1. Marketplace Business Goals - Clothing Brand

THE PROBLEM:

In Pakistan, premium quality apparel often feels out of reach for many citizens. While numerous clothing brands offer high-quality garments, they tend to cater exclusively to the upper class or are designed solely for export markets. This leaves the majority of our people turning to thrift shops for access to premium garments—ironically, garments that were originally produced in Pakistan or rely on Pakistani resources.

As the **4th largest cotton producer** in the world and home to a **thriving textile industry**, Pakistan plays a pivotal role in the global fashion supply chain. Many countries depend on our cheap labor and raw materials to craft premium products, yet our own citizens struggle to access the same quality at reasonable prices.

My Solution:

I aim to leverage the incredible potential of Pakistan's textile industry and abundant raw materials to create a marketplace that delivers affordable, premium quality clothing. By bridging this gap, I aspire to make high-quality apparel accessible to people across all socio-economic classes, empowering them with style, comfort, and pride in locally produced fashion.

TARGET AUDIENCE:

In the initial phase, my marketplace will focus solely on addressing the needs of **local consumers** in Pakistan, ensuring they have access to affordable, premium-quality clothing that meets their expectations.

As the business grows, the goal is to **expand globally**, bringing the richness of Pakistan's textile industry to audiences around the world and establishing a strong presence in the international market.

Market Insights:

1. **Growing Demand for Affordable Quality:**
   Consumers are increasingly looking for affordable yet premium-quality garments, especially in regions with rising living costs.
2. **Export-Oriented Industry:** The Pakistani textile industry largely caters to export markets, leaving local consumers with limited options for premium, locally sourced clothing.

Competitor Analysis:

1. Brands like **Gul Ahmed, Khaadi, and Sapphire:** These brands provide high-quality clothing but target higher-income groups with premium pricing.
2. International fast-fashion brands like **Zara, H&M, and Uniqlo**: Although popular, these brands are often unaffordable for the average Pakistani due to high pricing and import costs.

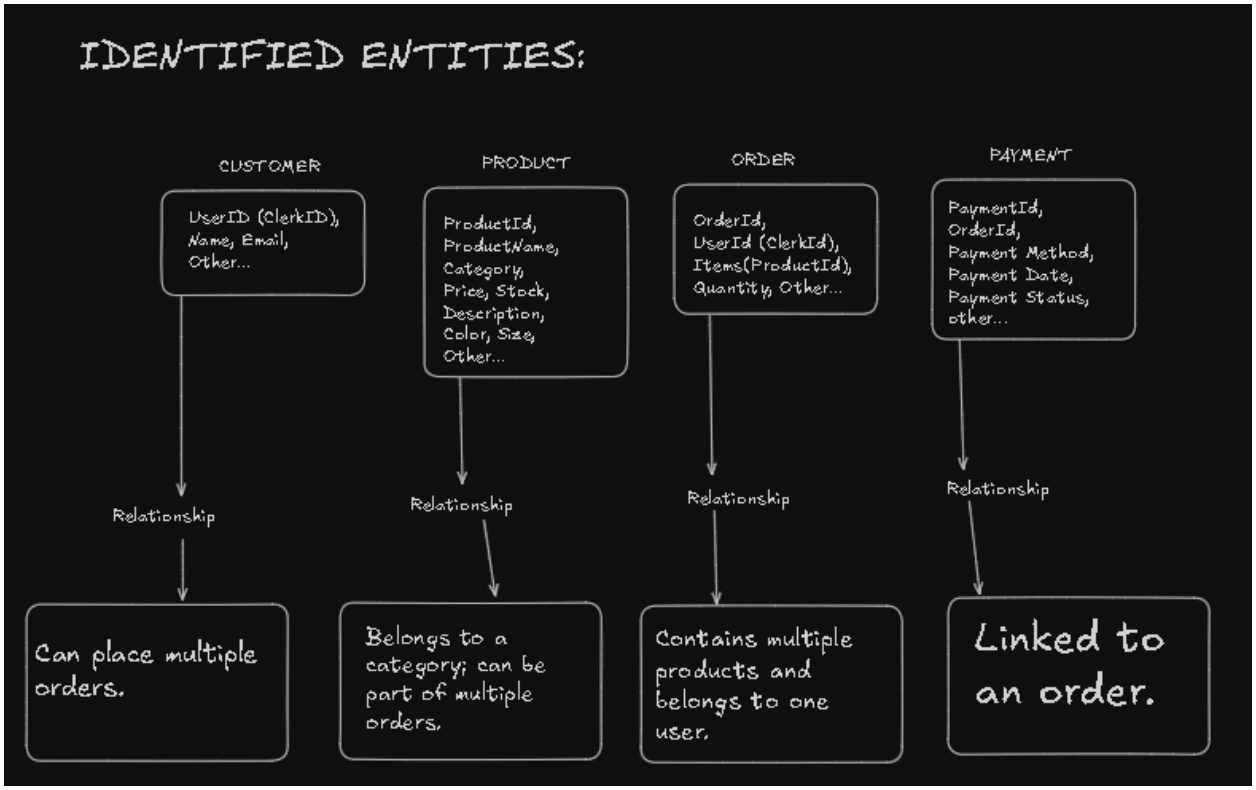
PRODUCTS:

Casual Wear: Comfortable everyday apparel for men, women, and children.

Formal Attire: Sophisticated and elegant clothing for professional and special occasions.

Activewear: High-quality, durable, and stylish options for fitness enthusiasts.

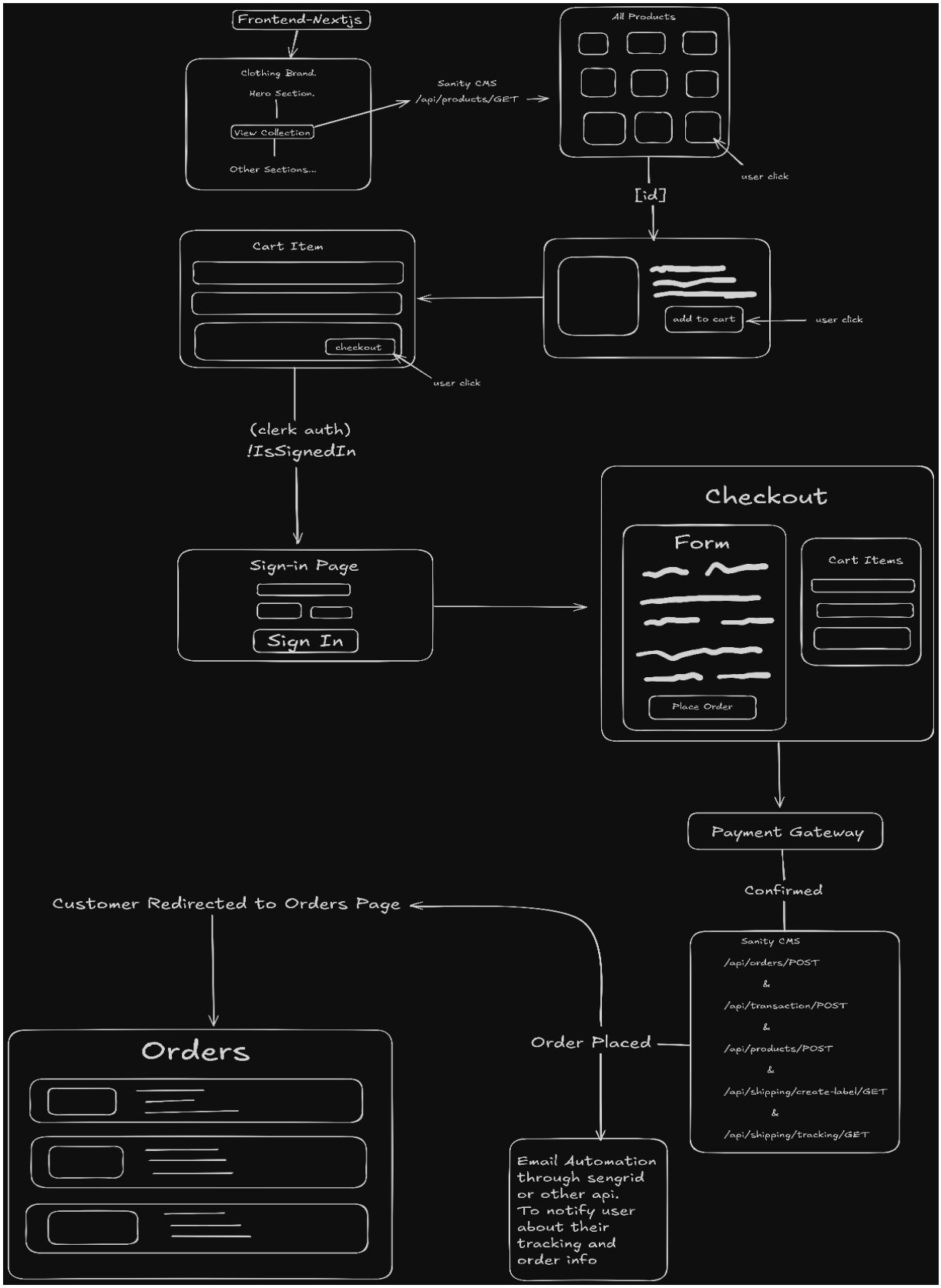
Accessories: Complementary items like scarves, shawls, and other garment-related essentials.



2. Marketplace Technical Foundation - Clothing Brand

---

2. System Architecture Overview:


Frontend-Nextjs
Clothing Brand.
Hero Section.
View Collection
Other Sections...
Sanity CMS
/api/products/GET
All Products
user click
[id]
Cart Item
checkout
user click
add to cart
user click
(clerk auth)
!IsSignedIn
Sign-in Page
Sign In
Checkout
Form
Cart Items
Place Order
Payment Gateway
Confirmed
Customer Redirected to Orders Page
Sanity CMS
/api/orders/POST
&
/api/transaction/POST
&
/api/products/POST
&
/api/shipping/create-label/GET
&
/api/shipping/tracking/GET
Orders
Order Placed
Email Automation
through sengrid
or other api.
To notify user
about their
tracking and
order info

1. **Key Workflows:**

1. User Arrival and Navigation:

- **Landing Page Experience:**
  The user lands on a clean, minimalist homepage with a Hero section that features a captivating banner image, a tagline, and a clear Call-to-Action (CTA) button, e.g., "View Collection."
  - **Header Navigation: Categories like "Men," "Women," "Kids," or "Accessories" are prominently displayed, allowing users to navigate directly to their desired sections.**
  - **User-Friendly Search Bar: Users can also search for specific items using keywords.**

2. Browsing Products:

- **Product Listing Page:**
  Upon clicking "View Collection," the user is redirected to the product listing page, where:
  - **Products are neatly categorized by type.**
  - **Each category has a "View All" button, allowing users to explore the complete collection of that section.**

3. Product Exploration:

- **Product Details Page:**
  When a user clicks on a product:
  - **High-quality images of the product in different colors and angles are displayed.**
  - **The user selects their preferred color, and available sizes for that color dynamically appear beneath it, along with real-time stock availability.**
  - **After choosing the color, size, and quantity, they click "Add to Cart," seamlessly adding the item to their cart.**

4. Cart Management:

- **Cart Page:**
  Clicking the Cart Icon in the header navigates the user to the cart page, which includes:

- **A list of all added items, each with a thumbnail, selected color, size, and quantity.**
- **Options to increment or decrement item quantity or remove items altogether.**
- **A “Proceed to Checkout” button to continue the purchase process.**

5. Checkout Process:

- **Checkout Page:**
  The user fills out a detailed form, including:
  - **Name, email, phone number, address, city, postal code, and state.**
  - **Beside the form, a Sticky Summary Section displays:**
    - **Items in the cart, along with their details.**
    - **The subtotal amount, updated in real-time.**
- **Payment Options:**
  Underneath the form, the user selects a payment method:
  - **Cash on Delivery (COD): The user selects COD as the payment method and clicks the “PLACE ORDER” button to confirm their order.**
  - **Bank Transfer: Initiates a secure payment gateway where the user completes the payment.**
- **Order Confirmation:**
  - **After payment confirmation, the system places the order, showers the screen with celebratory confetti, and redirects the user to the Orders Page.**

6. Post-Purchase Experience:

- **Orders Page:**
  Users can track their orders in real-time, with details such as:
  - **Items purchased (with selected color, size, and quantity).**
  - **Shipping status with live tracking.**
- **Confirmation Email:**
  A confirmation email is sent, including:
  - **Order ID and Tracking ID.**
  - **Thank you note and support contact details.**

3. API Requirements:

Endpoint	Method	Description	Payload Example	Response Example
/api/products	GET	Retrieve all products.	N/A	[{ "id": 1, "name": "T-Shirt", "category": "Casual", "price": 1200, variations: [{color: red, size: medium, stock: 50}] }]
/api/products/:id	GET	Retrieve details of a specific product.	N/A	{ "id": 1, "name": "T-Shirt", "category": "Casual", "price": 1200, variations: [{color: red, size: medium, stock: 50}] }
/api/products	POST	Add a new product (Admin).	{ "name": "T-Shirt", "category": "Casual", "price": 1200, variations: [{color: red, size: medium, stock: 50}] }	{ "message": "Product created successfully", "id": 101 }
/api/orders	POST	Proceed with posting orders.	{ "userId": 1, "cartItems": [{ "productId": 1, "quantity": 2 }], "address": "Street 123, Karachi",	{ "orderId": 5001, "message": "Order placed successfully", "trackingId": "TRACK1234" }

Endpoint	Method	Description	Payload Example	Response Example
			`"paymentMethod": "COD" }`	
`/api/orders`	`GET`	Retrieve all orders for a user.	N/A	`[ { "orderId": 5001, "status": "Shipped", "items": [ { "productId": 1, "name": "T-Shirt", "quantity": 2, "price": 1200 } ] } ]`

4. **Sanity Schema Design:**

• I will use sanity CMS as a database for my marketplace,
It will store data such as:

1. Products,
2. Orders Records,
3. Customers Details,
4. Transactions Records,
5. Shipment Records,

• [🏷️ Products Schema]:

```
import { defineType, defineField } from "sanity";

export default defineType({
  name: "product",
  title: "Product",
  type: "document",
  fields: [
    defineField({
      name: "id",
      title: "ID",
      type: "string",
```

```
  description: "A unique identifier for the product.",
}),
defineField({
  name: "name",
  title: "Product Name",
  type: "string",
  description: "The name of the product.",
}),
defineField({
  name: "price",
  title: "Price",
  type: "string",
  description: "The price of the product.",
}),
defineField({
  name: "images",
  title: "Product Images",
  type: "array",
  of: [{ type: "image" }],
  description: "Images of the product.",
}),
defineField({
  name: "ratings",
  title: "Ratings",
  type: "number",
  description: "The average ratings of the product (e.g., '5.0').",
}),
defineField({
  name: "tags",
  title: "Tags",
  type: "array",
  of: [{ type: "string" }],
  description: "Tags to categorize the product (e.g., 'Best Selling').",
}),
defineField({
  name: "discountPercentage",
  type: "number",
  title: "Discount Percentage",
}),
defineField({
  name: "priceWithoutDiscount",
  type: "number",
  title: "Price Without Discount",
  description: "Original price before discount",
```

```
}),
defineField({
  name: "ratingCount",
  type: "number",
  title: "Rating Count",
  description: "Number of ratings",
}),
defineField({
  name: "description",
  title: "Description",
  type: "text",
  description: "A detailed description of the product.",
}),
defineField({
  name: "variations",
  title: "Product Variations",
  type: "array",
  of: [
    defineField({
      name: "variation",
      type: "object",
      fields: [
        {
          name: "color",
          title: "Color",
          type: "string",
          description: "Color of the product variation.",
        },
        {
          name: "size",
          title: "Size",
          type: "string",
          description: "Size of the product variation.",
        },
        {
          name: "quantity",
          title: "Quantity",
          type: "number",
          description: "Available quantity for this variation.",
        },
      ],
    }),
  ],
  description: "List of variations for the product, including size, color, and quantity.",
```

```
    }),
  ],
});
```

