# **Day 4 - Dynamic Frontend Components Report**

**Clothing E-Commerce Website**

### **Name:** Ayman Shaheen

### **Roll No:** 00476421

### **Class Day:** Sunday Morning

## **1. Objective**

The goal of this report is to document the frontend development process for the **Clothing E-Commerce Website** using **Next.js**, API integration, and dynamic UI components. The project is based on the **Figma template** and converted into a **fully responsive and scalable Next.js application**.

## **2. Project Resources**

🔗 **GitHub Repository:**  [GitHub Project](https://github.com/Ayman-queen)  
🔗 **Live Website:** [Deployed Version](https://figma-to-nextjs-ecommerce.vercel.app/)  
🔗 **Figma Design:** [Figma Template](https://www.figma.com/design/YUR8kary0dXj39RSTmNolR)

## **3. Key Frontend Components Built**

✔ **Product Listing Component:** Displays products in a grid format with images, names, and prices.  
✔ **Product Detail Component:** Dynamic routing to individual product pages with descriptions and pricing.  
✔ **Search Bar:** Allows users to search products by name.  
✔ **Category Filter:** Enables filtering of products based on categories.  
✔ **Shopping Cart:** Adds/removes items and displays total price.  
✔ **Wishlist Component:** Allows users to save favorite products.  
✔ **Checkout Flow:** Multi-step checkout with billing and shipping details.  
✔ **User Profile:** Displays order history and user details.  
✔ **Reviews & Ratings:** Users can submit and view reviews for products.  
✔ **Pagination & Sorting:** Allows users to navigate product lists efficiently.

## **4. API Integration in Next.js**

✔ API calls were made to fetch product data dynamically from **Sanity CMS**.  
✔ Used getServerSideProps for SSR-based product rendering.  
✔ Implemented API error handling and fallback UI for better user experience.

## **5. Example Code Snippet - Fetching Products**

The following code fetches products from the API and renders them dynamically:

export async function getServerSideProps() {

const res = await fetch('https://api.example.com/products');

const products = await res.json();

return { props: { products } };

}

export default function ProductList({ products }) {

return (

<div className="grid">

{products.map((product) => (

<div key={product.id} className="product-card">

<img src={product.image} alt={product.name} />

<h3>{product.name}</h3>

<p>${product.price}</p>

</div>

))}

</div>

);

