# Day1\_Laying\_the\_Foundation\_for\_Your\_

# Marketplace\_Journey

**Market place Type:  
(General E-Commerce)**

**Business Goals:  
  
What problem does my marketplace aim to solve?**

· **Convenience:** A one-stop shop for a variety of products across different categories, reducing the need to visit multiple stores.

· **Affordability:** Providing high-quality products at competitive prices to meet the needs of budget-conscious customers.

· **Delivery speed:** Ensuring timely delivery within 2–5 business days, depending on the source of the product.

· **Ease of returns:** Introducing a hassle-free 7-day return policy to build trust and customer satisfaction.

**Who is My Traget Audience?**

**Geographic Focus:**

* Primary: Karachi and other major cities in Pakistan.
* Gender: All genders
* Ages:all ages

**What products or services will be offer?**

**Services:**

* Cash-on-Delivery (COD).
* Express shipping (local and international).
* 7-day hassle-free return policy.

**How will I offer te product and service?**

**Platform:**

* A user-friendly e-commerce website and mobile app with an intuitive search, filter, and payment system.

· **Payment Methods:**

* · Multiple payment options, including COD, debit/credit cards, and mobile wallets like Easypaisa and JazzCash.

· **Customer Support:**

* · 24/7 online chat for queries and returns.

**What is the unique selling point of my Market-place?**

· **Speedy Delivery:**

* · Products from Karachi are delivered within **2–3 business days**.
* International products (from China) are delivered within **4–5 business days**, much faster than many competitors.

· **Affordability:**

* · Lower prices without compromising quality, making the marketplace accessible to a broader audience.

· **Trust & Convenience:**

* · Easy 7-day return policy and multiple payment options build trust and ensure convenience.

**Access to Imported Goods:** Reliable and fast access to products from China, eliminating customer worries about delays and high costs.

**Speedy Delivery in Karachi:**

**Product from karachi 2 to 3 business days**

**Product from overseas (china) 4 to 5 business days  
  
Affordability:  
Lower prices and best products**

· **Competitive Pricing:** Offering lower prices compared to other marketplaces.

· **Product Quality:** Ensuring every product passes quality checks to reduce dissatisfaction and returns.

**Return Policy:  
Esay return in 7 days after requesting**

# Day2\_Planning\_the\_Technical\_

# Foundation

### ****1. Define Technical Requirements****

#### ****Frontend Requirements****

**User-Friendly Interface**

* 1. Framework: Use **Next.js** for fast rendering and SEO-friendly pages.
  2. CSS: Use **Tailwind CSS, Bootstrap, Lucid react, React Icons** for styling and responsiveness.
  3. State Management: Use **Redux Toolkit** or **Context API** to handle the global state (cart, wishlist, etc.).

**Responsive Design**

* 1. Ensure compatibility across all devices using media queries or Tailwind's responsive utilities.
  2. Test using tools like Google Chrome's device emulation.

**Essential Pages**

* 1. **Home:** Showcase featured products and promotions.
  2. **Product Listing:** Use category-based filters like Men, Women, and Sale.
  3. **Search:** Implement a real-time product search bar.
  4. **Wishlist:** Save favorite products using local storage or user profile data.
  5. **Cart and Checkout:** Include options for promo codes, order summaries, and payment.
  6. **Order Confirmation:** Display order details after a successful purchase.
  7. **Find a Store/ Pick From Store:** Use a location API to help users find nearby stores.
  8. **Need help? Chat With Us!:**
  9. **FAQ:**
  10. **Return & Exchnage:**
  11. **Profile/My Account:**
  12. **Gift Cards**
  13. **About Us**
  14. **Customer Service**
  15. **Our Policies**

#### ****Backend Requirements (Sanity CMS)****

* Use **Sanity CMS** to handle dynamic content, such as product information and blog posts.
* Define schema for the following:
  + **Product Data:** SKU, images, price, sizes, and stock status.
  + **Customer Data:** User profiles, order history, and saved addresses.
  + **Order Data:** Cart items, payment status, and shipping details.
  + **Shipment Data:** Tracking numbers, delivery status, and estimated delivery dates.
  + **Delivery Data:** delivery status
  + **Payment Data:**

Third-Party APIs to Integrate for Pakistan:

(Shipment tracking, payment gateways, sms/mail notification,authentication)

1. Payment Gateways (PKR Support)

**PayFast by APPS**

* Locally available gateway supporting PKR.
* Handles Visa, Mastercard, UnionPay, JazzCash, and Easypaisa transactions.

