Marketplace Technical Planning - Day 2

Project Overview

This is a technical plan for the eCommerce marketplace focused on selling products such as bridal suits, casual wear, and party wear. The project involves a user-friendly, responsive frontend using *Next.js, a backend powered by **Sanity CMS* for dynamic product management, and third-party API integrations for features like shipment tracking and payment gateways.

Table of Contents

Frontend Requirements

1. *User-Friendly Responsive Site:* The site should be mobile-friendly and easy to navigate, providing a seamless shopping experience across devices.
2. *General eCommerce Products for All Genders:* The marketplace should cater to products for both men and women.
3. *Pages:*
   - *Home*
   - *About*
   - *Shop*
   - *Team*
4. *SEO Optimization:*
   - Using *Next.js* ensures good SEO performance due to Server Side Rendering (SSR), which is fast and helps the site rank better on search engines.

Sanity CMS As Backend

1. *Product Data Management:* Sanity CMS will be used to dynamically manage product information, including details like name, description, price, and images.
2. *User and Order Details:* Sanity CMS will also manage user details and order information.
3. *Data Fetching Using GROQ Query:* Products and order details will be fetched dynamically using *GROQ queries* in Sanity CMS to be displayed on the frontend.

API Setup (Third Party)

1. *Third-Party APIs:*
   - APIs will be used for shipment tracking and payment gateways.
2. *Sanity Data Fetching:*
   - We will fetch data from Sanity CMS using API calls to display product details dynamically in the UI.
   - For example, a product’s API might return data like { name: 'Black Trouser For Men', title: 'Black Trouser', description: 'A stylish black trouser' }. The product details will be fetched and displayed on the UI.

Dynamic Routing

1. *Dynamic Routing for Products:*
   - We will use dynamic routing for product pages. For instance, a product title like "Black Trouser For Men" will have a slug like /black-trouser-for-men.
   - We can also use id or name in the URL like /productdetails/1 or /productdetails/blacktrouser.
2. *Best Practice for SEO:*
   - *Slug-based routing* is preferred as it is SEO-friendly, making the URL more readable and keyword-rich.

General eCommerce Example

1. *Endpoint Name:* /shop
2. *Explanation:*
   - When a user visits the /shop route, it fetches all product data either from the API or Sanity CMS.
3. *Example Response:*
   - Data for a product might look like: { productName: 'Black Trouser', productId: '1', productPrice: 3000 }.

Data Schema

Even if you have an API, you still need to define schemas in the Sanity folder to structure your data.

1. *Example Data Schema for Product:*

~~~
```javascript
{
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    { name: 'productId', title: 'Product ID', type: 'string' },
    { name: 'productName', title: 'Product Name', type: 'string' },
    { name: 'productDescription', title: 'Product Description', type: 'text' },
    { name: 'productPrice', title: 'Product Price', type: 'number' },
    { name: 'productImage', title: 'Product Image', type: 'image' }
  ]
}
~~~

2. Example Data Schema for Order:

```
{
  name: 'order',
  title: 'Order',
  type: 'document',
  fields: [
    { name: 'orderId', title: 'Order ID', type: 'string' },
    { name: 'customerName', title: 'Customer Name', type: 'string' },
    { name: 'orderDetails', title: 'Order Details', type: 'array', of: [{ type: 'string' }] }
  ]
}
```

API Endpoints

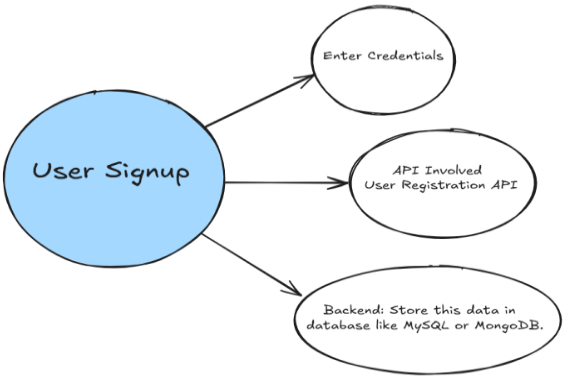
1. /products Endpoint

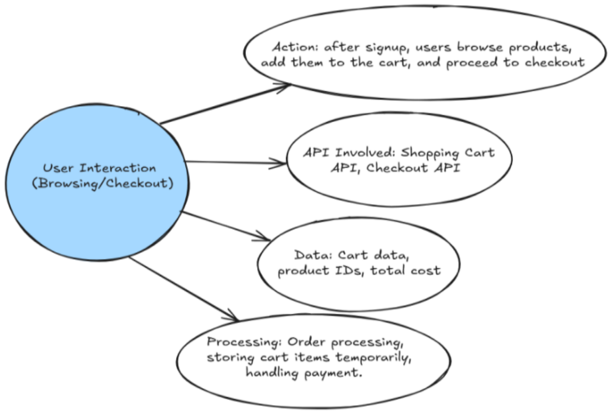
Method: GET

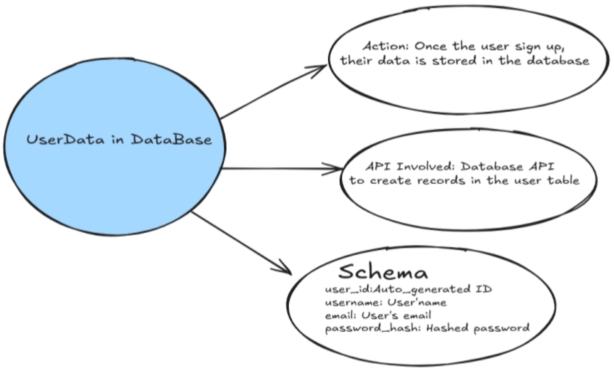
Description: Fetches all product data from Sanity CMS, returning fields like id, name, description, price, quantity, etc.
2. /orders Endpoint
Method: POST
Description: Creates a new user order in Sanity CMS, including customer information and order details.
3. /shipment Endpoint
Method: GET
Description: Tracks products using a third-party API and returns shipment details like shipment ID, status, and expected delivery date.

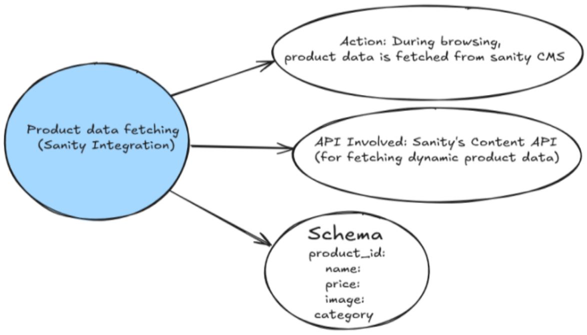
Payment Methods

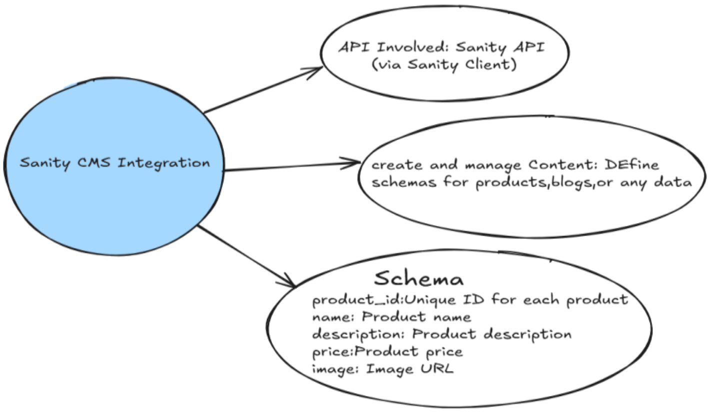
1. Worldwide:
Stripe and PayPal will be used for international payments.
2. In Pakistan:
Payment methods like Alfa Payment Gateway, HBLPay, JazzCash Merchant, and EasyPaisa Merchant will be implemented for local payments.
Sign In & Sign Up (Authentication)
Authentication is yet to be implemented, but we will update this section once we learn about handling sign-in and sign-up functionality.

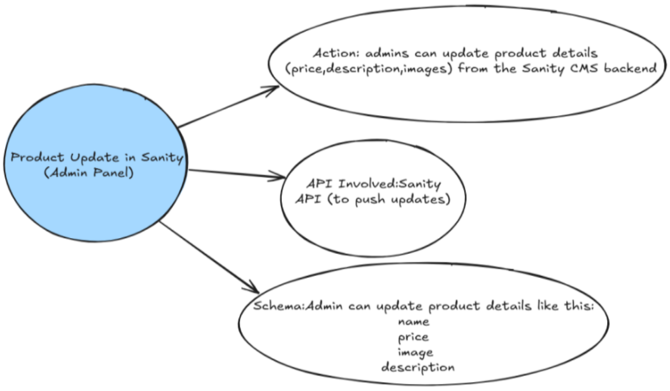



User Interaction
(Browsing/Checkout)
Action: after signup, users browse products,
add them to the cart, and proceed to checkout
API Involved: Shopping Cart
API, Checkout API
Data: Cart data,
product IDs, total cost
Processing: Order processing,
storing cart items temporarily,
handling payment.


UserData in DataBase
Action: Once the user sign up,
their data is stored in the database
API Involved: Database API
to create records in the user table
Schema
user_id:Auto_generated ID
username: User'name
email: User's email
password_hash: Hashed password


Product data fetching
(Sanity Integration)
Action: During browsing,
product data is fetched from sanity CMS
API Involved: Sanity's Content API
(for fetching dynamic product data)
Schema
product_id:
name:
price:
image:
category


Sanity CMS Integration
API Involved: Sanity API
(via Sanity Client)
create and manage Content: DEfine
schemas for products,blogs,or any data
Schema
product_id:Unique ID for each product
name: Product name
description: Product description
price:Product price
image: Image URL


Product Update in Sanity
(Admin Panel)
Action: admins can update product details
(price,description,images) from the Sanity CMS backend
API Involved:Sanity
API (to push updates)
Schema:Admin can update product details like this:
name
price
image
description