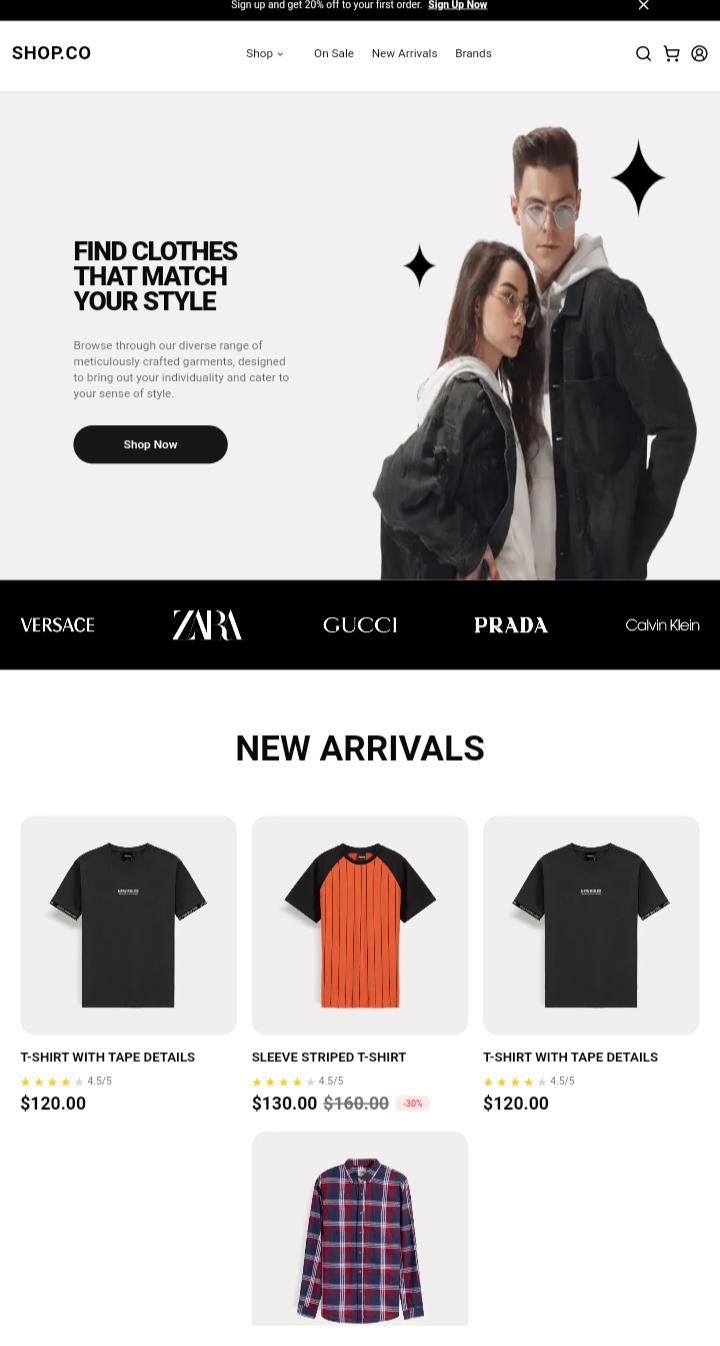
}

## **6. Error Handling & Best Practices**

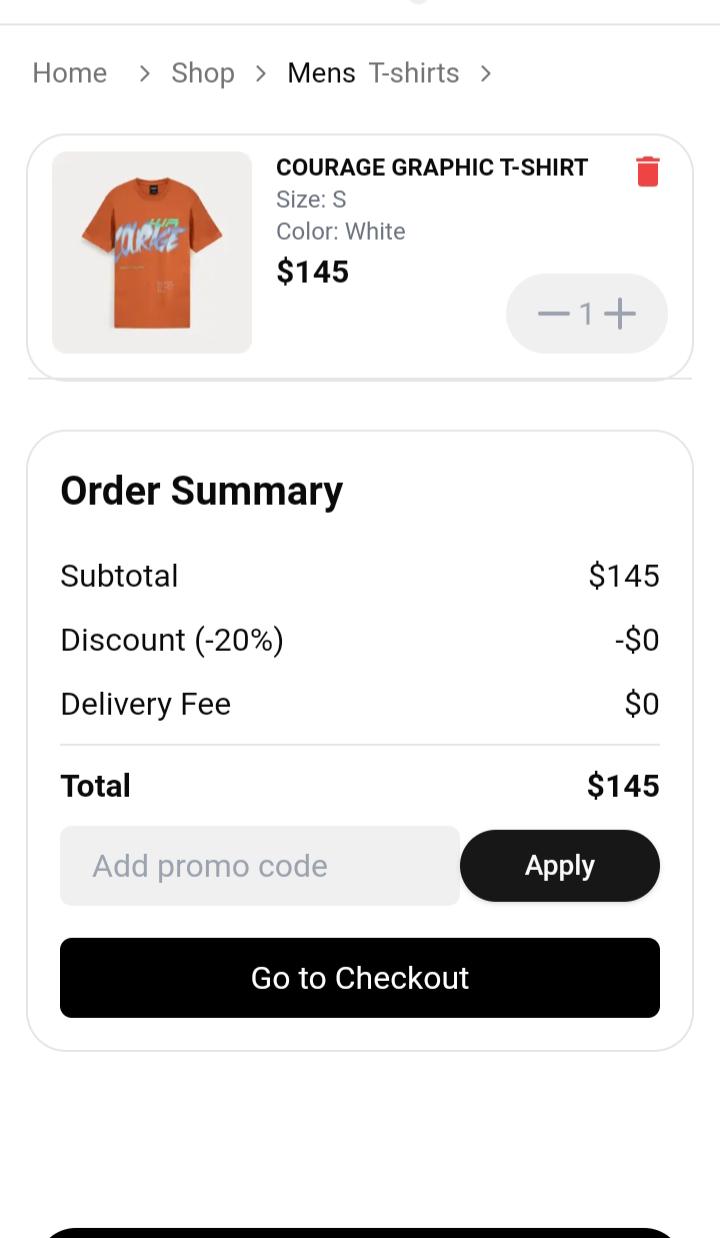
✔ Used try...catch blocks to handle API errors.  
✔ Implemented loading skeletons for better UX.  
✔ Optimized performance with **lazy loading images** and **caching responses**.

## **7. Screenshots of Implementation**

✅ **Product Listing Page - Dynamic Data Rendering**



✅ **Cart & Checkout Functionality**



## **8. Final Submission Checklist**

✔ **API Integration** ✅  
✔ **Dynamic Components Built** ✅  
✔ **Mobile-Responsive UI** ✅  
✔ **Next.js Routing** ✅  
✔ **Error Handling Implemented** ✅  
✔ **Screenshots Attached** ✅

## **9. Conclusion**

This report outlines the development of a **fully functional Clothing E-Commerce Website** using **Next.js, API integration, and dynamic UI components**. The site successfully **fetches products from the API, renders them dynamically, and ensures a smooth shopping experience**.

✔ **API successfully integrated**  
✔ **Products displayed dynamically**  
✔ **Data migrated securely**

### **📌 Submitted by:**

👤 **Ayman Shaheen**  
📅 **Date:** February 2, 2025  
🚀 **Project:** Clothing E-Commerce Website