

Day 2 Planning The Technical Foundation E-Commerce

1 Technical Requirements

Project Context:

- Build a general e-commerce marketplace focused on selling **Nike shoes**.

Frontend Requirements:

- Use **Next.js** for SSR/SSG and **Tailwind CSS** for responsive design.
- Essential pages:
 - **Home Page:** Showcase Nike shoe categories and promotions.
 - **Product Listing:** Grid layout with filters (size, price, color).
 - **Product Details:** Detailed descriptions, images, reviews, and add-to-cart option.
 - **Cart Page:** View and manage selected items.
 - **Checkout Page:** Collect shipping/payment details.
 - **Order Confirmation:** Display order summary and tracking info.

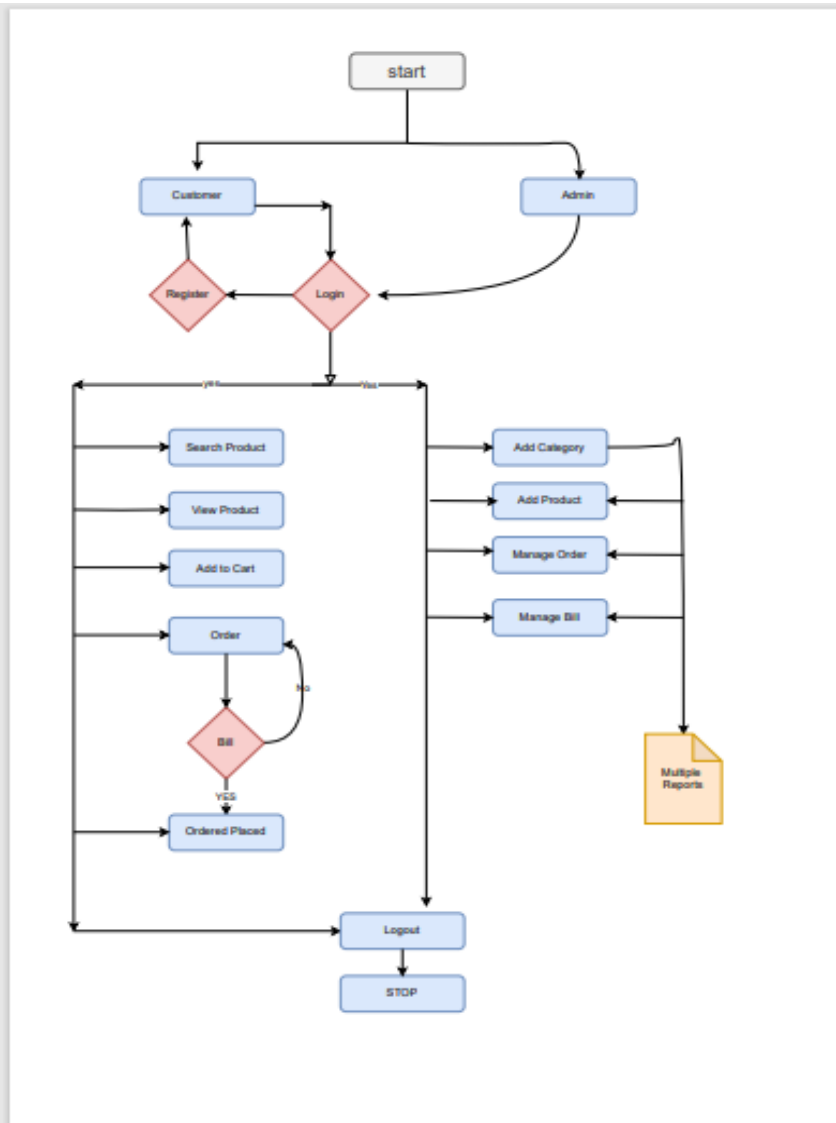
Backend Requirements (Sanity CMS):

- Create schemas for products (name, size, price, color, stock), customers (profile, address), and orders (details, status).
- Use Sanity GROQ queries for dynamic data.

Third-Party API Integrations:

- **Shipment Tracking:** Real-time tracking with services like Shippo.
- **Payment Gateway:** Secure payments via credit card or Easypisa.

