

HACKATHON DAY 2

Date _____

MARKETPLACE TECHNICAL FOUNDATION

SYSTEM ARCHITECTURE DESIGN:

Next.js for building
a dynamic, server-side rendered
(SSR) & seo friendly user interface.

Product catalog display

Product detail

Cart

Checkout

Order Confirmation

BACKEND

Product Data

Customer Data

Order Records

THIRD PARTY API'S

Payment Gateway

Shipment Tracking

AR Integration

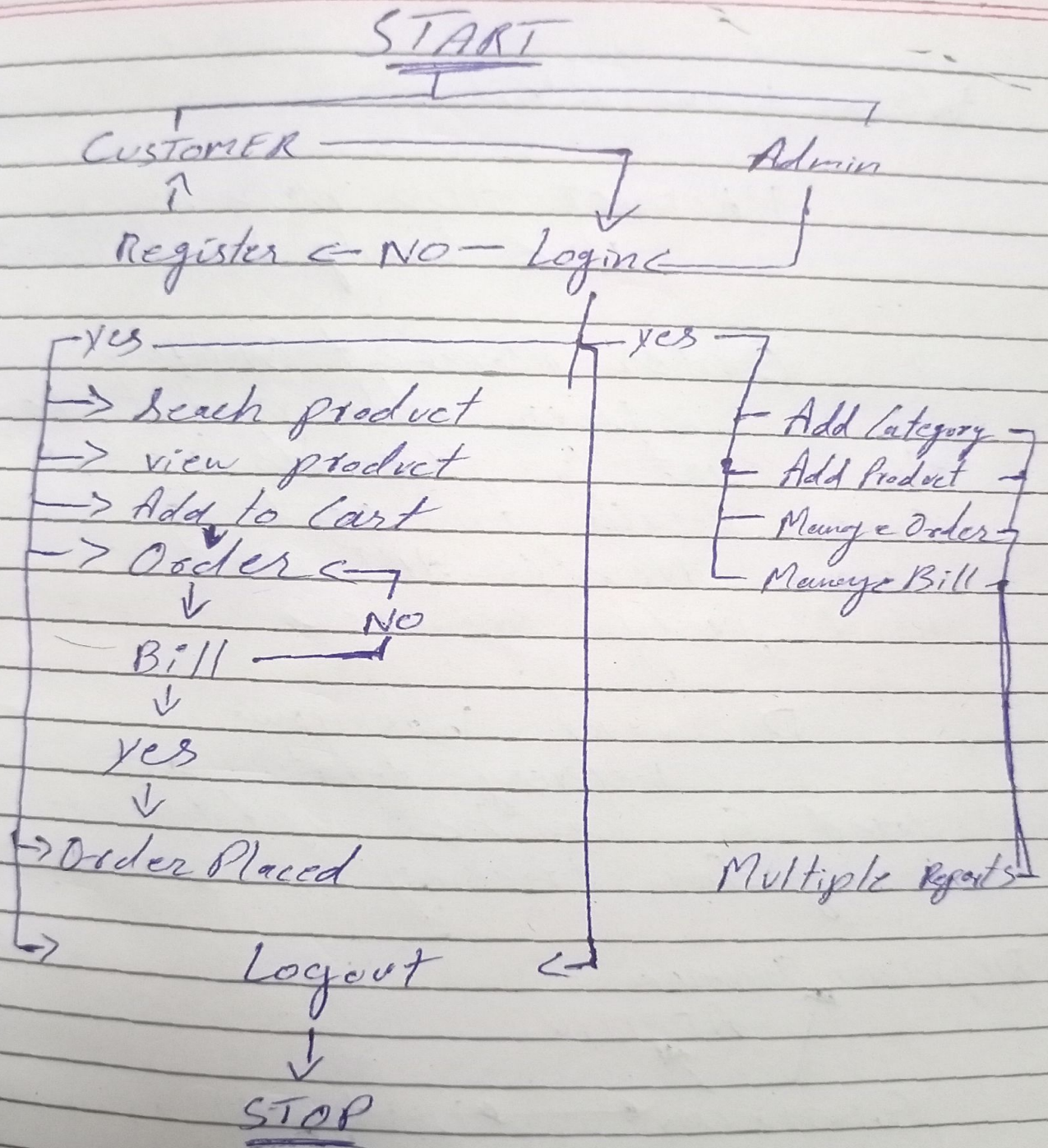
Signature _____

Q2

No. 1

Roll #: 00319323

Date: _____



Signature _____

Dr. _____

No. 2

Roll #: 00319323

Date: _____

WORK FLOW PLAN

PRODUCT ADD WORKFLOW

Use Sanity to add Product

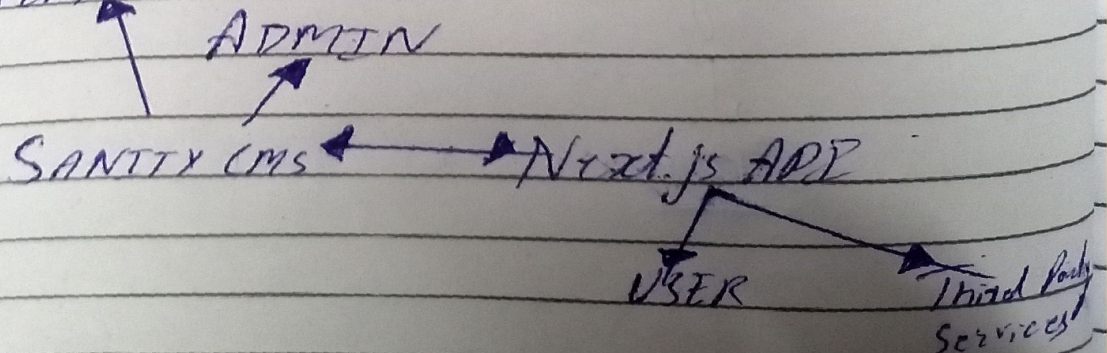
Order Placement Workflow

User adds products to the cart & proceeds to checkout. Order details are sent to a custom Next.js API route OR third party services.

DELIVERY WORKFLOW

Delivery details are stored & can be updated via Sanity for tracking purpose.

DELIVERY TRACKING



Signature: _____

RE

No

Roll # = 00319323

Date _____

API REQUIREMENT

SANITY API:

Querying Products:

Use Sanity GraphQL to fetch product data dynamically.

Order Management:

Use a custom API in Next.js or integrate a third-party service.

SANITY API



FRONTEND (NEXT.JS)



NEXT.JS API



THIRD-PARTY SERVICES

Date _____

API Requirement

Endpoint	Method	Des	Response
=> /products	GET	Fetch all from sanity	{ "id": 1, "name": "Product", "price": 100 }
