

# Data System Architecture Documentation Based on Requirements:

## 1. Frontend Requirements:

- **Update the Browser Section:**
  - Add specific frontend modules like:
    - **Home Page**
    - **Product Listing Page**
    - **Product Details Page**
    - **Cart Page**
    - **Checkout Page**
    - **Order Confirmation Page**
  - Clearly differentiate the admin panel (for CMS management) and the customer storefront UI.
- **Responsive Design:**
  - Indicate that both the Admin and Storefront gateways will be optimized for mobile and desktop.

## 2. Sanity CMS as Backend:

- Replace **Admin Gateway** with **Sanity CMS Gateway**:
  - Show how Sanity CMS interacts with the Catalog, Customer, and Order services by managing the product data, customer details, and order records.
  - Label the **Catalog Service API** to reflect that Sanity CMS is used to fetch product data.
- Add a **Schema Design Box**:
  - Highlight that schemas in Sanity are designed to:
    - Handle product categories, variants, and inventory.
    - Track customer profiles and their orders.
    - Support order records, promotions, and shipping details.

## 3. Third-Party APIs:

- In the **Integrations Section**:
  - Add modules for:

- **Payment Gateway API** (e.g., Stripe, PayPal).
- **Shipment Tracking API.**
- Highlight these integrations connecting to the **Checkout Service** and **Order Service** APIs.
- Modify **Checkout Service**:
  - Label it to include "Payment Integration" and "Cart Management."
- Modify **Order Service**:
  - Include "Shipment Tracking" in its responsibilities.

#### ***4. Database Layer:***

- Indicate that Sanity CMS acts as the primary database for the e-commerce system:
  - Highlight **Product Data**, **Customer Data**, and **Order Records** stored and managed by Sanity.

#### ***5. Aggregation Section:***

- Ensure the aggregation layer consolidates data from the CMS, APIs, and services to provide:
  - Reports for admin users.
  - Insights for frontend dashboards.