SHOP.CO – Farwa Batool

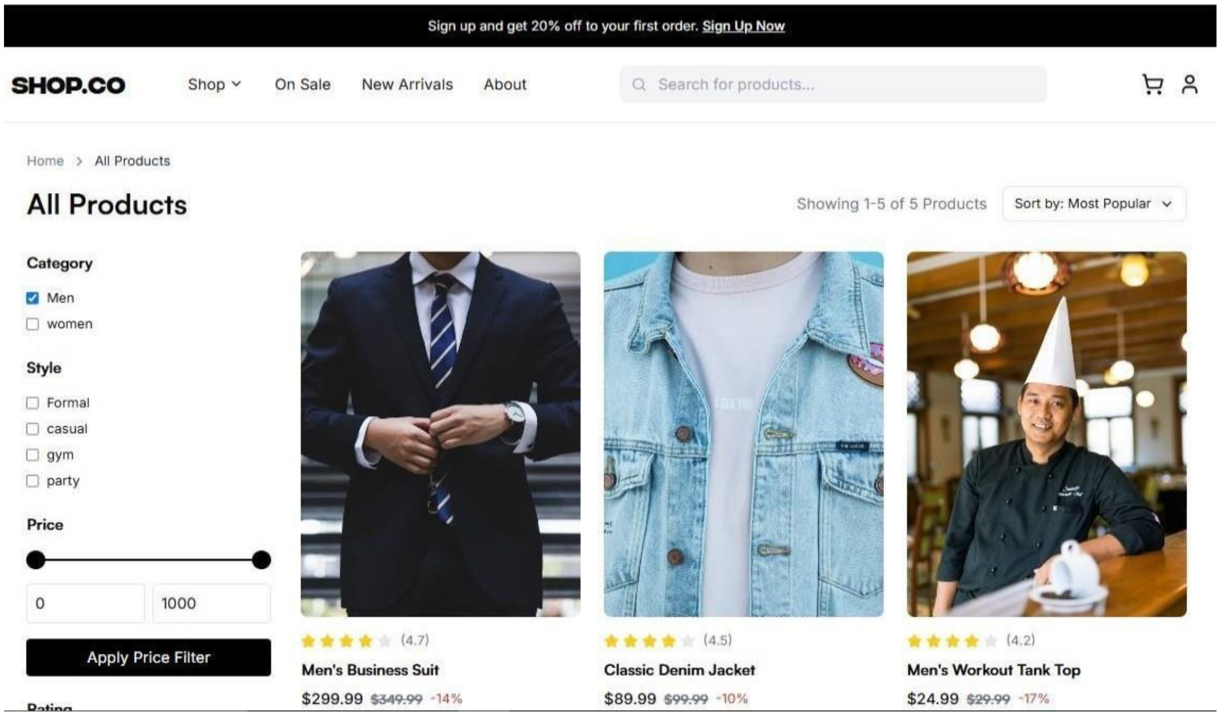
Marketplace Self Validation Checklist by
Farwa Batool

I. Functional Deliverables

1. **Product Listing Page:**

   A grid layout showcasing dynamic product data, including:

a. Product Name
b. Price
c. Image
d. Reviews
e. Ratings


Sign up and get 20% off to your first order. Sign Up Now

SHOP.CO · Shop · On Sale · New Arrivals · About · Search for products...

