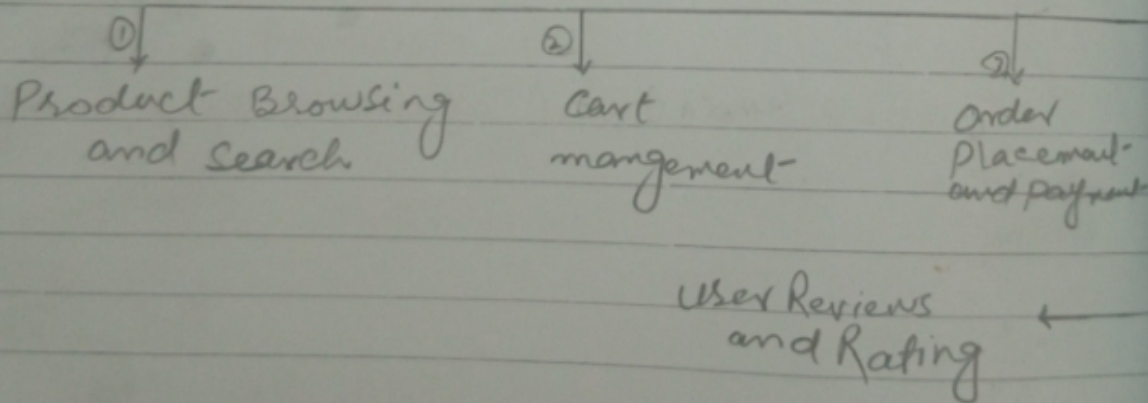# PLANNING THE TECHNICAL
FOUNDATION

Building a Strong Technical Foundation for My General E-commerce MarketPlace.

General E-commerce :-
General E-commerce Plateform are online spaces, where users can buy and sell Physical or digital Products These marketplace types are relatively simple realising basic functimalities like Products browsing, cart manegment, and order Placement.

FEATUR AND TECHNICAL REQUIREMENTS:-

① Product Browsing and Search.

② Cart mangement

③ Order Placement and Payment

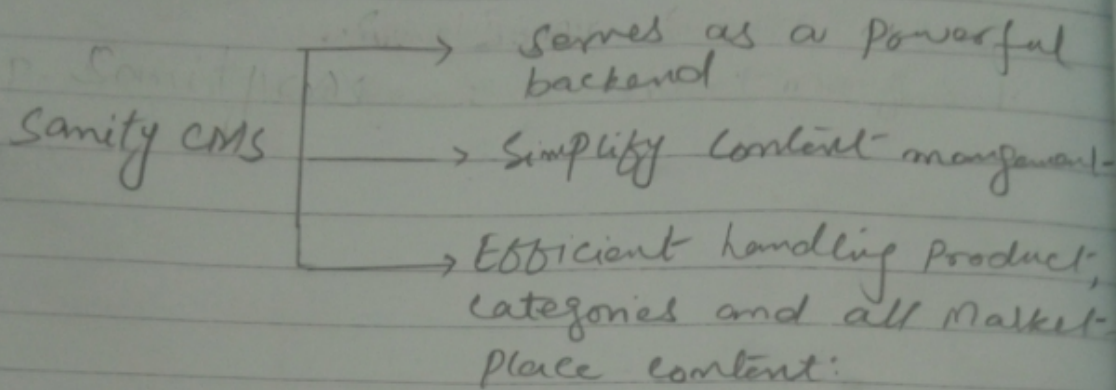User Reviews and Rating ←

SYSTEM ARCHITECURE FOR MARKETPLACE WITH SANITY CMS and NEST.JS :-

When designing the system architecture for market-place, it is important to clearly visualize the

intrection between the frontend (Next.Js)
and the backend (Sanity cms and the
third Party APIs)

Sanity CMS
- → Serves as a powerful backend
- → Simplify content mangement
- → Efficient handling Product, categories and all Market-Place content.

Key components of the Architecture :-
① Dymanic Product Page
② Cart and check out
③ filter and search
Frontend Next Js.

SYSTEM ARCHITECURE OVERVIEW-

The architecture ensure a
clear data flow between to components and
defines the responsibilities of each.

Sanity CMS → act as the backend

Next.Js → Serves as the frontend to display

APIs → External APis For Payment, delivery Proce

DETAILED API REQUIREMENTS FOR MARKETPLACE

API endPoints for market Place to fecilbt data Exchange between the frontend and backend.

