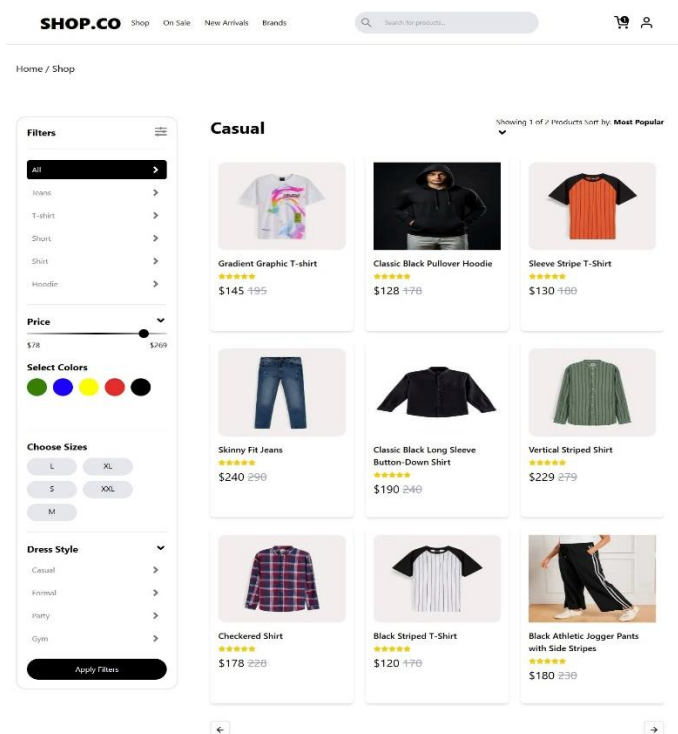# Day 4- Dynamic Frontend Components-Shop.Co

Introduction

This document outlines the development of dynamic frontend components for the marketplace website. The goal was to create a fully functional, user-friendly interface that integrates with the backend to display product listings, detailed product information, and allows users to filter and search through products seamlessly. The key components of the project include a dynamic product listing page, individual product detail pages, search bar, category filters, pagination, and the **Add to Cart** functionality. Additionally, I have implemented features like related products and a user profile section to enhance the overall user experience.
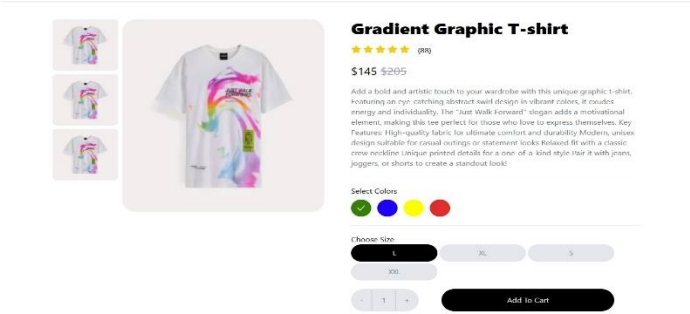
1. Functional Deliverables

