

Hackathon:

Day 2 – General Ecommerce

Prepared by:

Zahida Raees

Roll No:

00160756

Project Title:

Day2_Planning_The_Technical_Foundation

Hackathon Day 2: Planning the Technical Foundation

Recap of Day 1: Business Focus

On Day 1, we emphasized the importance of focusing on business requirements before jumping into technical implementation. Here's what we achieved:

1. **Primary Business:**

- ✓ I have designed a versatile platform for Balochi embroidery to offer diverse items with convenience, competitive pricing, and reliable delivery tailored to local and international customer needs.
- ✓ I have tried to put forward the demand of my customers through my project, use all the means of product delivery to meet their need and preference so that customers can easily come back to purchase.

2. **Data Schema Drafted:**

- Using paper and pencil, I have created a preliminary data schema outlining entities like categories, products, vendors, customers, orders, payments, delivery and their relationships.

3. **Single Focus:**

- By concentrating on e-commerce business requirements without technical distractions, I tried to set a solid foundation for the technical phase.

Day 2 Activities: Transitioning to Technical Planning

1. Technical Requirements:

- Frontend Requirements:

- ❖ User-friendly interface for browsing products.
- ❖ Responsive design for mobile and desktop users.

Essential pages:

- ❖ Home, Products, Catalog, Order, Login, Contact, About
- ❖ Supporting pages or forms:
payment form, delivery form, order tracking pages.

- Backend:

- To focus on my business goal I will use **Sanity CMS** to manage category data, vendor details, product data, customer details, order records and delivery record.
- In my project Sanity will act as the database for my marketplace. But if I feel need of different database then, I will surely use mysql or any other which can be easily link with my front end to ease my work.

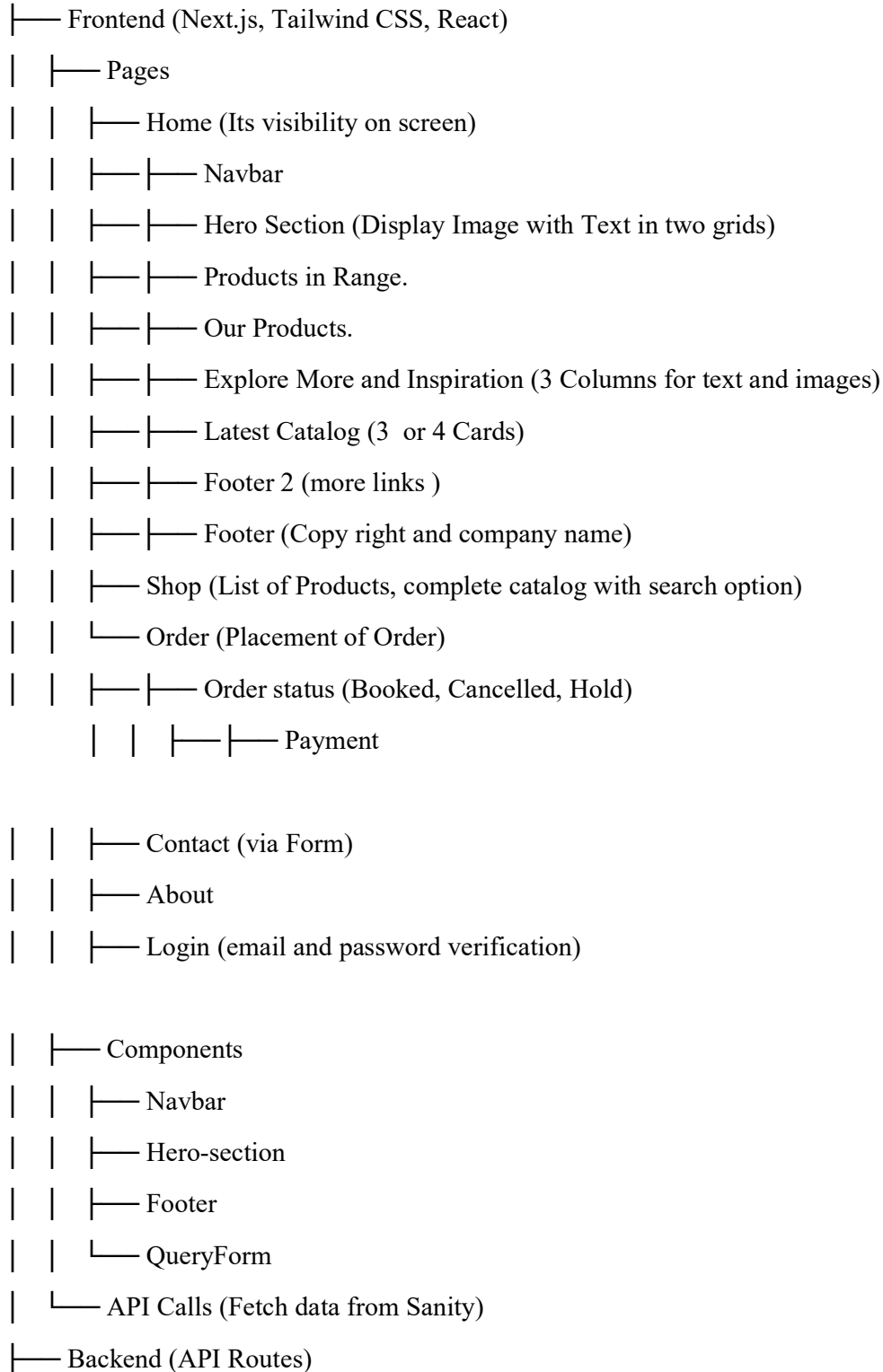
- Third-Party APIs:

