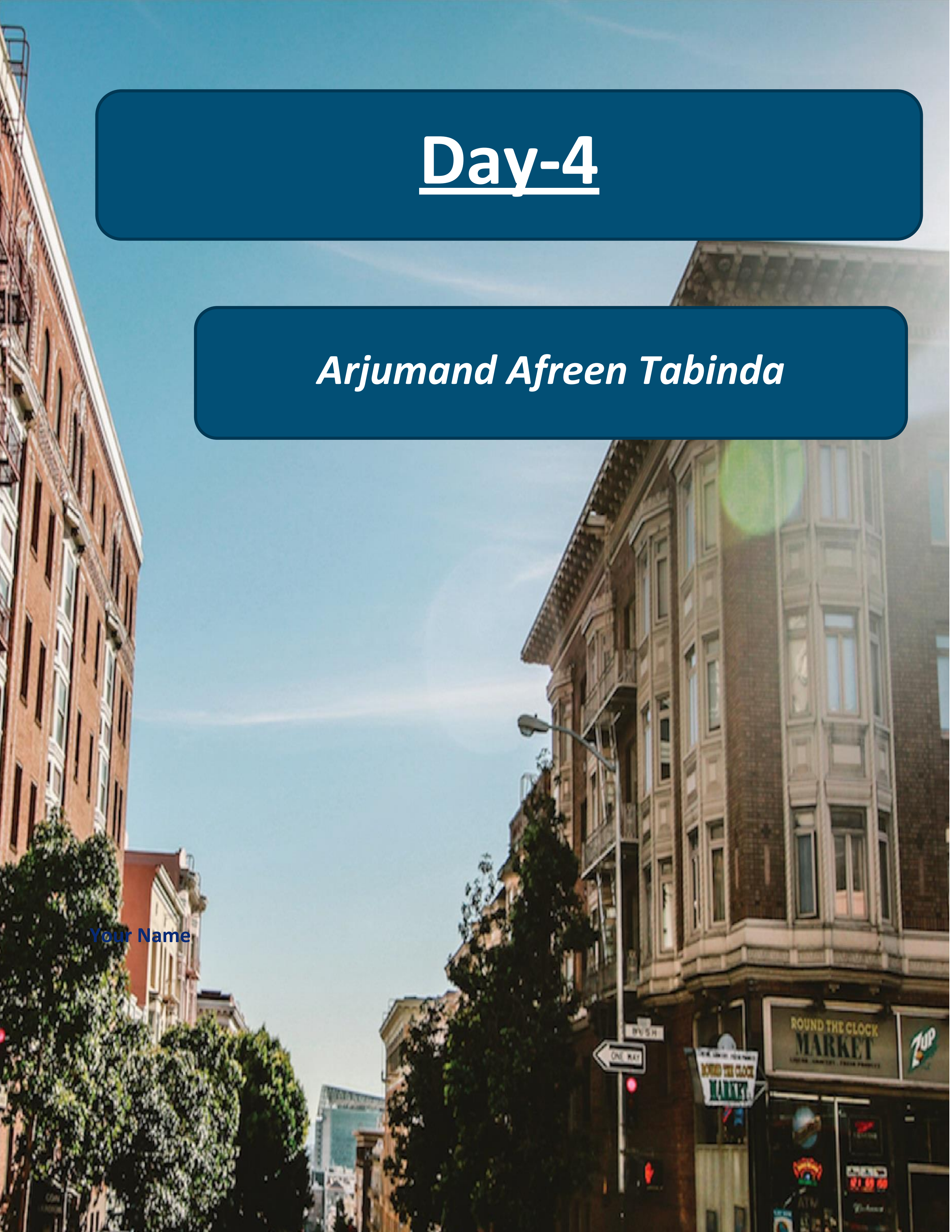
Day-4

Arjumand Afreen Tabinda



Your Name

Accomplishments on Day 4:

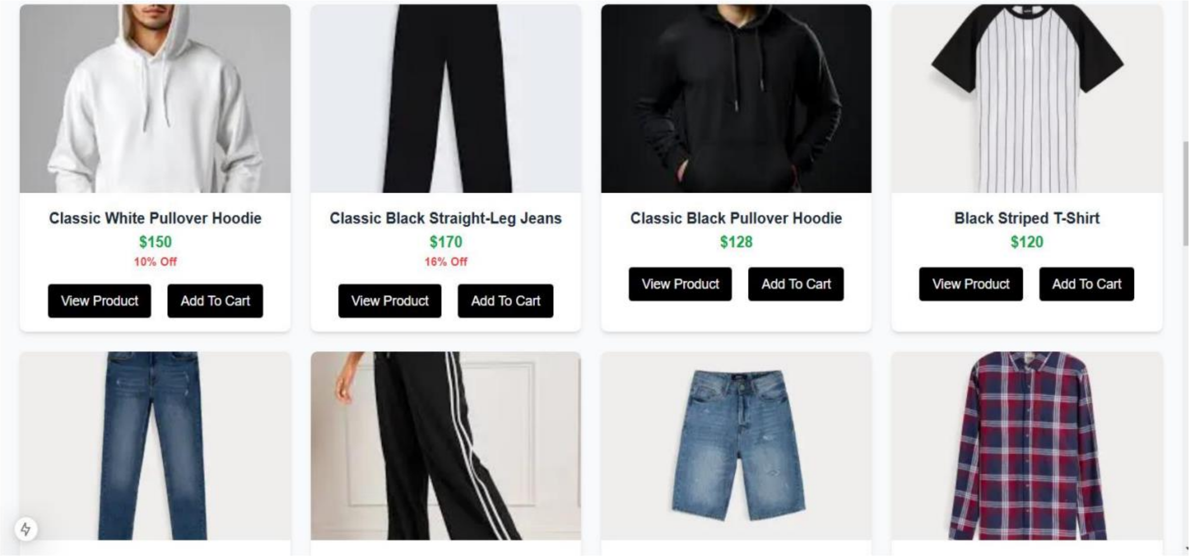
1. Products Listing Page

   • Developed a Products Listing Page by retrieving data dynamically from Sanity CMS.

   • Created product cards with hover effects, ensuring they were both visually attractive and interactive.

   • Employed reusable components for a tidy and well-structured codebase.

   • Optimized the page for responsiveness, ensuring it adapts seamlessly to various devices (e.g., mobile, tablet, desktop).



Classic White Pullover Hoodie
$150
10% Off

Classic Black Straight-Leg Jeans
$170
16% Off

Classic Black Pullover Hoodie
$128

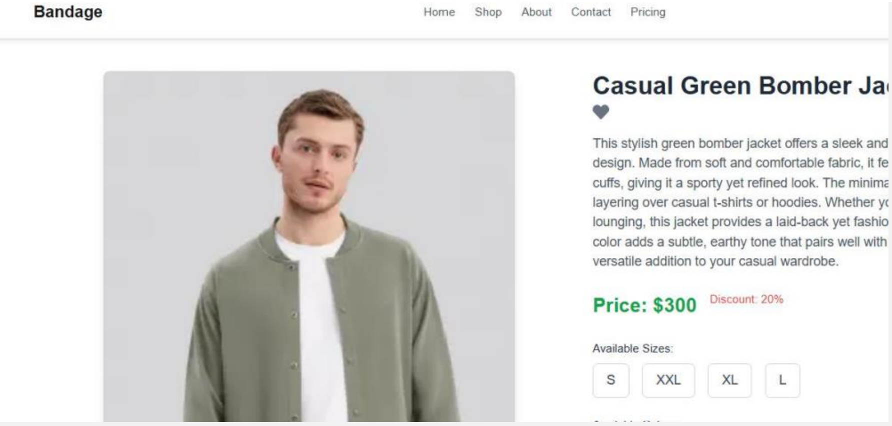
Black Striped T-Shirt
$120

2. Product Information Page

   • Developed a Product Information Page by defining a slug in Sanity schemas and generating a unique slug for each product.

   • Designed the page to dynamically present detailed product information retrieved from Sanity.

• Prioritized an intuitive layout to ensure smooth navigation and readability.



Casual Green Bomber Ja

This stylish green bomber jacket offers a sleek and
design. Made from soft and comfortable fabric, it fe
cuffs, giving it a sporty yet refined look. The minima
layering over casual t-shirts or hoodies. Whether yc
lounging, this jacket provides a laid-back yet fashio
color adds a subtle, earthy tone that pairs well with
versatile addition to your casual wardrobe.

Price: $300 Discount: 20%

Available Sizes:

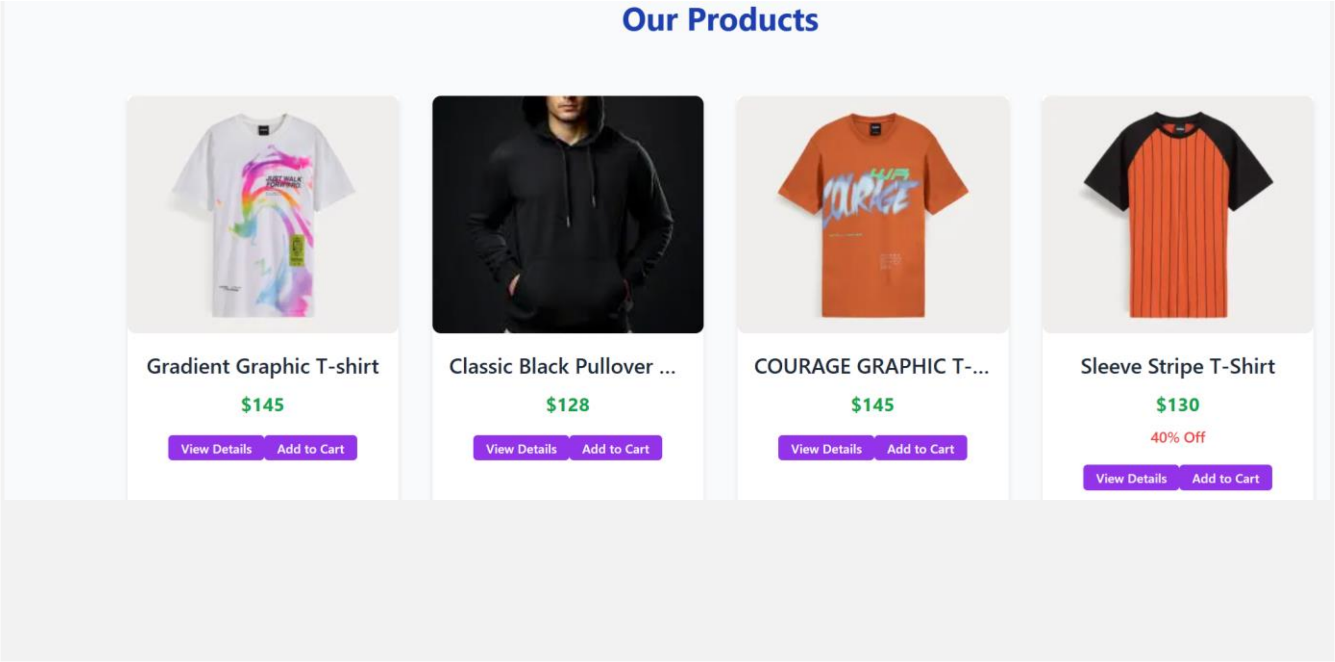
3. *Filter Implementation*

   • Included a Categories Filter to sort products such as Men's Shirts, T-Shirts, Hoodies, etc., based on categories defined in Sanity.

   • Added a Price Filter to enable users to sort products by price range.

   • Leveraged React hooks and state management for smooth filter functionality.

   • Ensured the filters were effective and seamlessly integrated with the user interface.



Our Products

Gradient Graphic T-shirt
$145

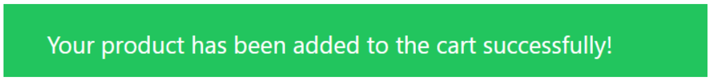
Classic Black Pullover ...
$128

COURAGE GRAPHIC T-...
$145

Sleeve Stripe T-Shirt
$130
40% Off

4: Add to cart –

I also incorporated cart functionality, so when a user clicks on the cart icon, they'll receive a confirmation message indicating that the item has been successfully added to the cart.



Your product has been added to the cart successfully!

5. ***UI Improvements and Animations***

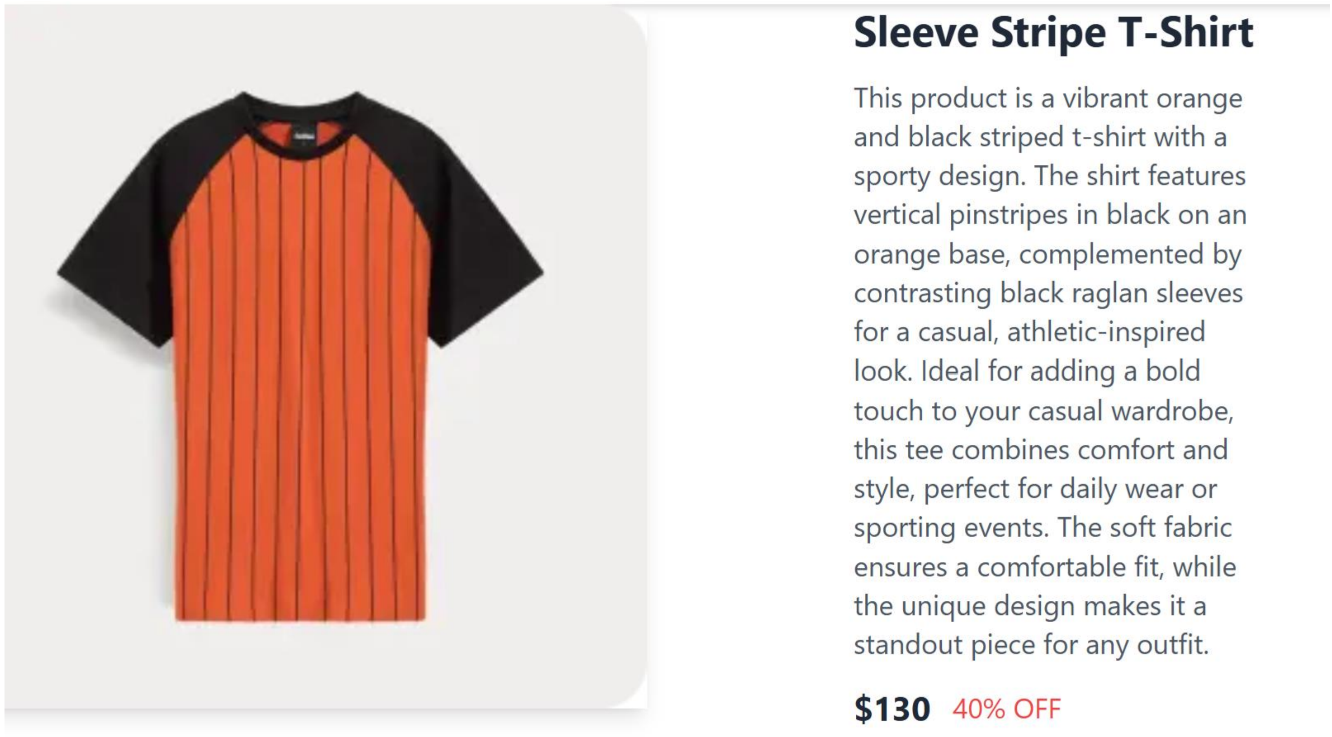
   • Focused on refining the user interface to enhance the overall user experience.

   • Incorporated animations and hover effects to make the website more interactive and engaging.

   • Maintained a consistent and visually appealing design throughout all pages.

6. ***Mobile-Friendly Design***

   • Made sure the entire website was fully responsive and optimized for various devices.



Sleeve Stripe T-Shirt

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.

$130 40% OFF

7. Modular Components

• Developed dynamic components for products and other elements to ensure consistency and reusability.

• Made these components reusable throughout the project, minimizing repetition and enhancing maintainability.

8.Code details

This code represents a React component for displaying products in a shopping interface. Here's an analysis of its structure and functionality:

Imports:

- ***Image****: A component from Next.js to handle optimized image rendering.*

- ***useState, useEffect***: *React hooks used for managing state and performing side effects (like fetching data).*
- ***useRouter***: *A Next.js hook used to navigate programmatically (though not used here for navigation, it's imported).*
- ***client, allProducts***: *Imports from Sanity to fetch data from the Sanity CMS.*
- ***urlFor***: *A utility from Sanity to generate image URLs.*
- ***Products***: *A TypeScript type representing the structure of product data.*
- ***Link***: *A Next.js component to handle navigation between pages.*

Component Breakdown:

1. ***State Variables:***
   - *products: An array of products fetched from the Sanity CMS.*
   - *successMessage: A string that shows a success message when an item is added to the cart.*
2. ***useEffect Hook:***
   - *The hook is used to fetch the products from the Sanity CMS on component mount ([] as dependency).*
   - *The products are fetched using client.fetch(allProducts), which calls the allProducts query from the Sanity CMS and stores the fetched data in the products state.*
3. ***handleClick Function:***
   - *This function is triggered when the "Add to Cart" button is clicked.*

  - *It logs the added product and sets a success message in the state.*
  - *After 3 seconds, the success message is cleared using setTimeout.*

4. ***UI Layout:***
    - *The layout includes a section with a title and a grid to display the products.*
    - *It conditionally renders either the product list or a loading message if the products are not yet fetched.*
    - *For each product, it displays the name, price, discount (if available), and an image (if available).*
    - *Each product has two buttons: one to view more details (linked to a product-specific page) and another to add the product to the cart.*
5. ***Styling:***
    - *Tailwind CSS classes are used for layout, colors, spacing, and hover effects.*
    - *Cards are created for each product, with hover effects and smooth scaling when interacting.*
    - *The "Add to Cart" button shows a success message when clicked, and this message is displayed as a fixed banner at the top of the page.*

Code Behavior:

- *On initial render, the component fetches product data and populates the products state.*
- *Each product is displayed in a responsive grid layout.*

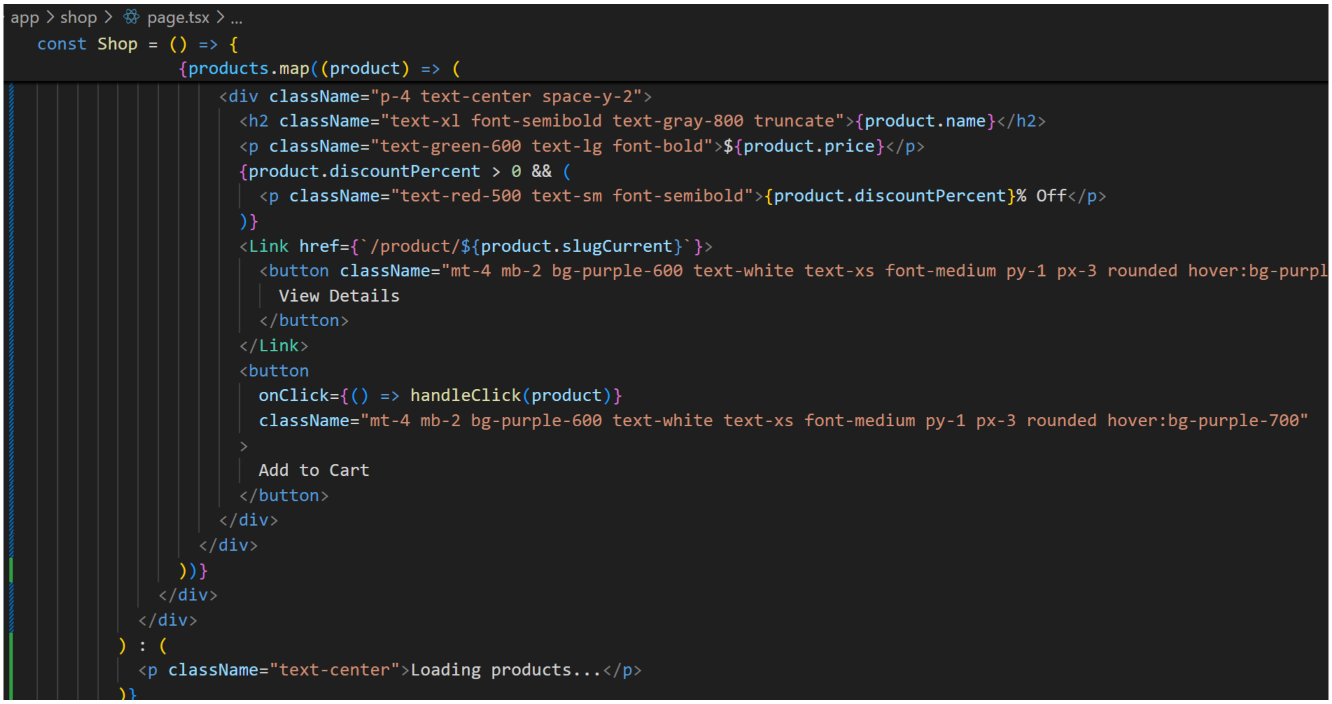
- *Users can click the "Add to Cart" button for each product, and a success message is shown temporarily at the top of the page.*

Observations and Suggestions:

- ***Unused router import****: The useRouter hook is imported but not used in this code, so it can be removed unless intended for future functionality.*
- ***Loading state****: A more visually appealing loading indicator could be implemented for better UX.*
- ***Success Message Placement****: The success message is fixed at the top of the page, which works well for visibility, but it could potentially overlap content if the user scrolls too much. Consider adding spacing or animation for smoother user experience.*

Key Features:

- ***Dynamic Product Rendering****: Fetches and displays products dynamically from Sanity CMS.*
- ***Cart Interaction****: Displays a success message when an item is added to the cart.*
- ***Responsive Design****: Uses Tailwind CSS to ensure the layout is responsive across various device sizes.*



```
app > shop > page.tsx > ...
const Shop = () => {
              {products.map((product) => (
                  <div className="p-4 text-center space-y-2">
                    <h2 className="text-xl font-semibold text-gray-800 truncate">{product.name}</h2>
                    <p className="text-green-600 text-lg font-bold">${product.price}</p>
                    {product.discountPercent > 0 && (
                      <p className="text-red-500 text-sm font-semibold">{product.discountPercent}% Off</p>
                    )}
                    <Link href={`/product/${product.slugCurrent}`}>
                      <button className="mt-4 mb-2 bg-purple-600 text-white text-xs font-medium py-1 px-3 rounded hover:bg-purpl
                        View Details
                      </button>
                    </Link>
                    <button
                      onClick={() => handleClick(product)}
                      className="mt-4 mb-2 bg-purple-600 text-white text-xs font-medium py-1 px-3 rounded hover:bg-purple-700"
                    >
                      Add to Cart
                    </button>
                  </div>
                </div>
              ))}
            </div>
          </div>
        ) : (
          <p className="text-center">Loading products...</p>
        )}
```

This code defines a React component that displays products within an e-commerce interface. Here's an overview of its structure and behavior:

Imports:

- ***Image****: A Next.js component for optimized image rendering.*
- ***useState, useEffect****: React hooks for managing component state and handling side effects (such as data fetching).*
- ***useRouter****: A Next.js hook for programmatic navigation (though not utilized here, it's imported).*
- ***client, allProducts****: These imports come from Sanity, allowing the component to fetch product data from Sanity CMS.*
- ***urlFor****: A helper function from Sanity for generating image URLs.*
- ***Products****: A TypeScript type defining the shape of product data.*

- ***Link****: A Next.js component that facilitates navigation between pages.*

Component Breakdown:

1. ***State Variables:***
   - *products: An array to hold the fetched product data from Sanity.*
   - *successMessage: A state variable to store a success message when a product is added to the cart.*
2. ***useEffect Hook:***
   - *The hook runs when the component is first mounted, fetching product data from Sanity CMS using client.fetch(allProducts).*
   - *The fetched products are stored in the products state variable.*
3. ***handleClick Function:***
   - *Triggered when a product's "Add to Cart" button is clicked.*
   - *It logs the added product to the console and sets a success message.*
   - *After 3 seconds, the success message is removed from the UI using setTimeout.*
4. ***UI Layout:***
   - *Displays a section with a heading and a grid layout for the products.*
   - *If products are loaded, it renders the product list; otherwise, it shows a loading message.*

- *Each product shows its name, price, discount (if any), and image (if available).*
- *Two buttons are included: one to view more details (linked to a product page) and another to add the product to the cart.*

5. ***Styling:***
    - *Tailwind CSS is used for styling, providing grid layout, spacing, colors, hover effects, and responsiveness.*
    - *Product cards are displayed with hover animations and scaling effects.*
    - *The "Add to Cart" button triggers a success message, which is displayed as a fixed banner at the top of the page.*

Code Functionality:

- *When the component is first loaded, it fetches products from the CMS and stores them in the products state.*
- *The products are displayed in a responsive grid layout.*
- *Clicking the "Add to Cart" button for a product shows a success message temporarily at the top of the page.*

Suggestions and Observations:

- ***Unused router import**: The useRouter hook is not used in the current code. It can be removed unless there are plans to use it for navigation.*
- ***Loading State**: A more visually appealing loading indicator could enhance the user experience.*
- ***Success Message Display**: The success message is displayed as a fixed banner, which is effective for visibility but could*

potentially overlap with content if the page is scrolled too much. A more refined animation or spacing could improve this experience.

Key Features:

- ***Dynamic Product Display****: Fetches and dynamically displays product data from Sanity CMS.*
- ***Cart Interaction****: Displays a success message upon adding a product to the cart.*
- ***Responsive Design****: The layout adapts to different screen sizes, ensuring it's user-friendly on mobile, tablet, and desktop.*