1. **Product Listing Page**
   - Displays all products fetched dynamically from Sanity CMS.
   - Each product has its image, name, price
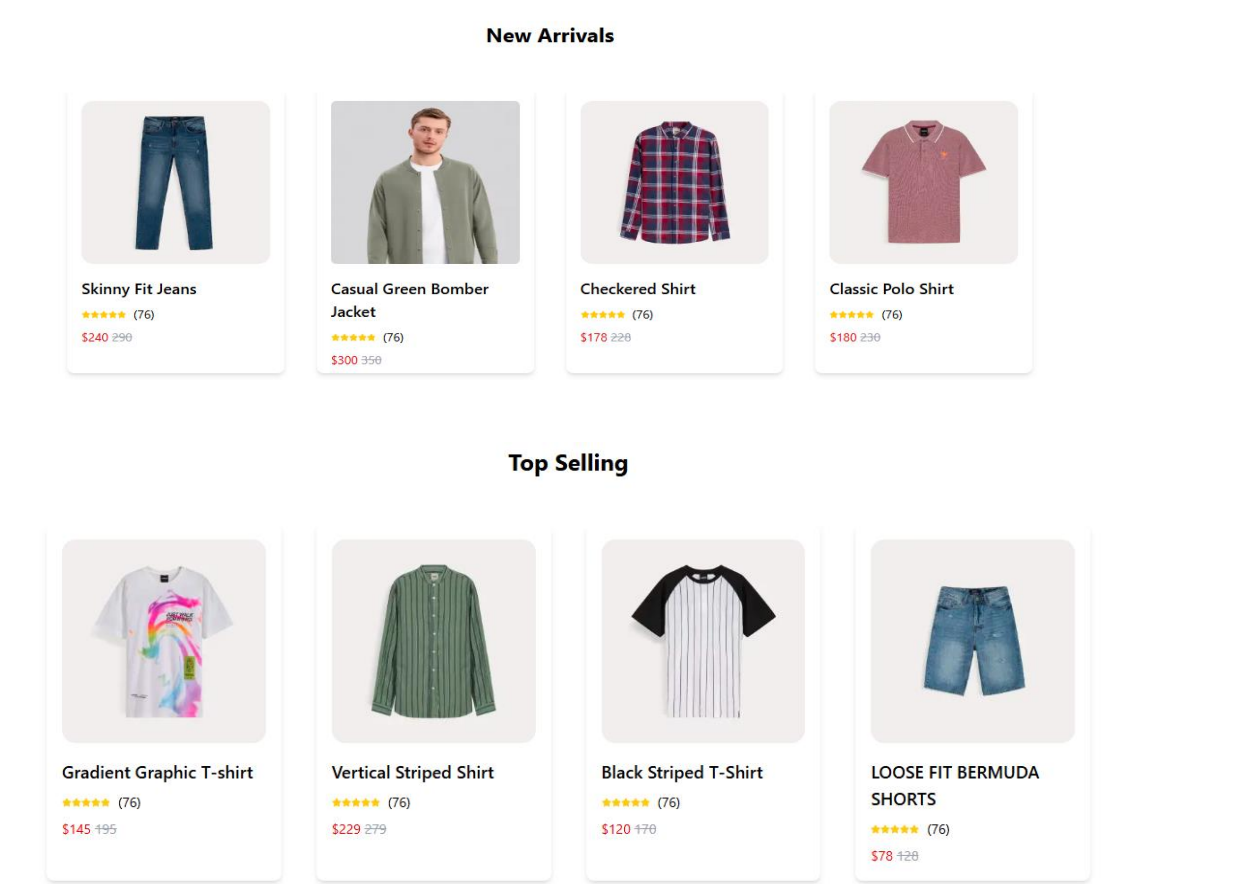   - Pagination ensures smooth navigation across multiple pages of products.