1. Shipping & Delivery

**TCS API**

* Provides shipment management, tracking, and delivery services across Pakistan.
* Contact TCS for API integration details.

1. Geolocation for Store Finder

**Google Maps API**

* Use Google Maps for accurate geolocation, store locators, and route planning.
* Set up a free or paid API key for integration.

#### **d. Authentication**

**Firebase Authentication**

* 1. Free tier available; allows user authentication via email, Google, or Facebook logins.

1. Real-Time Search

**Sanity CMS:**

Sanity provides a flexible API with filtering, querying, and sorting capabilities that can handle most eCommerce search functionalities.

#### ****f. Email Notifications****

**SendGrid**

* 1. Supports transactional emails like order confirmations and promotions.
  2. Free tier available.

**Mailchimp API**

* 1. Excellent for email marketing campaigns and customer engagement.
  2. Website

**Local SMS/Email Providers**

* 1. Look for local providers that support email and SMS notifications in PKR, such as **ClickSend** or **Zong BizStore**.

#### ****g. Analytics****

**Google Analytics**

* 1. Free tool for tracking user activity, behavior, and sales performance.
  2. [Documentation](https://analytics.google.com/" \t "_new)

**Hotjar**

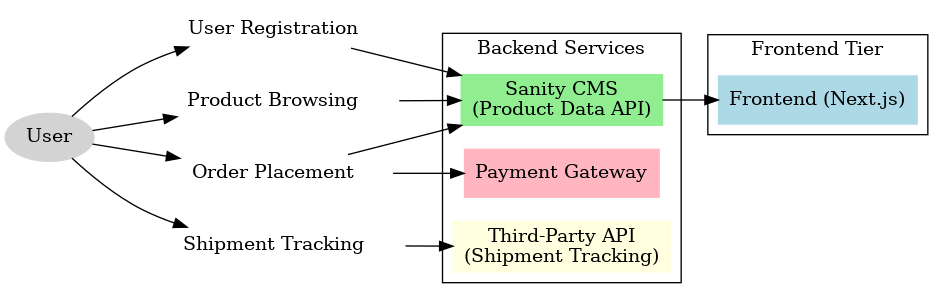
* 1. Provides insights into user interaction via heatmaps and recordings.
  2. [Website](https://www.hotjar.com/" \t "_new)

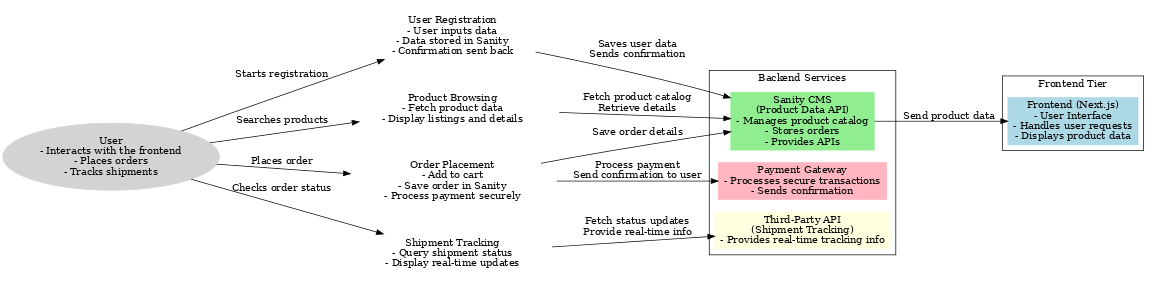
#### ****h. CMS API****

1. **Sanity CMS**
   1. Use Sanity's API to manage product, order, and customer data dynamically.
   2. Website

**DESIGN SYSTEM ARCHITECTURE:**

Highlevel diagram shoing how the system componets interacts and overall the uml diagram for this ecommerce clothing brand website working including the (frontend tier (next.js), sanity CMS with Product Data API, Third Party API with Shipment tracking API, Payment Gateway

****

****

1. **PLAN API REQUIREMENTS:**

**Based on the data schema, define API endpoints needed.**

### ****1. Core API Endpoints****

These are the core APIs needed to interact with the Sanity CMS and manage basic eCommerce workflows.

