DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS

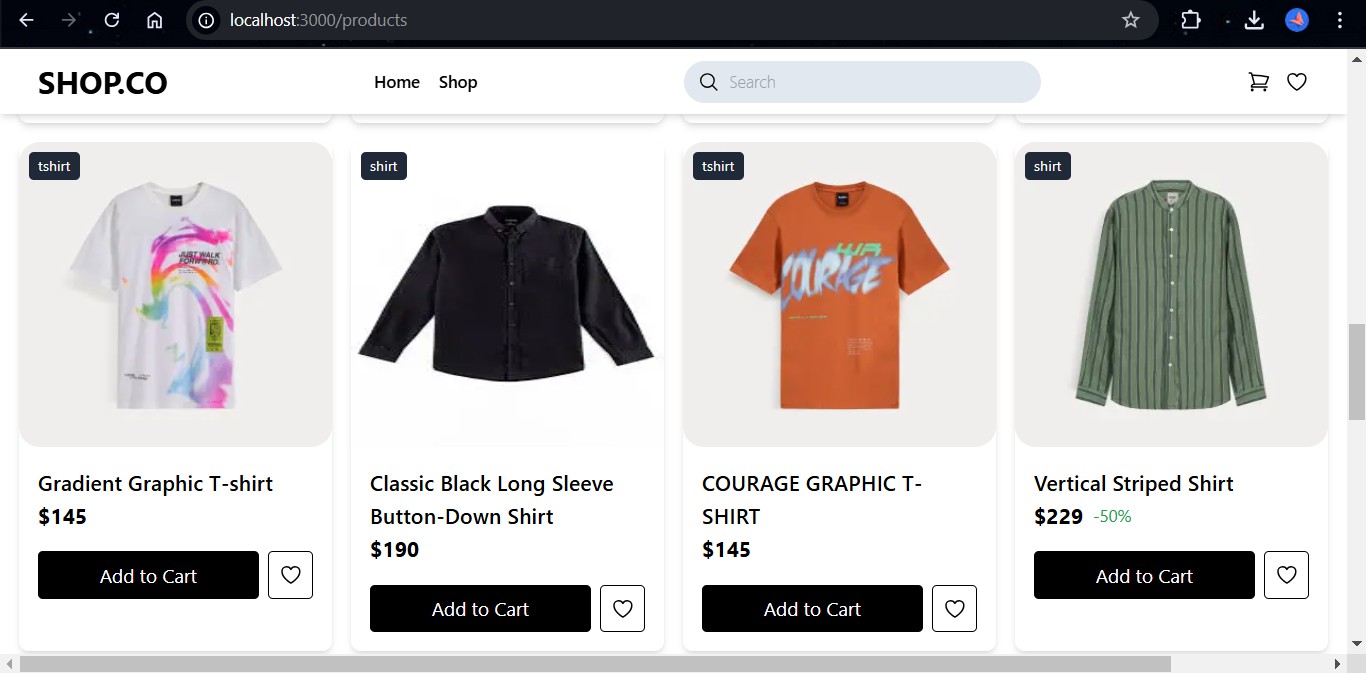
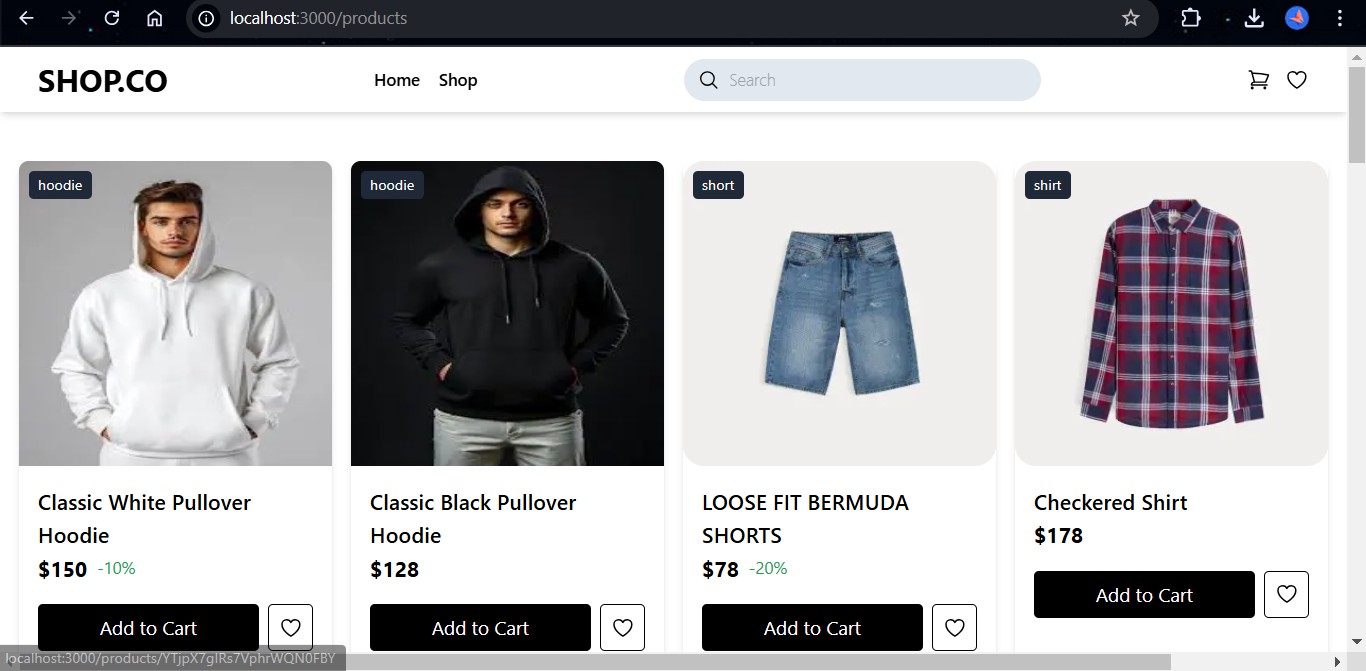
# Objective

