



Created by Laiba Noor

Roll no 52916

Hackathon Day 02: Planning The Technical Foundation

1-Technical Requirements:

User Management:

- User registration and login functionality to ensure secure access.
- Ability to manage user profiles including address, payment information, and order history.

Product Management:

- A comprehensive product catalog organized by categories, subcategories, and easy-to-use filters such as Category, Product Type, Price Range, and Brand.
- Detailed product pages that include high-quality images, descriptions, dimensions, available quantities, and key features.
- Efficient inventory management system that tracks product stock levels and updates in real-time.

Shopping Cart and Checkout:

- Users can easily add, edit, or remove items from their shopping cart.
- A smooth, secure, and responsive checkout process to enhance user experience.
- Integration of multiple payment gateways like credit/debit cards, Easypaisa, JazzCash, and digital wallets.
- Support for discount codes and promotional offers to attract customers.

Order Management:

- Customers can track their orders and view order history.

- Admin dashboard to monitor and process orders, update their status, and manage fulfillment.

Shipping and Delivery:

- Integration with trusted shipping partners for real-time delivery tracking updates.
- Offering multiple delivery options along with shipping cost calculation to suit customer preferences.

Search and Navigation:

- Advanced search functionality with filters like Category, Product Type, Price, and Brand for a personalized browsing experience.
- Breadcrumb navigation and user-friendly menu structure for easy site navigation.
- Moderation tools available to manage user reviews and ensure content quality.

Wishlist and Favorites:

- Customers can save their favorite products for future purchases by adding them to a personalized wishlist.

Content Management System (CMS):

- Ability to manage blog posts, promotional banners, and other content pages, enabling dynamic website updates and promotions.

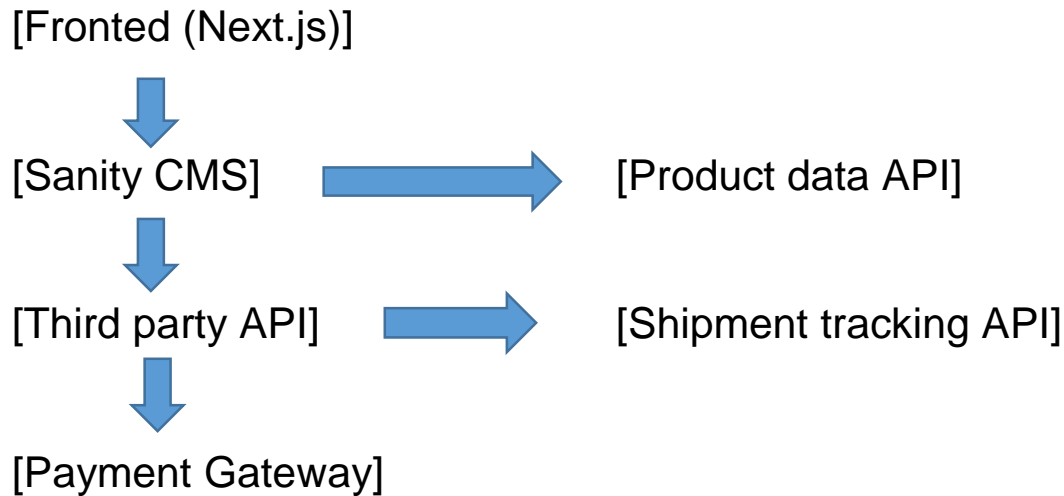
Responsiveness:

- A mobile-friendly design with a responsive UI/UX to ensure seamless browsing and shopping experience across all devices.

Third-Party Integrations:

- Integration with multiple payment gateways such as credit/debit cards, Easypaisa, JazzCash, and digital wallets for convenient transactions.
 - Collaboration with reliable shipping partners like TCS, M&P Express, and Leopards for smooth order delivery and tracking.
 - Integration of marketing tools for email campaigns and social media, helping to enhance customer engagement and promote products.
-

2.High-Level System Architecture Diagram



Project Overview:

- **Frontend (Next.js):**
This is the user-facing part of your project, where the website or web app is built using Next.js, providing a responsive and dynamic interface for customers.
 - **Sanity CMS:**
Sanity CMS acts as the content management system, allowing easy management and updates of your website's content like product details, blog posts, and banners.
 - **Product Data API:**
This API fetches and provides product-related data such as details, images, and pricing. It helps to display updated product information on the website.
 - **Third-party API:**
This could be used for integrating additional services or features, such as external product catalogs, user reviews, or other external data sources.
 - **Shipment Tracking API:**
Integrated with the website, this API fetches real-time shipping updates for orders, allowing customers to track their shipments.
 - **Payment Gateway:**
The payment gateway handles all payment transactions (credit/debit cards, digital wallets, etc.) to securely process payments for customer orders.
-

3.API Endpoints for Furniture E-commerce Website

User Authentication

1-User Registration

- **Endpoint:** /auth/register
- **Method:** POST
- **Description:** Registers a new user and creates an account in the system.
- **Response Example:** { "id": "1", "name": "Laiba Noor", "email": "positivethought2562@gmail.com" }

2-User Login

- **Endpoint:** /auth/login
- **Method:** POST
- **Description:** Logs the user in and returns an authentication token.
- **Response Example:** { "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9" }

Product Management

1-Get All Products

- **Endpoint:** /products
- **Method:** GET
- **Description:** Fetches all the available products in the store.
- **Response Example:** [{ "id": "1", "name": "Deluxe Wooden Table", "price": 1200, "stock": 10 }, { "id": "2", "name": "Elegant Dining Chair", "price": 550, "stock": 15 }]

2-Get Product Details by ID

- **Endpoint:** /products/{id}
- **Method:** GET
- **Description:** Retrieves the details of a specific product using its ID.
- **Response Example:** { "id": "1", "name": "Deluxe Wooden Table", "price": 1200, "description": "A beautiful solid wood table with a sleek design.", "stock": 10 }

Cart Management

Endpoint: /cart

Method: GET

Purpose: Fetches the user's current cart items

Response Example: [{ "productId": "2", "name": "Luxury wooden table", "quantity": 1, "price": 1200 }]

- **Endpoint:** /cart
- **Method:** POST

- **Description:** Adds a product to the cart
- **Response Example:** { "productId": "2", "quantity": 1, "status": "Successfully added to cart" }

Order Processing

- **Endpoint:** /orders
- **Method:** POST
- **Description:** place a new order
- **Response Example:** { "orderId": "8456", "status": "Processing", "total": 980 }

- **Endpoint:** /orders
- **Method:** GET
- **Description:** Retrieves order details
- **Response Example:** { "orderId": "8456", "status": "Shipped", "deliveryDate": "2025-01-16" }

Payment Processing

- **Endpoint:** /payments
- **Method:** POST
- **Purpose:** Processes the payment for an order
- **Response Example:** { "paymentId": "py-2345", "status": "Success", "amount": 980 }

Shipment Tracking

- **Endpoint:** /shipment/{orderId}
 - **Method:** GET
 - **Purpose:** Tracks the shipping status of an order
 - **Response Example:** { "orderId": "8456", "status": "Out for delivery", "estimatedDelivery": "2025-01-25" }
-

