

DAY 2 HACKATHON

DEFINE TECHNICAL REQUIREMENT

FRONT END REQUIREMENT :

WE HAVE CREATED VERY USER FRIENDLY INTERFACE FOR USER WHERE THEY CAN EASILY BUY AND VIEW PRODUCTS. WE HAVE CREATED THE SIDEBAR WHERE THEY CAN SEE THE CATEGORY OF PRODUCT. THEY CAN SELECT THE CATEGORY ACCORDING TO THEIR CHOICE. WE HAVE CREATED THE SEARCH BAR WHERE THEY CAN FIND THE PRODUCT THAT THEY WANT. WE HAVE CREATED THE DIFFERENT SECTION LIKE FLASH SALE, BEST SELLING PRODUCT, LIMITED TIME OFFER. THESE THINGS HELP USER TO BUY PRODUCT.

RESPONSIVENESS OF THE WEBSITE IS UP TO THE MARK. THE USER CAN VIEW OUR WEBSITE IN ANY KIND OF DEVICE WITH BEST QUALITY. WE HAVE ENSURED THAT USER CAN VIEW OUR WEBSITE IN MOBILE OR IN DESKTOP WITHOUT ANY DIFFICULTY.

ESSENTIAL PAGES:

HOME PAGE: WE HAVE CREATED THE HOME PAGE WHERE USER CAN SEE ALL KIND OF PRODUCT AND SECTION LIKE FLASH SALE, BEST SELLING PRODUCT, LIMITED TIME OFFER AND ALL KIND OF CATEGORY

PRODUCTS PAGES: WE HAVE CREATED DIFFERENT PAGES WHERE PRODUCT IS LIST. WE HAVE CREATED SEPARATE PAGE FOR MEN, WOMEN, ACCESSORIES, SALE. THESE PAGES HELP USER TO VISIT ONLY THOSE PAGES WHICH PRODUCT THEY WANT TO BUY.

PRODUCT DETAIL PAGE: WE HAVE CREATED INDIVIDUAL PRODUCT DETAIL PAGE FOR EVERY SINGLE PRODUCT. WHEN USER CLICK ON THE PRODUCT THE NEW PAGE OPEN WHERE INFORMATION REGARDING PRODUCT IS WRITTEN. THE USER ALSO SEE THE REVIEWS OF CUSTOMER.

CART PAGE: WE HAVE CREATED THE CART PAGE WHERE USER CAN SELECT THE PRODUCT WHICH THEY WANT TO BUY. THEY CAN INCREASE AND DECREASE THE QUANTITY THEIR AND SEE THE

TOTAL AMOUNT AND DISCOUNT. THEY CAN ALSO REMOVE THE SELECTED PRODUCT WHICH THEY DON'T WANT TO BUY.

CHECKOUT PAGE:WE HAVE CREATED A CHECK OUT PAGE WHERE USER CAN SEE THE DETAIL ABOUT THE AMOUNT.IN THESE WE HAVE ADDED THE ITEM AMOUNT, SHIPPING PRICE ,DISCOUNT PRICE,TOTAL PRICE AND TAX .IN THIS PAGE WE HAVE ALSO ADDED THE PAYMENT METHOD SO THAT USER CAN ADD THE PAYMENT METHOD ACCORDING TO THEIR CHOICE.

ORDER CONFIRMATION: WE HAVE HAVE CREATED THE CONFIRMATION ORDER PAGE WHERE USER CAN ADD THE INFORMATION ABOUT THE ADDRESS AND CONFIRM THE ORDER.AFTER THAT HE CAN CHECK THE SHIPPING DETAIL AND TRACK WHERE THE ORDER IS.

SANITY CMS AS BACKEND:

PRODUCT DATA MANAGEMENT:SANITY WILL STORE ALL THE DETAILS ABOUT THE NIKE SHOES FOR SALE. THIS INCLUDES THINGS LIKE PRODUCT NAMES, DESCRIPTIONS, PRICES, SIZES, COLORS, AND HIGH-QUALITY IMAGES. WE WILL CREATE SPECIAL SYSTEMS (SCHEMAS) THAT ORGANIZE THIS INFORMATION, MAKING IT EASY TO UPDATE AND FIND WHEN NEEDED.

CUSTOMER DETAILS:SANITY WILL ALSO KEEP TRACK OF CUSTOMER INFORMATION, SUCH AS NAMES, EMAIL ADDRESSES, AND THEIR ORDER HISTORY. THIS HELPS US GIVE CUSTOMERS A MORE PERSONAL SHOPPING EXPERIENCE AND MANAGE THEIR ACCOUNTS WELL.

