Created by Shafaq Saleem   Roll num : 00159196

Marketplace Technical Foundation -[Furniro E-Commerce]

Planning The Technical Foundation

Day 2 Activities: Transitioning to Technical Planning

1. Define Technical Requirements

Frontend Requirements

- Product Listing Page (filters: price, category, popularity)
- Product Details Page (images, descriptions, reviews)
- Cart Page (adjust quantities, total costs)
- Order Confirmation Page (order summary, tracking link)
- Checkout Page (user info, discount codes, finalize payment)

Design Considerations - Responsive Design (mobile, tablet, desktop)
- Performance Optimization (lazy loading, caching)
- Intuitive Navigation (categories, cart, search)
- SEO Optimization (crawlable, dynamic metadata)

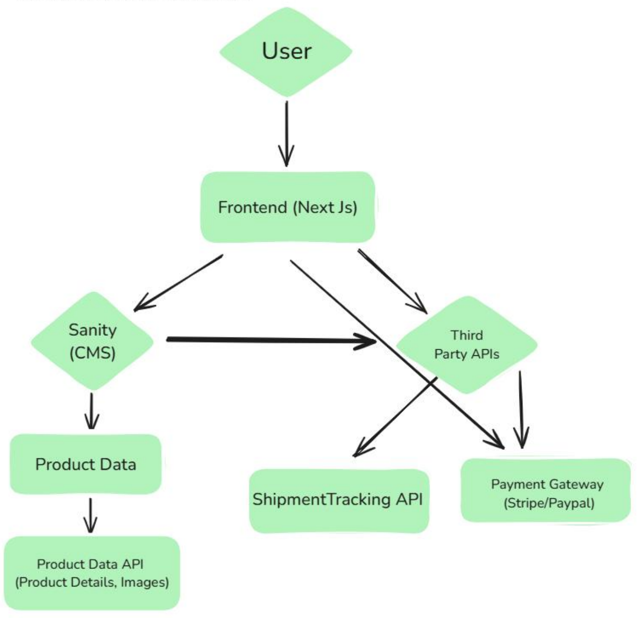
Backend Requirements (Sanity CMS)

Data Models - Products: Name, Description, Price, Images, SKU, Category, Inventory, Tags
- Categories: Name, Slug, Description, Hierarchy
- Customers: Name, Email, Address, Purchase History
- Orders: Order ID, User Info, Items Ordered, Payment/Shipment Status
- Secure Data Access with API keys

Third-Party API Integrations

Shipment Tracking API - Fetch real-time shipment status
- Provide tracking links for users
- Payment Gateway - Platforms: Stripe, PayPal
- Support multiple payment options (cards, wallets)
- Analytics API: Track user behavior, sales, traffic trends

2. Design System Architecture



Data Flow Explanation:

User Interaction: A customer visits your e-commerce store's frontend (Next.js).

Fetching Product Data: The frontend sends a request to Sanity CMS via Product Data API to retrieve product listings and their details (images, price, descriptions).

Third-Party Integrations: If needed, the frontend can also integrate with a Third-Party API (e.g., shipping services, product recommendations, or additional customer data).

Order Placement: Once a user selects products, the frontend integrates with a Payment Gateway (e.g., Stripe, PayPal) to process the payment.

Shipping Process: After payment, the Shipment Tracking API provides real-time tracking for the order's shipment.

3. Plan API Requirements

Structured table summarizing the API endpoints

Endpoint Name	Method	Description	Payload/Response
/products	GET	Fetch all available products from Sanity.	Response: { "id": 1, "name": "Syltherine","price": 2.500, "stock": 10, "image": "url" }
/orders	POST	Create a new order in Sanity.	Payload: { "customer": { "name": "Ali", "email": "Ali@example.com" }, "products": [{ "id": 1, "quantity": 2 }], "paymentStatus": "Paid" }
/shipment	GET	Track order status via third-party API	Response: { "shipmentId": SI05, "orderId": SKU-07, "status": "In Transit", "expectedDelivery": "2025-01-25" }

4-User Workflows

User Browses Products:

Frontend sends a GET request to /products.
Backend fetches product data from Sanity CMS or database.
Response is displayed on the frontend.

User Adds Product to Cart:

Frontend sends a POST request to /cart with product details.
Backend updates the user's cart in the database.
Confirmation response is sent back to the frontend.

User Places Order:

Frontend sends a POST request to /orders with cart and user details.

Backend:

Validates the order.
Processes payment through a third-party API.
Stores order details in the database.
Returns an order confirmation to the frontend.