Student Name : Manahil

Father's Name : M. Nadeem

Roll No. : 73028

Slot : Sunday
(9:00 - 12:00).

7-01-25

HACKATHON # 03

DAY # 02: PLANNING THE TECHNICAL FOUNDATION

DAY # 02 ACTIVITIES: Transitioning To Technical Planning.

1- Define Technical Requirements:

Now I translate my business goal into clear technical requirements.

* Frontend Requirements:

1- User-friendly interface: Our website is very simple and clear for browsing. So, the user are easily find it and enjoy our services.

Responsive design: The website of my products are user friendly and also responsive for mobile and desktop users. It seems like good design.

Essential pages:

Home
Product Listing
Product Details
Cart
Checkout
Order Confirmation

Sanity CMS as Backend:

Sanity.io is a cloud based open source content platform that can be used as backend because for content management.

It is a headless content management system (CMS) that allows users to manage content and deliver to across devices.

PRODUCT SCHEMA (for managing product)

Fields :

name : string (product name)
price : number (product price)
stock : number (Quantity in stock)

Schema's / product.js :

```
export default {
  name : 'product',
  title : 'product',
  type : 'document',
  fields : [
    { name : 'name', type : 'string' },
    { name : 'price', type : 'number' },
    { name : 'stock', type : 'number' },
  ]
}
```

Order Schema

Fields :

1) Order ID 2) Product ID
3) total amount 4) Quantity

Schema's / Order.js :

```
export default {
  name : 'order',
  title : 'Order',
  type : 'document',
  fields : [
    {name : 'OrderNumber', type : 'string'},
    { name : 'product', type : array of :
        [{type : 'reference', to : [{type :
          'product'}]}]}

    { name : 'totalAmount', type : 'number'},
    { name : 'Quantity', type : 'number'};
        ], };
```

Customer Schema

fields:

- name
- email
- address
- cell number

schema / customer.js :

```
export default {
name : 'customer',
title : 'Customer',
type : 'document',
fields :[
 { name : 'name', title : 'Name', type: 'string'},
 { name : 'email', title: 'Email', type : 'string'},
 { name : 'address', title : 'Address', type: 'string'},
 { name : 'cell number', title: 'Cell Number', type: 'number'},
  ],
  };
```

Third - Party APIs :

Third-party APIs are providing necessary data for frontend functionality. Which is used for integrate APIs in shipment tracking, payment gateways. In - short using

1) shipment tracking API
2) payment gateway API :

2- Design System Architecture :

(i) Frontend :

- Built using Next.js
- Handles user interactions and displays data fetched from APIs.

→ Also including main pages, like; Homepage, product listing, product details, cart and checkout, orders tracking.

(ii) Sanity CMS :

Sanity CMS is used for backend, the purpose is to manage content and data handling.