WORFLOW

/Product (GET
└── This EndPoint Product details, including name, Price, stock, ing

/orders (POST)

```
{ "id": " 45670",
  "name"; " chair",
  "Price": 200.00,
  "image": " https://exaple
  /image/Product-45670.j
  "category"; " Furnatire"
},
"
)
```
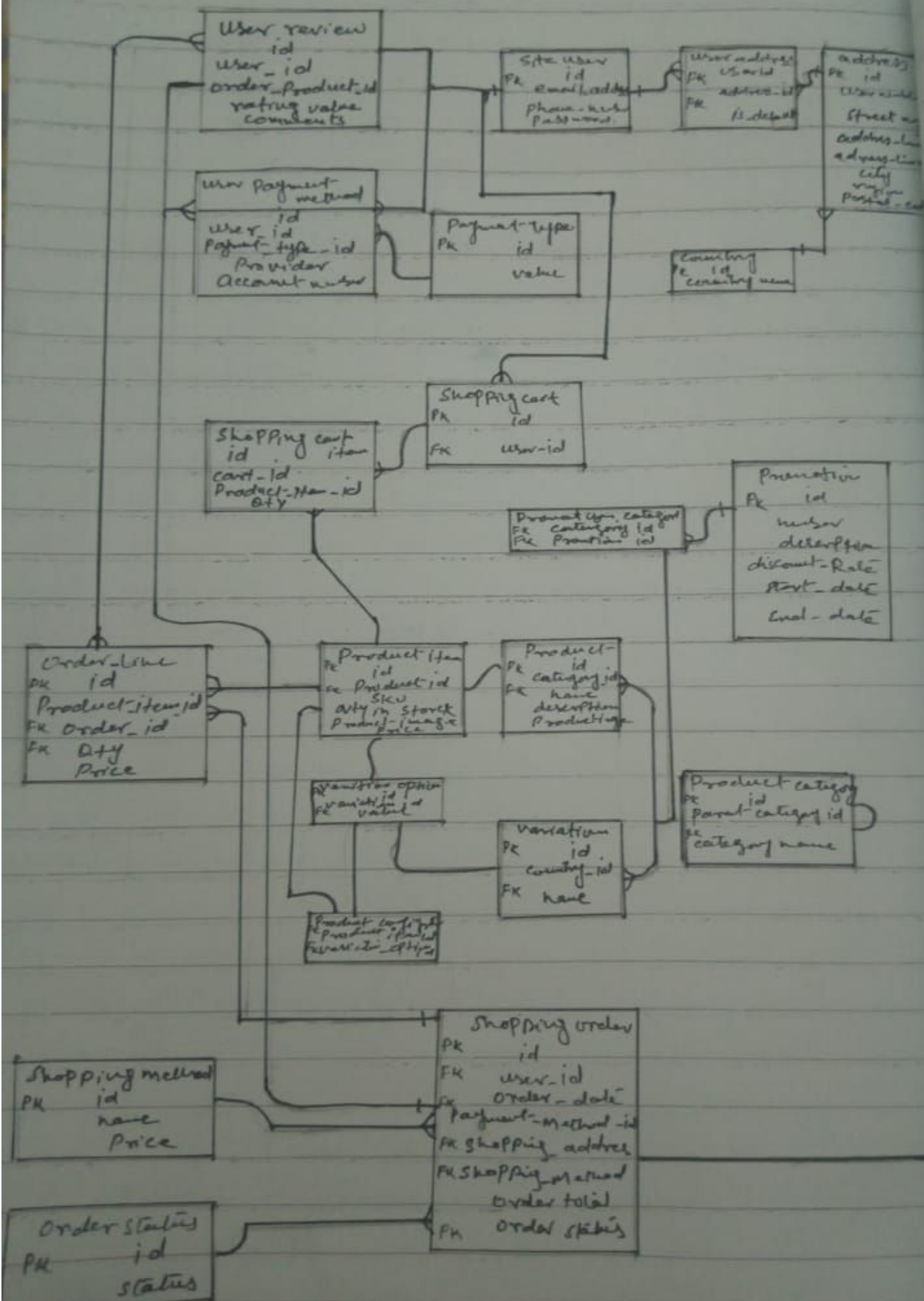
/shipment (GET)

/cart (POST/GET)

/Payment (POST)

Response Example:

```
{ "status"; " Payment successful",
  "Payment Id"; " Pay 45678"
  "orderId"; " cabe 4567"
}
```

E-commerce DataBase Structure

User review
- id
- user_id
- order_product_id
- rating value
- comments

Site user
- PK id
- email address
- phone number
- password

User address
- PK user id
- address id
- FK is default

address
- PK id
- user id
- Street
- address line
- address line 2
- city
- region
- postal code

User payment method
- id
- user_id
- Payment_type_id
- Provider
- Account number

Payment-type
- PK id
- value

Country
- PK id
- country name

Shopping cart
- PK id
- FK user-id

Shopping cart item
- id
- cart_id
- Product_item_id
- Qty

Promotion
- PK id
- name
- description
- discount-Rate
- start_date
- End_date

Product Cup. category
- FK category id
- FK Promotion id

Order_line
- PK id
- Product_item_id
- FK Order_id
- FK Qty
- Price

Product Item
- PK id
- FK Product_id
- SKU
- qty in stock
- Product_image
- Price

Product
- PK id
- category_id
- FK name
- description
- Production

Variation option
- PK variation id
- FK value

Variation
- PK id
- country_id
- FK name

Product category
- PK id
- Parent category id
- FK category name

Product config
- Product item id
- variation option id

Shopping order
- PK id
- FK user-id
- FK order_date
- Payment_Method id
- FK shopping address
- FK shopping method
- order total
- PK order status