Home > All Products

All Products

Showing 1-5 of 5 Products · Sort by: Most Popular

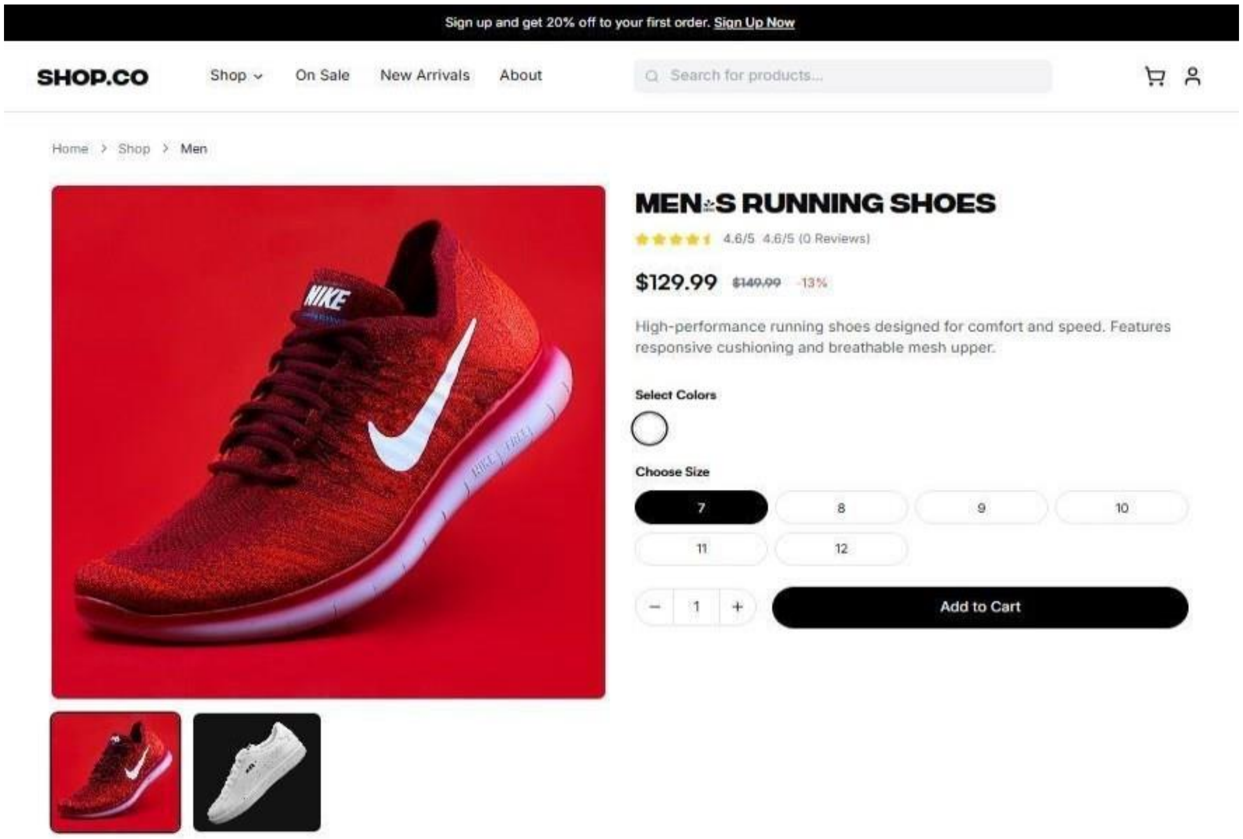
Category: ☑ Men · ☐ women
Style: ☐ Formal · ☐ casual · ☐ gym · ☐ party
Price: 0 – 1000 · Apply Price Filter
Rating

Rating	Product	Price
(4.7)	Men's Business Suit	$299.99 ~~$349.99~~ -14%
(4.5)	Classic Denim Jacket	$89.99 ~~$99.99~~ -10%
(4.2)	Men's Workout Tank Top	$24.99 ~~$29.99~~ -17%

2. **Product Detail Page**

Dynamic product pages implemented with routing. Displayed details include:

a. Product Name
b. Description
c. Price
d. Sizes
e. Colors
f. Images
g. Inventory
h. Category
i. Style
j. Tags



MEN:S RUNNING SHOES

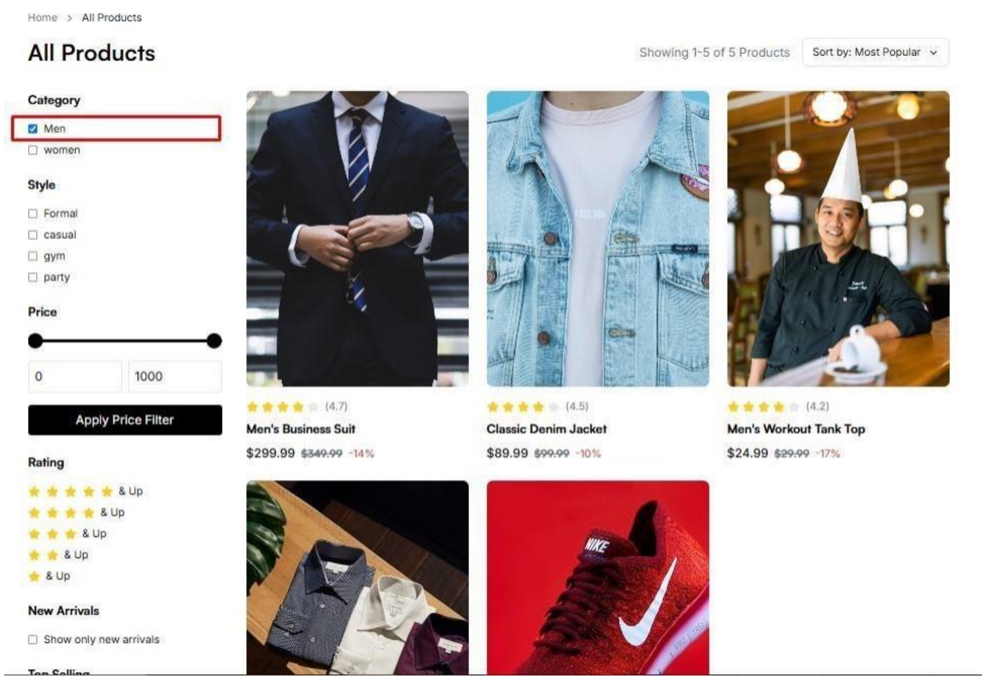
4.6/5 4.6/5 (0 Reviews)

$129.99 ~~$149.99~~ -13%

High-performance running shoes designed for comfort and speed. Features responsive cushioning and breathable mesh upper.

