browsing products.

- Responsive design for mobile and desktop users.

- Key pages: Home, Product Listing, Product Details, Cart, Checkout, Order Confirmation.

Backend: Sanity CMS

- Use Sanity CMS to manage product data, customer details, and order records.

- Sanity schemas must align with the data structure for products, orders, and users.

Third-Party APIs:

- Integrate APIs for shipment tracking, payment gateways, and order management.

- APIs will provide the necessary data to populate the frontend with product and order information.

2. Design System Architecture

The system architecture for the Furniture Marketplace includes the following components:

Frontend (Next.js) -> Sanity CMS -> Product Data API

```
          |        |
       Shipment Tracking API
          |
    Payment Gateway API
```

Key Workflows:

1. User Registration:

  - User signs up -> Data is stored in Sanity CMS -> Confirmation sent.

2. Product Browsing:

  - User views product categories -> Data fetched from Sanity API -> Products displayed.

3. Order Placement:

  - User adds products to cart -> Order placed -> Data recorded in Sanity.

4. Shipment Tracking:

  - Order status updates fetched via 3rd-party API -> Displayed to user.

5. Payment Processing:

  - Payment details processed securely via Payment Gateway -> Confirmation sent and recorded.

3. Plan API Requirements

The following API endpoints are required for the marketplace to function:

1. /products (GET): Fetch all available product details from Sanity.

   - Response Example: { "id": 1, "name": "Product A", "price": 100, "stock": 10 }

2. /orders (POST): Create a new order with customer info and product details.

   - Payload Example: { "customerName": "John Doe", "productId": 1, "quantity": 2 }

   - Response Example: { "orderId": 123, "status": "Success" }

3. /shipment (GET): Track shipment status for an order.

- Response Example: { "orderId": 123, "status": "In Transit", "ETA": "15 mins" }

4. Write Technical Documentation

The technical documentation includes the following key documents:

1. System Architecture Overview: Diagram and description of how the frontend, Sanity CMS, third-party APIs, and payment systems interact.

2. API Specification Document: Details of each endpoint, method, response, and usage.

3. Data Schema Design: Structure and relationships for Sanity CMS entities, such as products, orders, and customers.

4. Technical Roadmap: Step-by-step milestones to complete the marketplace, including timelines and key deliverables.

By Ummefizza