

## Technical Plan for Shop.co

### 1. Website Purpose

Shop.co ek e-commerce platform hai jo clothing brands ke products sell karne ke liye design ki gayi hai.

### 2. Frontend Technologies

- **Framework:** Next.js
- **Styling:** Tailwind CSS
- **UI Library:** React
- **Features:**
  - Responsive design
  - Pixel-perfect layout

### 3. Features

- **Product Listing:**
  - Different categories for clothing brands
  - Filters for price, size, color, etc.
- **Search Functionality:**
  - Keyword-based product search
- **Product Details Page:**
  - Detailed view of a single product (images, description, price, etc.)
- **Cart Management:**
  - Add to cart and remove from cart
  - View total price
- **Checkout System:**
  - Integration with payment gateway (e.g., Stripe or PayPal)
  - Billing and shipping details form
- **User Accounts:**
  - Login, signup, and user dashboard for order tracking

### 4. Content and Pages

- **Home Page:** Featured products, promotions, and banners.
- **Shop Page:** Product categories and listings.

- **Product Details Page:** Detailed view for individual products.
- **Cart Page:** Overview of selected items.
- **Checkout Page:** Payment and order confirmation.
- **About Us Page:** Information about the brand.
- **Contact Us Page:** Contact form and social links.

## 5. Backend and Database

- **Backend CMS:** Sanity (for managing product data, categories, and banners).
- **Database:** Sanity will act as the database for structured content.

## 6. Design Requirements

- Pixel-perfect and responsive design.
- Smooth user experience with intuitive navigation.
- Consistent branding and styling across the website.

## 7. Future Enhancements

- Adding customer reviews and ratings for products.
- Wishlist functionality for users.
- Advanced analytics integration to track user behavior.

## API Integration Plan for Shop.co

### 1. Purpose of API Integration

API integration ka maqsad frontend or backend ke darmiyan smooth communication ko ensure karna hai. Ye APIs user data, product information, cart management, or payment processing ke liye use hongi.

---

### 2. API Integrations

#### A. Product Management APIs

- **Purpose:** Products ko fetch karna aur display karna.
- **Endpoints:**
  1. **Get Products:**
    - Endpoint: /api/products
    - Method: GET
    - Description: All products ko retrieve karega.

## 2. **Get Product by ID:**

- Endpoint: /api/products/:id
- Method: GET
- Description: Specific product details fetch karega.

## 3. **Filter Products:**

- Endpoint: /api/products?category=clothing&priceRange=0-1000
- Method: GET
- Description: Filters apply karne ke liye.

## **B. User Authentication APIs**

- **Purpose:** User login, signup, aur authentication ke liye.

- **Endpoints:**

### 1. **Signup User:**

- Endpoint: /api/auth/signup
- Method: POST
- Payload: { "email": "user@example.com", "password": "12345" }

### 2. **Login User:**

- Endpoint: /api/auth/login
- Method: POST
- Payload: { "email": "user@example.com", "password": "12345" }

### 3. **Get User Details:**

- Endpoint: /api/auth/user
- Method: GET
- Headers: { Authorization: "Bearer <token>" }

## **C. Cart Management APIs**

- **Purpose:** Cart mein items add karna, remove karna, aur update karna.

- **Endpoints:**

### 1. **Add to Cart:**

- Endpoint: /api/cart/add
- Method: POST

- Payload: { "productId": "123", "quantity": 1 }
- 2. **Remove from Cart:**
  - Endpoint: /api/cart/remove
  - Method: DELETE
  - Payload: { "productId": "123" }
- 3. **Get Cart Items:**
  - Endpoint: /api/cart
  - Method: GET

#### D. Payment Gateway Integration

- **Purpose:** Online payments process karne ke liye.
- **Service:** Stripe (recommended for simplicity and security).
- **Endpoints:**
  1. **Create Payment Session:**
    - Endpoint: /api/payment/create-session
    - Method: POST
    - Payload: { "cartItems": [{ "id": "123", "quantity": 1 }] }
  2. **Verify Payment:**
    - Endpoint: /api/payment/verify
    - Method: POST
    - Payload: { "paymentIntentId": "pi\_abc123" }

#### E. Order Management APIs

- **Purpose:** Orders ko track aur manage karna.
- **Endpoints:**
  1. **Place Order:**
    - Endpoint: /api/orders
    - Method: POST
    - Payload: { "userId": "abc123", "cartItems": [...], "paymentId": "pi\_abc123" }
  2. **Get User Orders:**
    - Endpoint: /api/orders/user/:userId

- Method: GET

## F. CMS Integration (Sanity)

- **Purpose:** Products aur banners ko manage karne ke liye Sanity API use hogi.
- **Endpoints:**
  1. **Fetch Content:**
    - Endpoint: `https://<project_id>.api.sanity.io/v1/data/query/production`
    - Query Example: `{ "*[_type == 'product']" }`
  2. **Add Content:**
    - Endpoint: `https://<project_id>.api.sanity.io/v1/data/mutate/production`
    - Payload: `{ "mutations": [{ "create": { "_type": "product", "title": "New Product" } } ] }`