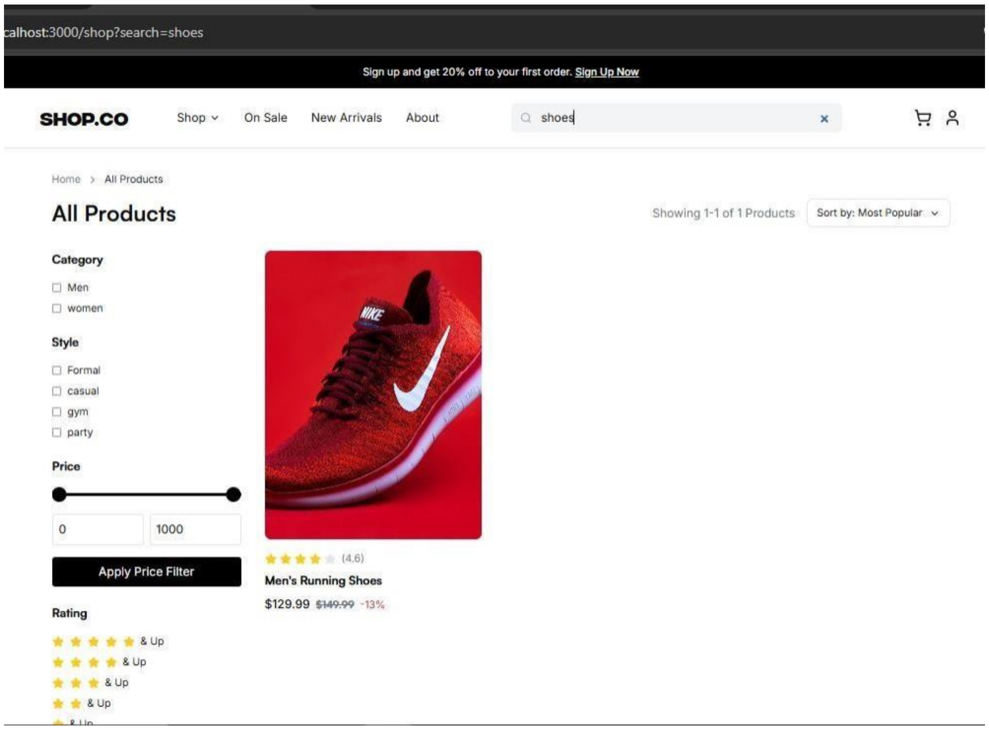
3. **Category Filters**

Fully functional category filters allowing users to filter products dynamically



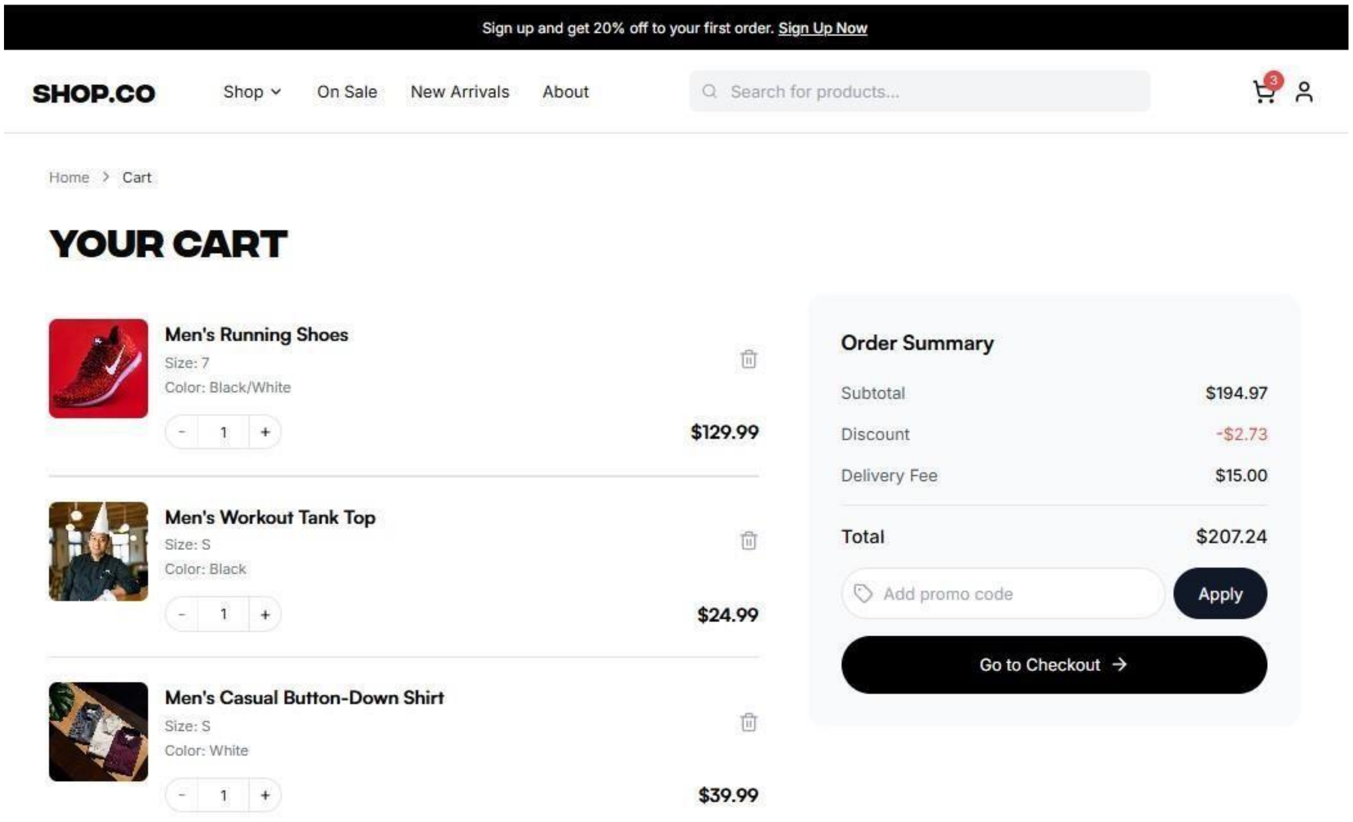
4. **Search Bar**

Implemented search functionality to filter products by name or tags



5. **Add To Cart**

   Users can add products to the cart, view them, and remove items dynamically.



Sign up and get 20% off to your first order. Sign Up Now

SHOP.CO Shop On Sale New Arrivals About Search for products... 3