• [🤝 Customer Schema]:

```
import { defineType, defineField } from 'sanity'

export default defineType({
  name: 'customer',
  title: 'Customer',
  type: 'document',
  fields: [
    defineField({
       name: 'customerId',
       title: 'Customer ID',
       type: 'string',
       readOnly: true,
    }),
    defineField({
      name: 'name',
      title: 'Name',
      type: 'string',
    }),
    defineField({
      name: 'email',
      title: 'Email',
      type: 'string',
    }),
    defineField({
      name: 'phone',
      title: 'Phone',
      type: 'string',
    }),
    defineField({
      name: 'address',
      title: 'Address',
      type: 'object',
      fields: [
        { name: 'street', type: 'string', title: 'Street' },
        { name: 'city', type: 'string', title: 'City' },
        { name: 'state', type: 'string', title: 'State' },
        { name: 'zipCode', type: 'string', title: 'Zip Code' },
```

```
      { name: 'country', type: 'string', title: 'Country' },
    ],
  }),
 ],
});
```

• [📝 Order Schema]:

```
import { defineType, defineField } from 'sanity'

export default defineType({
 name: 'order',
 title: 'Order',
 type: 'document',
 fields: [
  defineField({
   name: 'orderId',
   title: 'Order ID',
   type: 'string',
  }),
  defineField({
   name: 'customer',
   title: 'Customer',
   type: 'reference',
   to: [{ type: 'customer' }],
  }),
  defineField({
   name: 'items',
   title: 'Items',
   type: 'array',
   of: [{ type: 'reference', to: [{ type: 'product' }] }],
  }),
  defineField({
   name: 'totalAmount',
   title: 'Total Amount',
   type: 'number',
  }),
  defineField({
   name: 'status',
   title: 'Order Status',
   type: 'string',
   options: {
    list: [
```

```
        { title: 'Pending', value: 'pending' },
        { title: 'Processing', value: 'processing' },
        { title: 'Shipped', value: 'shipped' },
        { title: 'Delivered', value: 'delivered' },
      ],
    },
  }),
  defineField({
    name: 'shippingAddress',
    title: 'Shipping Address',
    type: 'object',
    fields: [
      { name: 'street', type: 'string', title: 'Street' },
      { name: 'city', type: 'string', title: 'City' },
      { name: 'state', type: 'string', title: 'State' },
      { name: 'zipCode', type: 'string', title: 'Zip Code' },
      { name: 'country', type: 'string', title: 'Country' },
    ],
  }),
  defineField({
    name: "shipment",
    title: "Shipment",
    type: "reference",
    to: [{ type: "shipment" }], // Reference to shipment schema
  }),
 ],
})
```

