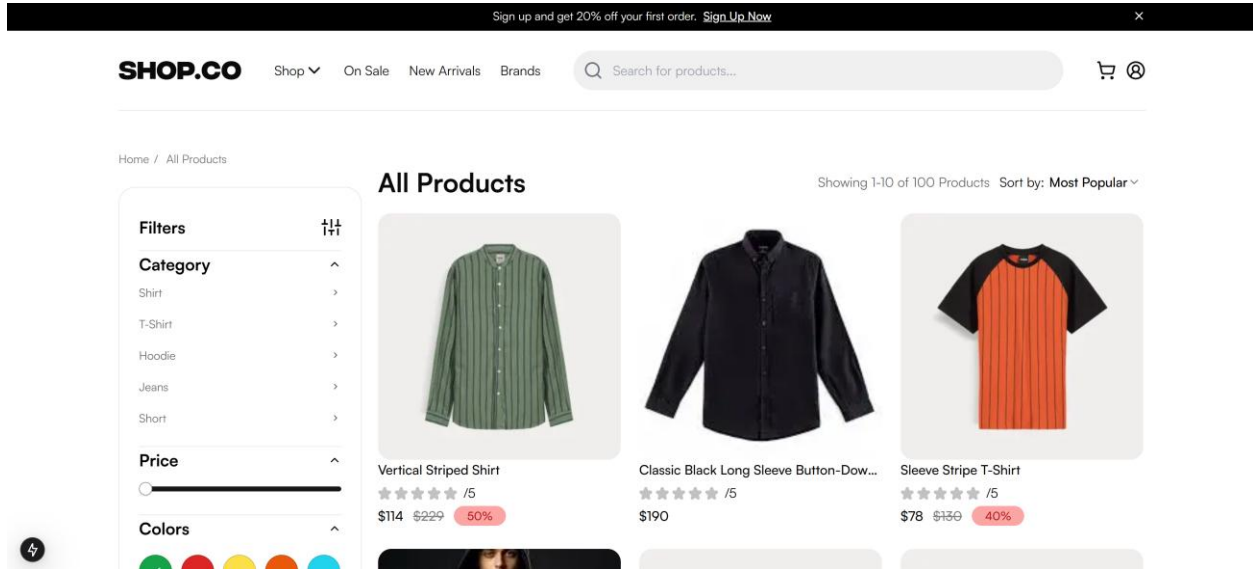


Day 4 - Dynamic Frontend Components - [SHOP.CO]

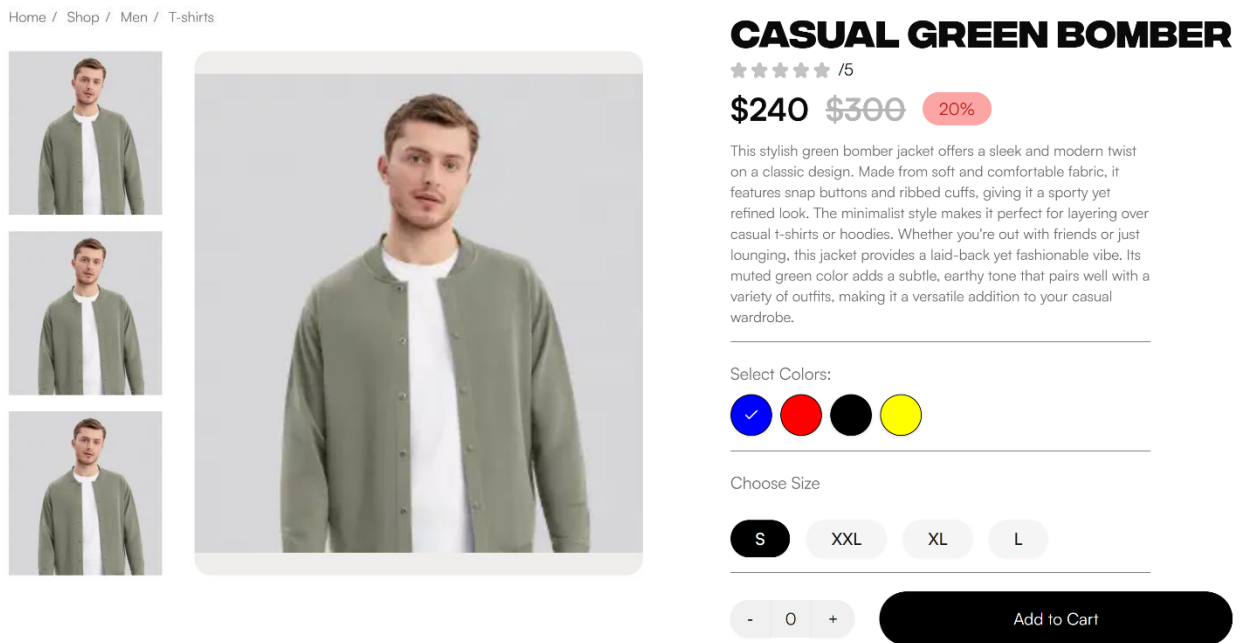
Functional Deliverables:

The product listing page with dynamic data.



