High-Level Diagram

[Frontend (Next.js)]

|

v

[Sanity CMS] <--------> [Product Data (Mock) API]

|

^

|

[Third-Party APIs] <----> [(ShipEngine) Shipment Tracking API]

|

|

v

[Payment Gateway (Stripe)]

|

v

[Authentication (Clerk)]

Component Descriptions

Frontend (Next.js)

- Develops a user-friendly interface for browsing products, managing orders, and logging in.
- Dynamically displays data by connecting to backend APIs.

Sanity CMS

- Manages product details, user information, orders, and inventory.
- Shares data with the frontend via APIs.

Third-Party APIs

- **Shipment Tracking API (ShipEngine):** Provides real-time shipping updates and tracking information.
- **Payment Gateway (Stripe):** Securely processes payments and confirms transactions.

Authentication (Clerk)

- Manages user sign-up, login, and session handling.
- Works with Sanity CMS to securely store user details.

Key Workflows

1. User Registration

- Users sign up via the frontend using Clerk.
- Registration details are securely stored in Sanity CMS.

2. Product Browsing

- Users explore product categories on the frontend.
- Sanity CMS API provides product details such as name, price, stock, description, and images.
- The frontend dynamically displays product listings.

3. Order Placement

- Users add products to their cart and proceed to checkout.
- Order details (products, quantities, shipping address) are sent to Sanity CMS.
- Payments are securely processed via Stripe.

4. Shipment Tracking

- Shipment details are updated using ShipEngine after an order is placed.
- Real-time tracking information is displayed to users on the frontend.

5. Inventory Management

- Stock levels are managed in Sanity CMS.
- Real-time updates ensure that only in-stock products can be added to the cart.

API Documentation

Endpoint	Method	Purpose	Response Example
/products	GET	Fetch all product details	[{ "name": "Product Name", "slug": "product-slug", "price": 100 }]
/order	POST	Submit new order details	{ "orderId": 123, "status": "success" }
/shipment-tracking	GET	Fetch real-time tracking updates	{ "trackingId": "AB123", "status": "In Transit" }
/delivery-status	GET	Fetch express delivery details	{ "orderId": 456, "deliveryTime": "30 mins" }
/inventory	GET	Fetch real-time stock levels	{ "productId": 789, "stock": 50 }
/cart	POST	Add product to cart	{ "cartId": 101, "items": [...] }