


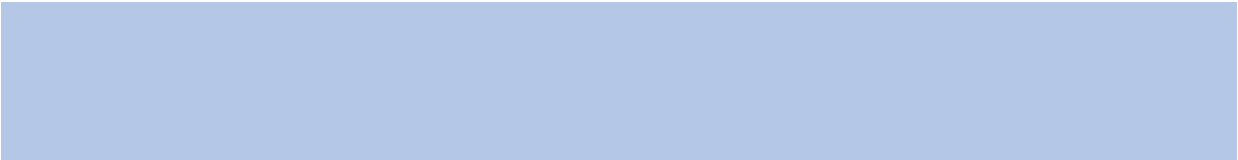





SELF-VALIDATION CHECKLIST FOR DAY 1 AND DAY 2



Prepared By: Zahida Raees



Self-Validation Checklist for Day 1 and Day 2

Day 1: Business Focus Outcome Checklist

Objective: Defining the business foundation of your marketplace.

Checklist:

Business Goals:

I have planned a small e-commerce website to sell all products that made up with Balochi embroidery such as chadars, shawls, scarves, dresses, purses, shoes, and jewelry. I believe this market could have significant potential in areas where Baloch people reside or where people appreciate Balochi traditions. Initially, the website will function as a testing platform, with our own stock being featured first. However, third-party vendors will also have the opportunity to list their products. A reasonable fee will be charged for listing items, but we will waive this fee for the first order.

1. **Market Research:**

Doch, known as embroidery in the Balochi language, holds a unique name and recognition within Baloch society across the world. As a Baloch it's also my own day to day experience to visit shops of Balochi embroidery in Karachi. So if I provide a platform for the small and large vendors to sell their products, it will grow soon. To reach them I will use all means like directly visiting their places or shops in nearest areas. Rest of the marketing will be done on social media. If I found any gap in my system I'll sure add them into it.

2. **Data Schema Draft:**

1. Venders:

Each vender will have a record in the `vendor` table, with the following fields:

- Vender ID
- Name
- NIC No
- Phone No
- Address

2. **Categories:**

Each product will belong to a specific category. These categories will be stored in a table named `category`, containing the following fields:

- Category ID
- Category Name
- Image representing the category

3. **Items:**

Each product will have a record in the `item` table, with the following fields:

- Item ID
- Name
- Description
- Size
- Stock Quantity
- Rate
- Discount
- Image
- Vendor ID
- Delivery Time
- Delivery Region

4. **Customers:**

Customer information will be stored in a `customer` table with the following fields:

- Customer ID (Auto-generated based on Purchase Record ID)
- Name
- Phone Number
- Address
- Email Address
- Preferred Payment Gateway

5. Orders:

The order form will include the following fields:

- Order ID
- Customer ID (with the customer name auto-filled based on the Customer table)
- Selected Category (displayed on-screen upon selection)
- Selected Item (displayed with its current stock quantity, basic rate, and discount)
- Order Quantity (calculated dynamically with the basic rate minus the discount, showing the net price)
- Delivery Charges (calculated based on the customer's postal address)
- Gross Total (net price + delivery charges)

6. Delivery Form:

The delivery form will include following fields:

- Order Number
- Delivery Medium
- Delivery Document Number
- Delivery Date

Day 2: Technical Planning Outcome Checklist

Objective: Transition to the technical foundation for my marketplace.

Checklist:

1. Technical Plan:

. Frontend:

- ❖ 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.