The objective of this report is to document the integration of core functional features and components for the e-commerce marketplace. This includes the development of product listing pages, individual product detail pages, and dynamic features such as category filters, search functionality, pagination, and related product displays.

# Functional Deliverables

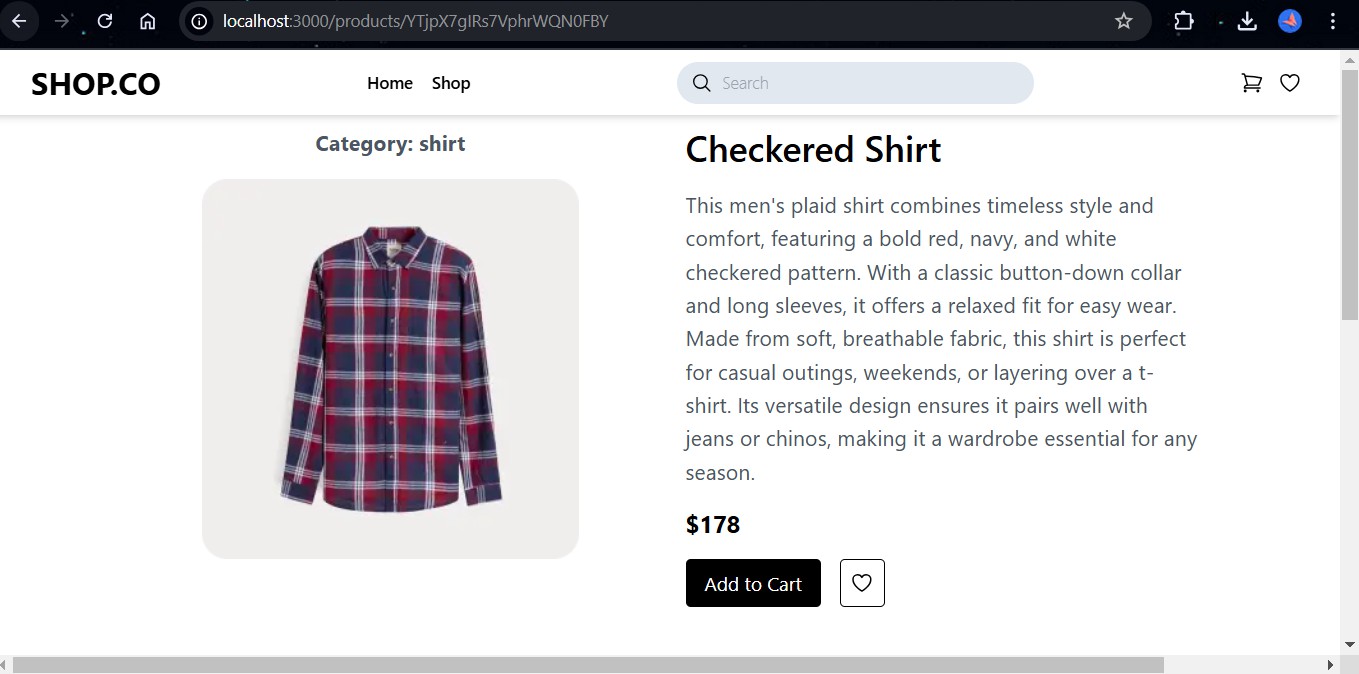
## Product Listing Page

* + **Description**: The product listing page displays dynamically fetched products, including their images, categories, and pricing.