=> /orders	POST	Create a New Order	{ "orderId": 123, "status": "confirm" }
=> /shipment	GET	Track Order	{ "orderId": 123, "status": "In Transit", "ETA": "3 hours" }
=> /payment	POST	Process Payment	{ "payment ID": 456, "status": "Success" }

Roll # 00 319323

Date _____

SANITX SCHEMA

Product & Order

Product Schema

```
export default {  
  name: "product",  
  type: "document",  
  fields: [
```

```
    { name: "id", type: "string",  
      title: "Product ID" },
```

```
    { name: "images", type: "array",  
      of: [{ type: "image" }],  
      title: "Product Images" },
```

```
    { name: "name", type: "string", title: "Product Name" },  
    { name: "price", type: "number", title: "Price" },  
    { name: "stock", type: "number", title: "Stock Level" },  
    { name: "description", type: "text", title: "Product  
      Description" } ] }
```


ORDER SCHEMA

```
export default {  
  name: "order",  
  type: "document",  
  fields: [  
    { name: "Order Id", type: "string",  
      title: "Order ID" },  
    { name: "CustomerName", type: "string",  
      title: "Customer Name" },  
    { name: "Email", type: "string",  
      title: "Email address" },  
    { name: "product Details", type: "array",  
      title: "Product Details" },  
    { name: "totalAmount", type: "number",  
      title: "Total Amount" },  
    { name: "orderStatus", type: "string",  
      title: "Order Status" }  
  ]  
};
```