

Marketplace Technical Foundation - E-Commerce Shop.co.pk

Purpose: This document outlines the technical foundation for Shop.co.pk, a premium ecommerce platform offering high-quality Pakistani products and top local brands to both local and international buyers.

1. System Architecture Overview

A high-level architecture diagram illustrating interactions between the frontend (Next.js), Sanity CMS, and third-party APIs (e.g., shipment tracking, payment gateways, and currency conversion APIs) will be included.

Architecture Components:

Frontend (Headless UI):

- Built with Next.js for dynamic rendering, localization, and SEO optimization.
- Styled using Tailwind CSS, DaisyUI, and Font Awesome for modern, responsive, and user-friendly interfaces.

Sanity CMS:

- Manages product catalog, categories, orders, and customer profiles dynamically.
- Handles multiple currencies and supports product details in different languages (e.g., English, Urdu, Arabic).

Third-Party APIs:

- Shipping Providers: Integration with Pakistani shipping services (e.g., Leopard Courier, TCS, BlueEx) and international carriers (e.g., DHL, FedEx, Aramex) for worldwide delivery tracking.

- Payment Gateways: Supports JazzCash, EasyPaisa, credit/debit cards, PayPal, and other global payment solutions for international buyers.
- Currency Conversion: Real time exchange rates via APIs (e.g., Fixer.io, OpenExchangeRates) to display prices in USD, GBP, EUR, and other major currencies.

2. Key Workflows

User Registration (Local & International Buyers):

1. Users (local or international) visit the registration page.
2. Fill in details (name, email, phone number, country, and password).
3. Data is sent to the authentication service.
4. Upon successful registration, users receive a confirmation email/SMS and can log in.

Product Browsing & Viewing:

1. Dynamically fetches product data, including high-quality images, descriptions, pricing, and stock availability, from Sanity CMS.
2. Displays product details in multiple currencies based on the user's location or selection.
3. Allows users to filter by brand, price range, categories, and shipping availability.
4. Provides multilingual support for product descriptions (e.g., English, Urdu, Arabic).

Order Placement (Local & International):

1. Users add products to the cart with selected quantities and preferences (e.g., size, color).
2. Checkout adapts based on user location:
 - Local Buyers: Payment via JazzCash/EasyPaisa or cash on delivery.
 - International Buyers: Payment via credit cards, PayPal, or international gateways.
3. Order data (user details, product IDs, quantities, shipping address, and total amount) is stored in Sanity CMS.

4. API triggers an order confirmation email/SMS and provides a delivery timeline.

Global Shipment Tracking:

- 1. Shipping details, including tracking ID, are sent to the third-party API (local or international carrier).
- 2. International users can view real-time delivery status in their preferred language and currency.

3. API Requirements

Endpoint	Method	Payload	Response
/products	GET	-	List of products
/product/{id}	GET	Product ID	Product details
/order	POST	User info, product IDs, quantities, country	Order confirmation
/shipment/{orderId}	GET	Order ID	Real-time tracking info
/currency-conversion	GET	Currency code (e.g., USD)	Exchange rate
/languages/{langCode}	GET	Language code (e.g., en, ur)	Translated content

4. Sanity Schema Design

Product Schema:

- 1. title: string
- 2. description: text (supports multiple languages)
- 3. price: number (base currency in PKR)
- 4. brand: string (e.g., Khaadi, Gul Ahmed, Outfitters)
- 5. category: string
- 6. image: URL

7. stock: number
8. shippingAvailability: array (e.g., Pakistan, US, UK)

Order Schema:

1. user: reference
2. products: array
3. totalPrice: number (calculated in user's currency)
4. status: string (Pending, Shipped, Delivered)
5. country: string (used for shipping and taxes)

5. Error Handling in Sanity Queries

1. Implement try...catch: Handles API request failures gracefully.
2. User-Friendly Messages:
 - For local users: "Unable to load products. Please try again later."
 - For international users: "We're experiencing connectivity issues. Please check back shortly."

6. Sanity Product Upload Script

1. Develop a Node.js script to bulk upload product data, including multi-language descriptions, pricing, and shipping details, to Sanity CMS.
2. Ensure proper validation for pricing and availability for global markets.

7. Collaboration Notes

1. Collaborated with team members to finalize schemas for local and international requirements.
2. Integrated feedback to improve UX for currency conversion and multilingual support.
3. Debugged API integrations for real-time tracking and international payment gateways.

8. File Format & Submission

1. Format: The document will be submitted in PDF format.
2. Content: Includes high-resolution diagrams for architecture, workflows, and multilingual support.
3. Repository: Uploaded to the Documentation folder with proper naming conventions.