

2. System Architecture Overview

The system architecture consists of the following components:

1. **Frontend (Next.js)**: Handles user interactions and displays dynamic content.
2. **Sanity CMS**: Manages product data, customer details, and order records.
3. **Third-Party APIs**: Integrates payment processing and shipment tracking.
4. **Database (Sanity CMS)**: Stores all data (products, orders, customers, riders).

High-Level Diagram

```
plaintext
[Frontend (Next.js)]
|||
[Sanity CMS] [Third-Party APIs]
```

Detailed Workflow

Here's how the components interact in a real-world scenario:

1. User Registration

- **Frontend**: User signs up by entering their details (name, contact info, address).
- **Sanity CMS**: Stores the customer data in the **Customer schema**.
- **Response**: Confirmation message is sent to the user.

2. Product Browsing

- **Frontend**: User views product categories (e.g., Pizza, Beverage, Dessert).
- **Sanity CMS**: Fetches product data from the **Product schema**.
- **Frontend**: Displays the products dynamically on the website.

3. Order Placement

- **Frontend**: User adds items to the cart and proceeds to checkout.
- **Sanity CMS**: Saves the order details in the **Order schema**.
- **Third-Party API (Payment Gateway)**: Processes the payment.
- **Response**: Order confirmation is displayed to the user.

4. Shipment Tracking

- **Frontend**: User checks the status of their order.
- **Third-Party API (Shipment Tracking)**: Fetches real-time delivery updates.

- **Frontend:** Displays the shipment status (e.g., "In Transit", "Delivered").
-

Component Details

1. Frontend (Next.js)

- **Pages:**
 - **Home:** Displays featured products and promotions.
 - **Product Listing:** Shows all products with filters (e.g., category, price).
 - **Product Details:** Displays detailed information about a product.
 - **Cart:** Allows users to view and modify their cart.
 - **Checkout:** Collects delivery details and processes payments.
 - **Order Confirmation:** Displays order summary and tracking information.
- **Features:**
 - Responsive design for mobile and desktop.
 - Dynamic content fetched from Sanity CMS.
 - Integration with third-party APIs for payment and shipment tracking.

2. Sanity CMS

- **Schemas:**
 - **Product:** Manages product data (name, description, price, category, image, customization options, availability).
 - **Order:** Tracks customer orders (order ID, customer ID, products, total price, order status, delivery address, rider ID).
 - **Customer:** Stores customer information (name, contact info, address, order history, preferences).
 - **Rider:** Manages delivery personnel (name, contact info, vehicle type, availability status, delivery zone).
- **API:**
 - Provides endpoints to fetch and update data (e.g., /products, /orders, /customers, /riders).

3. Third-Party APIs

- **Payment Gateway:**
 - Processes payments securely (e.g., Stripe, PayPal).
 - Endpoint: /process-payment.
 - Payload: Order details (e.g., total price, customer ID).
 - Response: Payment status (e.g., "Success", "Failed").
- **Shipment Tracking:**
 - Provides real-time delivery updates (e.g., Google Maps API).
 - Endpoint: /track-shipment.
 - Payload: Order ID.
 - Response: Shipment status (e.g., "In Transit", "Delivered").

Key Workflows

1. User Registration

plaintext

[Frontend] → [Sanity CMS] → [Database]

1. User enters details (name, contact info, address).
2. Frontend sends data to Sanity CMS.
3. Sanity CMS stores data in the Customer schema.
4. Confirmation message is sent to the user.

2. Product Browsing

plaintext

[Frontend] → [Sanity CMS] → [Database]

1. User views product categories.
2. Frontend fetches product data from Sanity CMS.
3. Sanity CMS retrieves data from the Product schema.
4. Frontend displays products dynamically.

3. Order Placement

plaintext

[Frontend] → [Sanity CMS] → [Third-Party API] → [Database]

1. User adds items to the cart and proceeds to checkout.
2. Frontend sends order details to Sanity CMS.
3. Sanity CMS saves data in the Order schema.
4. Frontend sends payment details to the Payment Gateway API.
5. Payment Gateway processes the payment and sends a response.
6. Order confirmation is displayed to the user.

4. Shipment Tracking

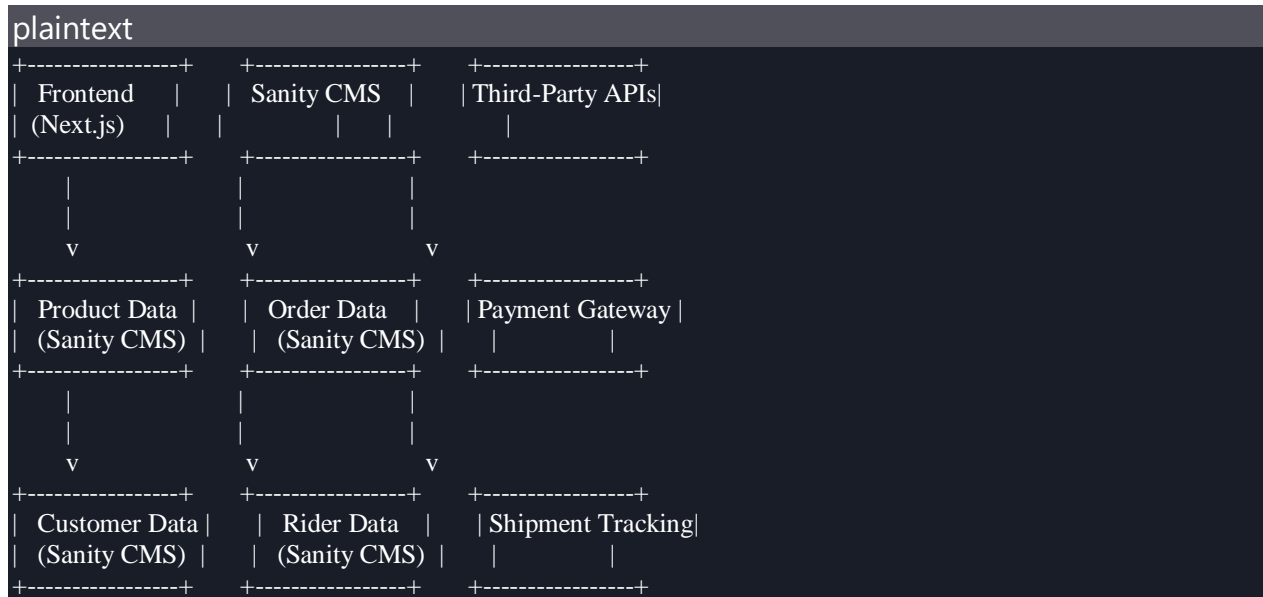
plaintext

[Frontend] → [Third-Party API] → [Database]

1. User checks the status of their order.
 2. Frontend sends a request to the Shipment Tracking API.
 3. Shipment Tracking API fetches real-time updates.
 4. Frontend displays the shipment status to the user.
-

System Architecture Diagram

Here's a **visual representation** of the system architecture:



Summary

- **Frontend (Next.js):** Handles user interactions and displays dynamic content.
- **Sanity CMS:** Manages product data, customer details, and order records.
- **Third-Party APIs:** Integrates payment processing and shipment tracking.
- **Workflows:** User registration, product browsing, order placement, and shipment tracking.