2. **Individual Product Detail Pages**
   o Accurate dynamic routing using slugs for each product.
   o Pages display product-specific details like name, price, description, and an 'Add to Cart' button.
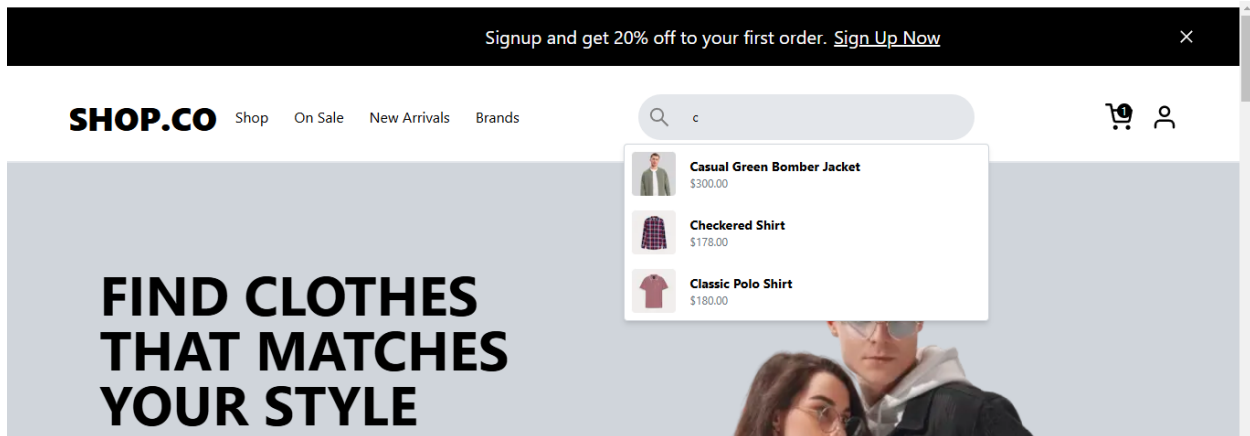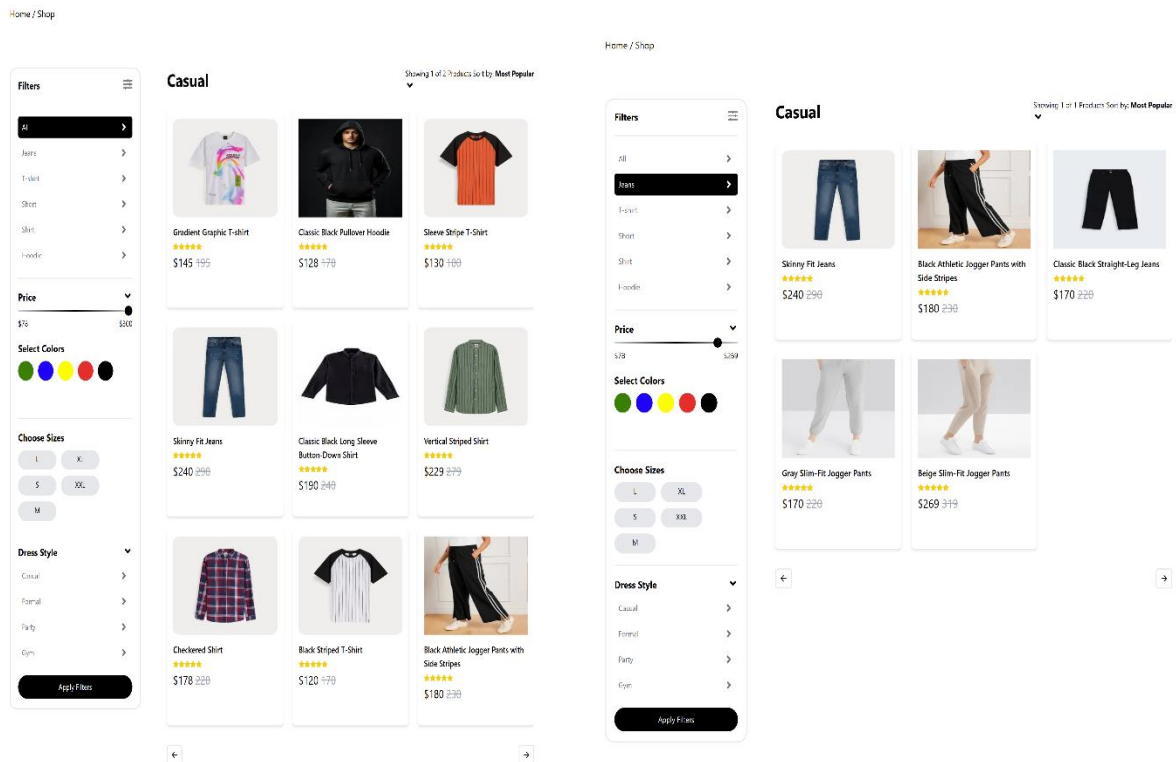

3. **Category Filters, Search Bar, and Pagination**
   o Functional category filter to display products based on specific categories.

New Arrivals

Skinny Fit Jeans	Casual Green Bomber Jacket	Checkered Shirt	Classic Polo Shirt
★★★★★ (76)	★★★★★ (76)	★★★★★ (76)	★★★★★ (76)
$240 290	$300 350	$178 228	$180 230

Top Selling

Gradient Graphic T-shirt	Vertical Striped Shirt	Black Striped T-Shirt	LOOSE FIT BERMUDA SHORTS
★★★★★ (76)	★★★★★ (76)	★★★★★ (76)	★★★★★ (76)
$145 195	$229 279	$120 170	$78 120

o Search bar to find products by name or keyword.

o Pagination to manage and display product lists efficiently.