1. Product Detail Page:

Individual product detail pages with accurate routing and data rendering.



2. Category Filters:

Working category filters, search bar, and pagination.

a) Sort with High price:

All Products

Showing 1-10 of 100 Products Sort by: **High Price** ▾



Casual Green Bomber Jacket

★★★★★ /5

\$240 ~~\$300~~ 20%



Beige Slim-Fit Jogger Pants

★★★★★ /5

\$242 ~~\$269~~ 10%



Skinny Fit Jeans

★★★★★ /5

\$240

b) Sort with Low Price:

All Products

Showing 1-10 of 100 Products Sort by: **Low Price** ▾



LOOSE FIT BERMUDA SHORTS

★★★★★ /5

\$62 ~~\$78~~ 20%



Black Striped T-Shirt

★★★★★ /5

\$120



Classic Black Pullover Hoodie

★★★★★ /5

\$128

c) Displaying Products from Shirt Category:

Filters

Category

Shirt

T-Shirt

Hoodie

Jeans

Short

Price


Colors

Size


Dress Style

Shirt


Showing 1-10 of 100 Products Sort by: Most Popular



Vertical Striped Shirt
★★★★★ /5
\$114 ~~\$229~~ 50%



Classic Black Long Sleeve Button-Down...
★★★★★ /5
\$190



Checkered Shirt
★★★★★ /5
\$178

d) Displaying Products from Hoodie Category:

Filters

Category

Shirt

T-Shirt

Hoodie

Jeans

Short

Price


Colors

Size


Dress Style

Hoodie


Showing 1-10 of 100 Products Sort by: Most Popular



Classic Black Pullover Hoodie
★★★★★ /5
\$128



Casual Green Bomber Jacket
★★★★★ /5
\$240 ~~\$300~~ 20%



Classic White Pullover Hoodie
★★★★★ /5
\$135 ~~\$150~~ 10%

e) Products Filtration based on Price Range:

Home / All Products

Filters

Category

Price

Colors


Size

Dress Style


Apply Filter

All Products

Showing 1-10 of 100 Products Sort by: Most Popular



Black Striped T-Shirt
★★★★★ /5
\$120



LOOSE FIT BERMUDA SHORTS
★★★★★ /5
\$62 ~~\$78~~ 20%


3. Search Bar:

Fully functional search bar feature, that searches product based on their names.


SHOP.COShop ▾On SaleNew ArrivalsBrands

Search results for **SHIRT**


[Home / Search](#)




Black Striped T-Shirt
★★★★★ /5
\$120



COURAGE GRAPHIC T-SHIRT
★★★★★ /5
\$145



Checkered Shirt
★★★★★ /5
\$178



Classic Black Long Sleeve Button-Down...
★★★★★ /5
\$190

4. Additional Features:


a) Add To Cart:

Add to cart feature with real time valuation tracking.

SHOP.COShop ▾On SaleNew ArrivalsBrands

[Home / Cart](#)

YOUR CART




Checkered Shirt
Size: XL
Color: White
\$178

-

1

+



Classic Black Long Sleeve Button-Down Shirt
Size: S
Color: Yellow
\$190

-

1

+

Order Summary

Subtotal

\$368

Discount (-20%)

-\$74

Delivery Fee

\$15

Total

\$279

Add promo code

Apply

Go to Checkout →

b) Product Details Section:

Product details section displaying details of each product.





Product Details	Rating & Reviews	FAQs
Product specifications		
Material composition	100% Cotton	
Care instructions	Machine wash warm, tumble dry	
Fit type	Classic Fit	
Pattern	Solid	

c) Ratings & Reviews:

Ratings & reviews section with filtration as per users need.


