HAMMAD ALI

Kolachi.pk Ecommerce Technical Foundation

HAMMAD ALI

Frontend

Technologies :

Next.js for server-side rendering and faster page loads.

Tailwind CSS for styling and responsive design.

Key Features :

- Build a user-friendly interface for browsing products and placing orders.

- Design responsive layouts that work on both mobile and desktop devices.

- Implement core pages:

  Home , Product Listing , Product Details ,

  Cart and Checkout , Order Confirmation

Sanity CMS as Backend

Purpose :

- Use Sanity CMS to manage product data, customer records, and order information.
- Define structured content schemas for:
- Product Details (ID, Name, Price, Stock, Images).
- Orders (Order ID, Customer ID, Product IDs, Quantity, Status).
- Customers (Customer ID, Contact Info, and Purchase History).
- Real-time updates to reflect stock changes and order processing.

HAMMAD ALI

Third-Party APIs

ShipEngine :

For shipment tracking and logistics.

Enables assigning delivery riders and tracking shipment zones.

Payment Gateway:

Support Cash on Delivery and Digital Payments (Stripe, PayPal, or

JazzCash).

Notification APIs :

Use services like Twilio or Firebase for SMS/Email

notifications about order status.

API Requirements

1.) Product Management API

Endpoint Name: /products

Method : GET

Description : Fetch all available products from the Sanity CMS.

Response Example:

```
[{ "id": 1, "name": "Product A", "price": 100 },
 { "id": 2, "name": "Product B", "price": 200 }]
```

API Requirements

2. Order Management API

Endpoint Name: /orders

Method : POST

Description : Create a new order in the system

Response Example:

{ "order_id": 456, "status": "Success",

"message": "Order created successfully" }

API Requirements

3. Shipment Management API

Endpoint Name: /shipment

Method : GET

Description : Track the order status via a third-party API.

Response Example:

{"shipment_id": 789,"order_id": 456,"status": "In Transit"

,"expected_delivery_date": "2025-01-20"}

API Requirements

4. Express Delivery Status API

Endpoint Name: /express-delivery-status

Method : GET

Description : Fetch real-time delivery updates for perishable items.

Response Example:

```
{"order_id": 123,"status": "In Transit",
"ETA": "45 minutes"}
```
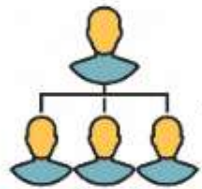
Users

Frontend (Next.js)

Sanity as Backend

SANITY

Order Data

APIs

Shipment and tracking

Delivery

Payment Methods

Marketplace Technical Foundation for Kolachi.pk

1. System Architecture Overview

This section provides an overview of how the different components of your e-commerce platform interact.

Frontend:

- **Technology:** Next.js

- **Features:**

  - Responsive and user-friendly design.

  - Core pages:

    - **Home:** Show featured products and categories.

    - **Product Listing:** Display all products in a category.

    - **Product Details:** Details of a specific product with reviews and ratings.

    - **Cart and Checkout:** Add/remove products, view total price, and finalize orders.

    - **Order Confirmation:** Display order success and tracking information.

Backend:

- **Technology:** Sanity CMS

- **Purpose:** Manage all content and data:

  - Products, orders, customers, and shipment details.

Third-Party APIs:

- **ShipEngine:** Track shipments, assign riders, and fetch delivery zone coverage.

- **Payment Gateway:** JazzCash/Stripe for secure payments.

- **Notification API:** Twilio/Firebase for SMS/email updates about orders.

Product Schema

- **ID:** Unique identifier for the product

- **Name:** Name of the product

- **Price:** Price of the product

- **Stock:** Number of items available in stock

- **Images:** Array of image URLs

Order Schema

- **Order ID:** Unique identifier for the order

- **Customer ID:** ID of the customer who placed the order

- **Products:**

  - **Product ID:** ID of the product

  - **Quantity:** Quantity ordered

- **Status:** Current status of the order (e.g., Pending, Shipped, Delivered)

- **Payment Status:** Status of payment (e.g., Paid, Unpaid)

- **Delivery Address:** Address for order delivery

Customer Schema

- **Customer ID:** Unique identifier for the customer

- **Name:** Name of the customer

- **Contact Info:** Contact information of the customer (e.g., phone, email)

- **Purchase History:** Array of order IDs representing the purchase history

Shipment Schema

- **Shipment ID:** Unique identifier for the shipment

- **Order ID:** Associated order ID

- **Zone Name:** Name of the delivery zone

- **Coverage Area:** Area covered by the delivery zone

- **Status:** Current status of the shipment (e.g., In Transit, Delivered)

- **Assigned Rider:** Name of the rider assigned for delivery

Delivery Schema

- **Delivery ID:** Unique identifier for the delivery

- **Order ID:** Associated order ID

- **Assigned Rider:** Name or ID of the rider responsible for delivery

- **Delivery Status:** Current status of the delivery (e.g., Dispatched, Out for Delivery, Delivered)

- **Expected Delivery Time:** Estimated delivery time or date

- **Actual Delivery Time:** Timestamp of when the delivery was completed

- **Delivery Notes:** Optional notes about the delivery (e.g., "Left at the doorstep")

4. Key Backend Features

1. Order Management:

   - Handle order creation, status updates, and cancellations.
   - Payment integration for COD and online payments.

2. Inventory Management:

   - Update stock levels based on orders.
   - Alerts for low stock levels.

3. Returns and Refunds:

   - Simplify returns with easy-to-use workflows.
   - Automate refund processes.

4. Customer Reviews:

   - Enable uploading of photos and real feedback.

---

5. Security

- **Authentication:** Use JWT for login/signup processes.
- **Encryption:** HTTPS for secure communication.
- **Validation:** Sanitize inputs to prevent SQL injection or XSS attacks.

---

6. Deployment and Monitoring

1. Hosting:

   - Deploy frontend on **Vercel.**
   - Host backend (Sanity CMS) on **Sanity.io.**

2. Monitoring Tools:

   - Use **LogRocket** or **Sentry** for performance monitoring.
   - Enable **Google Analytics** to track user engagement.