Home > Cart

YOUR CART

Men's Running Shoes
Size: 7
Color: Black/White
- 1 + $129.99

Men's Workout Tank Top
Size: S
Color: Black
- 1 + $24.99

Men's Casual Button-Down Shirt
Size: S
Color: White
- 1 + $39.99

Order Summary

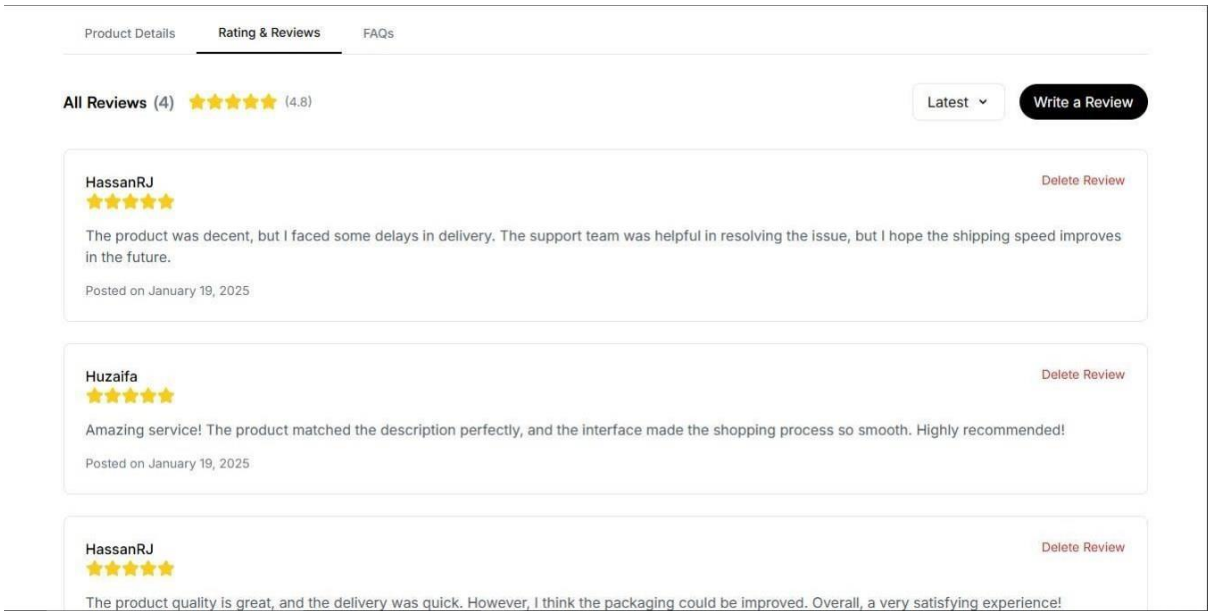
Subtotal	$194.97
Discount	-$2.73
Delivery Fee	$15.00
Total	$207.24

Add promo code Apply

Go to Checkout →

6. **Reviews and Ratings**

   Users can add, view, and delete reviews dynamically.



