

Day 2: Planning the Technical Foundation for Furniro

Introduction

This document outlines the technical foundation for the Furniro marketplace, focusing on the design, implementation, and workflows of the platform. The document covers technical requirements, system architecture, API endpoints, and future plans.

1. Technical Requirements

Frontend Requirements

- Framework: Next.js for server-side rendering and client-side interactivity.
- Features:
 - Product listing.
 - Cart management.
 - Product details page.
 - Responsive design for mobile and desktop.

Backend Requirements

- Content Management System (CMS): Sanity CMS for managing product and order data.
 - Database: No separate database; Sanity CMS acts as the primary data store.
-

2. System Architecture

High-Level Design

Key Components:

- **Frontend (Next.js):** User-facing application for browsing and managing products.
- **Backend (Sanity CMS):** Stores and retrieves data for products and orders.

Workflow:

1. User browses products on the frontend.
2. Product details are fetched from Sanity CMS and displayed dynamically.
3. User adds items to the cart and proceeds to checkout.
4. Payment information is sent to the gateway for processing.

Example Architecture Diagram:

[Frontend (Next.js)] <--> [Sanity CMS] <--> [Payment Gateway]

3. API Requirements

Endpoints Overview

GET /api/products

- **Description:** Fetches all available products.

Response Example:

```
[
  { "id": 1, "name": "Sofa Set", "price": 500, "stock": 10 },
  { "id": 2, "name": "Dining Table", "price": 300, "stock": 5 }
]
```

POST /api/cart

- **Description:** Adds items to the user's cart.
 - **Request Example:**
{ "productId": 1, "quantity": 2 }
 - **Response Example:**
{ "status": "success", "message": "Item added to cart." }
-

4. Technical Documentation

Sanity CMS Schema

Product Schema Example:

```
export default {
  name: 'product',
  type: 'document',
  fields: [
```

```
{ name: 'name', type: 'string', title: 'Product Name' },  
{ name: 'price', type: 'number', title: 'Price' },  
{ name: 'stock', type: 'number', title: 'Stock Level' }  
],  
};
```

The End
