# Technical Documentation

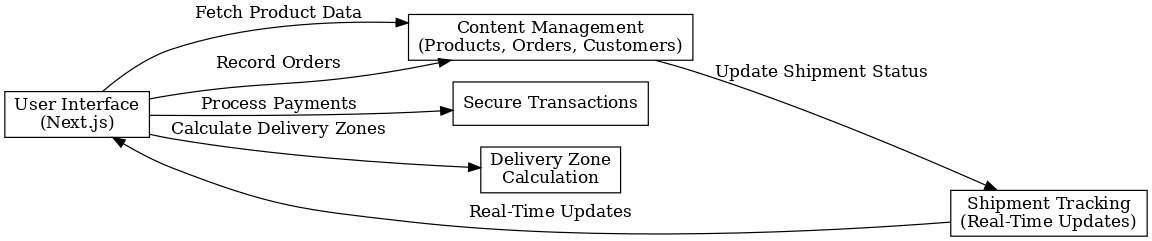
## 1. System Architecture

The system architecture showcases the interaction between the frontend, backend (Sanity CMS), and third-party APIs.

### Components:

- Frontend: Built with Next.js for a responsive and dynamic user interface.  
- Sanity CMS: Backend to manage product data, orders, and customer information.  
- Third-Party APIs:  
 - Shipment Tracking: Real-time delivery updates.  
 - Payment Gateway: Secure transaction processing.  
 - Geolocation: Delivery zone and time calculations.

### System Architecture Diagram:



## 2. Workflows

### Key Workflows:

1. Product Browsing:  
 - The user visits the platform.  
 - Frontend requests product data from Sanity CMS.  
 - Products are displayed dynamically on the Product Listing page.  
  
2. Order Placement:  
 - User adds products to the cart and proceeds to checkout.  
 - Order details are saved in Sanity CMS.  
 - Payments are securely processed using the Payment Gateway API.  
 - The user receives an order confirmation.  
  
3. Shipment Tracking:  
 - Shipment status is updated in Sanity CMS via the Shipment Tracking API.  
 - Users can view real-time shipment updates.  
  
4. Delivery Zone Calculation:  
 - User address is sent to the Geolocation API.  
 - API calculates the delivery zone and provides an estimated time.

## 3. Data Schema

Entities and Relationships:  
1. Products:  
 - Attributes: ProductID, Name, Price, Stock, Category, Description, Image URL.  
2. Customers:  
 - Attributes: CustomerID, Name, Email, Address, OrderHistory.  
3. Orders:  
 - Attributes: OrderID, CustomerID, ProductDetails, TotalAmount, OrderStatus.  
4. Shipment:  
 - Attributes: ShipmentID, OrderID, DeliveryStatus, ETA.  
5. Delivery Zone:  
 - Attributes: ZoneID, ZoneName, Distance, ETA.

## 4. API Endpoints

1. Fetch Products:  
 - Endpoint: /products  
 - Method: GET  
 - Response:  
 {  
 "id": 101,  
 "name": "Fresh Apples",  
 "price": 150,  
 "stock": 20,  
 "category": "Groceries",  
 "image": "apple.jpg"  
 }  
  
2. Create an Order:  
 - Endpoint: /orders  
 - Method: POST  
 - Payload:  
 {  
 "customerId": 1,  
 "products": [  
 { "productId": 101, "quantity": 2 },  
 { "productId": 102, "quantity": 1 }  
 ],  
 "totalAmount": 450  
 }  
 - Response:  
 { "orderId": 12345, "status": "Order Created" }  
  
3. Track Shipment:  
 - Endpoint: /shipment-status  
 - Method: GET  
 - Response:  
 {  
 "orderId": 12345,  
 "status": "In Transit",  
 "ETA": "15 mins"  
 }  
  
4. Process Payment:  
 - Endpoint: /process-payment  
 - Method: POST  
 - Payload:  
 {  
 "orderId": 12345,  
 "amount": 450,  
 "paymentMethod": "CreditCard"  
 }  
 - Response:  
 { "status": "Payment Successful", "transactionId": "TXN789456" }  
  
5. Calculate Delivery Zone:  
 - Endpoint: /calculate-distance  
 - Method: GET  
 - Response:  
 {  
 "distance": "5 km",  
 "ETA": "20 mins",  
 "zone": "Zone 1"  
 }