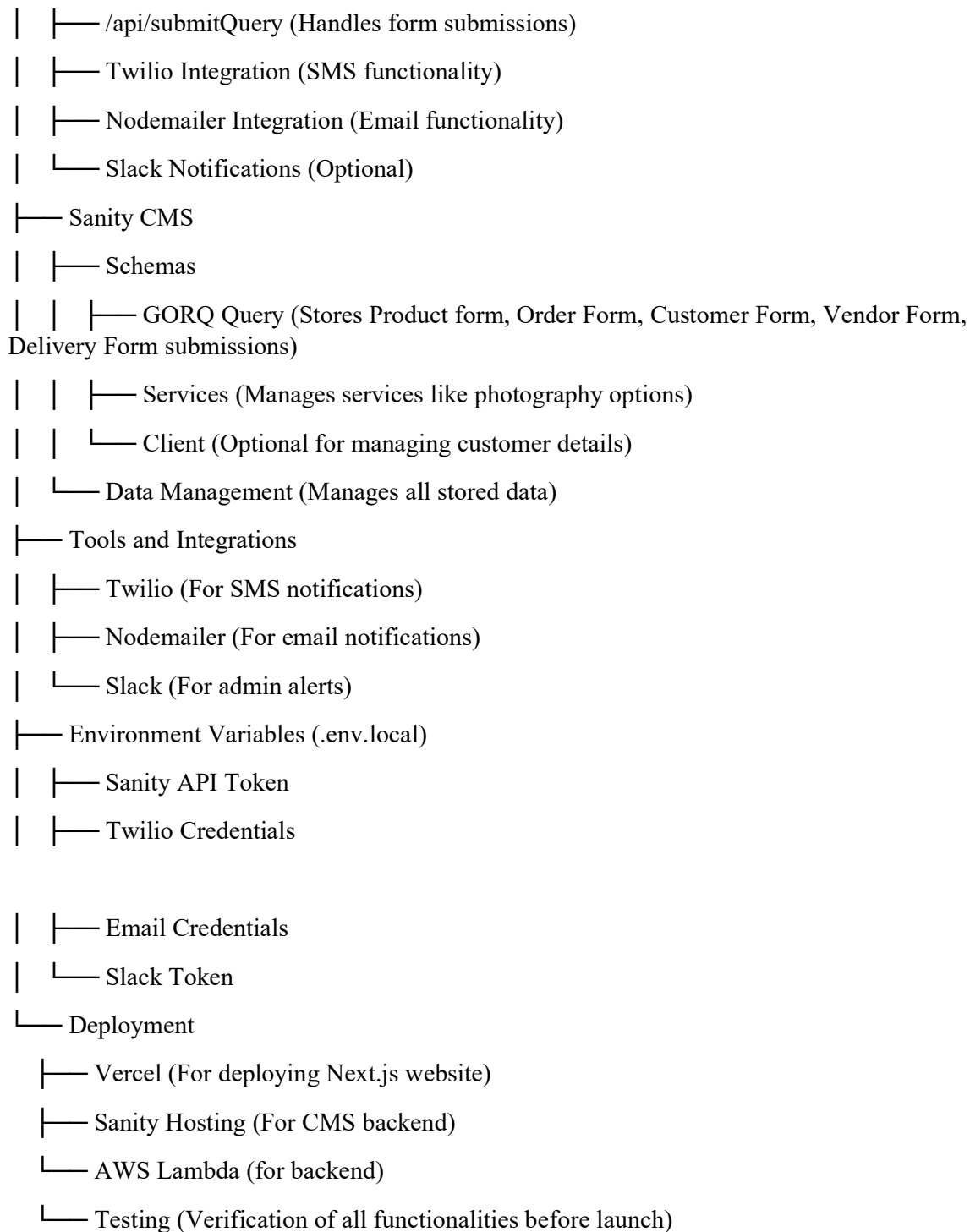
Up till now as I am aware about mockapi, sanity API and Stripe so for product, customer, vendor, order tracking I do use sanity or in some extent mockapi but for shipment tracking I do use **ShipEngine**, I'll also integrate **Stripe** for payment gateways.

