

DAY 4: BUILDING DYNAMIC FRONTEND COMPONENTS

Report on E-Commerce MarketPlace Website Development

Introduction:

The developed e-commerce website is a fully functional and responsive marketplace built with modern technologies, providing a seamless user experience for browsing, searching, and purchasing products. The project incorporates advanced features like dynamic product listing, category-based filtering, payment gateway integration, and an intuitive cart system, ensuring high performance and scalability.

Technologies Used

1. Next.js Framework:

Description: Next.js, a React-based framework, was chosen for its server-side rendering (SSR) capabilities, static site generation (SSG), and API route handling. These features ensure optimal performance and SEO-friendly architecture.

Implementation: Pages and components are modularized, leveraging Next.js' dynamic routing to enhance navigation between categories, products, and checkout pages.

2. Tailwind CSS:

Description: Tailwind CSS, a utility-first CSS framework, was employed for designing responsive and mobile-friendly layouts with minimal CSS code.

Usage: Custom classes provided flexibility in implementing reusable and adaptive design components, such as product cards, navigation bars, and grid layouts.

3. Sanity CMS

Description: Sanity, a headless content management system (CMS), was integrated to manage product data dynamically. It provides a robust platform for data management and real-time updates.

Updates Implemented:

Price Field Update: The product schema was updated to include a `price` field for dynamically setting and updating product prices.

- Ensures accurate and consistent product pricing across the website.

Hero Section Fields: A new schema for the Hero Section was created to include fields like `title`, `subtitle`, and `image`.

This allows easy customization of the Hero Section's content directly from Sanity CMS without requiring code updates.

Implementation:

- API queries fetched data from Sanity for products, categories, and hero sections.
- Sanity's asset pipeline efficiently handled images, ensuring fast loading.

4. API Integration:

Description: APIs were utilized to fetch external data and sync it with the CMS.

Implementation: A custom API was developed to fetch categorized product data.

- Client-side rendering with fallback mechanisms ensured seamless data fetching and error handling.

5. Stripe Payment Gateway:

Description: Stripe, a secure and reliable payment gateway, was integrated for managing transactions.

Usage: The checkout flow includes secure token generation and server-side processing for payment handling.

- Stripe's APIs facilitated real-time updates and notifications for successful transactions.

6. Shopping Cart with `use-shopping-cart`:

Description: The `use-shopping-cart` library provided a pre-built solution for managing cart functionality.

Features:

- Add, remove, and update items in the cart.
- Persistent cart state across sessions for enhanced user experience.

7. Shadcn Library:

Description: Shadcn was used to integrate pre-designed and accessible UI components, expediting development without compromising design consistency.

Implementation: Buttons, modals, and form inputs were styled and customized to align with the project theme.

Website Features:

1. Responsive Design:

Description:

- The website is fully responsive and adapts seamlessly across devices of various screen sizes, including desktops, tablets, and mobile phones.
- Tailwind CSS utilities (`grid`, `flex`, and `media queries`) were used to create a fluid layout that adjusts components dynamically based on screen width.
- Product grids, navigation bars, and hero sections are optimized for mobile-first design, providing an excellent user experience regardless of the device.

2. Dynamic Product Listings:

Description: Products are dynamically displayed using data fetched from Sanity CMS, categorized based on user preferences.

Implementation:

- The data fetching function fetches products by category and displays them using a grid layout styled with Tailwind CSS.
- Products are updated in real-time with category filtering and search functionality for enhanced navigation.

3. Dynamic Category Page:

Description:

- A dedicated dynamic category page allows users to view products based on their selected category.
- The `getProductsByCategory` function fetches data from Sanity CMS by category parameter and dynamically renders products in the selected category.
- Next.js' dynamic routing (`[category]`) enables the creation of a single reusable page component for all categories.

Features:

- Displays category-specific products in a visually appealing grid layout.
- Handles cases where no products are available in a category with user-friendly messaging.
- The page is responsive, ensuring a consistent experience on all devices.

4. Product Detail Page:

Description: Each product includes a detailed page displaying attributes such as name, price, discount percentage, description, and images.

Implementation:

- Dynamic routing in Next.js enables efficient rendering of individual product details.
- Users can easily navigate from product listings to detail pages.

5. Search and Filter Functionality:

Description: Users can search for products using a keyword-based search bar and filter them by categories for a streamlined shopping experience.

6. Hero Section:

Description:

- A dynamic Hero Section was built using data fetched from Sanity CMS.
- The section displays visually captivating content, including a title, subtitle, and banner image, which can be updated directly through the CMS.
- Automatic transitions were implemented for a slideshow effect using ``useState`` and ``useEffect`` hooks.

7. Payment and Checkout:

Description:

- A robust checkout page facilitates secure payment processing via Stripe, enhancing user trust and security.
- User cart items are synchronized during the checkout process.

Challenges and Solutions:

Challenge 1: Managing Real-Time Product Updates:

Solution: Sanity's GROQ queries and webhooks allowed real-time data synchronization.

Challenge 2: Complex Cart State Management:

Solution: The `use-shopping-cart` library simplified state management for cart items and user interactions.

Challenge 3: Secure Payment Integration:

Solution: Stripe APIs were implemented with server-side token validation and error handling to ensure transaction security.

Challenge 4: Dynamic Hero Section Integration:

Solution: A separate schema for the Hero Section in Sanity enabled easy updates and allowed flexible control over its content.

Challenge 5: Responsive Design Complexity:

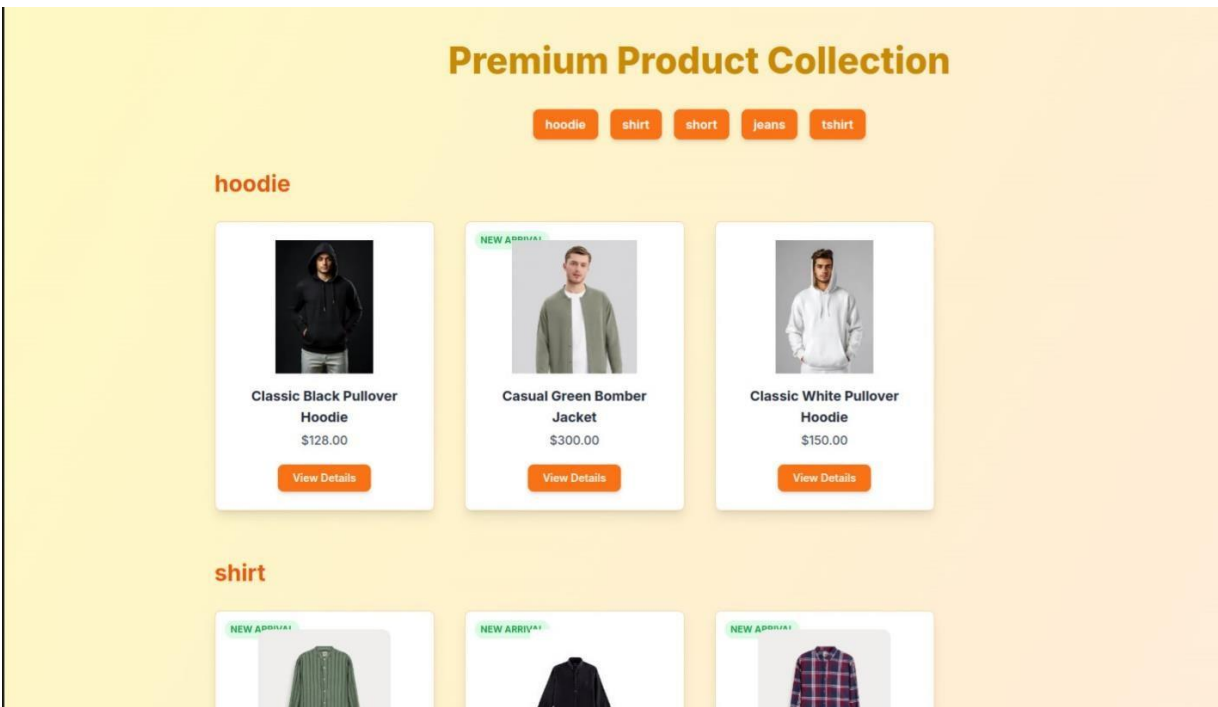
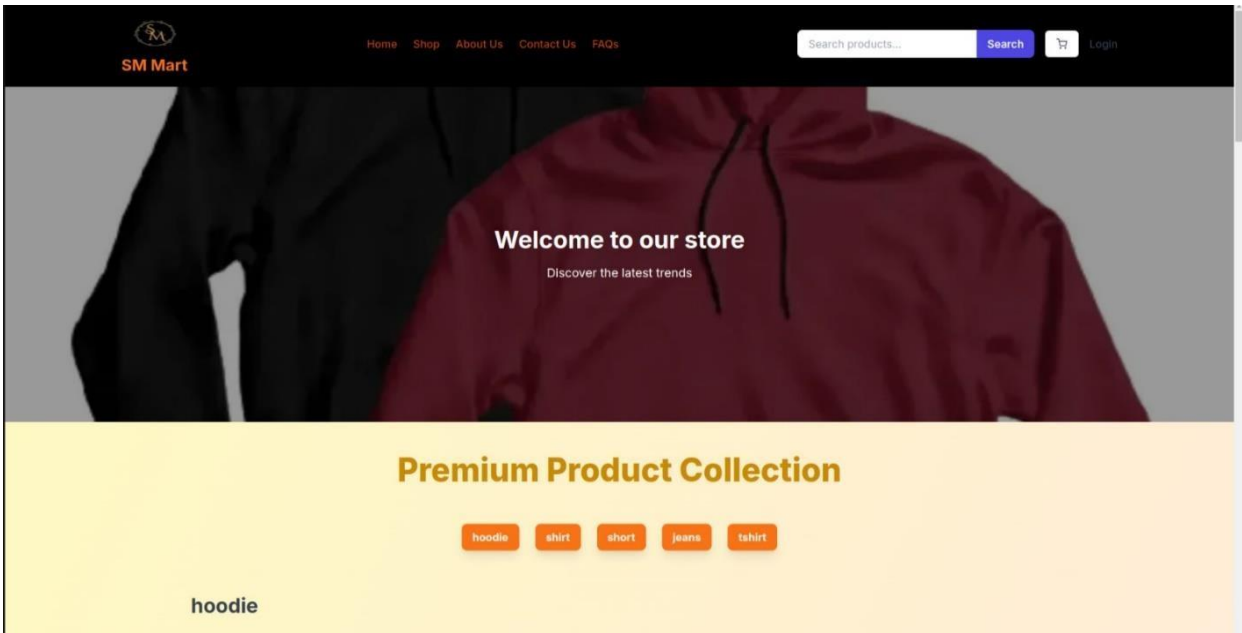
Solution: Tailwind CSS's utility classes and media queries simplified the implementation of responsive designs for grids and navigation.

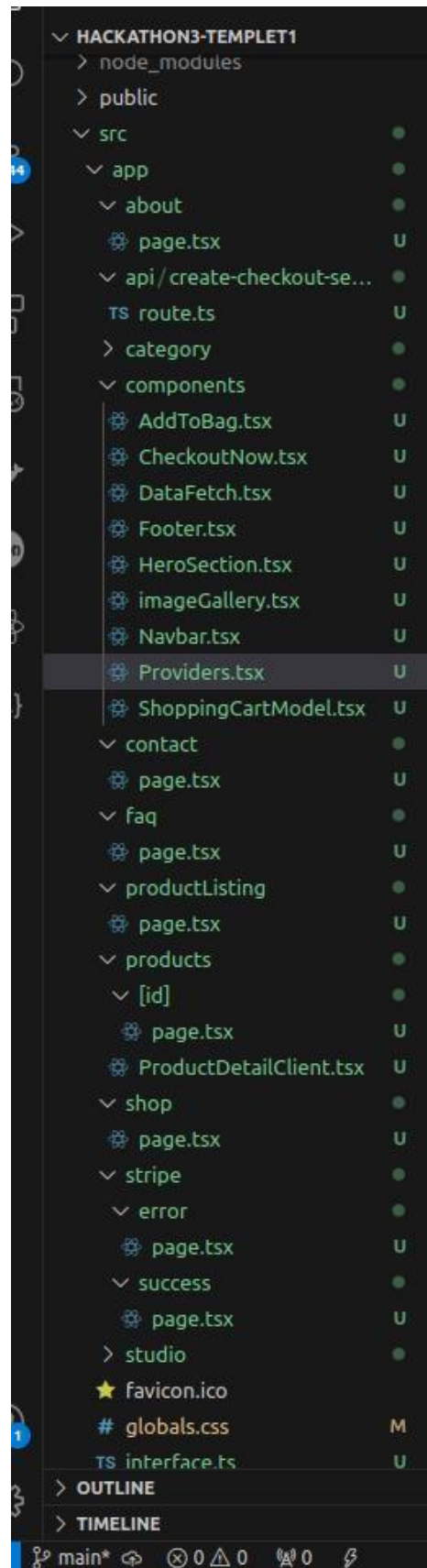
Conclusion

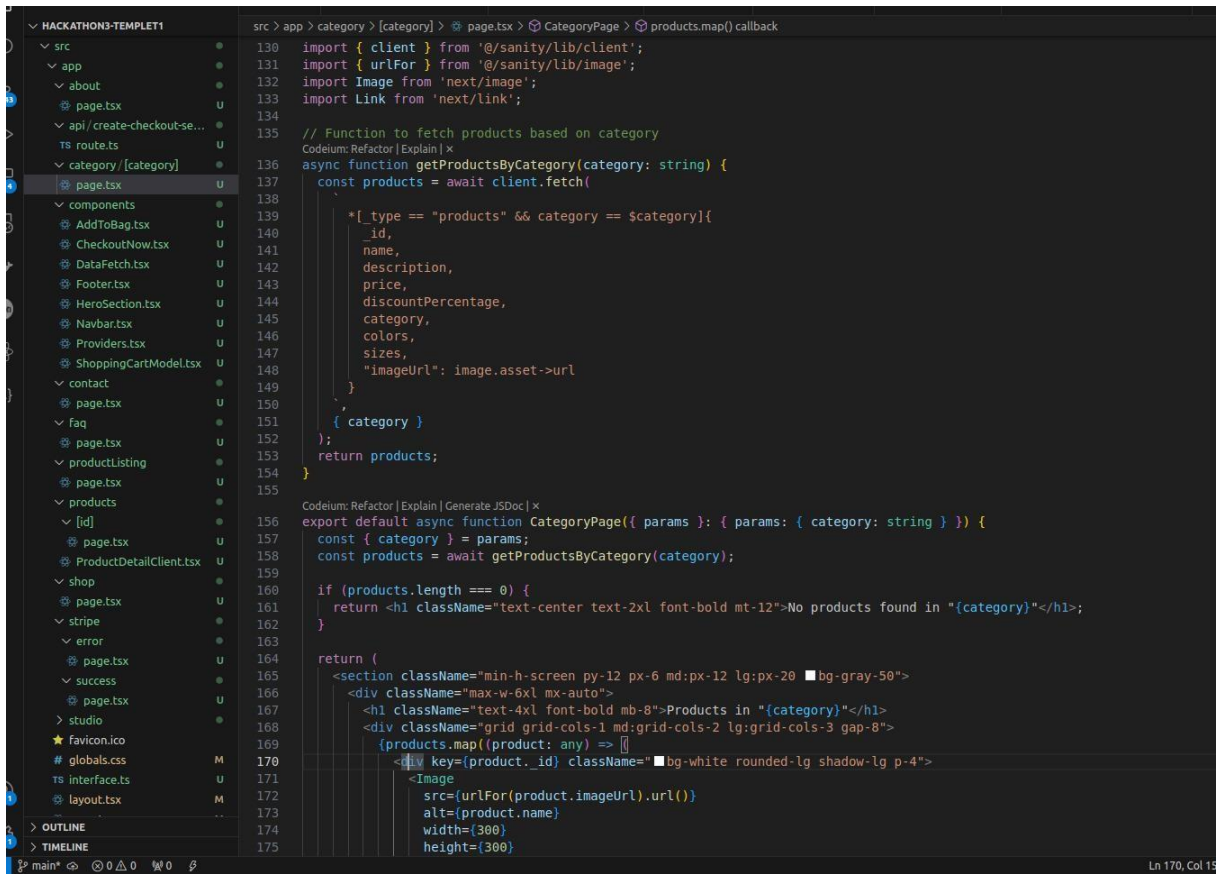
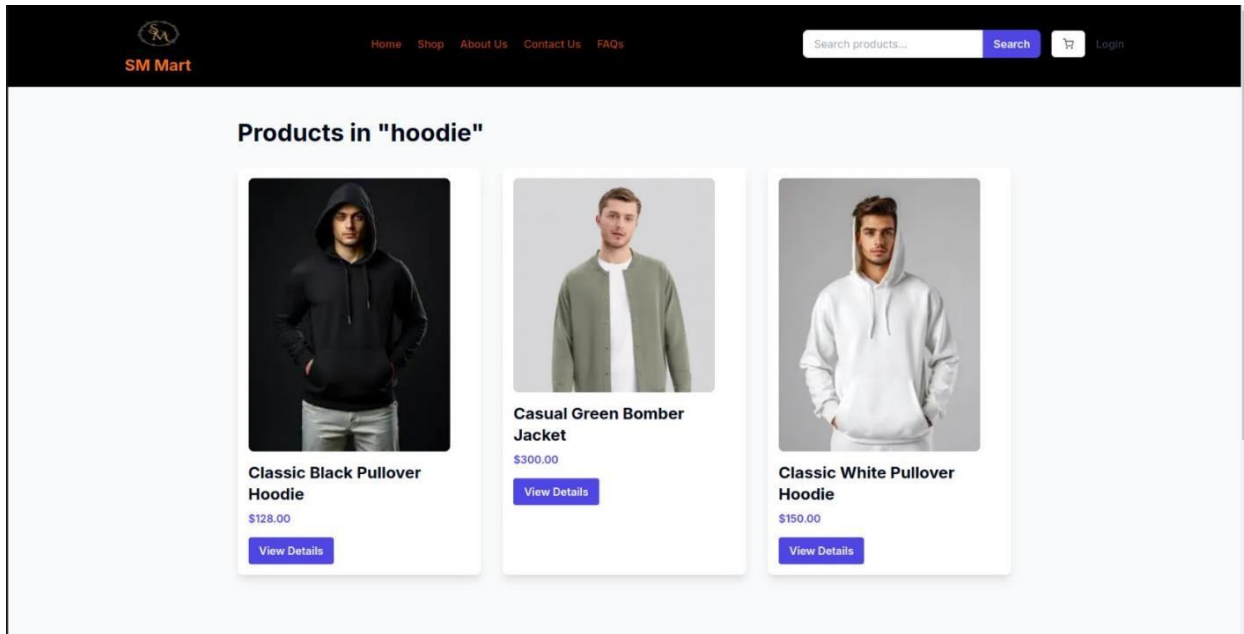
This e-commerce website leverages modern web development frameworks and tools, providing a robust, scalable, and user-friendly platform for online shopping. Its dynamic category page, responsive design, secure payment processing, and real-time data updates make it suitable for marketplace-based applications. The addition of a dynamic Hero Section and price field updates

enhances the website's customization capabilities. Future improvements may include user authentication, multi-currency support, and enhanced analytics.

2. Functional Deliverable









Casual Green Bomber Jacket

This stylish green bomber jacket offers a sleek and modern twist on a classic design. Made from soft and comfortable fabric, it features snap buttons and ribbed cuffs, giving it a sporty yet refined look. The minimalist style makes it perfect for layering over casual t-shirts or hoodies. Whether you're out with friends or just lounging, this jacket provides a laid-back yet fashionable vibe. Its muted green color adds a subtle, earthy tone that pairs well with a variety of outfits, making it a versatile addition to your casual wardrobe.

Available Sizes:

S

XXL

XL

L

\$300.00

Add To Cart

Checkout Now

```

src > app > products > @ ProductDetailClient.tsx > @ ProductDetailClient
export default function ProductDetailClient({ product }: { product: any }) {
  /* Sizes Section */
  {product.sizes && product.sizes.length > 0 && [
    <div className="mb-6 text-center">
      <h2 className="text-lg font-bold text-gray-800 mb-2">Available Sizes:</h2>
      <div className="flex justify-center gap-2">
        {product.sizes.map((size: string, index: number) => (
          <span
            key={index}
            className="px-4 py-2 border border-gray-300 rounded-lg text-gray-600"
            {size}
          >
        ))}
      </div>
    </div>
  ]}

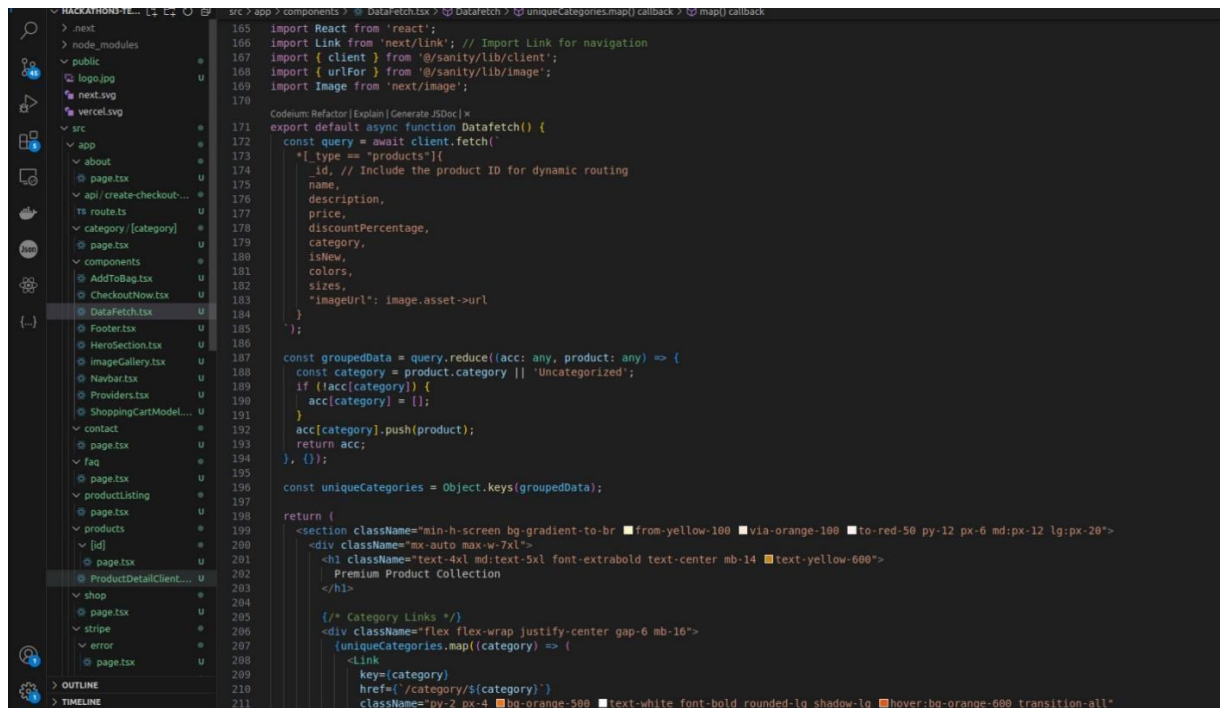
  /* Price */
  <p className="text-2xl font-semibold text-indigo-600 mb-6 text-center">
    ${product.price.toFixed(2)}
  </p>

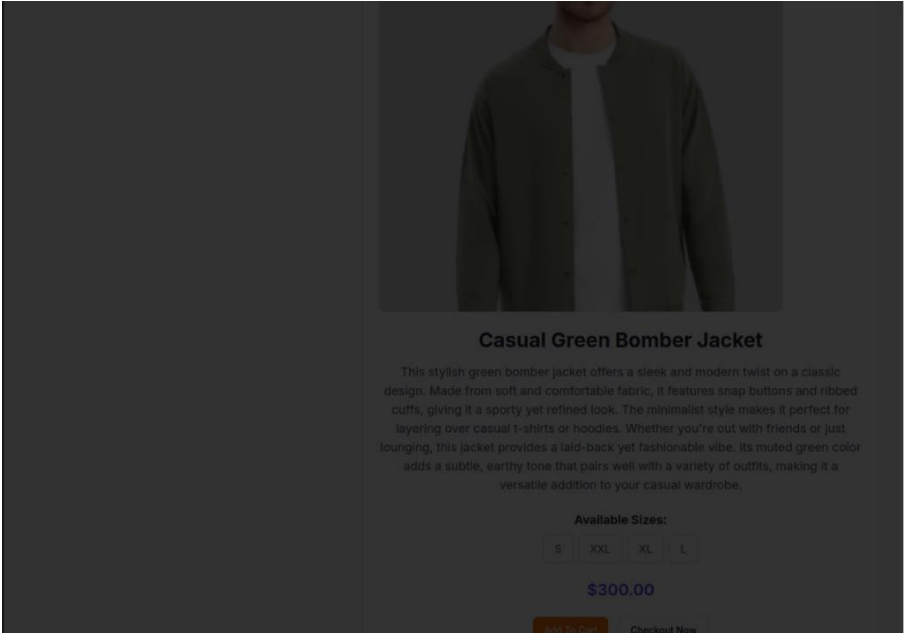
  /* Add to Cart and Checkout Buttons */
  <div className="flex gap-4 justify-center">
    <AddToBag
      currency="USD"
      description={product.description}
      image={product.imageUrl}
      name={product.name}
      price={product.price}
      key={product.id}
      price_id={product._id}
    />
    <CheckoutNow
      currency="USD"
      description={product.description}
      image={product.imageUrl}
      name={product.name}
      price={product.price}
      key={product.id}
      price_id={product._id}
    />
  </div>
</div>
</section>
);
}
```

```

    price: 145,
    discountPercentage: null,
    imageUrl: 'https://cdn.sanity.io/images/nss7ldml/production/18be6c1b158aacd1475323d7544d2bf5640a5b4d-295x298.png'
  },
  {
    _id: 'jQWkZ7WRTG5Z6CroBTK',
    discountPercentage: null,
    category: 'tshirt',
    isNew: true,
    colors: [ 'Red', 'Black', 'Blue', 'White' ],
    sizes: [ 'S', 'L', 'XL', 'M' ],
    imageUrl: 'https://cdn.sanity.io/images/nss7ldml/production/b47f45391dc091331e87f6afa3522e2468e65ed-295x298.png',
    name: 'Sleeve Stripe T-Shirt',
    description: 'This product is a vibrant orange and black striped t-shirt with a sporty design. The shirt features vertical pinstripes in black on an orange base, complemented by contrasting black raglan sleeves for a casual, athletic-inspired look. Ideal for adding a bold touch to your casual wardrobe, this tee combines comfort and style, perfect for daily wear or sporting events. The soft fabric ensures a comfortable fit, while the unique design makes it a standout piece for any outfit.',
    price: 130
  },
  {
    discountPercentage: null,
    category: 'shirt',
    isNew: true,
    colors: [ 'White', 'Black', 'Yellow', 'Blue' ],
    imageUrl: 'https://cdn.sanity.io/images/nss7ldml/production/55e845afa21185a2443f7067fa2911ccd4c68e58-295x298.png',
    _id: 'jQWkZ7WRTG5Z6CroCBN',
    name: 'Checkered Shirt',
    sizes: [ 'XL', 'L', 'XXL', 'M' ],
    description: 'This men's plaid shirt combines timeless style and comfort, featuring a bold red, navy, and white checkered pattern. With a classic button-down collar and long sleeves, it offers a relaxed fit for easy wear. Made from soft, breathable fabric, this shirt is perfect for casual outings, weekends, or layering over a t-shirt. Its versatile design ensures it pairs well with jeans or chinos, making it a wardrobe essential for any season.',
    price: 175
  },
  {
    description: 'Elevate your casual style with this sporty and timeless raglan t-shirt. Featuring a crisp white base adorned with vertical pinstripes, it blends a retro vibe with modern appeal. The contrasting black raglan sleeves add a bold, athletic touch, making it a versatile piece for any wardrobe.'
  },
  {
    'Key Features:\n' +
    '\n' +
    'Soft and breathable fabric for all-day comfort\n' +
    'Classic pinstripe design for a clean, stylish look\n' +
    'Contrasting raglan sleeves for a sporty edge\n' +
    'Regular fit with a crew neckline\n' +
    'Perfect for casual outings or active days, pair this tee with your favorite jeans or joggers for effortless',
    category: 'tshirt',
    isNew: false,
    sizes: [ 'M', 'L', 'S', 'XXL' ],
    _id: 'jQWkZ7WRTG5Z6CroCQ',
    name: 'Black Striped T-Shirt',
    price: 120,
    discountPercentage: null,
    colors: [ 'Blue', 'White', 'Black', 'Red' ],
    imageUrl: 'https://cdn.sanity.io/images/nss7ldml/production/07dc8647a64f4a93a9d1fbb41e0a81fc940b903d-295x298.png'
  }
)

```





Shopping Cart

Classic Black Long Sleeve Button-Down Shirt

This sleek black button-down shirt is the epitome of style and versatility. Designed with a tailored fit, it...

QTY: 1

Remove

Vertical Striped Shirt

This product is a stylish vertical striped shirt, featuring alternating green and darker green stripes...

QTY: 1

Remove

Casual Green Bomber Jacket

This stylish green bomber jacket offers a sleek and modern twist on a classic design. Made from soft...

QTY: 1

Remove

Subtotal:

\$719

Shipping and taxes are calculated at checkout.

Checkout

OR Continue Shopping

```
src > app > components > shoppingCartModel.tsx > ...
267 "use client";
268 import { Button } from "@components/ui/button";
269 import {
270   Sheet,
271   SheetContent,
272   SheetHeader,
273   SheetTitle,
274 } from "@components/ui/sheet";
275
276 import Image from "next/image";
277 import { useShoppingCart } from "use-shopping-cart";
278
279 export default function ShoppingCartModal() {
280   const {
281     cartCount,
282     shouldDisplayCart,
283     handleCartClick,
284     cartDetails,
285     removeItem,
286     totalPrice,
287     redirectToCheckout,
288   } = useShoppingCart();
289
290   async function handleCheckoutClick(event: any) {
291     event.preventDefault();
292     try {
293       const result = await redirectToCheckout();
294       if (result?.error) {
295         console.error("Checkout error:", result.error);
296       }
297     } catch (error) {
298       console.error("Error during checkout:", error);
299     }
300   }
301
302   return (
303     <Sheet open={shouldDisplayCart} onOpenChange={() => handleCartClick()}>
304       <SheetContent className="sm:max-w-lg w-[90vw]">
305         <SheetHeader>
306           <SheetTitle>Shopping Cart</SheetTitle>
307         </SheetHeader>
308
309         <div className="h-full flex flex-col justify-between">
310           <div className="mt-8 flex-1 overflow-y-auto">
311             <ul className="my-6 divide-y divide-gray-200">
312               {cartCount === 0 ? (
```

```
src > sanity > schemaTypes > ts products.ts > @ default > @ fields
1 import { defineType } from "sanity"
2
3 export default defineType({
4   name: 'products',
5   title: 'Products',
6   type: 'document',
7   fields: [
8     {
9       name: 'name',
10      title: 'Name',
11      type: 'string',
12    },
13    {
14      name: 'price',
15      title: 'Price',
16      type: 'number',
17    },
18    {
19      name: 'price id',
20      title: 'Stripe Price ID',
21      type: 'string',
22    },
23    {
24      name: 'description',
25      title: 'Description',
26      type: 'text',
27    },
28    {
29      name: 'image',
30      title: 'Image',
31      type: 'image',
32    },
33    {
34      name: 'category',
35      title: 'Category',
36      type: 'string',
37      options: {
38        list: [
39          {title: 'T-Shirt', value: 'tshirt'},
40          {title: 'Short', value: 'short'},
41          {title: 'Jeans', value: 'jeans'},
42          {title: 'Hoodie', value: 'hoodie'},
43          {title: 'Shirt', value: 'shirt'},
44        ]
45      }
46    },
47    {
48      name: 'discountPercent',
```

```
src > sanity > schemaTypes > ts indexes.ts > ...
1 import { type SchemaTypeDefinition } from 'sanity'
2 import products from './products'
3 import heroslides from './heroslides'
4
5 export const schema: { types: SchemaTypeDefinition[] } = {
6   types: [products, heroslides],
7 }
8
```