

E-Commerce System Architecture Documentation

1. Introduction

Purpose: To provide a high-level overview of the architecture and design of the eCommerce system, detailing how components interact to deliver seamless functionality.

Scope: Focuses on the interaction between the frontend, CMS, APIs, and external services (payment and shipment) to enable smooth workflows.

2. Architecture Diagram

Components and Flow:

1. **User Interaction:** Users interact with the platform via the frontend, built on Next.js, ensuring a responsive, server-rendered experience.
2. **Frontend Communication:** Fetches product and order data from Sanity CMS, sends orders, and queries shipment status through custom API endpoints. Processes payments securely via a third-party payment gateway.
3. **API Gateway:** Acts as a bridge between the frontend and the backend.
4. **Payment Gateway:** Ensures secure payment transactions.
5. **Shipment Tracking API:** Provides real-time shipment updates to users.

3. System Component Responsibilities

- **Frontend (Next.js):**

Handles user interactions such as browsing, cart updates, and payment confirmations.

- **Sanity CMS:**

Stores and manages data such as products, orders, and customer information.

- **API Gateway:**

Provides key endpoints like `/products`, `/products/:id`, `/orders`, and `/shipment/:id`.

- **Payment Gateway:**

Processes payments securely and provides transaction confirmation.

- Shipment Tracking API:

Integrates with logistics providers for real-time shipment tracking.

4. Data Flow

1. Browsing Products: Users browse products -> Frontend fetches data from /products.
2. Product Details: Users view product details -> Frontend queries /products/:id.
3. Cart & Checkout: Users add products to the cart and proceed to checkout -> Order data is sent to /orders -> Payment processed by the payment gateway.
4. Shipment Tracking: Users track their orders -> Frontend fetches shipment status via /shipment/:id.

5. API Endpoints

```
{  
  "endpoints": [  
    {  
      "endpoint": "/products",  
      "method": "GET",  
      "description": "Fetch all products",  
      "requestPayload": null,  
      "responseExample": {  
        "id": 1,  
        "name": "Kurta",  
        "price": 2500  
      }  
    },  
    {  
      "endpoint": "/products/:id",
```

```
"method": "GET",

"description": "Fetch product details",

"requestPayload": null,

"responseExample": {

  "id": 1,

  "name": "Kurta",

  "price": 2500,

  "stock": 10

},

{

  "endpoint": "/orders",

  "method": "POST",

  "description": "Submit a new order",

  "requestPayload": {

    "customer": "Ali",

    "items": [

      {

        "id": 1,

        "name": "Kurta",

        "quantity": 2

      }

    ],

    "total": 5000

  },

  "responseExample": {

    "status": "success",
```

"orderId": 123

}

}

]

}