All Reviews (4) (4.8)

HassanRJ

The product was decent, but I faced some delays in delivery. The support team was helpful in resolving the issue, but I hope the shipping speed improves in the future.

Posted on January 19, 2025

Huzaifa

Amazing service! The product matched the description perfectly, and the interface made the shopping process so smooth. Highly recommended!

Posted on January 19, 2025

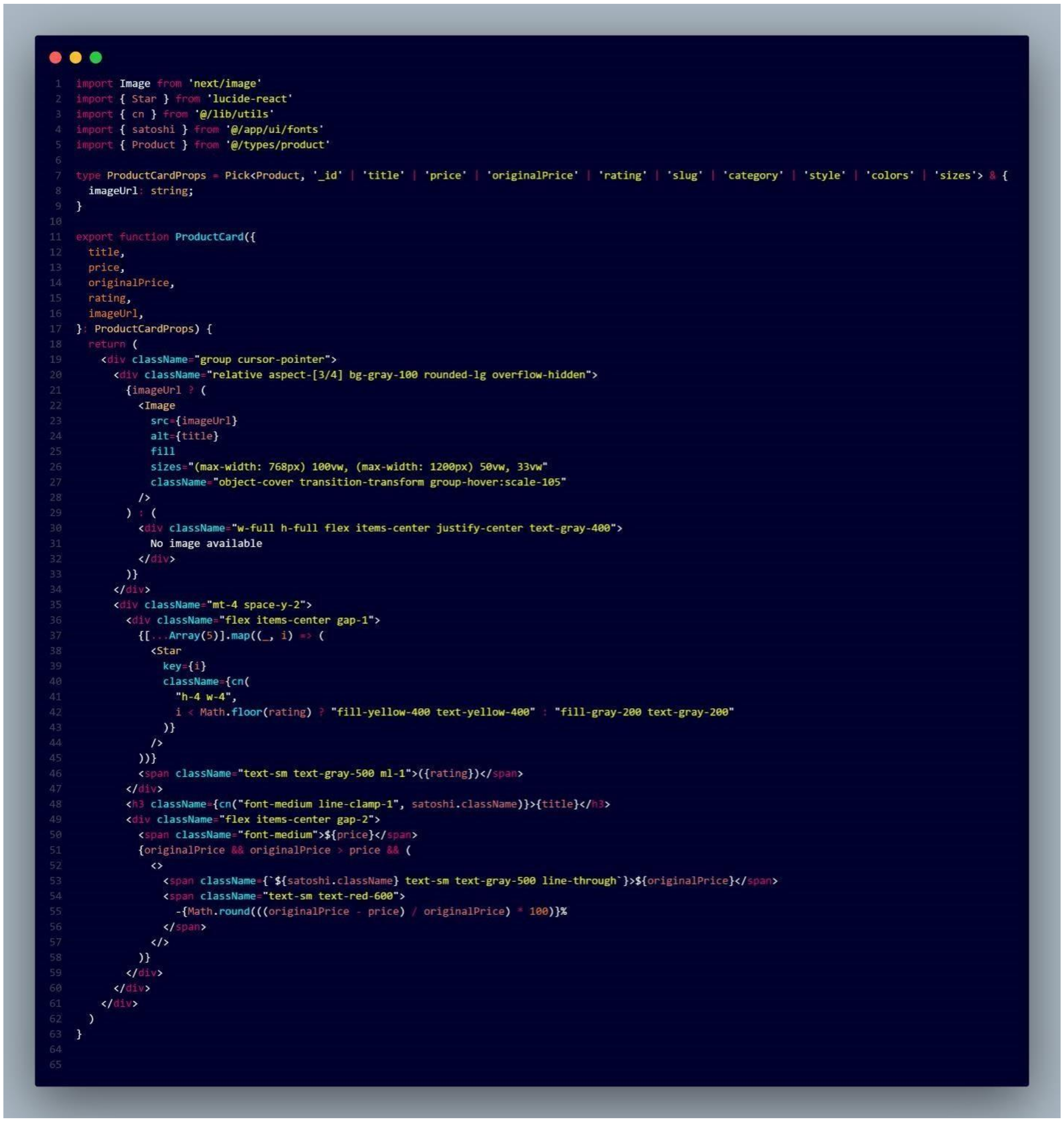
HassanRJ

The product quality is great, and the delivery was quick. However, I think the packaging could be improved. Overall, a very satisfying experience!