• [🚚 Shipment Schema]:

```
import { defineType, defineField } from 'sanity'

export default defineType({
  name: 'shipment',
  title: 'Shipment',
  type: 'document',
  fields: [
    defineField({
      name: 'tracking_id',
      title: 'Tracking ID',
      type: 'string',
    }),
```

```
defineField({
  name: "order",
  title: "Order",
  type: "reference",
  to: [{ type: "order" }], // Reference to order schema
}),
defineField({
  name: 'shipment_status',
  title: 'Shipment Status',
  type: 'string',
  options: {
    list: [
      { title: 'Pending', value: 'pending' },
      { title: 'In Transit', value: 'in_transit' },
      { title: 'Delivered', value: 'delivered' },
    ],
  },
}),
defineField({
  name: 'estimated_delivery_date',
  title: 'Estimated Delivery Date',
  type: 'date',
}),
defineField({
  name: 'carrier',
  title: 'Carrier',
  type: 'string',
}),
defineField({
  name: 'shipment_origin',
  title: 'Shipment Origin',
  type: 'string',
}),
defineField({
  name: 'shipment_destination',
  title: 'Shipment Destination',
  type: 'string',
}),
defineField({
  name: "customer",
  title: "Customer",
  type: "reference",
  to: [{ type: "customer" }], // Reference to customer schema
}),
```

```
  ],
})
```

• [💳 Transaction Schema]:

```
export default {
    name: 'transaction',
    type: 'document',
    title: 'Transaction',
    fields: [
      {
       name: "order",
       type: "reference",
       to: [{ type: "order" }], // Reference to order schema
       title: "Order",
       description: "The order associated with this transaction",
      },
      {
       name: 'user',
       type: 'reference',
       to: [{ type: 'customer' }], // Reference to a customer schema
       title: 'User',
       description: 'User who made the transaction',
      },
      {
       name: 'productDetails',
       type: 'array',
       of: [{ type: 'reference', to: [{ type: 'product' }] }], // Reference to a product schema
       title: 'Product Details',
       description: 'Products purchased in the transaction',
      },
      {
       name: 'amount',
       type: 'number',
       title: 'Amount',
       description: 'Total amount of the transaction',
      },
      {
       name: 'paymentStatus',
```

```
      type: 'string',
      title: 'Payment Status',
      options: {
        list: [
          { title: 'Success', value: 'success' },
          { title: 'Pending', value: 'pending' },
          { title: 'Failed', value: 'failed' },
        ],
      },
    },
    {
      name: 'transactionDate',
      type: 'datetime',
      title: 'Transaction Date',
    },
  ],
};
```

● Summary of Schema Relationships:

• Product:
Referenced by order and transaction.

• Customer:
Referenced by order, shipment, and transaction.

• Order:
References customer, product, and shipment.
Referenced by shipment and transaction.

• Shipment:
References order and customer.

• Transaction:
References order, customer, and product.

Collaboration Notes (Solo Project)

1. **Problem I Identified:**
    - **The main problem I noticed in the current marketplace for clothing and garments in Pakistan is the unavailability of affordable premium quality apparel for all socio-economic classes.**
    - **Despite Pakistan being one of the largest producers of cotton and home to a thriving textile industry, local consumers struggle to find high-quality clothing that is reasonably priced.**
    - **The lack of access to premium quality garments forces many people to resort to thrift shops or imported goods, while locally produced, high-quality garments are largely reserved for export markets or the upper class.**

My Solution:

- **I created this marketplace to bridge the gap and offer affordable, high-quality clothing to people from all walks of life, especially in Pakistan.**
- **Leverage local resources: By tapping into Pakistan's robust textile industry and raw materials, the goal is to lower production costs, thus making quality garments affordable for everyone.**
- **The marketplace will offer timeless designs and premium fabrics at accessible prices, ensuring that everyone, regardless of their socio-economic background, can have access to fashionable, well-made garments.**

Day 1: Business Focus Outcome Checklist Objective: Define the business foundation of your marketplace.

Checklist:

- ☑ **1. Business Goals:**
- ☑ **2. Market Research**
- ☑ **3. Data Schema Draft**
- ☑ **4. Submission from Day 1**

Day 2: Technical Planning Outcome Checklist Objective: Transition to the technical foundation for your marketplace.

Checklist:

- ☑ **1. Technical Plan**
- ☑ **2. Workflows**
- ☑ **3. API Requirements**
- ☑ **Sanity Schema**
- ☑ **5. Collaboration Notes**
- ☑ **6. Submission from Day 2**