

# Marketplace Technical Foundation – Nike Shoes

## Key Objectives:

### 1. User Authentication:

Implement Clerk for secure and seamless user authentication.

### 2. Content Management:

Use Sanity CMS to manage and store product data, user details, and order information efficiently.

### 3. Frontend Development:

Build a responsive and user-friendly interface using Next.js, ensuring smooth navigation across all devices.

### 4. Product Browsing and Cart Management:

Dynamically fetch and display product details from Sanity CMS using API integrations.

Enable users to add products to the cart and proceed to checkout.

### 5. Payment Integration:

Integrate a reliable payment gateway like Stripe or PayPal.

### 6. Order Management:

Use Sanity CMS to store and track order details, including product, user, and payment information.

### 7. Shipping and Delivery:

Integrate a third-party shipping API (e.g., ShipEngine) to provide real-time order tracking and delivery status updates.

## Define Technical Requirements:

### 1. Frontend Requirements

- **User-friendly interface:**

Design an intuitive and attractive interface for browsing shoe products.

Include easy navigation for users to find shoes by categories.

- **Responsive Design:**

Ensure the website is fully responsive, providing an optimal viewing experience across mobile, tablet, and desktop devices.

- **Essential pages:**

**Homepage:**

Display featured shoe products, seasonal trends, featured products, bestselling products, gear up and a search bar for easy navigation.

**Product Listing:**

Include filters for shoe categories, price range, size, color and popularity.

**Product Details Page:**

Provide detailed information about each shoe product, including images, descriptions prices, sizes, colors, and customer reviews.

**Cart Page:**

Allow users to view and edit their selected products before proceeding to checkout.

**Checkout Page:**

Collect user details, shipping information, and payment options securely.

### 2. Sanity CMS as Backend:

Use **Sanity** for:

- **Managing product data:** Name, price, size, tags, category, stock, image
- **Storing order details:** orderId, customerId, productDetails, status, Timestamp.
- **Storing Customer Data:** Customer ID, Name, Contact Info, Address, Order history.
- **Managing Shipments:** Shipment Id, Order Id, Delivery Date, Status.
- **Managing Payments:** Payment Id, Order Id, Amount, method, status.

### 3. Third-Party APIs:

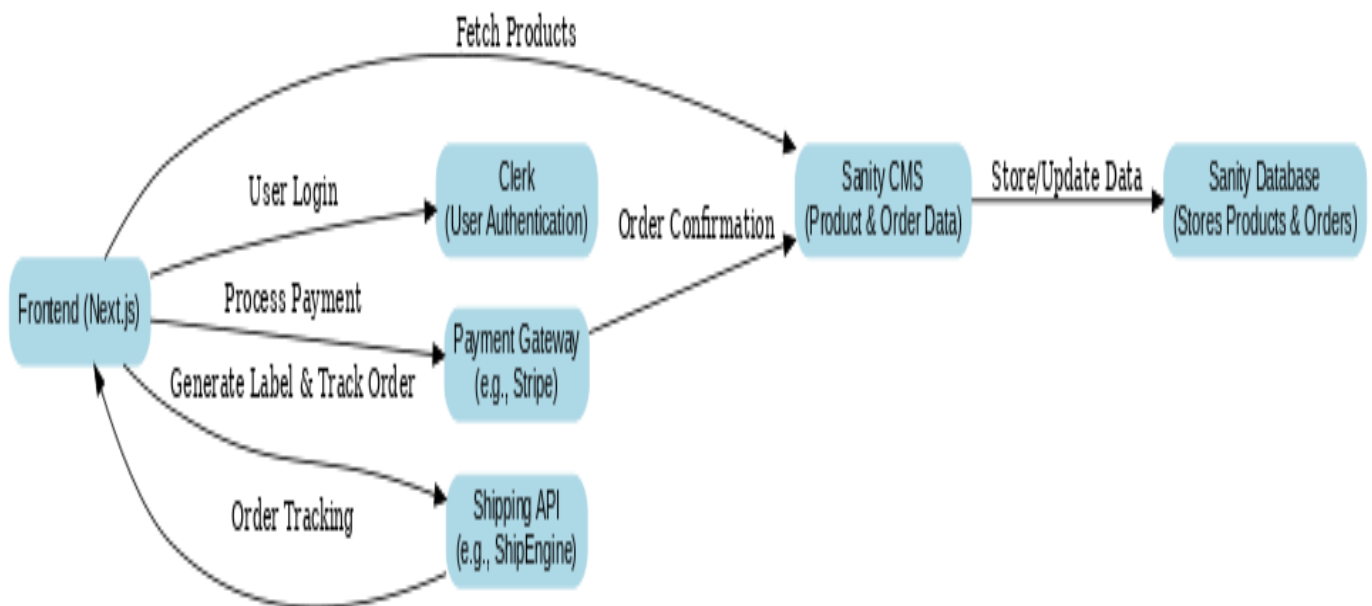
- **Shipment Tracking:**

Use **ShipEngine** API for shipment tracking.

- **Payment Gateway:**

Use **Stripe** and **PayPal** for secure payment processing.

### System Architecture



## Workflow:

### 1. User Authentication:

- User logs in via Clerk → Clerk authenticates the user → Access granted to the frontend.

### 2. Product Browsing:

- User browses products → Next.js fetches product data from Sanity CMS → Displays the product list.

### 3. Cart Management:

User adds products to the cart → Frontend sends cart data `/api/cart`  
Cart details are saved in Sanity.

### 4. Order Placement:

- User places an order → Order details are sent to Sanity CMS → Sanity stores the order.
- Stripe processes payment securely → Confirmation is sent to the user.

### 5. Shipping:

- Order data is sent to ShipEngine → Shipping label generated → Tracking details are stored in Sanity and shared with the user.