Product Details	Rating & Reviews	FAQs
All Reviews (10)		
Latest Write a Review		
Filter Reviews		
✓ Latest		
Most Relevant		
Oldest		
★★★★★		
Sarah M.		
I'm blown away by the quality and style of the clothes I received from Shop.co. From casual wear to elegant dresses, every piece I've bought has exceeded my expectations.		
Sun Oct 01 2023		
★★★★★		
Alex K.		
Finding clothes that align with my personal style was a challenge until I discovered Shop.co. The range of options they offer is truly remarkable, catering to a variety of tastes and occasions.		
Fri Nov 01 2024		

d) New Arrivals Section:

NEW ARRIVALS			
			
Vertical Striped Shirt	Classic Black Long Sleeve Button-Down...	Sleeve Stripe T-Shirt	Checkered Shirt
★★★★★ /5	★★★★★ /5	★★★★★ /5	★★★★★ /5
\$114 \$229 50%	\$190	\$78 \$130 40%	\$178
View All			

e) Top Selling Products:


TOP SELLING



Classic Black Pullover Hoodie

★★★★★ /5


\$128



Black Striped T-Shirt

★★★★★ /5


\$120



Classic Black Straight-Leg Jeans

★★★★★ /5

\$142 ~~\$170~~ 16%



LOOSE FIT BERMUDA SHORTS


★★★★★ /5

\$62 ~~\$78~~ 20%

View All

f) You Might also Like section:


YOU MIGHT ALSO LIKE



Black Athletic Jogger Pants with Side...

★★★★★ /5


\$180



Classic Black Straight-Leg Jeans

★★★★★ /5


\$142 ~~\$170~~ 16%



Gray Slim-Fit Jogger Pants

★★★★★ /5

\$144 ~~\$170~~ 15%



Skinny Fit Jeans

★★★★★ /5

\$240

Code Deliverables:

1. Product Card Component:

```
src > components > itemcards > Card.tsx > ...
1  import Image from "next/image";
2  import React from "react";
3  import Link from "next/link";
4  import StarRating from "../feedback/StarRating";
5  import { CATALOG_QUERYResult } from "sanity.types";
6  import { urlFor } from "@sanity/lib/image";
7
8  function Card({ catalog }: { catalog: CATALOG_QUERYResult[0] }) {
9    return (
10     catalog && (
11       <Link
12         href={` /shop/${catalog._id}`}
13         className="cursor-pointer mb-2 w-full"
14       >
15         <div className="group relative p-4 lg:p-0 bg-[#F0EEED] rounded-2xl flex justify-center items-center xl:w-[295px] xl:h-[298px] w-[198px] h-[200px] overflow-hidden">
16           <Image
17             src={urlFor(catalog.imageUrl).url()}
18             alt={` ${catalog.name?.split(" ").join("").toLowerCase}`}
19             width={172}
20             height={152}
21             className="object-cover h-full w-full hover:scale-125 transition-all duration-700"
22           />
23         </div>
24         <div className={`flex flex-col gap-1`} >
25           <p className="font-medium text-base leading-6 line-clamp-1">
26             {catalog.name}
27           </p>
28           <div className={`flex flex-col gap-1`} >
29             <div className="flex flex-row items-center gap-2">
30               <StarRating rating={catalog.rating} />
31               {catalog.rating}/5
32             </div>
33             <div className="font-medium text-base leading-6 flex gap-3 items-center">
34               <span>
35                 $
36                 {catalog.discountPercent! > 0
37                   ? catalog.price! -
38                     Math.ceil((catalog.price! * catalog.discountPercent!) / 100)
39                   : catalog.price}
40               </span>
41               <span
42                 className={`text-text2 opacity-50 ${catalog.discountPercent! > 0 ? "line-through"
43                   : "hidden"}}`
44               >
45                 {catalog.price}
46               </span>
47               {catalog.discountPercent! > 0 ? (
48                 <div className="bg-red-300 px-4 text-sm py-0.5 text-red-700 rounded-full">
49                   {catalog.discountPercent}%
50                 </div>
51               ) : (
52                 ""
53               )}
54             </div>
55           </div>
56         </Link>
57       </div>
58     );
59   }
60
61   export default Card;
62
```


2. Shop Page displaying All Products:

```
src > app > shop > page.tsx > Shop
1  import { BreadCrumb } from "@components/BreadCrumb";
2  import DisplayedProducts from "@components/DisplayedProducts";
3  import Container from "@components/ui/Container";
4  import { getAllCategory, getProductsCatalog } from "@sanity/helpers";
5
6  export default async function Shop() {
7    const catalog = await getProductsCatalog();
8    const categoriesData = await getAllCategory();
9
10   return (
11     <Container>
12       <hr className="h-[1px] border-t-black/10 mb-5 sm:mb-6" />
13       <BreadCrumb location={["Home", "All Products"]} />
14       <DisplayedProducts catalog={catalog} categoriesData={categoriesData} />
15     </Container>
16   );
17 }
18
```

