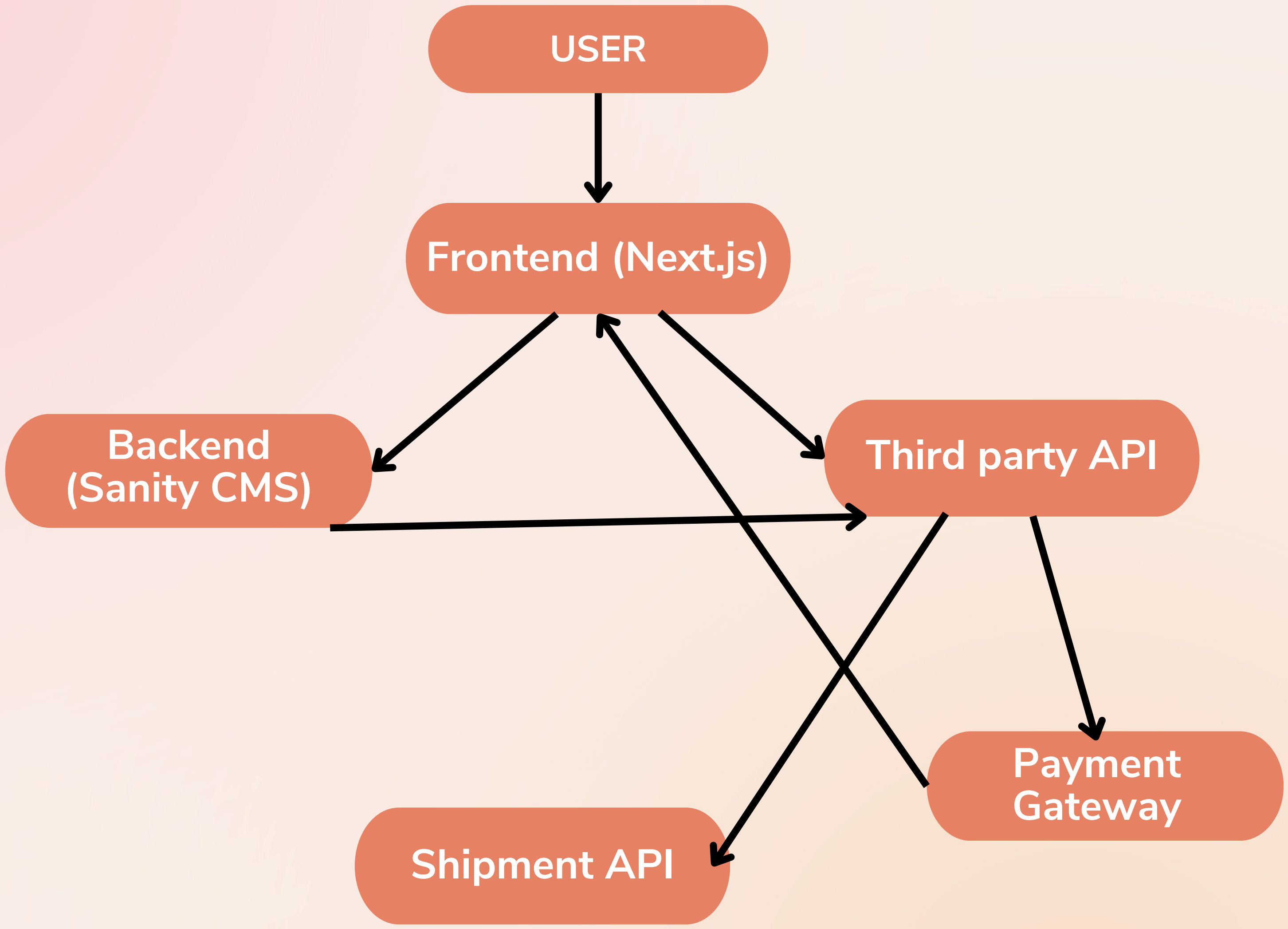