#### ****Product Management****

**Endpoint Name**: /products

* + **Method**: GET
  + **Description**: Fetch all available products.
  + **Query Parameters** (optional):
    - category (e.g., Men, Women, Sale)
    - search (keyword for filtering)
    - sort (price, popularity)
  + **Response Example**:

[

{

"id": "product-123",

"name": "Product A",

"price": 1000,

"stock": 20,

"image": "url-to-image",

"category": "Men"

}

]

**Endpoint Name**: /products/:id

* + **Method**: GET
  + **Description**: Fetch details of a specific product.
  + **Response Example**:

{

"id": "product-123",

"name": "Product A",

"description": "High-quality product",

"price": 1000,

"stock": 20,

"images": ["url-to-image1", "url-to-image2"],

"category": "Men"

}

#### ****Order Management****

**Endpoint Name**: /orders

* + **Method**: POST
  + **Description**: Create a new order.
  + **Payload Example**:

{

"customer": {

"name": "John Doe",

"email": "john@example.com",

"phone": "+923001234567"

},

"products": [

{

"id": "product-123",

"quantity": 2

},

{

"id": "product-456",

"quantity": 1

}

],

"paymentStatus": "pending",

"shipmentAddress": "Karimabad, Karachi"

}

* + **Response Example**:

{

"orderId": "order-123",

"status": "created",

"paymentStatus": "pending"

}

**Endpoint Name**: /orders/:id

* + **Method**: GET
  + **Description**: Get order details.
  + **Response Example**:

{

"orderId": "order-123",

"status": "shipped",

"products": [

{ "id": "product-123", "name": "Product A", "quantity": 2 },

{ "id": "product-456", "name": "Product B", "quantity": 1 }

],

"shipmentStatus": "in-transit"

}

#### ****Search****

* **Endpoint Name**: /search
  + **Method**: GET
  + **Description**: Search products dynamically using Sanity's API filtering.
  + **Query Parameters**:
    - query: Search keyword.
    - filter: Additional filters (e.g., category, price range).
  + **Response Example**:

[

{

"id": "product-123",

"name": "Product A",

"price": 1000,

"stock": 20,

"image": "url-to-image"

}

]

### ****2. Third-Party API Integrations****

These endpoints interact with third-party services for specific functionalities.

#### ****Payment Gateways (PayFast by APPS)****

* **Endpoint Name**: /payment
  + **Method**: POST
  + **Description**: Initiates a payment request via PayFast.
  + **Payload Example**:

{

"orderId": "order-123",

"amount": 3000,

"currency": "PKR",

"method": "JazzCash"

}

* + **Response Example**:

{

"paymentId": "payment-123",

"status": "pending",

"redirectUrl": "https://payfast.pk/payment-redirect"

}

#### ****Shipping & Delivery (TCS API)****

* **Endpoint Name**: /shipment
  + **Method**: POST
  + **Description**: Create a shipment with TCS.
  + **Payload Example**:

{

"orderId": "order-123",

"recipientName": "John Doe",

"address": "Karimabad, Karachi",

"phone": "+923001234567"

}

* + **Response Example**:

{

"shipmentId": "tcs-123",

"trackingUrl": "https://tcs.com/tracking?tcs-123",

"status": "created"

}

#### ****Shipment Tracking****

* **Endpoint Name**: /shipment/:id
  + **Method**: GET
  + **Description**: Fetch shipment tracking details.
  + **Response Example**:

{

"shipmentId": "tcs-123",

"status": "in-transit",

"expectedDelivery": "2025-01-20"

}

### ****3. Additional Features****

#### ****Authentication (Firebase)****

* **Endpoint Name**: /auth/register
  + **Method**: POST
  + **Description**: Register a new user.
  + **Payload Example**:

{

"email": "john@example.com",

"password": "securepassword123",

"name": "John Doe"

}

#### ****Email Notifications (SendGrid)****

* **Endpoint Name**: /email/send
  + **Method**: POST
  + **Description**: Send email notifications (order confirmation, promotions).
  + **Payload Example**:

{

"to": "john@example.com",

"subject": "Order Confirmation",

"body": "Thank you for your order!"

}

### ****4. Analytics Integration****

#### ****Google Analytics****

Google Analytics is ideal for tracking user activities, including page views, product clicks, checkout processes, and more.

Configure custom events to track specific actions:

* Example for product clicks:

window.gtag('event', 'product\_click', {

category: 'Products',

label: 'Product Name',

value: 'Product ID',

});

#### ****Hotjar****

Hotjar is a powerful tool for heatmaps, session recordings, and user feedback.

### ****5. Real-Time Search****

Since Sanity CMS provides filtering, sorting, and querying capabilities, real-time search can be achieved without the need for external tools like Meilisearch or ElasticSearch.

