# Key Components

## 1. Frontend (Next.js)

- User Actions:
  - o User Signup/Login:
    - Users fill out the registration form.
    - Form data is validated on the frontend.
    - Validated data is sent to Sanity CMS via API.
    - A confirmation message is displayed to the user.
  - o Product Browsing:
    - Users can view product categories.
    - Categories and products are fetched from Sanity CMS using an API.
    - Fetched data is displayed dynamically on the frontend.
  - o Checkout and Order Placement:
    - Users fill in billing and shipping information.
    - Order details are validated and sent to Sanity CMS via API.
    - Order summary is displayed to the user.

## 2. Sanity CMS (Backend)

- Centralized Data Management:
  - o User Data: Stores user registration details.

- Product Data: Stores product categories and product details.
- Order Data: Stores all order-related details, including billing, shipping, and payment status.

- API Integration:
  - Handles incoming requests from the frontend to fetch or update data.
  - Serves as the bridge between frontend and third-party APIs.

## 3. Third-Party APIs

- Payment Processing:
  - Payment details are securely sent to the third-party payment gateway.
  - Payment confirmation or error response is returned.
  - Confirmation details are saved in Sanity CMS.
- Shipment Tracking:
  - Shipment details are fetched from the third-party logistics API.
  - Real-time tracking updates are displayed to the user.

## Detailed Workflow

1. Registration:
   - User fills out the signup form.
   - Frontend validates data and sends it to Sanity CMS.
   - Sanity stores the user data and returns a confirmation response.

2. Product Browsing:

   o Frontend sends a request to Sanity CMS to fetch product categories and details.

   o Fetched data is displayed as a dynamic product catalog on the frontend.

3. Order Placement:

   o User adds items to the cart and proceeds to checkout.

   o Billing and shipping information is submitted to Sanity CMS.

   o Order summary is generated and displayed.

4. Payment:

   o Payment details are sent to the third-party payment gateway.

   o Payment confirmation is saved in Sanity CMS.

5. Shipment Tracking:

   o Shipment status is fetched from the third-party logistics API.

   o Real-time tracking updates are displayed to the user on the order page.

## System Architecture

1. Frontend (Next.js)

- Handles user interactions and displays data dynamically.
- Fetches and sends data to/from Sanity CMS and third-party APIs.

2. Sanity CMS

- Centralized storage for all application data.
- Acts as a middleware to interact with external APIs.

3. Third-Party APIs

- Responsible for specialized tasks:
  - o Payment gateway API: Handles secure payment transactions.
  - o Logistics API: Provides real-time shipment tracking.

# API Requirements

## Example API Endpoints:

1. User Registration:
   - o Endpoint: /API/register
   - o Method: POST
   - o Description: Saves user data to Sanity CMS.
2. Fetch Products:
   - o Endpoint: / API /products
   - o Method: GET
   - o Description: Retrieves product categories and details from Sanity CMS.
3. Order Placement:
   - o Endpoint: / API /order

- Method: POST
- Description: Saves order details to Sanity CMS.

4. Payment Confirmation:
   - Endpoint: / API /payment
   - Method: POST
   - Description: Sends payment details to the payment gateway and saves confirmation in Sanity CMS.
5. Shipment Tracking:
   - Endpoint: / API /shipment
   - Method: GET
   - Description: Fetches shipment tracking details from a third-party API.