3. Dynamic Product Displaying Page:

```
src > app > shop > [id] > page.tsx > ProductPage > catalog.map() callback
1  // src/app/product/[id]/page.tsx
2  import { BreadCrumb } from "@components/BreadCrumb";
3  import Product from "@components/product/Product";
4  import FeedbackTabs from "@components/feedback/FeedbackTabs";
5  import DisplayCard from "@components/itemcards/DisplayCard";
6  import Container from "@components/ui/Container";
7  import { getProductsCatalog } from "@sanity/helpers";
8  // export interface Item { ...
177 export default async function ProductPage(props: {
178   params: Promise<{ id: string }>;
179 }) {
180   // You can directly access the `id` from the `params` object
181   const params = await props.params;
182   const id = params.id;
183   const catalog = await getProductsCatalog();
184   return (
185     <Container className="relative overflow-x-hidden">
186       {catalog.map((product) =>
187         product._id === id ? (
188           <div key={product._id} className="flex flex-col">
189             <BreadCrumb location={["Home", "Shop", "Men", "T-shirts"]} />
190             <Product product={product} />
191           </div>
192         ) : (
193           ""
194         )
195       )}
196       <FeedbackTabs />
197       <DisplayCard heading="You might also like" button={false} id={id} />
198     </Container>
199   );
200 }
201
```


4. Cart Functionalities Using Zustand:

```
src > store > ts CartStore.ts > ...
1  import { CATALOG_QUERYResult } from "sanity.types";
2  import { create } from "zustand";
3
4  interface CartItems {
5    product: CATALOG_QUERYResult[0],
6    quantity: number
7    sizes: string[]
8    colors: string[]
9  }
10
11 interface CartState {
12   cartItems: CartItems[],
13   color: string,
14   setColor: (newColor: string) => void
15   size: string,
16   setSize: (newSize: string) => void
17   addCartItem: (product: CATALOG_QUERYResult[0], size: string, color: string) => void
18   reduceItemCount: (productId: string) => void
19   removeCartItem: (productId: string) => void
20   cartItemsCount: (productId: string) => number
21   getTotalPrice: () => number
22 }
23
24 const useCartStore = create<CartState>((set, get) => ({
25   cartItems: [],
26   color: "",
27   setColor: (newColor) => set({ color: newColor }),
28   size: "",
29   setSize: (newSize) => set({ size: newSize }),
30   addCartItem: (product, size, color) =>
31     set((state) => {
32       const existingItem = state.cartItems.find(
33         (item) => item.product._id === product._id
34       );
35       if (existingItem) {
36         return {
37           cartItems: state.cartItems.map((item) =>
38             item.product._id === product._id
39               ? { ...item, quantity: item.quantity + 1, sizes: [...item.sizes, size], colors: [...item.colors, color] }
40               : item
41           ),
42         };
43       } else {
44         return { cartItems: [...state.cartItems, { product, quantity: 1, sizes: [size], colors: [color] }] as CartItems[] };
45       }
46     }),
47   reduceItemCount: (productId: string) =>
48     set((state) => {
49       const existingItem = state.cartItems.find((item) => item.product._id === productId)
50       if (existingItem) {
51         if (existingItem!.quantity > 1) {
52           return {
53             cartItems: state.cartItems.map((item) =>
54               item.product._id === productId ? { ...item, quantity: item.quantity - 1, sizes: item.sizes.slice(0, -1), colors: item.colors.slice(0, -1) } : item
55             ),
56           };
57         } else {
58           return { cartItems: state.cartItems.filter((item) => item.product._id !== productId) }
59         }
60       } else {
61         return { cartItems: state.cartItems }
62       }
63     }),
64   removeCartItem: (productId: string) =>
65     set((state) => ({ cartItems: state.cartItems.filter((item) => item.product._id !== productId) })),
66   cartItemsCount: (productId: string) => {
67     const item = get().cartItems.find((item) => item.product._id === productId)
68     return item ? item.quantity : 0
69   },
70   getTotalPrice: () => {
71     const totalPrice = get().cartItems.map((item) => (item.product.discountPercent! > 0 ? Math.ceil(item.product.price! - (item.product.price! * item.product.discountPercent! / 100)) * item.quantity : item.product.price! * item.quantity)).reduce((prev, curr) => (prev + curr), 0)
72     return totalPrice
73   },
74 })))
75
76 export default useCartStore
77
```