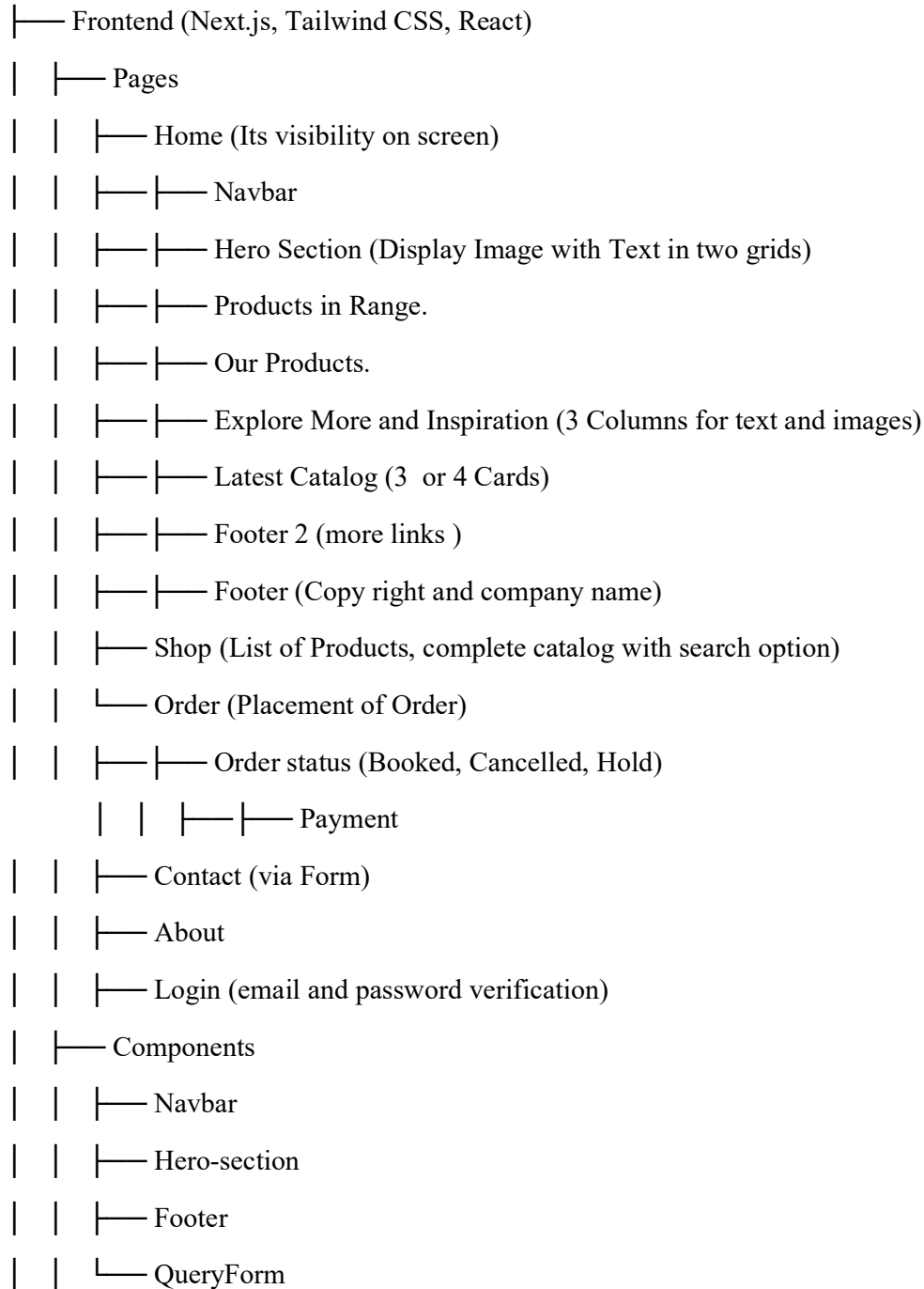
. Backend:

- 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.

System Architecture

A detailed architecture or a tree structure is here:

“Douch” E-commerce Project



- | └─ API Calls (Fetch data from Sanity)
- | └─ Backend (API Routes)
 - | └─ /api/submitQuery (Handles form submissions)
 - | └─ Twilio Integration (SMS functionality)
 - | └─ Nodemailer Integration (Email functionality)
 - | └─ Slack Notifications (Optional)
- | └─ Sanity CMS
 - | └─ Schemas
 - | | └─ GORQ Query (Stores Product form, Order Form, Customer Form, Vendor Form, Delivery Form submissions)
 - | | └─ Services (Manages services like photography options)
 - | | └─ Client (Optional for managing customer details)
 - | └─ Data Management (Manages all stored data)
- | └─ Tools and Integrations
 - | └─ Twilio (For SMS notifications)
 - | └─ Nodemailer (For email notifications)
 - | └─ Slack (For admin alerts)
- | └─ Environment Variables (.env.local)
 - | └─ Sanity API Token
 - | └─ Twilio Credentials
 - | └─ Email Credentials
 - | └─ Slack Token
- | └─ Deployment
 - | └─ Vercel (For deploying Next.js website)
 - | └─ Sanity Hosting (For CMS backend)
 - | └─ AWS Lambda (for backend)
 - | └─ Testing (Verification of all functionalities before launch)

2. Workflows:

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.

3. API Requirements

Based on my e-commerce platform data schema, here are the definition 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

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' },

  ]
};
```

3. Collaboration Notes:

Since most of the girls in my group are new to the technical field, they often need my help, and I do my level best to guide them. However, whenever they come across better videos or messages or receive help and guidance from someone else, they share it with me and the other girls in the group. For example, my marketing sense isn't very strong because I am primarily a technical person, but my friends' skills in this area have encouraged me to think differently.

Whether it's Quarter One or Quarter Two, we have worked together as a team, prepared together, and supported each other. If any friend lagged behind in any way, we made an effort to wake them from their slumber. And if they showed courage, we tried to move forward together with them.

The challenges I faced were greater when I genuinely needed technical help and couldn't find anyone around to assist me, whether in chat groups or Discord official groups. If my friends had sufficient knowledge, they would definitely share it because we all believe that knowledge grows when shared.