II. Code Snippets

1. **ProductCard Component:**

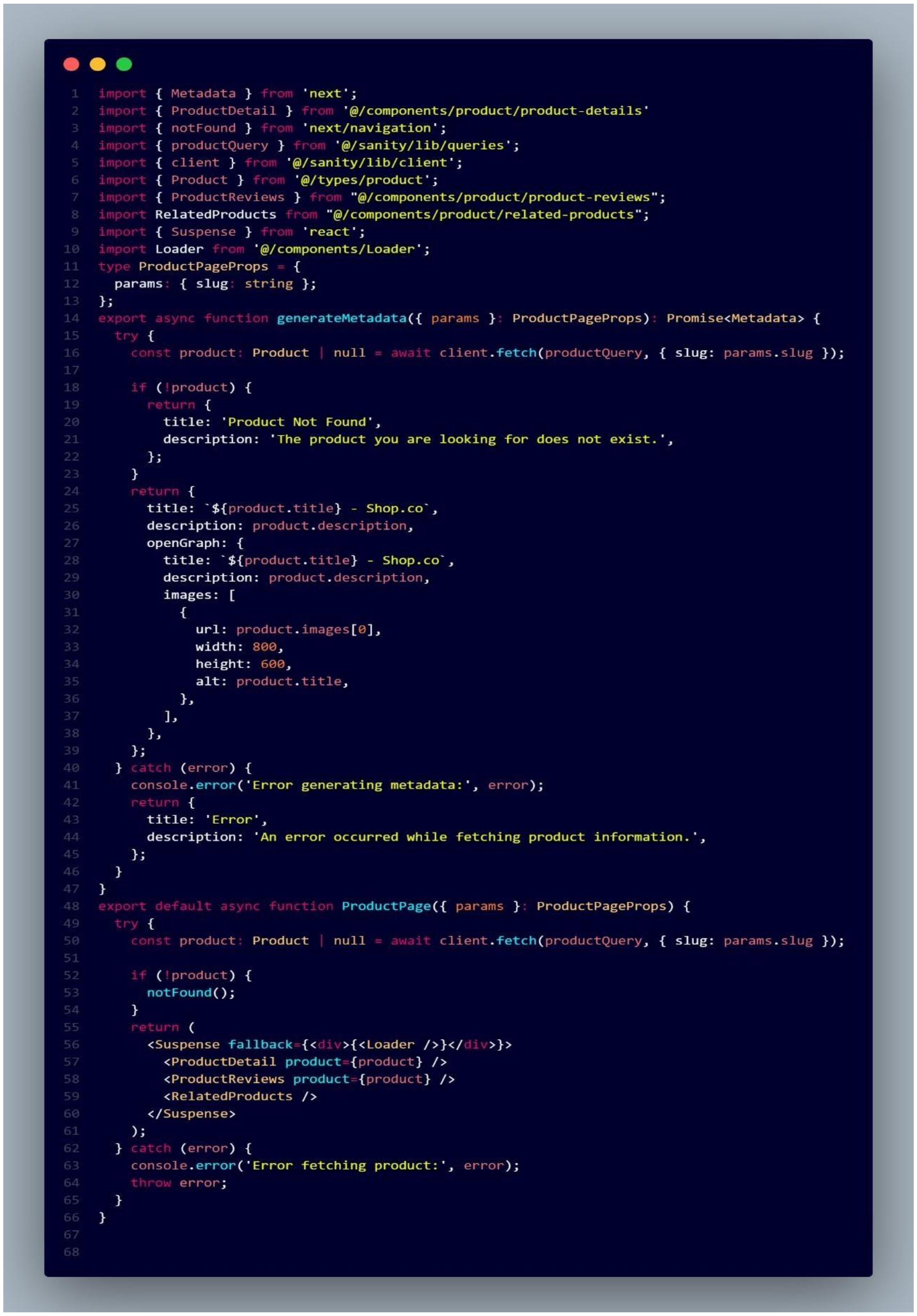
Used to render individual product cards in the listing



```tsx
import Image from 'next/image'
import { Star } from 'lucide-react'
import { cn } from '@/lib/utils'
import { satoshi } from '@/app/ui/fonts'
import { Product } from '@/types/product'

type ProductCardProps = Pick<Product, '_id' | 'title' | 'price' | 'originalPrice' | 'rating' | 'slug' | 'category' | 'style' | 'colors' | 'sizes'> & {
  imageUrl: string;
}

export function ProductCard({
  title,
  price,
  originalPrice,
  rating,
  imageUrl,
}: ProductCardProps) {
  return (
    <div className="group cursor-pointer">
      <div className="relative aspect-[3/4] bg-gray-100 rounded-lg overflow-hidden">
        {imageUrl ? (
          <Image
            src={imageUrl}
            alt={title}
            fill
            sizes="(max-width: 768px) 100vw, (max-width: 1200px) 50vw, 33vw"
            className="object-cover transition-transform group-hover:scale-105"
          />
        ) : (
          <div className="w-full h-full flex items-center justify-center text-gray-400">
            No image available
          </div>
        )}
      </div>
      <div className="mt-4 space-y-2">
        <div className="flex items-center gap-1">
          {[...Array(5)].map((_, i) => (
            <Star
              key={i}
              className={cn(
                "h-4 w-4",
                i < Math.floor(rating) ? "fill-yellow-400 text-yellow-400" : "fill-gray-200 text-gray-200"
              )}
            />
          ))}
          <span className="text-sm text-gray-500 ml-1">({rating})</span>
        </div>
        <h3 className={cn("font-medium line-clamp-1", satoshi.className)}>{title}</h3>
        <div className="flex items-center gap-2">
          <span className="font-medium">${price}</span>
          {originalPrice && originalPrice > price && (
            <>
              <span className={`${satoshi.className} text-sm text-gray-500 line-through`}>${originalPrice}</span>
              <span className="text-sm text-red-600">
                -{Math.round(((originalPrice - price) / originalPrice) * 100)}%
              </span>
            </>
          )}
        </div>
      </div>
    </div>
  )
}
```