ORDER RECORDS:EVERY TIME A CUSTOMER MAKES A PURCHASE, SANITY WILL RECORD THE ORDER DETAILS. THIS WILL HELP US TRACK SALES, MANAGE STOCK, AND ANSWER CUSTOMER QUESTIONS ABOUT THEIR ORDERS. IT WILL ALSO ENSURE THAT CUSTOMERS RECEIVE ACCURATE ORDER UPDATES AND CONFIRMATIONS.

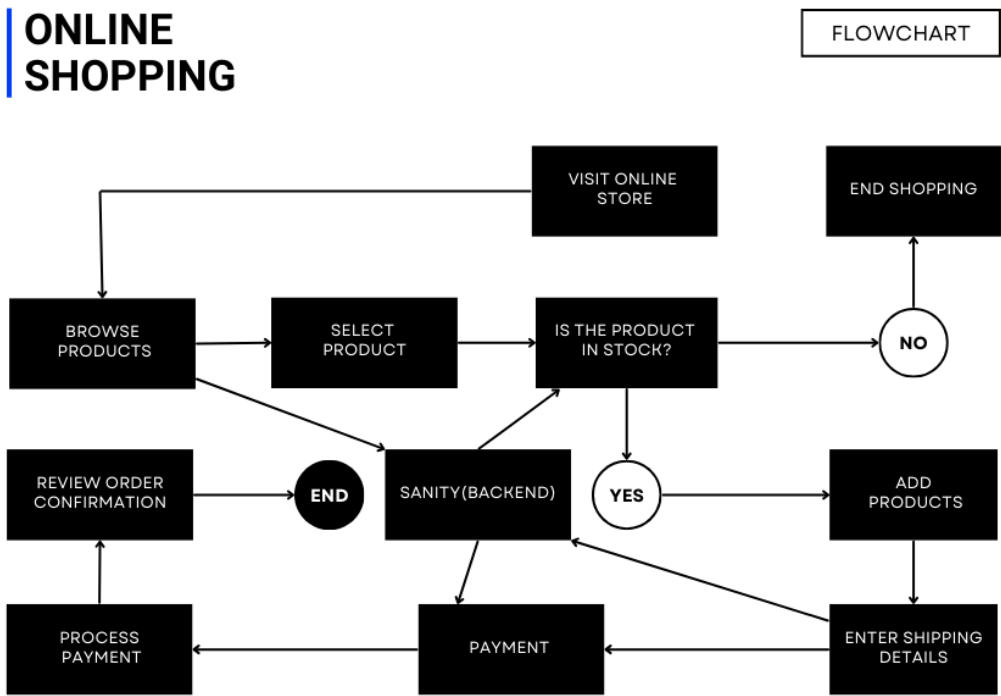
WE HAVE CREATED THE SCHEMA ACCORDING TO OUR PROJECT IN THIS ALL THE THING LIKE PRODUCTS DATA , CUSTOMER DATA ,ORDER DATA ,SHIPMENT DATA, DELIVERY DATA

THIRD PARTY API:

SHIPMENT TRACKING APIS: THESE APIS WILL LET CUSTOMERS TRACK THEIR ORDERS IN REAL-TIME. THEY CAN SEE THE SHIPPING STATUS, DELIVERY DATE, AND TRACKING NUMBER ON THE ORDER CONFIRMATION PAGE AND IN THEIR ACCOUNTS.

PAYMENT GATEWAY APIS:WE WILL USE SECURE PAYMENT APIS SO CUSTOMERS CAN PAY EASILY WITH CREDIT/DEBIT CARDS OR DIGITAL WALLETS. THIS ENSURES SAFE AND HASSLE-FREE TRANSACTIONS.

2. Design System Architecture



PLAN APIS REQUIREMENT

Fetch All Products

- **Endpoint Name:** `/products`
- **Method:** GET
- **Description:** GET a list of all available products from the Sanity CMS.

Response Example:

```
[
  {
    "id": 1,
    "name": "Product A",
    "price": 100,
    "stock": 20,
    "image": "https://example.com/product-a.jpg"
  },
  {
    "id": 2,
    "name": "Product B",
    "price": 150,
    "stock": 15,
    "image": "https://example.com/product-b.jpg"
  }
]
```

Create a New Order

Endpoint Name: `/orders`

Method: POST

Description: Add a new order to the Sanity CMS, including customer information, product details, and payment status.