**Search Endpoint**: Use the GROQ (Graph-Relational Object Queries) query language to define the search query in Sanity.

* 1. Example:

**const query = `\*[\_type == "product" && name match $keyword]{**

**\_id,**

**name,**

**price,**

**image**

**}`;**

**const params = { keyword: "shirt" };**

**const products = await sanityClient.fetch(query, params);**

### ****6. Notifications****

#### ****SMS Notifications (Zong BizStore)****

1. Contact Zong BizStore for API access.
2. Use their API to send SMS updates:
   1. Example payload:

{

"phone": "+923001234567",

"message": "Your order has been shipped. Track it here: https://tracking-url.com"

}

**SANITY SCHEMA EXAMPLE;**

### ****1. Products Schema****

export default {

name: 'product',

type: 'document',

title: 'Product',

fields: [

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

{ name: 'price', type: 'number', title: 'Price' },

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

{ name: 'category', type: 'string', title: 'Category' },

{ name: 'description', type: 'text', title: 'Description' },

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

],

};

### ****2. Orders Schema****

export default {

name: 'order',

type: 'document',

title: 'Order',

fields: [

{ name: 'customer', type: 'reference', to: [{ type: 'customer' }], title: 'Customer' },

{ name: 'products', type: 'array', of: [{ type: 'reference', to: [{ type: 'product' }] }], title: 'Products' },

{ name: 'totalAmount', type: 'number', title: 'Total Amount' },

{ name: 'status', type: 'string', title: 'Order Status', options: { list: ['Pending', 'Shipped', 'Delivered', 'Cancelled'] } },

{ name: 'createdAt', type: 'datetime', title: 'Order Date' },

],

};

### ****3. Customers Schema****

export default {

name: 'customer',

type: 'document',

title: 'Customer',

fields: [

{ name: 'name', type: 'string', title: 'Customer Name' },

{ name: 'email', type: 'string', title: 'Email' },

{ name: 'phone', type: 'string', title: 'Phone Number' },

{ name: 'address', type: 'text', title: 'Address' },

],

};

### ****4. Payments Schema****

export default {

name: 'payment',

type: 'document',

title: 'Payment',

fields: [

{ name: 'order', type: 'reference', to: [{ type: 'order' }], title: 'Order' },

{ name: 'amount', type: 'number', title: 'Amount' },

{ name: 'paymentMethod', type: 'string', title: 'Payment Method', options: { list: ['Credit Card', 'Easypaisa', 'JazzCash', 'Bank Transfer'] } },

{ name: 'status', type: 'string', title: 'Payment Status', options: { list: ['Pending', 'Completed', 'Failed'] } },

{ name: 'transactionId', type: 'string', title: 'Transaction ID' },

{ name: 'paymentDate', type: 'datetime', title: 'Payment Date' },

],

};

### ****5. Shipments Schema****

export default {

name: 'shipment',

type: 'document',

title: 'Shipment',

fields: [

{ name: 'order', type: 'reference', to: [{ type: 'order' }], title: 'Order' },

{ name: 'trackingId', type: 'string', title: 'Tracking ID' },

{ name: 'carrier', type: 'string', title: 'Carrier', options: { list: ['TCS', 'DHL', 'FedEx', 'Leopard'] } },

{ name: 'status', type: 'string', title: 'Shipment Status', options: { list: ['In Transit', 'Delivered', 'Pending', 'Returned'] } },

{ name: 'estimatedDelivery', type: 'datetime', title: 'Estimated Delivery Date' },

],

};

### ****6. Delivery Schema****

export default {

name: 'delivery',

type: 'document',

title: 'Delivery',

fields: [

{ name: 'order', type: 'reference', to: [{ type: 'order' }], title: 'Order' },

{ name: 'deliveryPerson', type: 'string', title: 'Delivery Person Name' },

{ name: 'deliveryDate', type: 'datetime', title: 'Delivery Date' },

{ name: 'deliveryStatus', type: 'string', title: 'Delivery Status', options: { list: ['Delivered', 'Failed', 'Rescheduled'] } },

{ name: 'customerFeedback', type: 'text', title: 'Customer Feedback' },

],

};

# Day\_3\_API\_Integration\_and\_Data\_Migration

**STEPS FOR DAY3:**

**UNDERSTAND THE PROVIDED API:**

**VALIDATE AND ADJUST YOUR SCHEMA:**

**DATA MIGRATION OPTIONS:**

**API INTEGRATION IN NEXT.JS:**