2. Design System Architecture

A detailed architecture or a tree structure:

“Douch” E-commerce Project





In this architecture, a typical data flow could look like this:

1. A user visits the marketplace frontend to browse products.
2. The frontend makes a request to the Product Data API (powered by Sanity CMS) to fetch product listings and details, which are displayed dynamically on the site.

3. When the user places an order via Order form or via Cart on Product image, the order details are sent to Sanity CMS via an API request, where the order is recorded.
4. Shipment tracking information will be fetched through a Third-Party API to be displayed to the user in real-time.
5. Payment details are securely processed through the Payment Gateway, and a confirmation is sent back to the user and recorded in Sanity CMS.

Key Workflows to Include:

1. User Registration (Visitor, Customer or Vendor):
 - o User signs up -> Data is stored in Sanity -> Confirmation sent to the user.
2. Product Append by the vendor:

After sign in or login user view a form to insert his product to the website for selling.
3. Product Browsing:
 - o User views product categories -> Sanity API fetches data -> Products displayed on frontend.
4. Order Placement:
 - o User adds items to the cart or order form -> Proceeds to checkout -> Order details saved in Sanity.
5. Shipment or delivery Tracking:
 - o Order status updates fetched via 3rd-party API -> Displayed to the user.

2. API Requirements

Based on my e-commerce platform data schema, here are the definitions of the API endpoints needed. Include:

Endpoint Name	Method	Description
/register	Post	Register a new user/customer/vendor
/login	Post	Authenticate existing user
/logout	Post	Sign out user

/product	Get	Display product details or catalog
/order	Get	Append new order
/order	Post	Retrieve Previous Orders.
/payment	Post	Process Payment gateway
/cart	Get	Add to cart fetching
/cart	Get	Retrieve Previous Carts
/shipment	Post	Add shipment details

3. Sanity Schema:

Category:

```
export default {  name:
'category',
  type: 'document',

  fields: [

    { name: 'category_id', type: 'string', title: 'Category ID' },

    { name: 'category_name', type: 'string', title: 'Category Name' },

    { name: 'category_image', type: 'image', title: 'Category Image' },

  ]
};
```

Product:

```
export default {  name:
'product',
  type: 'document',

  fields: [

    { name: 'product_id', type: 'string', title: 'Product ID' },

    { name: 'product_name', type: 'string', title: 'Product Name' },

    { name: 'product_description', type: 'string', title: 'Product Description' },

    { name: 'product_image', type: 'image', title: 'Product Image' },

    { name: 'rate', type: 'number', title: 'Rate' },

    { name: 'stock', type: 'number', title: 'Stock Level' }

  ]
};
```

Vendor:

```
export default {  name:
'vendor',
  type: 'document',

  fields: [

    { name: 'vendor_id', type: 'string', title: 'Vendor ID' },
```



```

    { name: 'vendor_name', type: 'string', title: 'Vendor Name' },
    { name: 'vendor_address', type: 'string', title: 'Vendor Address' },
    { name: 'vendor_email', type: 'string', title: 'Vendor Email' },
    { name: 'vendor_phone', type: 'string', title: 'Vendor Phone' },
  ]
};

```

Customer:

```

export default {  name:
'customer',
  type: 'document',
  fields: [
    { name: 'customer_id', type: 'string', title: 'Customer ID' },
    { name: 'customer_name', type: 'string', title: 'Customer Name' },
    { name: 'customer_address', type: 'string', title: 'Customer Address' },
    { name: 'customer_email', type: 'string', title: 'Customer Email' },
    { name: 'customer_phone', type: 'string', title: 'Customer Phone' },
  ]
};

```

Order:

```

export default {  name: 'order',
  type: 'document',
  fields: [
    { name: '_id', type: 'string', title: 'Order ID' },
    { name: 'customer_id', type: 'string', title: 'Customer ID' },
    { name: 'category_id', type: 'string', title: 'Category ID' },
    { name: 'product_id', type: 'string', title: 'Product ID' },
    { name: 'order_quantity', type: 'string', title: 'Order Quantity' },
    { name: 'product_id', type: 'string', title: 'Product ID' },
    { name: 'order_status', type: 'string', title: 'Order Status' },
  ]
};

```

```
};
```

delivery:

```
export default {  name:
```

```
'delivery',
```

```
  type: 'document',
```

```
  fields: [
```

```
    { name: '_id', type: 'string', title: 'Delivery ID' },
```

```
    { name: 'delivery_medium', type: 'string', title: ' Delivery Medium ' },
```

```
    { name: 'delivery_doc_no', type: 'string', title: ' Delivery Document No ' },
```

```
    { name: 'delivery_date', type: 'date', title: ' Delivery Date ' },
```

```
  ]
```

```
};
```