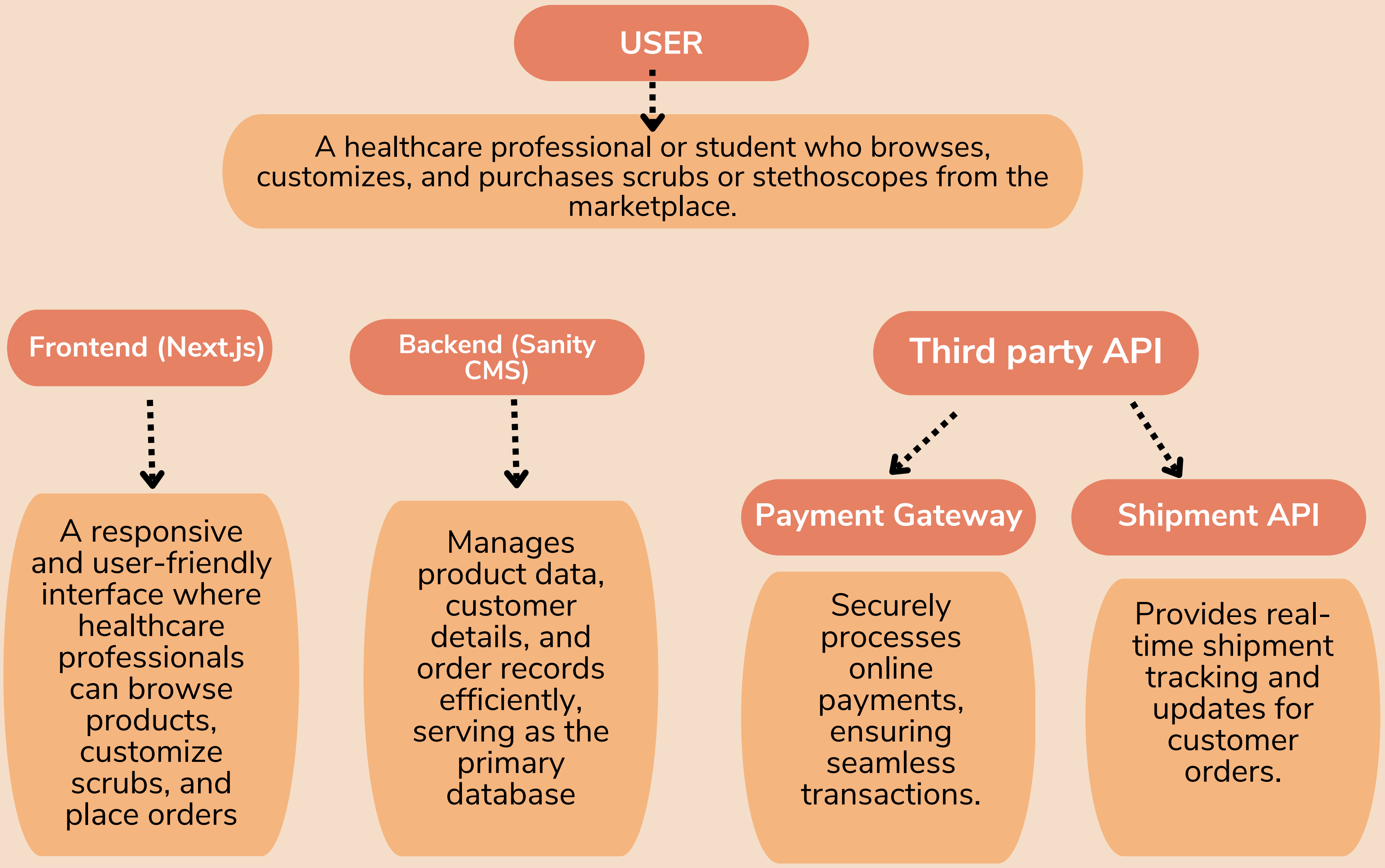
System Architecture




