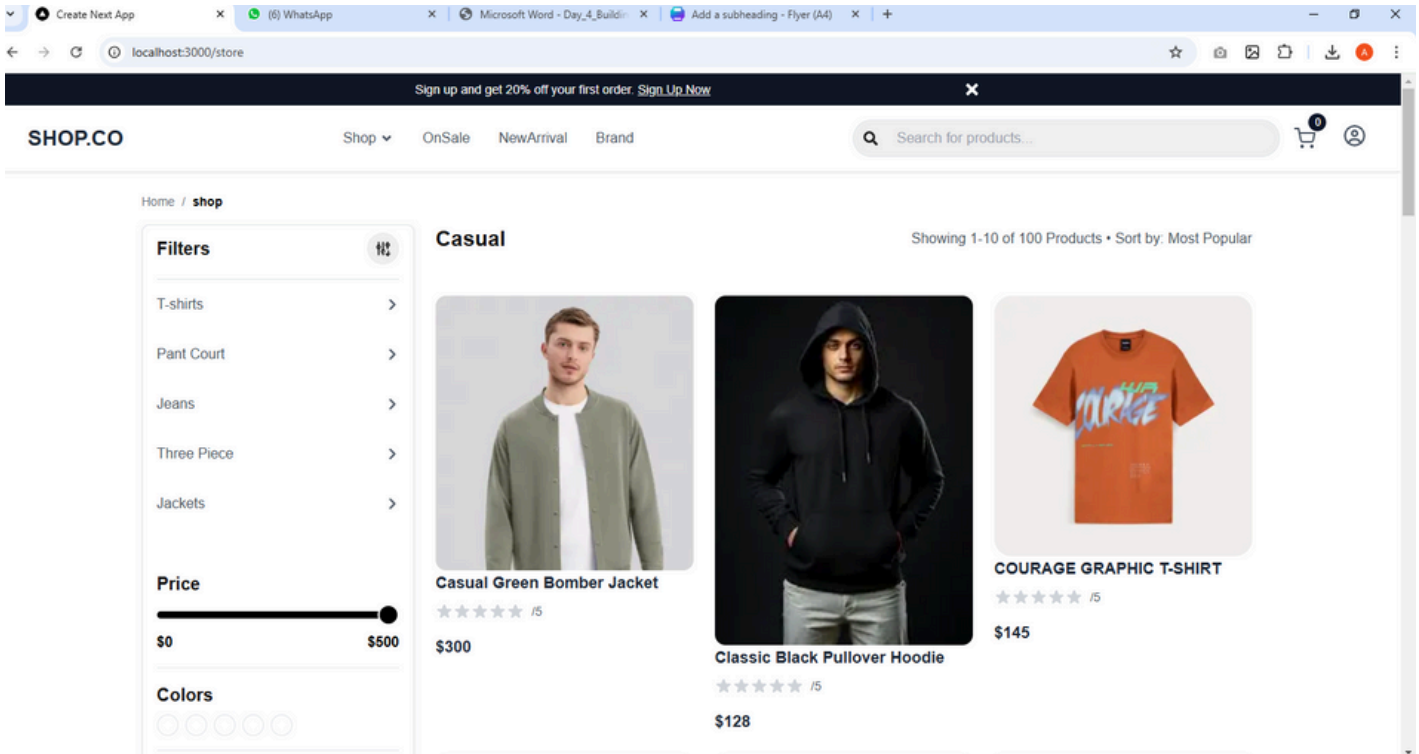
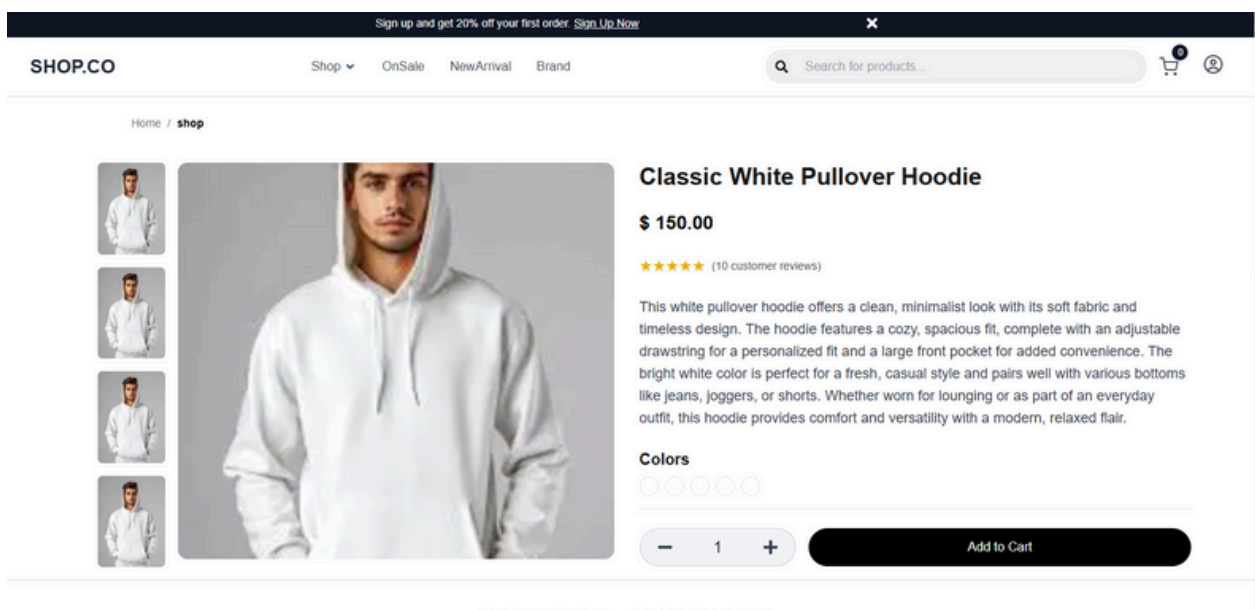


Day 4 - Dynamic Frontend Components - shop.co

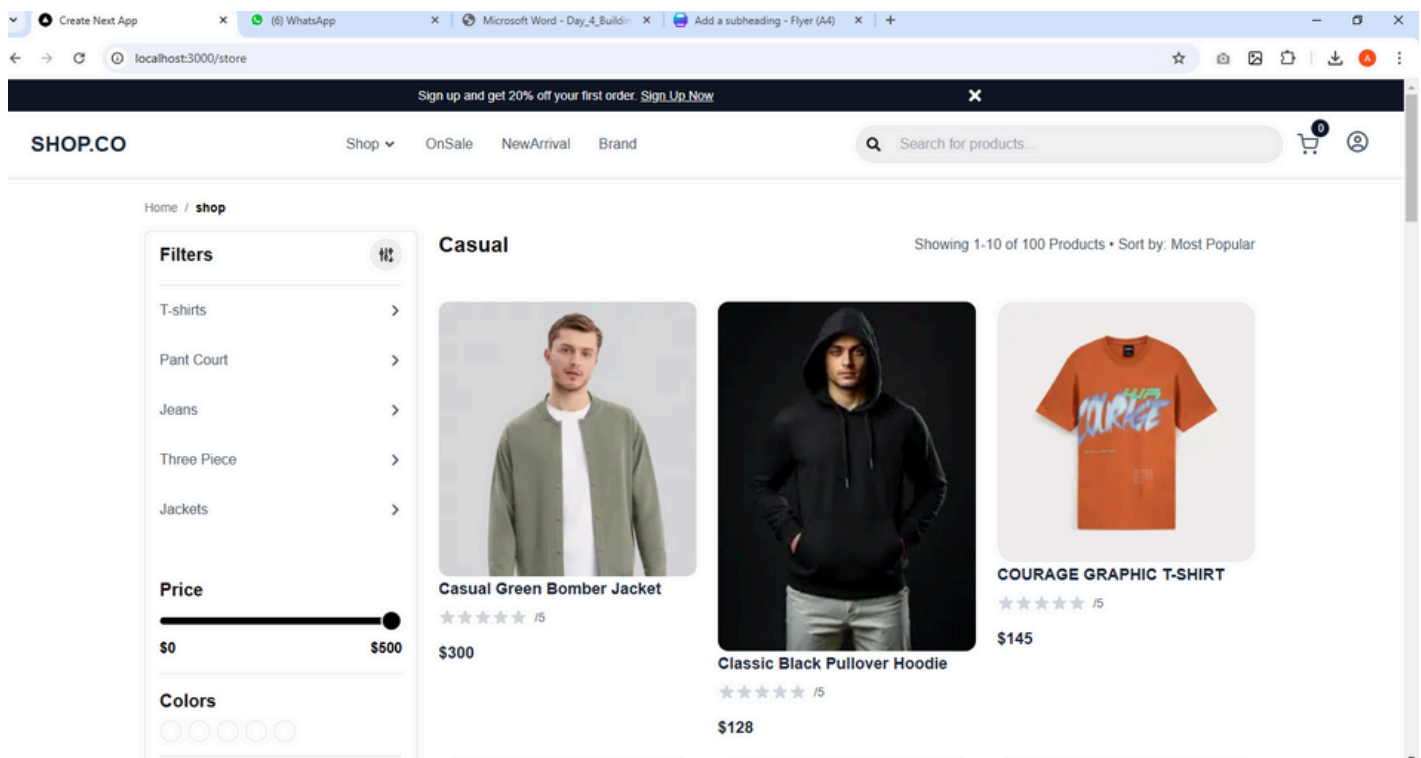
1.The product listing page with dynamic data.



2. product detail page



3.Working category filters



Product Listing

To display a list of products, I started by creating a GROQ query in Sanity Vision to fetch the required data. Using useEffect and useState, I retrieved the data and stored it in a state variable. In the return statement, I mapped over this state variable to render product cards. After designing the cards, the products displayed correctly on the output.

A challenge I encountered was implementing a skeleton loader for the product cards to display during the loading state.

Product Detail Page

For the product detail page, I set up a dynamic route. Using the id parameter from the route, I constructed a GROQ query in Sanity with a condition to fetch and display the product matching the route parameter.

Additional Functionalities

- Filter Functionality:
- I implemented filtering using useState. Conditions for filter items were added to the GROQ query, and the filtered products were rendered by mapping over the results.
- Search Functionality:
- A search bar was added to the navbar. The search query matches the input value with product titles stored in Sanity and displays the corresponding results.
- Pagination:
- To manage large datasets, I implemented pagination logic. If the total number of items (data.length) exceeded 20, items from the 21st onward were displayed on the next page, accessible via a second tab.