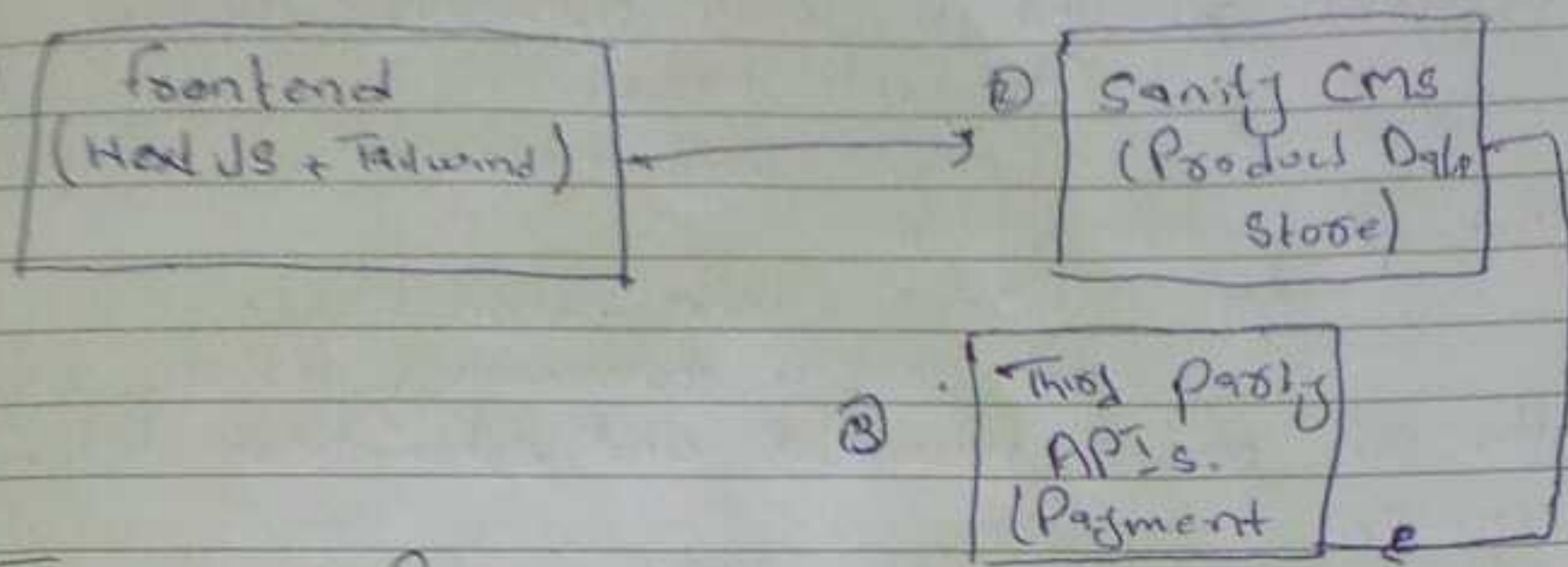
Day 2
Hackathon 3

Date 16/1/25

Name: Tooqeer Ahmed Soomro
Roll No: 00306062
Slot: Thursday, Morning 9 to 12

1) System Architecture Overview:



2) Technical Requirements:
~~User Need~~

Frontend Requirements:

1) User-friendly Interface:
- Intuitive navigation for seamless browsing
- Clean and minimal design enhance the user experience.

2. Responsive Design:-
- Fully optomized for both mobile and desktop users.
- Ensure proper ~~scaling~~ scaling and usability across ~~screen~~ screen sizes

3. Essential Pages:-
- Home Page:
Display featured products, promotion and categories.
- Product Listing Pages:
Show a grid or list of products with filters (price, category, ratings).

Product Detail Page:
- Display detailed information (name price description, reviews, and images).

Cart Page:
- Allow users to view selected products, adjust quantities, and remove items.

Checkout Page:
- Collect user information (shipping and payment details.

Order Confirmation Page:
- Display a summary of the order with an order number and estimated delivery time.

② Sanity CMS:

Sanity is where the product information like: Product name, prices, descriptions, categories, stock, etc is stored. It helps the backend ~~fetch~~ fetch and display this data on the frontend.

③ Third Party APIs:

Third Party APIs can greatly enhance e-commerce platform by adding advanced features without building them for scratch.