USER
A healthcare professional or student who browses, customizes, and purchases scrubs or stethoscopes from the marketplace.
Frontend (Next.js)
A responsive and user-friendly interface where healthcare professionals can browse products, customize scrubs, and place orders
Backend (Sanity CMS)
Manages product data, customer details, and order records efficiently, serving as the primary database
Third party API
Payment Gateway
Securely processes online payments, ensuring seamless transactions.
Shipment API
Provides real-time shipment tracking and updates for customer orders.

TECHNICAL REQUIREMENTS

Frontend Requirements

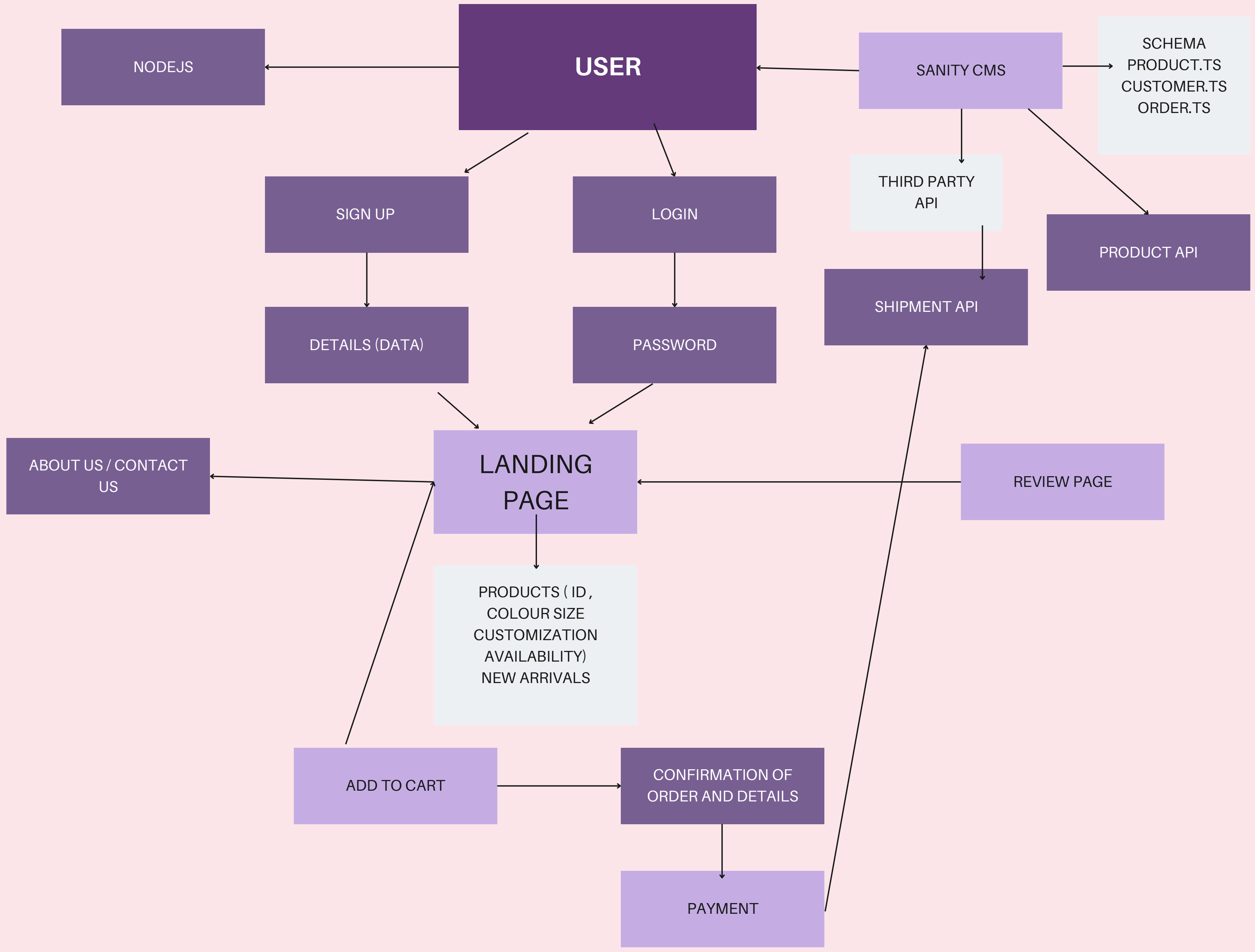
1. User Interface:
   - Clean, user-friendly design tailored for medical professionals and students.
   - Responsive and compatible with mobile, tablet, and desktop devices.
2. Key Pages:
   - Home: Highlights featured products and promotions.
   - Product Listing: Offers filters like size, color, and price.
   - Product Details: Displays product information, images, and customization options.
   - Cart: Summarizes selected items, quantities, and total price.
   - Checkout: Captures delivery address and payment details.
   - Order Confirmation: Shows order summary and estimated delivery.
3. Key Features:
   - Dynamic filtering and sorting of products.
   - Clear navigation with breadcrumb trails.
   - Real-time notifications for actions like adding to cart and order placement.
4. Technologies:
   - Frontend Framework: Next.js for optimized performance.
   - Styling: Tailwind CSS for efficient and responsive design.