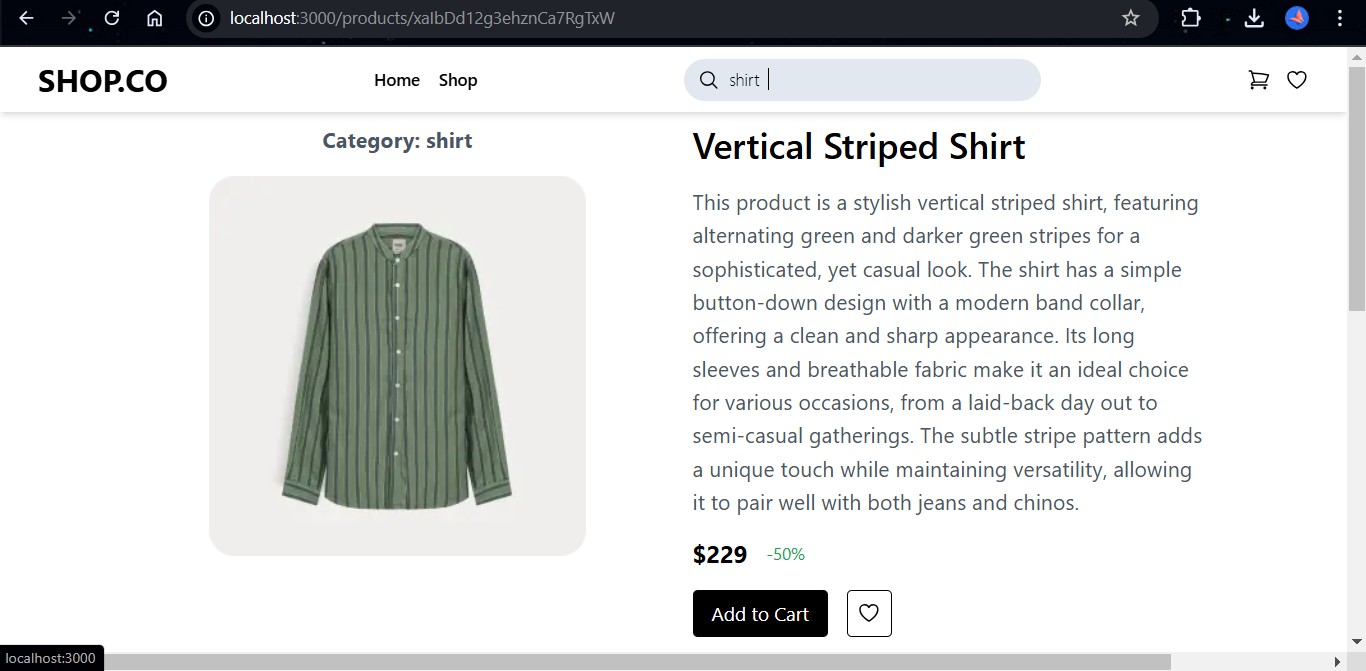
## Product Detail Pages

* + **Description**: Individual product pages are accessible through dynamic routing, showing detailed information about each product.