2. **Dynamic Product Detail Page Routing:**

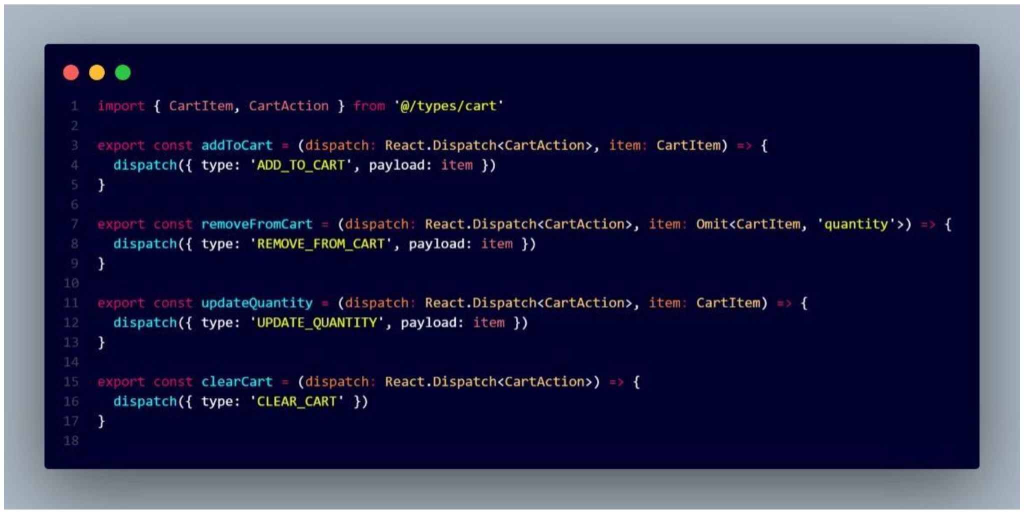
   Logic for dynamic product pages:



```
import { Metadata } from 'next';
import { ProductDetail } from '@/components/product/product-details'
import { notFound } from 'next/navigation';
import { productQuery } from '@/sanity/lib/queries';
import { client } from '@/sanity/lib/client';
import { Product } from '@/types/product';
import { ProductReviews } from "@/components/product/product-reviews";
import RelatedProducts from "@/components/product/related-products";
import { Suspense } from 'react';
import Loader from '@/components/Loader';
type ProductPageProps = {
  params: { slug: string };
};
export async function generateMetadata({ params }: ProductPageProps): Promise<Metadata> {
  try {
    const product: Product | null = await client.fetch(productQuery, { slug: params.slug });

    if (!product) {
      return {
        title: 'Product Not Found',
        description: 'The product you are looking for does not exist.',
      };
    }
    return {
      title: `${product.title} - Shop.co`,
      description: product.description,
      openGraph: {
        title: `${product.title} - Shop.co`,
        description: product.description,
        images: [
          {
            url: product.images[0],
            width: 800,
            height: 600,
            alt: product.title,
          },
        ],
      },
    };
  } catch (error) {
    console.error('Error generating metadata:', error);
    return {
      title: 'Error',
      description: 'An error occurred while fetching product information.',
    };
  }
}
export default async function ProductPage({ params }: ProductPageProps) {
  try {
    const product: Product | null = await client.fetch(productQuery, { slug: params.slug });

    if (!product) {
      notFound();
    }
    return (
      <Suspense fallback={<div>{<Loader />}</div>}>
        <ProductDetail product={product} />
        <ProductReviews product={product} />
        <RelatedProducts />
      </Suspense>
    );
  } catch (error) {
    console.error('Error fetching product:', error);
    throw error;
  }
}
```

