SHOP.CO

MARKETPLACE TECHNICAL FOUNDATION

Nida Wahid Baig
Roll No: 00471011

Technical Requirements – SHOP.CO

Frontend Requirements

- Home, On Sale, New Arrivals, Brands, and Cart pages.
- Mobile, tablet, and desktop compatibility using Tailwind CSS.
- Search bar, filters, and sorting options.

Sanity CMS as Backend

- Act as the primary backend to manage product data, customer details and order records.
- Provides API for the frontend to fetch and update data.

Product Data API

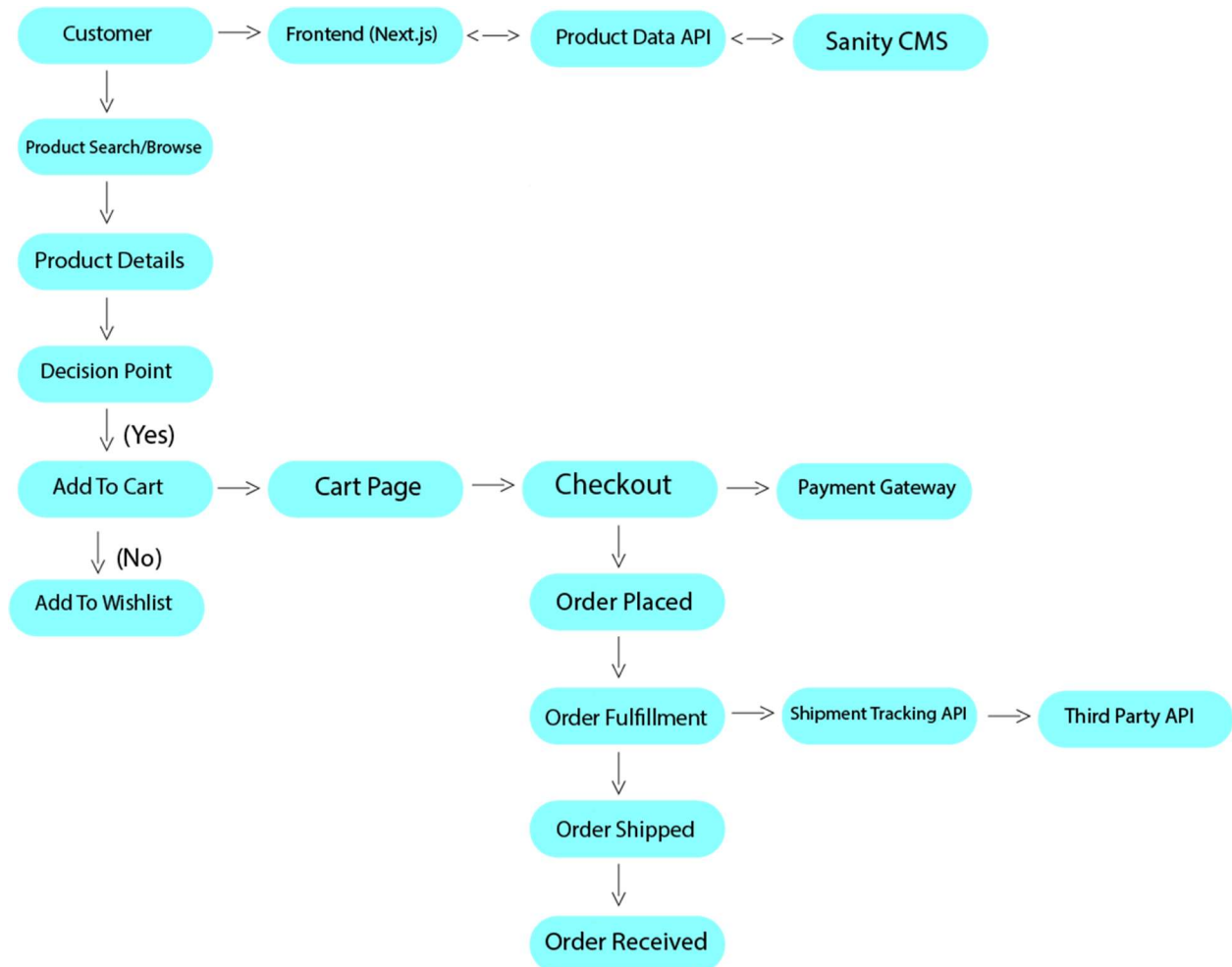- Provides endpoints to fetch product listings, details, and orders record.

Third Party APIs

- Integrates services like shipment tracking and payment processing.

Payment Gateway

- Processes user payments securely and provides transactions confirmations.

Flowchart: Shop Co. E-commerce Sales Process

Customer → Frontend (Next.js) <—> Product Data API <—> Sanity CMS

Customer ↓

Product Search/Browse ↓

Product Details ↓

Decision Point ↓ (Yes)

Add To Cart → Cart Page → Checkout → Payment Gateway

Add To Cart ↓ (No)

Add To Wishlist

Checkout ↓

Order Placed ↓

Order Fulfillment → Shipment Tracking API → Third Party API

Order Fulfillment ↓

Order Shipped ↓

Order Received

Flowchart Explanation

1. Customer Interaction:

- **Entry Point:** The customer begins by interacting with the **Frontend (Next.js)** of our shop.
- **Product Exploration:** The customer can **search** for specific products or **browse** through available categories.
- **Product Detail View:** The customer can view detailed information about a chosen product, including images, descriptions, and pricing.

2. Decision and Cart:

- **Decision Point:** The customer decides whether to:
    - **Purchase:** Proceed to add the product to their **cart**.
    - **Wish list:** Save the product for later consideration.
- **Add to Cart:** The selected product is added to the customer's shopping cart.
- **Cart Page:** The customer can review the items in their cart, adjust quantities, and proceed to **checkout**.

3. Checkout and Payment:

- **Checkout:** The customer enters their shipping and billing information.
- **Payment Gateway:** The customer provides payment details, and the **Payment Gateway** processes the transaction securely.
- **Order Placed:** Upon successful payment, an **order** is created and stored in the **Sanity CMS**.

4. Order Fulfillment and Shipping:

- **Order Fulfillment:** The system processes the order, updates inventory, and prepares the shipment.
- **Shipment Tracking:** The system interacts with the **Third-Party API** (e.g., FedEx, UPS) to generate a shipping label and initiate shipment.
- **Order Shipped:** The order is shipped, and the system updates the order status in the Sanity CMS. The customer receives a shipping confirmation with tracking information via the frontend.

5. Order Delivery:

- **Order Received:** The customer receives the order.

Key Components:

- **Frontend (Next.js):** The customer-facing interface of the e-commerce store.
- **Product Data API:** Facilitates the exchange of product information between the frontend and the Sanity CMS.
- **Sanity CMS:** Stores product data, customer information, and order history.
- **Payment Gateway:** Processes online payments securely.
- **Third-Party API:** Provides shipment tracking services.

.

API Endpoints

Endpoint	Method	Purpose	Response Example
/products	GET	Fetches all product details	{ "id": 1, "name": "Product A", "price": 100 }
/orders	POST	Creates a new order	{ "order_id": 123, "status": "Order Created" }
/shipment	GET	Tracks order shipment status	{ "shipment_id": "SHIP123", "order_id": 789, "status": "In Transit" }