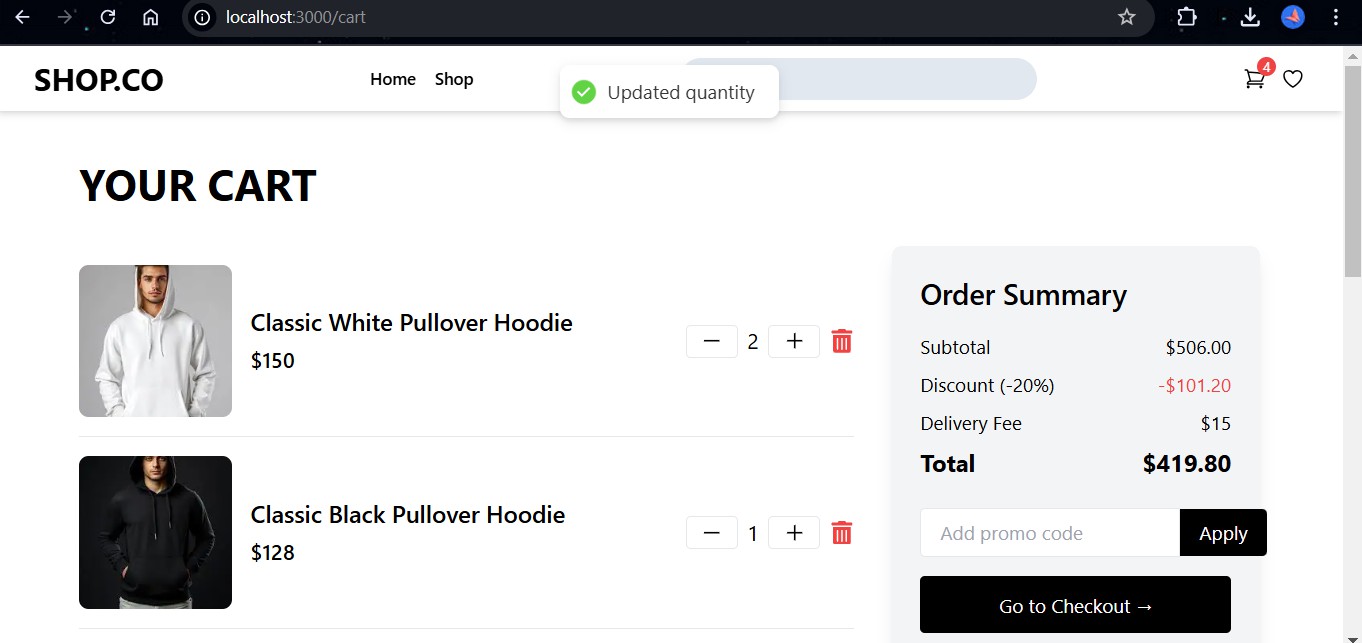
## Category Filters, Search Bar, and Pagination

* + **Description**: Implemented category filters, and a search bar for querying products for seamless navigation.