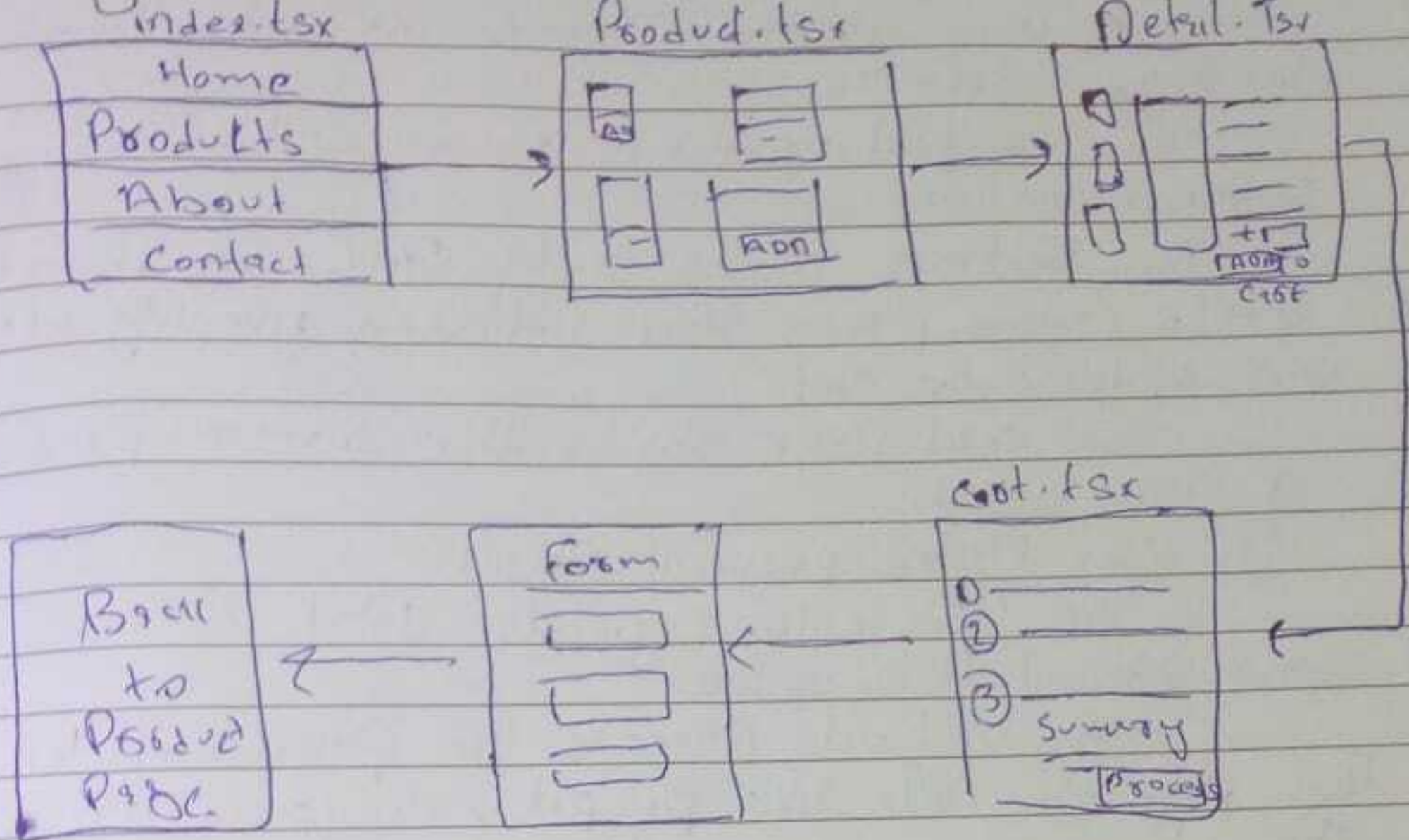
API Uses:

Paypal: Enables Credit card and PayPal account payment

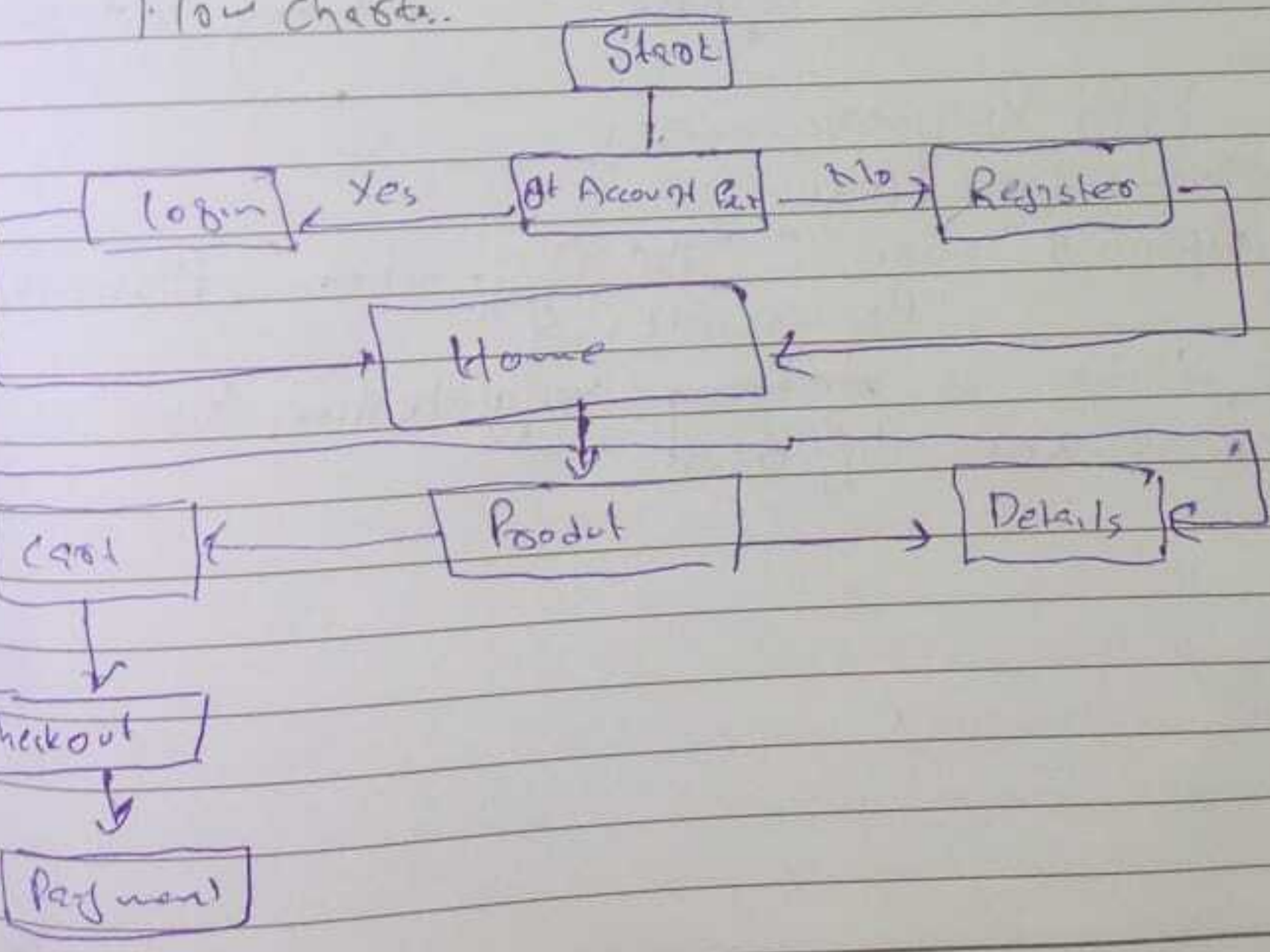
Easypaisa: Ideal for Pakistan-based business multiple payment methods.

Date

Key User flows:



Flow Charts..



Work felows Details:
① User Adds Products to cart:
- A user browse products and adds them to their cart.
- The frontend updates the cart with the selected products.
- The backend checks Sanity CMS for Products details (name, price, stock, category, quantity, etc) and updates the cart.
- The cart page shows the current products the user added.
② User Makes payment Details
- The User enter payments detail form and submit the order.
- The backend process the payment using the payment API. like. paypal, easypaisa, Jazzcash etc.
- The User see update the delivery status Stock.

API Requirement:
Method:- Post /login
Description: { "email," "abe@gmail.com", Pass-word
"Password123" }
— Same as ~~different~~ registration but Data Schema are different

Date _____

Product details.

Method: Get /Products/{id}

Description: Fetch detail of a single Products.

Example: {"id": 1, "name": "Burgers", "Price": 300, "Description": "Delicious Burgers"}

Order /Card.

Method: Post /Card

Description. New Orders please.

Example: {{"Productid": 1, "Quantity": 2}}

- Get method se old history fetch hoyega user ki.

Date

SANITY SCHEMA EXAMPLE.

Product.

export default {

name: "Product",

type: "document",

field: [

{name: 'name', type: 'string' title: 'Product

name' }},

{name: "Price", type: "number', titittle,

'Price"},

{name: 'decription", type: "string", title: "Category

Cart:

export default {

name: "Cart",

type: 'document',

Field: [

{name: "CustomerID', type: number, title: "CID"

{name: "Products", type: "array",

fields: [{name: "ProductID", type: "number"}]

title: "Product"}]}

Date

Orders:

```
export default {
  name: 'Orders',
  type: 'document',
  fields: [
    { name: "CustomerId", type: "number" },
    { name: "CartItem", type: "array",
      field: [{ name: "ProductID", type: 'number' },
      { name: "Quantity", type: 'number' }],
      title: "Cart Items" },
    { name: 'Status', type: 'string', title: "Order Status" },
    { name: "Payment", type: "string", title: "Payment" }
  ]
}
```

E.R Diagram: