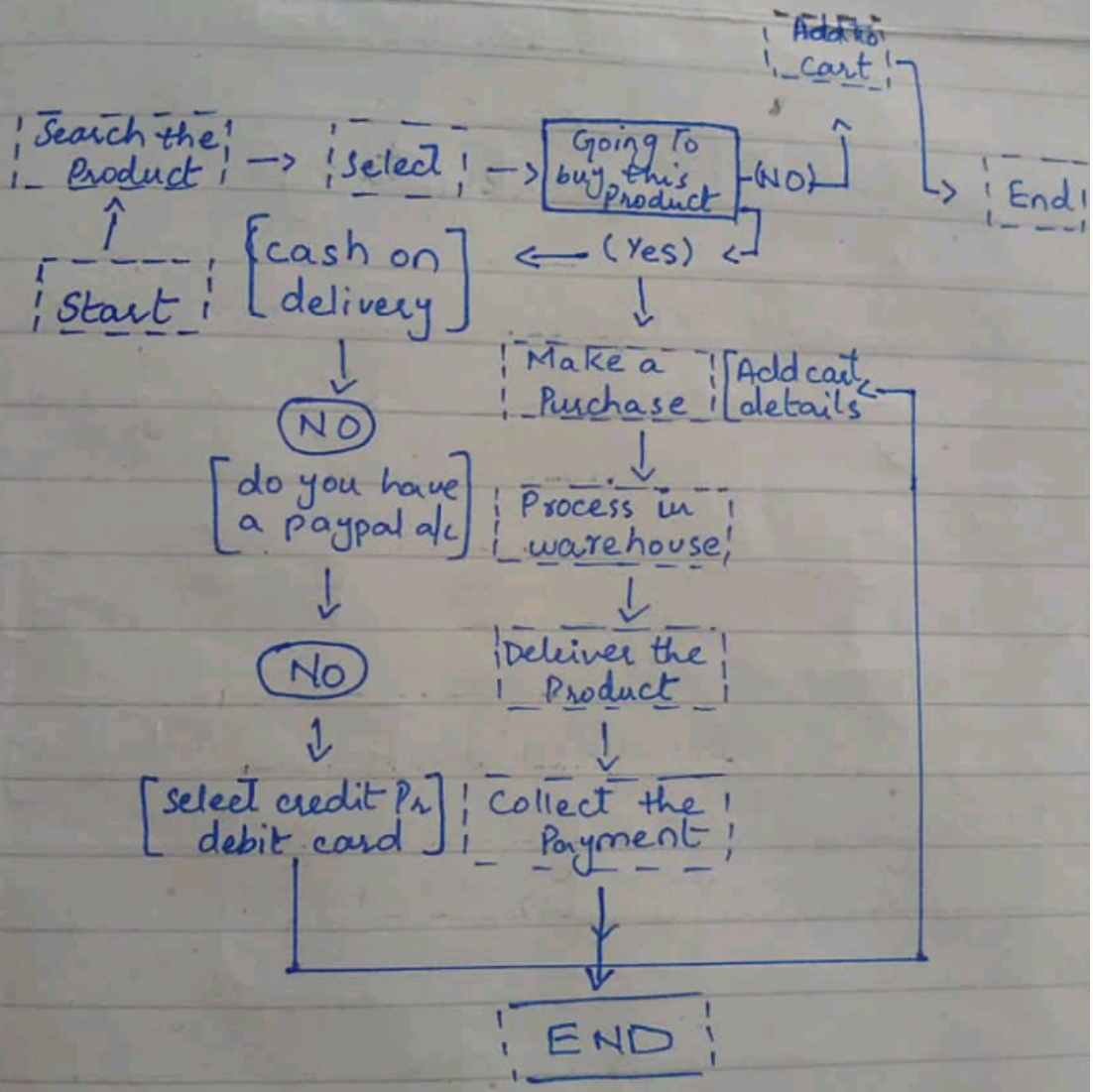
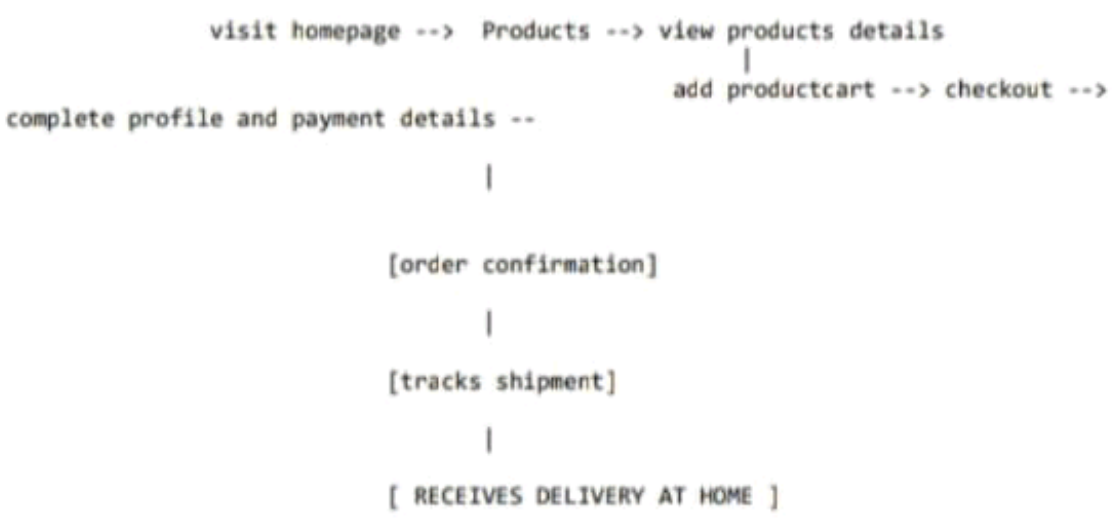
DAY-2 : MARKETPLACE TECHNICAL-FOUNDATION.

