

DAY 2 TASK

Planning the Technical Foundation

DEFINE TECHNICAL REQUIREMENTS

THE NAME OF THE E-COMMERCE WEBSITE IS "COMFORTY."

project introduction: Dynamic
E-commerce website
overview:

This project is fully responsive and dynamic e-commerce platform built using next.js and styled using Tailwind CSS, designed to build a seamless shopping experience. The platform is feature-rich and integrate with sanity CMS for dynamic content management, ensuring easy updates for products, categories and other site content. The website is tailored for a furniture store' showcasing both household and office furniture.

COMPONENTS AND INTERACTIONS:

1. Frontend (Next.js):
 - o Acts as the user interface for the system.
 - o Fetches data from APIs and displays it to the user.
2. Sanity CMS:
 - o A headless CMS to manage content such as product details.
 - o Provides data to the frontend via APIs.
3. Product Data API:
 - o A dedicated API for handling product information fetched from Sanity CMS.
4. Third-Party APIs: Includes external services for shipment, tracking, payment and other functionalities.
5. Shipment Tracking API:
Provides real-time updates on shipment status.
6. Payment Gateway:
Handles secure payment transactions.

FRONTEND ARCHITECTURE & IMPLEMENTATION PLAN

Objective:

Design and implement a user-friendly, efficient, and visually appealing frontend interface that includes the following pages:

1. Homepage
2. Product Listing Page
3. Cart Page
4. Checkout Page

1. Technologies and Tools: The project will use the following tech stack:

- Frontend:
 - Framework: Next.js
 - Styling: Tailwind CSS
 - State Management: Redux Toolkit or Context API
 - Icons & UI Components: Heroicons, Headless UI
- Backend:
 - Sanity CMS for managing product data and orders.
 - Third-Party APIs:
 - Stripe for payment gateway integration.
 - Shippo or AfterShip for shipment tracking.

2.2. PRODUCT LISTING PAGE

2. PAGE DESCRIPTIONS AND WIREFRAMES:

2.1. Homepage

Purpose:

: Introduce the store and highlight featured products, promotions, or categories.

Key Features:

- Hero banner (e.g., seasonal promotions).
- Display of featured products with a "View Details" button.
- Dynamic sections controlled by Sanity CMS (e.g., categories, promotional content)

Purpose:

Showcase all available products with filters and sorting options

. Key Features:

- Filters: Price range, categories, availability.
- Sorting options: Price (low-to-high/high-to-low), ratings
- Pagination or infinite scrolling for better performance
- Each product card will have:

- o Product image.

- o Name, price, rating, and "Add to Cart" button.

2.3. Cart Page

Purpose:

Enable users to view the products added to the cart and modify the quantity or remove items

Key Features:

- Display of product name, image, price, and quantity.
- Total price calculation with tax and shipping (if applicable).
- "Continue Shopping" and "Proceed to Checkout" buttons. Checkout

Page Purpose: Allow users to enter shipping information, select payment methods, and place an order.

Key Features:

- Shipping details form (name, address, contact number).
- Payment integration (via Stripe)
- Order summary section showing the cart total and breakdown.
- "Place Order" button to confirm and submit the order.

SANITY CMS CONFIGURATION

3.1. Setting Up Schemas

We will create the following schemas in Sanity CMS:

1. Product Schema:
 - o Fields: name, price, description, image, category, stock
2. Category Schema:
 - o Fields: name, description, image.

3. Order Schema:

- o Fields: order, id, user details (name, address, email, etc.), products (array of product IDs with quantities), total price, status (pending, shipped, delivered).

3.2. Querying Data from Sanity

- Use GROQ (Graph-Relational Object Queries) to fetch data for:
- Products on the homepage and listing page.
 - Individual product details for the "Product Details Page".
 - Orders on the admin dashboard (if applicable).

4. Third-Party API Integration

4.1. Stripe for Payment Gateway

1. Install Stripe SDK
2. Integrate Stripe Checkout API for secure payment processing.
3. Use webhooks to handle payment confirmation.

4.2. Shipment Tracking:

1. Choose a shipment tracking API (e.g., Shippo, AfterShip).
2. Use the API to:
 - o Generate shipping labels.
 - o Track shipment status in real-time.

THANK YOU