4. **Additional Features**
   o Related Products section on product detail pages for better user engagement.

5. Add to Cart Functionality

o   Alerts user when a product is added to the cart.
o   Stores product details in local state or localStorage for the Cart page.
o   Displays cart contents dynamically on the Cart page.

6. Checkout Pages:
   o A Checkout page allows users to review their orders and input payment or shipping details.

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

SHOP.CO   Shop   On Sale   New Arrivals   Brands   🔍 Search for products...   🛒 ②   👤

Checkout   Back to Cart

Order Summary

| | Gradient Graphic T-shirt | 145 |
| | Size: L | Qty: 1 |

| | Vertical Striped Shirt | 229 |
| | Size: XL | Qty: 1 |

Subtotal: $314.2
Tax: $2.5
Total:316.7

Checkout

First Name   Last Name

Email Address

Phone Number

Shipping Address

Street Address

City   ZIP Code

Country

Payment Information

Card Number

Expiration Date   CVV   Cardholder Name

☐ I agree to the Terms and Conditions.

Place Order

1 Steps Taken to Build and Integrate Components

1. **Project Setup:** Initialized the project using Next.js to handle routing and server-side rendering.
2. **API Integration:** Set up Sanity CMS as the data source, using the `fetch` API to retrieve product information.
3. **Component Development:** Created modular components like `ProductCard`, `ProductList`, and `SearchBar` to maintain clean and reusable code.
4. **Dynamic Routing:** Implemented dynamic routes using Next.js' file-based routing system, fetching product data based on the URL slug.
5. **UI/UX Implementation:** Applied Tailwind CSS for styling and responsive design to ensure the page is optimized for all devices.
6. **Add to Cart Implementation:** Added cart functionality using React state to allow users to add products to a cart and view them later.

2 Challenges Faced and Solutions Implemented

- **Challenge 1:** Handling dynamic routing with Next.js and ensuring data for each product was fetched and rendered correctly.
    - o **Solution:** Used Next.js' dynamic routing and fetched the product data based on the slug parameter.
- **Challenge 2:** Implementing category filters and pagination while ensuring performance.
    - o **Solution:** Used React state management to handle filter selections and pagination, optimizing API calls to fetch only necessary data.
- **Challenge 3:** Implementing the Add to Cart functionality while maintaining a smooth user experience.
    - o **Solution:** Used React state to manage the cart and ensured that the cart data was persisted across components.

3. Best Practices Followed During Development

- **Modular Components:** Each UI element (product card, list, search bar) was built as a separate reusable component.
- **Efficient Data Fetching:** Used async/await for fetching data and only queried necessary data, reducing load times.

Checklist for Day 4: Dynamic Frontend Components

1. **Frontend Component Development**
   - o   Product Listing Page with dynamic data: ✔ / ✗
   - o   Individual Product Detail Pages with accurate routing: ✔ / ✗
   - o   Functional Category Filters, Search Bar, and Pagination: ✔ / ✗
   - o   Additional Features (e.g., Related Products, User Profile): ✔ / ✗
   - o   Add to Cart functionality integrated: ✔ / ✗
   - o   Checkout Page with Order Summary and Form: ✔ / ✗
2. **Styling and Responsiveness**
   - o   Mobile-friendly product listing and detail pages: ✔ / ✗
   - o   Proper spacing, alignment, and typography: ✔ / ✗
   - o   Components adjust seamlessly across devices: ✔ / ✗
3. **Code Quality**
   - o   Components are reusable and modular: ✔ / ✗
   - o   Followed DRY principles and clean coding practices: ✔ / ✗
   - o   Error handling and state management implemented: ✔ / ✗
4. **Documentation and Submission**
   - o   Technical report completed in a professional format: ✔ / ✗
   - o   Screenshots or screen recordings attached: ✔ / ✗
   - o   Key code snippets included for critical components: ✔ / ✗

Final Review

- All functional and code deliverables meet requirements: ✔ / ✗
- Project tested for usability and bug fixes: ✔ / ✗
- Submission format verified (PDF, Markdown, etc.): ✔ / ✗