Backend Requirements

1. Data Management:
   - Use Sanity CMS to handle product, order, and customer data effectively.
2. Core Functionalities:
   - Manage inventory, including real-time stock updates.
   - Store and retrieve order details securely.
   - Support user registration and login with encrypted credentials.
3. Security Measures:
   - Ensure data encryption for sensitive information.
   - Implement access controls for managing data securely.
4.

APIs

1. Product Management API:
   - Enables retrieval and management of product details such as name, price, stock, and category.
2. Order Management API:
   - Handles order creation, updating, and fetching order details.
3. Shipment Tracking API:
   - Integrates real-time shipment updates to track the delivery status of orders.
4. Payment Gateway API:
   - Ensures secure processing of online transactions, supporting various payment methods.


NODEJS
USER
SANITY CMS
SCHEMA
PRODUCT.TS
CUSTOMER.TS
ORDER.TS
SIGN UP
LOGIN
THIRD PARTY API
PRODUCT API
SHIPMENT API
DETAILS (DATA)
PASSWORD
ABOUT US / CONTACT US
LANDING PAGE
REVIEW PAGE
PRODUCTS (ID , COLOUR SIZE CUSTOMIZATION AVAILABILITY) NEW ARRIVALS
ADD TO CART
CONFIRMATION OF ORDER AND DETAILS
PAYMENT

APIS

PRODUCT DETAILS API
METHOD: GET
ENDPOINT: /API/PRODUCTS/{ID}
DESCRIPTION: RETRIEVES DETAILS OF A SPECIFIC PRODUCT BY ID.
EXAMPLE CODE:

```
FETCH('/API/PRODUCTS/SCRB123'
).THEN(RES => RES.JSON());
```

ORDER CREATION API
METHOD: POST
ENDPOINT: /API/ORDERS
DESCRIPTION: CREATES A NEW ORDER.
EXAMPLE CODE:

```
FETCH('/API/ORDERS', {
 METHOD: 'POST',
 BODY: JSON.STRINGIFY(ORDER)});
```

SHIPMENT TRACKING API
METHOD: GET
ENDPOINT: /API/SHIPMENTS/{ID}
DESCRIPTION: TRACKS THE DELIVERY STATUS OF AN ORDER.
EXAMPLE CODE:

```
FETCH('/API/SHIPMENTS/ORD123')
.THEN(RES => RES.JSON());
```

PAYMENT PROCESSING API
METHOD: POST
ENDPOINT: /API/PAYMENT
DESCRIPTION: PROCESSES A PAYMENT TRANSACTION SECURELY.
EXAMPLE CODE:

```
FETCH('/API/PAYMENT', {
```