2) System Architecture Diagram



1. **Frontend (Next.js):** This is where the user interacts with the system. It communicates with other components to fetch product data and handle orders.
2. **Sanity CMS:** Manages product data and stores order details.
3. **Product Data API:** Fetches product data from Sanity CMS and sends it to the frontend.
4. **Third-Party APIs:** These APIs could handle shipment tracking, inventory updates, or other external services.
5. **Payment Gateway:** Handles the payment transaction securely and confirms the order once processed.

Data Flow Work flow

1. User Registration:

- The user signs up via the frontend.
- The registration data is stored in **Sanity CMS**.
- A confirmation is sent to the user.

2. Product Browsing:

- The user browses different product categories.
- The **Product Data API** queries **Sanity CMS** for product data (e.g., names, descriptions, prices).
- Products are displayed on the frontend.

3. Order Placement:

- The user adds items to their cart and proceeds to checkout.
- Order details are sent to **Sanity CMS** through an API request, where it is recorded.
- The user receives a confirmation message.

4. Shipment Tracking:

- Shipment status is updated through the **Third-Party APIs**.
- The shipment tracking information is displayed to the user in real-time.

5. Payment Processing:

- The user enters payment details at checkout.
- The payment gateway securely processes the payment.
- After successful payment, a confirmation is sent to the user and the order is recorded in **Sanity CMS**.

Example Diagram

[Frontend (Next.js)]

| | |

[Sanity CMS] [3rd Party APIs] [Payment Gateway]

| | |

[Product Data API] [Shipment Tracking API]

3 API End points

1. Fetch All Products

- **Endpoint:** `/products`
- **Method:** `GET`
- **Description:** Fetch all available products from Sanity CMS.
- **Response Example:**
 - Product details: ID, name, price, stock, image.

2. Create New Order

- **Endpoint:** `/orders`
- **Method:** `POST`
- **Description:** Create a new order in Sanity.
- **Payload Example:**
 - Customer information (name, email, address).
 - Product details (productId, quantity).
 - Payment status (e.g., pending, completed).
- **Response Example:**
 - Order ID and status.

3. Track Shipment

- **Endpoint:** `/shipment`
- **Method:** `GET`
- **Description:** Track order status via third-party Shipment Tracking API.
- **Response Example:**
 - Shipment ID, order ID, status, expected delivery date.

4 Data schema design

1. Product Schema

- ID :
- Name:
- Description:
- Price:
- Stock:
- Category:
- Image:

Fields:

- id: String
 - name: String
 - description: String
 - price: Number
 - stock: Number
 - category: String
 - image: Image
-

2. Order Schema

- Order ID:
- Customer ID:
- Products:
- Total Price:
- Order Date:
- Status:

Fields:

- orderId: String
 - customerId: String
 - products: Array of products (objects with product ID and quantity)
 - totalPrice: Number
 - orderDate: Date
 - status: String
-

3. Shipment Schema

- Shipment ID:

- Order ID:
- Shipment Date:
- Tracking Number:
- Status:
- Expected Delivery Date:

Fields:

- shipmentId: String
 - orderId: String
 - shipmentDate: Date
 - trackingNumber: String
 - status: String
 - expectedDeliveryDate: Date
-

4. Customer Schema

- Customer ID:
- Name:
- Email:
- Phone:
- Shipping Address:
- Billing Address:

Fields:

- customerId: String
 - name: String
 - email: String
 - phone: String
 - shippingAddress: String
 - billingAddress: String (optional)
-
-

5) Payment Schema

- Payment ID:
- Order ID:
- Payment Status:
- Amount:
- Payment Date:
- Payment Method:

Fields:

- **paymentId: String**
 - **orderId: String**
 - **paymentStatus: String**
 - **amount: Number**
 - **paymentDate: Date**
 - **paymentMethod: String**
-

6. Delivery Zone Schema

- **Zone ID:**
- **Zone Name:**
- **Shipping Cost:**
- **Available Areas:**
- **Fields:**
- **zoneld: String**
- **zoneName: String**
- **shippingCost: Number**
- **availableAreas: Array of strings**