[Prepared by Sadia Tariq]

WORKFLOW KEYS

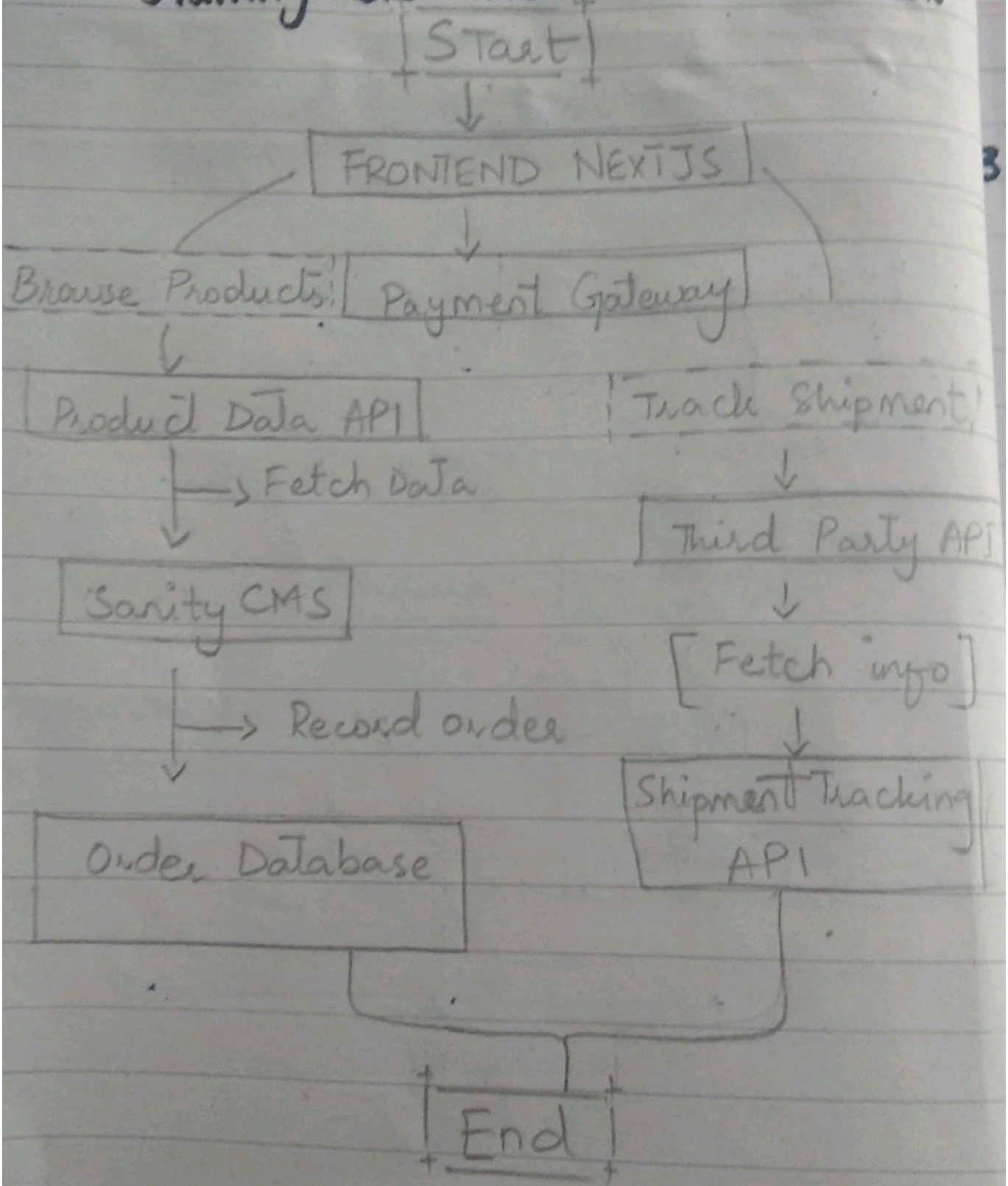


WORKFLOW DIAGRAM:

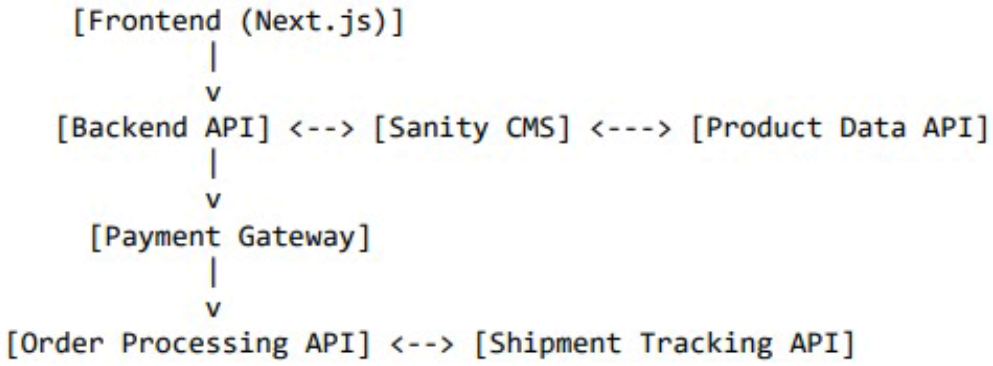


System Architecture Overview

Planing The Technical Foundation



System Architecture:



3. Plan API Requirements

API Endpoints

1- Endpoint Name : (/Products)

- Method : Get
- Description : Fetch all available product details in sanity.
- Response :

```
[
  { "id" = 1,
    "name" = "Product 1",
    "Price" = 100,
    "stock" = 20,
    "image" = "URL",
  },
```

2- Endpoint Name : (/orders)

- Method : Post
- Description : Create a new order sanity.
- Response :

```
{
  "customer_info": {
    "name": "Sadia",
    "email": "abc@gmail.com",
    "address": "abc street, city Karachi"
  },
  "Products": [
    {
      "id": 1,
      "name": "Product A"
      "quality": 2,
      "price": 100.
    },
  ],
    "id": 2,
    "name": Product
  "payment_status": "paid"
}
{
  "order_id": "ORD1234",
  "status": "order created",
  "customer_info": {
    "name": "Sadia",
    "email": "abc@gmail.com",
    "address": "abc street, city Karachi"
  },
    "order_Total": 350,
    "payment_status: "paid",
  },
```

3- ENDPOINT NAME: (/shipment)

- Method: Get
- Description: Track the order shipment status using a Third-Party API
- Response:

```
{
  "shipment_id": "SHIP1234",
  "order_id": "ORD1234",
  "status": "In Transit",
  "expected_Delivery_Date": "2025-01-01",
},
```

```
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Product Name',
      validation: (Rule) =>
        Rule.required()
          .max(100)
          .error('Product name is required and cannot exceed 100 characters.'),
    },
    {
      name: 'description',
      type: 'text',
      title: 'Description',
      validation: (Rule) =>
        Rule.required()
          .min(20)
          .max(500)
          .error('Description must be between 20 and 500 characters.'),
    },
    {
      name: 'price',
      type: 'number',
      title: 'Product Price',
      validation: (Rule) =>
        Rule.required()
          .min(0)
          .error('Product price must be a positive value.'),
    },
    {
      name: 'oldPrice',
      type: 'number',
      title: 'Old Price',
      description: 'The original price of the product before any discount.',
      validation: (Rule) =>
        Rule.min(0).error('Old price must be a positive value or zero.'),
    },
    {
      name: 'rating',
      type: 'number',
      title: 'Rating',
      description: 'Average rating of the product.',
      validation: (Rule) =>
```

```
        Rule.min(0)
          .max(5)
          .precision(1)
          .error('Rating must be between 0 and 5.'),
    },
    {
      name: 'sizes',
      type: 'array',
      title: 'Sizes',
      of: [{ type: 'string' }],
      options: {
        layout: 'tags',
      },
    },
    {
      name: 'image',
      type: 'image',
      title: 'Product Image',
      options: {
        hotspot: true,
      },
      validation: (Rule) => Rule.required().error('Product image is required.'),
    },
  ],
};
```