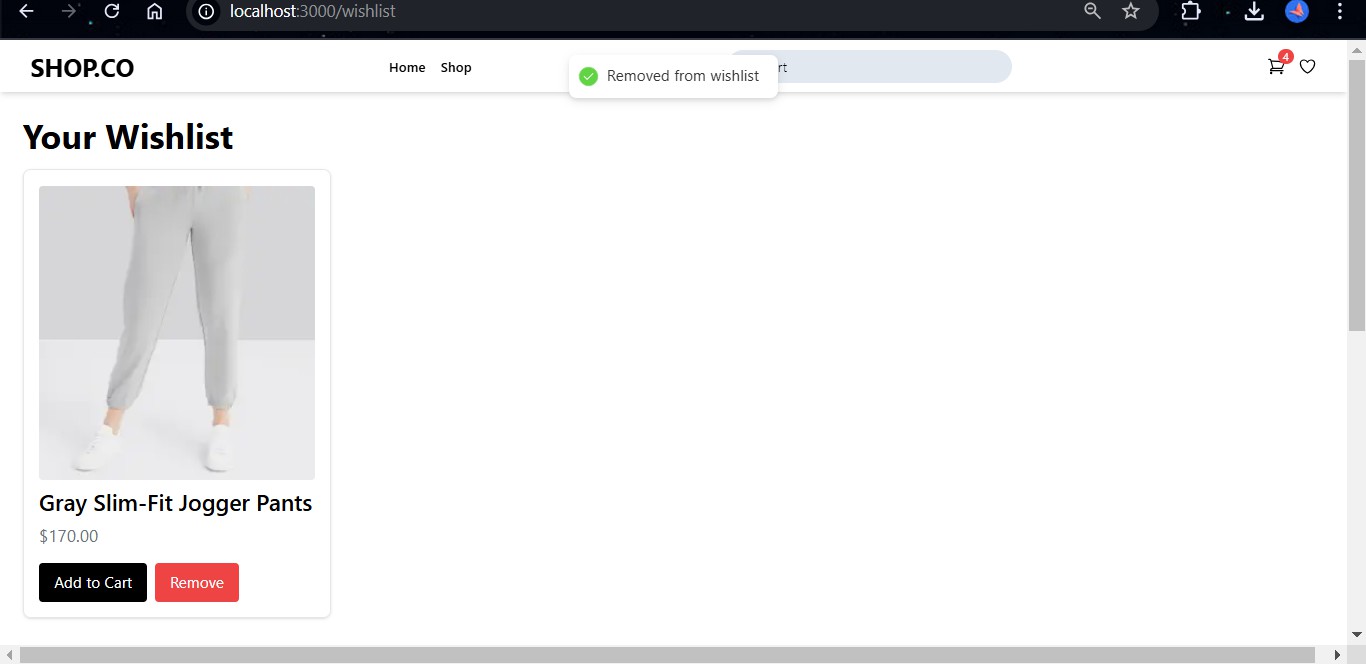
## Cart Functionality

* + **Description**: A cart component was developed to manage product selections, including adding and removing items and updating quantities.



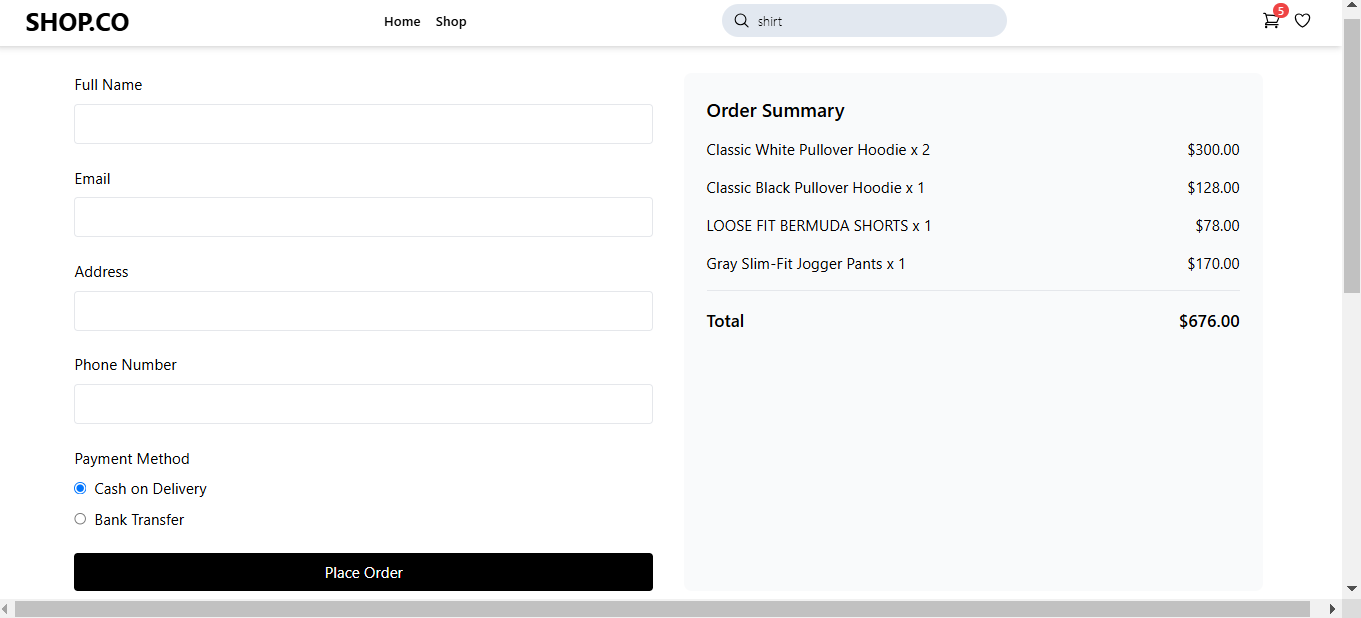
## Wishlist Functionality

* + **Description**: Users can add items to their wishlist for future purchases.



## Checkout Process

* + **Description**: A simple and secure checkout process was implemented, including collecting shipping information, payment methods, and order confirmation.



## Steps Taken

* + Developed the product listing page with dynamic data fetched via API.
  + Implemented dynamic routing for individual product detail pages.
  + Integrated category filters, and search functionality to enhance navigation.

## Challenges and Solutions

* + **Challenge**: Implementing dynamic routing for product detail pages.
    - **Solution**: Used Next.js dynamic routing and page generation to handle individual product detail pages based on their ID.

## Best Practices

* + Emphasized code modularity by creating reusable components
  + Implemented error handling and loading states during API data fetching to ensure a smooth user experience.
  + Followed React best practices, ensuring clean, maintainable code.