Shopping method
- PK id
- name
- Price

Order status
- PK id
- status

Sam
(For bac

Orde

SHORT

FRONTE

Backen
(Data

Ship

Paym

Meth

EndP

USER

|

Frontend (using Next. Js)

Sanity CMS
(For backend database)

Order data

APIs
(Third Party APis)

Shipment tracking
APis
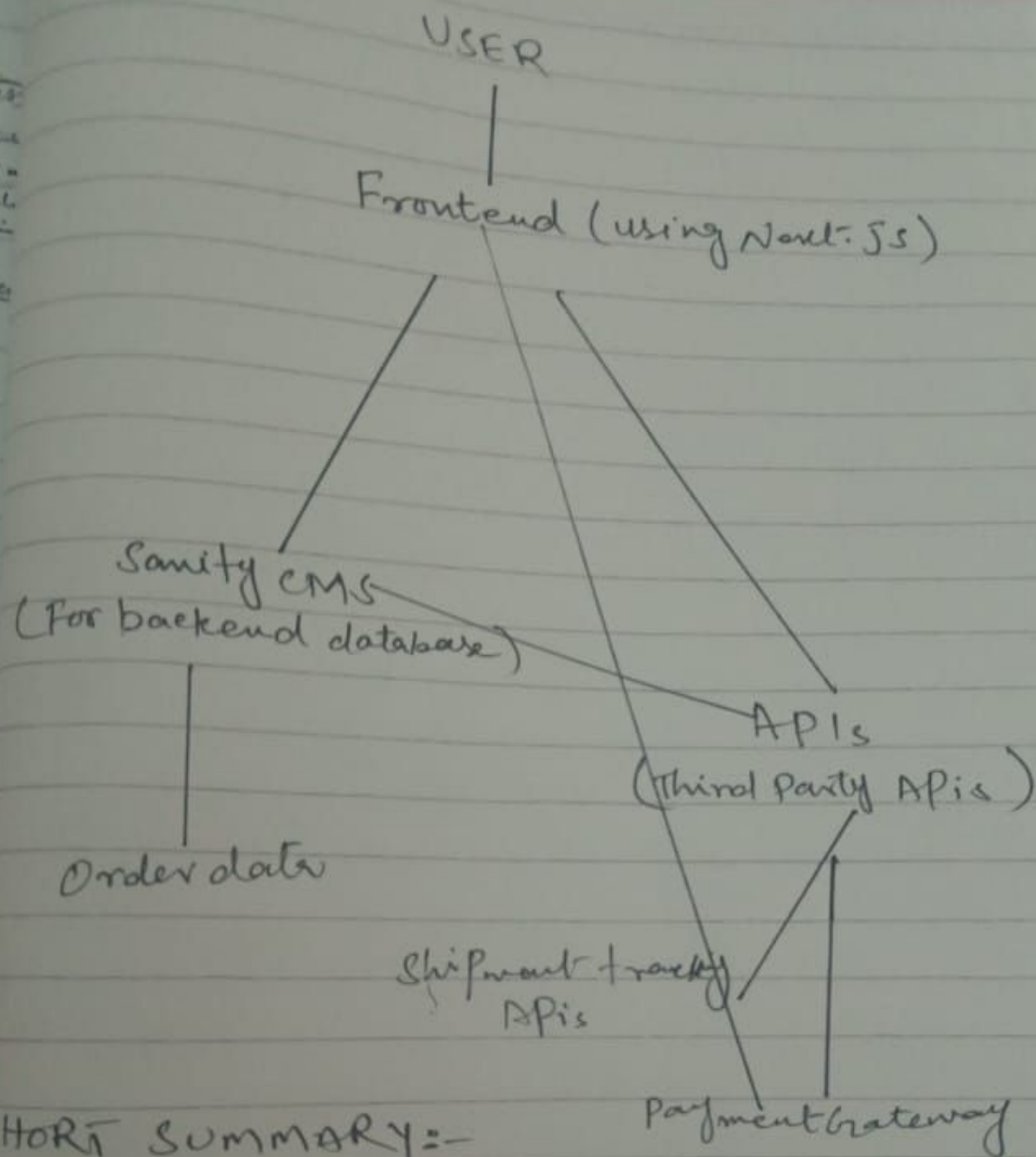
Payment Gateway

SHORT SUMMARY:-

FRONTEND = Next.Js

Backend = Sanity CMS
(Database)

Shipment Tracking: Third Party APi's (ShipEngine)

Payment Gateway: third Party APi's (Stripe)

Method of APi: (GET, Post, Put)

EndPoints of APi: / Product , / shipment
/ cart , / Payment

System Architecture Diagram

USER DEVICE
↓
FRONTEND (Next.js)
↓
SANITY CMS (Third Party APIs)

USER DEVICE

↓

FRONTEND (Next.js) —— (API calls) —— SANITY CMS

↕

THIRD-PARTY APIS (Payment, Shipping, Inventory)