DAY 2

MARKETPLACE TECHNICAL FOUNDATION FOR Living Luxe E-COMMERCE

Frontend

Next.js + TailwindCss

Responsive Design:

- Mobile-first, responsive layout.
- Implement a mobile-first approach using Tailwind CSS utilities.

Essential Pages:

- **Home Page:** Displays Best products and new arrivals.
- **Product Listing Page:** Displays all products with filters and search functionality.
- **Product Details Page:** Detailed information and add-to-cart functionality.
- **Cart Page:** Summary of selected items with editable quantities.

- **Checkout Page:** User input for shipping and payment details.
- **Order Confirmation:** Displays order details and a tracking link.

Backend

Sanity CMS

Schema Design:

- **Product Fields :** ID, name, description, price, stock, images, and categories.
- **OrdersFields :** customer details, order status, payment status, delivery zone, and products ordered.
- **CustomersFields:** ID, name, email, phone, and shipping address.

Custom Queries:

- Enable dynamic filtering, sorting, and searching capabilities.

Admin Control:

- Creating, updating, and deleting products, managing orders, and viewing customer data.

Integration with Frontend:

- Use Sanity's GROQ (Graph-Relational Object Queries) for fetching and displaying content dynamically on the frontend.

Non-Functional:

1. Performance

- Ensure fast page load times (aim for <3 seconds) using image optimization and lazy loading.

2. Security

- Use HTTPS for all connections.
- Implement user authentication for admin access to Sanity CMS.
- Encrypt sensitive customer data, especially during payments.

3. Scalability

- Design the system to accommodate future product expansions and increased user traffic.

4. Reliability

- Maintain consistent uptime through robust hosting solutions like Vercel

Third-Party APIs

1. **ShipEngine:**

    - Integration with logistics APIs to track orders in real-time.
    - Display delivery status updates on the Order Confirmation and Order Tracking pages.

2. **Payment Gateway API:**

    - Use Stripe or PayPal for processing secure online transactions.

    Features:

    - Payment confirmation.
    - Support for multiple payment methods (credit/debit cards).

Key Workflows

User Registration

1. Users sign up via the frontend.
2. Data is sent to the Sanity CMS API and stored in the `user` collection.
3. Confirmation email is sent to the user using an email API.

Product Browsing

1. User navigates product categories on the frontend.
2. Sanity CMS fetches product data via the `/products` endpoint.
3. Data is displayed dynamically with sorting and filtering options

   .

Order Placement

1. User adds products to the cart and proceeds to checkout.
2. Order details (user info, product details, total, and payment status) are sent to the `/orders` API.
3. Payment is processed via the payment gateway.
4. Sanity CMS updates the order status to "Paid."

Shipment Tracking

1. User accesses the shipment tracking page.
2. Frontend calls the `/shipment` API with the tracking number.
3. Shipment tracking data is fetched from a third-party API and displayed to the user.

Sanity Schemas:

Product Schema

- **`id`**: Unique identifier for the product.
- **`name`**: Name of the product.
- **`price`**: Price of the product.
- **`stock`**: Stock availability.
- **`image`**: Product image URL.

- **`category`**: Category the product belongs to.

Order Schema

- **Order ID**: Unique identifier for each order.
- **Customer ID**: Link to the customer who placed the order.
- **Product IDs**: List of products included in the order.
- **Quantity**: The quantity of each product ordered.
- **Total Price**: The total cost of the order.
- **Order Date**: Date and time when the order was placed.
- **Order Status**: The current status of the order (e.g., pending, shipped, delivered).

Customers:

- **Customer ID:** Unique identifier for each customer.
- **Name:** The full name of the customer.
- **Email:** Customer’s email address for communication and order updates.
- **Phone Number:** Customer’s contact number.
- **Shipping Address:** The address to deliver the products.
- **Order History:** List of past orders made by the customer.

Delivery Zones:

- **Zone Name:** Name of the delivery zone (e.g., Central, North, South).
- **Coverage Area:** The geographical area that the zone covers (e.g., city, region).
- **Delivery Fees:** The cost for delivery within this zone.
- **Delivery Time Slots:** Available time frames for delivery in the zone.

Shipment Schema:

- **Shipment ID:** Unique identifier for the shipment.
- **Order ID:** Links the shipment to the corresponding order for tracking.
- **Delivery Method:** Details of the delivery service (e.g., standard, express).
- **Delivery Address**: The shipping address provided by the customer.
- **Shipment Status:** Real-time updates (e.g., processing, in transit, delivered).
- **Tracking Number**: Provided by ShipEngine for order tracking.
- **Payment Status**: Ensures payment confirmation before shipment is initiated.

API Endpoints:

1. Product

- **Endpoint Name**: /products
- **Method**: GET
- **Description**: Fetches all product details available in the marketplace.
- **Response Example**:

```
    [
    { "id": 1, "name": "Sofa A", "price": 999,
     "stock": 10, "image": "url" },


  { "id": 2, "name": "Sofa B", "price": 799,
    "stock": 5, "image": "url" }
                                  ]
```

2. Order

- **Endpoint**: /orders
- **Method**: POST
- **Purpose**: Creates a new order

- **Response Example**:
  ```
  { "orderId": 12345, "status": "Order Placed",
  "estimatedDelivery": "2025-01-20" }
  ```

3. Shipment

- **Endpoint:** `/shipment`
- **Method:** `GET`
- **Purpose: Tracks shipment status**
- **Response Example:**
  ```
  { "shipmentId": "SHIP123", "orderId": 12345, "status":
  "In Transit", "expectedDelivery": "2025-01-20" }
  ```