HACKATHON DAY 2

MARKETPLACE INTRODUCTION:

Our marketplace will serve as a comprehensive platform for a modern clothing brand, offering a seamless and intuitive shopping experience. It will showcase a curated selection of stylish and high-quality apparel designed to cater to diverse tastes and preferences. this marketplace will align with the goals of Hackathon 2 to create a scalable and efficient e-commerce solution.

1. Define Technical Requirements:

Frontend Requirements:

- User-friendly interface for browsing products.
- Responsive design for mobile and desktop users.
- Essential pages:
- Shop Page: Displays all products with sorting and filtering options.
- About Page: Highlights the brand's mission, vision, and story.
- Listing Page: Displays categorized products or specific product sets.
- Cart Page: Shows selected products with options to adjust quantity or remove items.
- Product Detail Page: Provides in-depth details of individual products, such as descriptions, images, and reviews

  .

Tools and Frameworks:

- Tailwind CSS: For building responsive and visually appealing UI components quickly.

- Next.js: For creating server-side rendered (SSR) and static site-generated (SSG) pages, ensuring fast load times and SEO optimization.

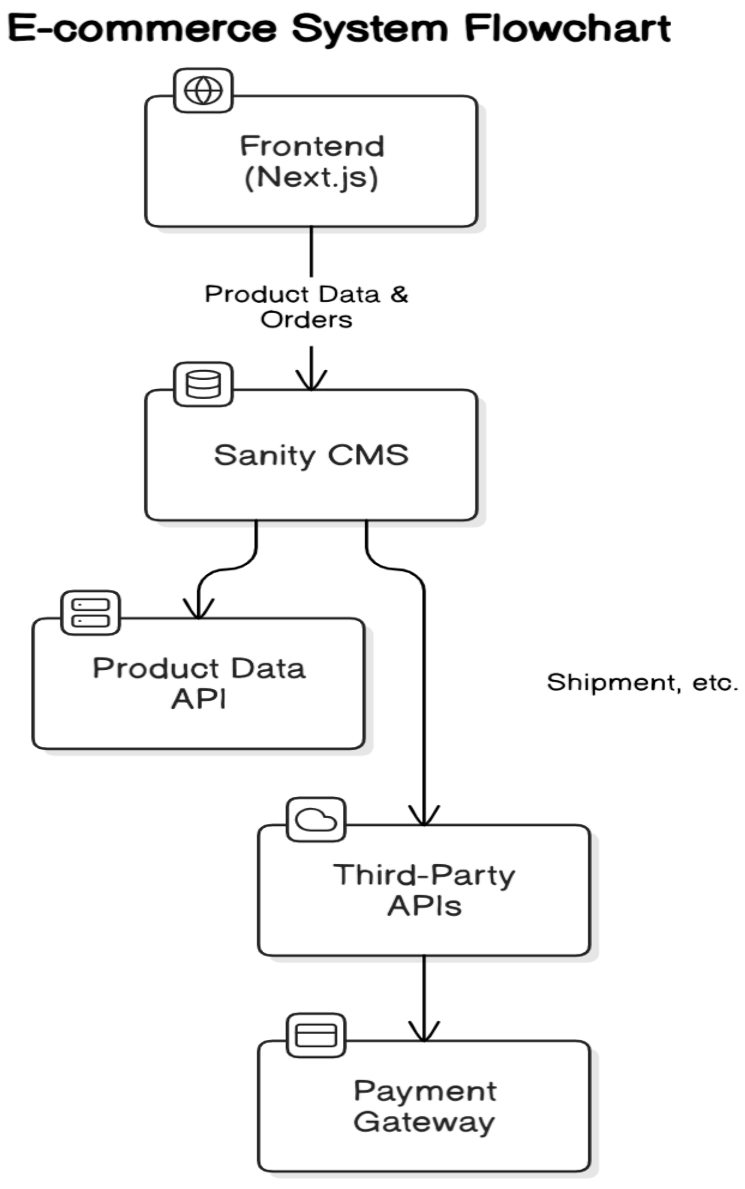
Sanity CMS as Backend:

- Use Sanity CMS to manage product data, customer details, and order records.
- Design schemas in Sanity CMS to align with the business goals defined on Day 1.

Third-Party APIs:

- Integrate APIs for shipment tracking, payment gateways, and other required backend services.
- Ensure APIs provide the necessary data for frontend functionality.

2. Design System Architecture:



Workflow Breakdown:

- **User Interaction:**
  - User interacts with the Frontend (Next.js) to browse products, view details, and place orders.
  - Frontend requests and displays data using APIs connected to Sanity CMS.

- **Backend Data Management:**
  - Sanity CMS handles product data, user orders, and other key records.
  - Sanity APIs are used to fetch and update this information.

- **Third-Party Integrations:**
  - APIs handle external functions like shipment tracking or Payment Processing Securely managed via a Payment Gateway, ensuring transactions are seamless.

API Requirements:

1. Endpoint for Fetching Products

Endpoint Name: `/products`

Method: GET

Description: Fetch all available products from Sanity CMS.

Response Example:

```
{
   "id": 1,
   "name": "Product A",
   "price": 100,
   "stock": 20,
   "image": "productA.jpg"

  }
```

2. Endpoint for Creating Orders

- Endpoint Name: `/orders`
- Method: POST
- Description: Create a new order in Sanity CMS.

Payload:

```
{
  "customerInfo": {
    "name": "John Doe",
    "email": "john.doe@example.com",
    "address": "123 Street Name"
  },
  "productDetails": {
    "productId": 1,
    "quantity": 2
  },
  "paymentStatus": "Pending"
}
```

Response Example:

```
      {
      "orderId": 456,
      "status": "Success"
        }
```

3. Endpoint for Shipment Tracking

- Endpoint Name: `/shipment`
- Method: GET
- Description: Track order status via third-party API.

Response Example:

```
        {
          "shipmentId": "ABC123",
          "orderId": 456,
          "status": "Shipped",
          "expectedDeliveryDate": "2025-01-20"
              }
```