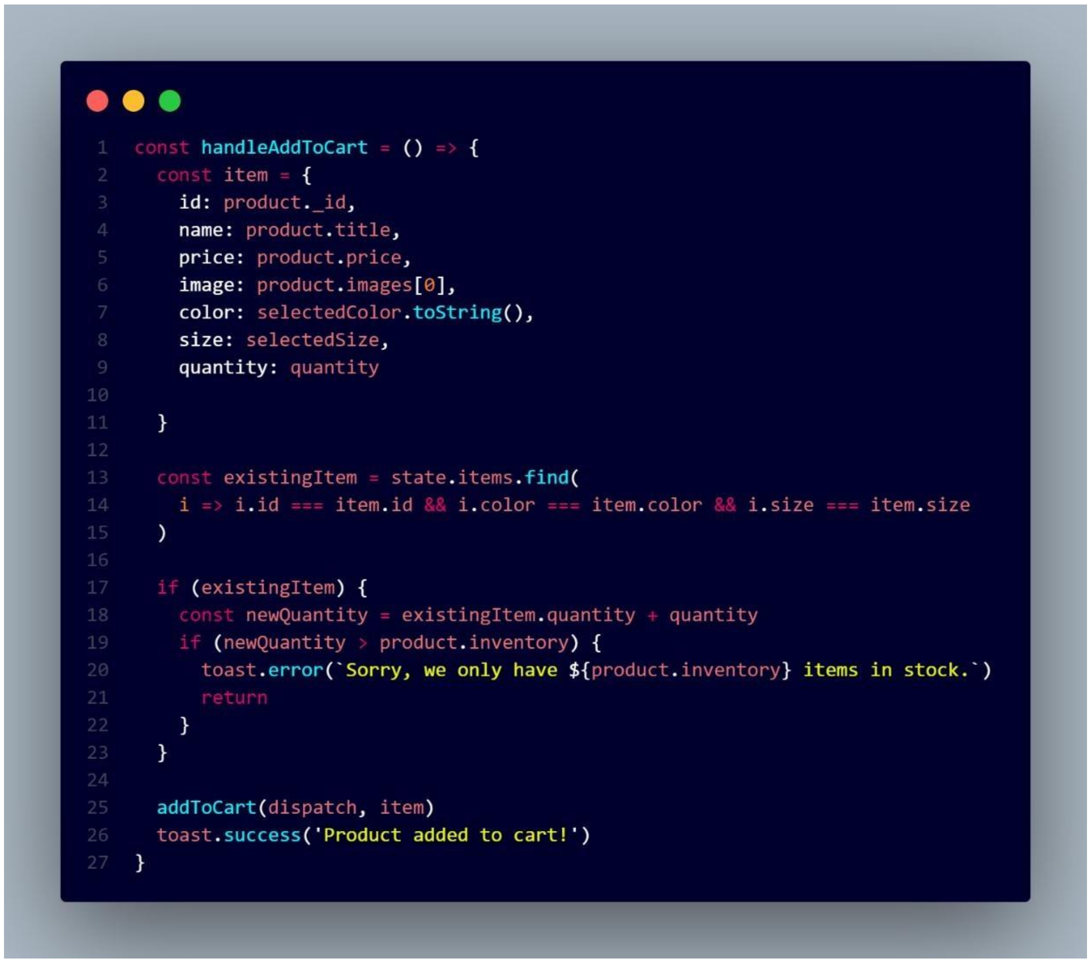
3. **Add to Cart Functionality:**

Logic for adding and removing items from the cart: ***actions.ts***



```ts
import { CartItem, CartAction } from '@/types/cart'

export const addToCart = (dispatch: React.Dispatch<CartAction>, item: CartItem) => {
  dispatch({ type: 'ADD_TO_CART', payload: item })
}

export const removeFromCart = (dispatch: React.Dispatch<CartAction>, item: Omit<CartItem, 'quantity'>) => {
  dispatch({ type: 'REMOVE_FROM_CART', payload: item })
}

export const updateQuantity = (dispatch: React.Dispatch<CartAction>, item: CartItem) => {
  dispatch({ type: 'UPDATE_QUANTITY', payload: item })
}

export const clearCart = (dispatch: React.Dispatch<CartAction>) => {
  dispatch({ type: 'CLEAR_CART' })
}

```

Function from Frontend HandleAddToCart



```ts
const handleAddToCart = () => {
  const item = {
    id: product._id,
    name: product.title,
    price: product.price,
    image: product.images[0],
    color: selectedColor.toString(),
    size: selectedSize,
    quantity: quantity

  }

  const existingItem = state.items.find(
    i => i.id === item.id && i.color === item.color && i.size === item.size
  )

  if (existingItem) {
    const newQuantity = existingItem.quantity + quantity
    if (newQuantity > product.inventory) {
      toast.error(`Sorry, we only have ${product.inventory} items in stock.`)
      return
    }
  }

  addToCart(dispatch, item)
  toast.success('Product added to cart!')
}
```

Dynamic Public Marketplace – SHOP.CO

By Farwa Batool