Payload Example:

```
{
  "customer": {
    "name": "John Doe",
    "email": "john.doe@example.com"
  },
  "products": [
    { "id": 1, "quantity": 2 },
    { "id": 2, "quantity": 1 }
  ],
  "paymentStatus": "Paid"
}
```

```
}
```

Response Example:

```
{  
  "orderId": 12345,  
  "status": "Order Placed",  
  "totalAmount": 350  
}
```

Track Order Shipment

Endpoint Name: `/shipment`

Method: GET

Description: Fetch shipment tracking details for an order via a third-party API.

Response Example:

```
{  
  "shipmentId": "SH123456",  
  "orderId": 12345,  
  "status": "In Transit",  
  "expectedDelivery": "2025-01-20"  
}
```

4. Write Technical Documentation

Introduction

This document outlines the technical requirements and API endpoints for the e-commerce marketplace, which includes the following components:

1. **Frontend:** A user-friendly interface built to help users browse, purchase, and track products easily.

2. **Backend (Sanity CMS):** A content management system used for managing product data, customer details, and order records.
 3. **Third-Party APIs:** APIs for shipment tracking and payment processing.
-

System Architecture Overview

Frontend Requirements

- **User Interface:** Simple and intuitive design with a sidebar for product categories, search bar, and sections like Flash Sale, Best Selling Products, and Limited Time Offers.
- **Responsiveness:** The website is optimized for all devices, providing an excellent experience on mobile and desktop.
- **Essential Pages:**
 - **Home Page:** Displays product categories and sections (Flash Sale, Best Sellers, etc.).
 - **Products Pages:** Separate pages for Men, Women, Accessories, and Sale items.
 - **Product Detail Page:** Displays detailed information, reviews, and images of individual products.
 - **Cart Page:** Allows users to select quantities, view total amounts, and make changes to their cart.
 - **Checkout Page:** Displays item amounts, shipping price, discounts, total price, and payment methods.
 - **Order Confirmation Page:** Allows users to input their address and confirm orders.

Sanity CMS as Backend

- **Product Data Management:** Manages all product information, including names, descriptions, prices, sizes, colors, and images.
- **Customer Details:** Stores customer information such as names, emails, and order history for personalized shopping.
- **Order Records:** Tracks orders and ensures that stock and order details are accurately managed.
- **Schemas:** Defined for product data, customer data, order data, shipment data, and delivery data.

Third-Party API Integrations

- **Shipment Tracking API:** Provides real-time shipment tracking details such as status, expected delivery date, and tracking number.
 - **Payment Gateway API:** Facilitates secure transactions through credit/debit cards and digital wallets.
-

API Requirements and Endpoints

1. Fetch All Products

- **Endpoint Name:** `/products`
- **Method:** GET
- **Description:** Fetch a list of all available products from Sanity CMS.

Response Example:

```
[  
  {  
    "id": 1,  
    "name": "Product A",  
    "price": 100,  
    "stock": 20,  
    "image": "https://example.com/product-a.jpg"  
  },  
  {  
    "id": 2,  
    "name": "Product B",  
    "price": 150,  
    "stock": 15,  
    "image": "https://example.com/product-b.jpg"  
  }  
]
```

2. Create a New Order

- **Endpoint Name:** `/orders`
- **Method:** POST
- **Description:** Add a new order to Sanity CMS, including customer information, product details, and payment status.

Payload Example:

```
{
  "customer": {
    "name": "John Doe",
    "email": "john.doe@example.com"
  },
  "products": [
    { "id": 1, "quantity": 2 },
    { "id": 2, "quantity": 1 }
  ],
  "paymentStatus": "Paid"
}
```

Response Example:

```
{
  "orderId": 12345,
  "status": "Order Placed",
  "totalAmount": 350
}
```

-

3. Track Order Shipment

Endpoint Name: `/shipment`

Method: GET

Description: Fetch shipment tracking details for an order via a third-party API.

Response Example:

```
{  
  
  "shipmentId": "SH123456",  
  
  "orderId": 12345,  
  
  "status": "In Transit",  
  
  "expectedDelivery": "2025-01-20"  
}
```

Data Flow and Interactions

- 1. Fetching Products:**
 - The frontend makes a GET request to the `/products` endpoint to fetch product data from Sanity CMS.
 - The data is returned in JSON format, which is then displayed on the frontend.
- 2. Placing Orders:**
 - A customer adds products to their cart and proceeds to checkout.
 - The frontend sends a POST request to the `/orders` endpoint with customer details, selected products, and payment status.
 - The order is saved in Sanity CMS, and a response is returned with the order ID and total amount.
- 3. Tracking Shipments:**
 - Once an order is placed, the frontend makes a GET request to the `/shipment` endpoint to track the order's shipment status via a third-party API.
 - The shipment tracking information is returned and displayed to the user in real-time.