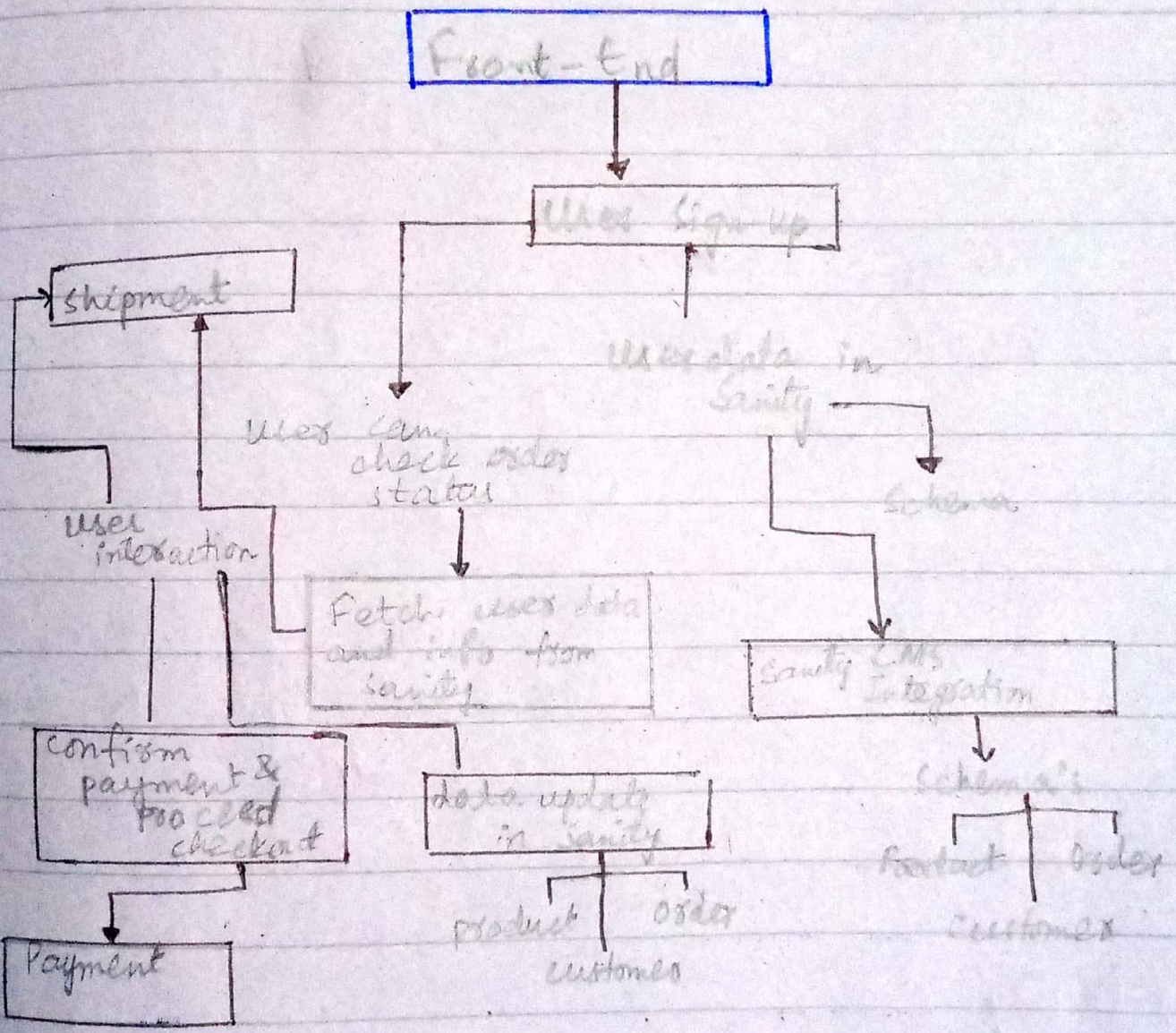
(iii) Third-party APIs :

Integrated with third party APIs services for payments and shipping.

(iv) Authentication:

Integrated ~~for~~ with clerk for secure user authentication.

Data Schema Design;



WORKFLOW:

1) Frontend: Users via browsering products, add items to their cart and proceed to checkout

2) Backend: Processing API request from the frontend.

3) Payment gateway: Backend sends order and payment details to the gateway.

4) Shipment tracking:
Backend interacts with tracking APIs to fetch the order status.

5) Delivery Zones:
Backend logic calculates delivery zones as fetches

driver details which is based on customer location (using Google Map API).

iv) Notification: Backend triggers notification through email or SMS upon for order confirmation or shipment updates.

3- Plan API Requirements :

API endpoints required for the marketplace, based on the described schema and workflow.

i) Products: /products :
(Fetch all data) (Response Example)

```
[
{
"id" : 1,
"name" : "Product A",
"price" : 5000,
"stock" : 05,
"image" : "url_to_image"
},
```

Endpoint Name: /products
→ Method = GET
→ Description: Fetch all available products from variety CMS

```
{
  "id": 2,
  "name": "Product B",
  "price": 10,000,
  "stock": 05,
  "image": "url-to-image"
}
]
```

(ii) Products : /products/{id}

```
{
  "id": 1,
  "name": "Product A",
  "price": 1000,
  "stock": 25,
  "description": "product detail",
  "image": "url-to-image"
}
```

Endpoint Used:
{/products/id}
→ Method : GET
→ Description: Fetch detail info for a specific product by its ID.

Endpoint: /orders:

```
{
"customer" : {
"name" : "ABC",
"email" : "abc@gmail.com",
"cell number" = 03112243128
},

"products" : [
{"id" : 1 , "quantity : 1},
{"id" : 4 , quantity : 2}
],
"payment Status" : "Paid",
"total Amount" : 10,000
}
```

Endpoint Name: /orders
→ Method: POST
→ Description: Create a new order in sanity CMS

Shipment Tracking: /shipment:

```
{
"shipment Id" : "ship345",
"Order Id" : "345",
"status" : "process",
"delivery" : 01-02-2025,
"time" : 04:00 pm
```

Endpoint Name: /shipment
→ Method: GET
→ Description: Track order status via a third-party API
→ Query Parameter: order ID.

c) Users Authentication :

Endpoint Name : /auth / login

→ Method : POST

→ Description : Log-in as a user and return an authentication token.

» Response Example :

```
{
  "token" : "autt-token",
  "message" : "login"
}
```

———— x ———— x ————