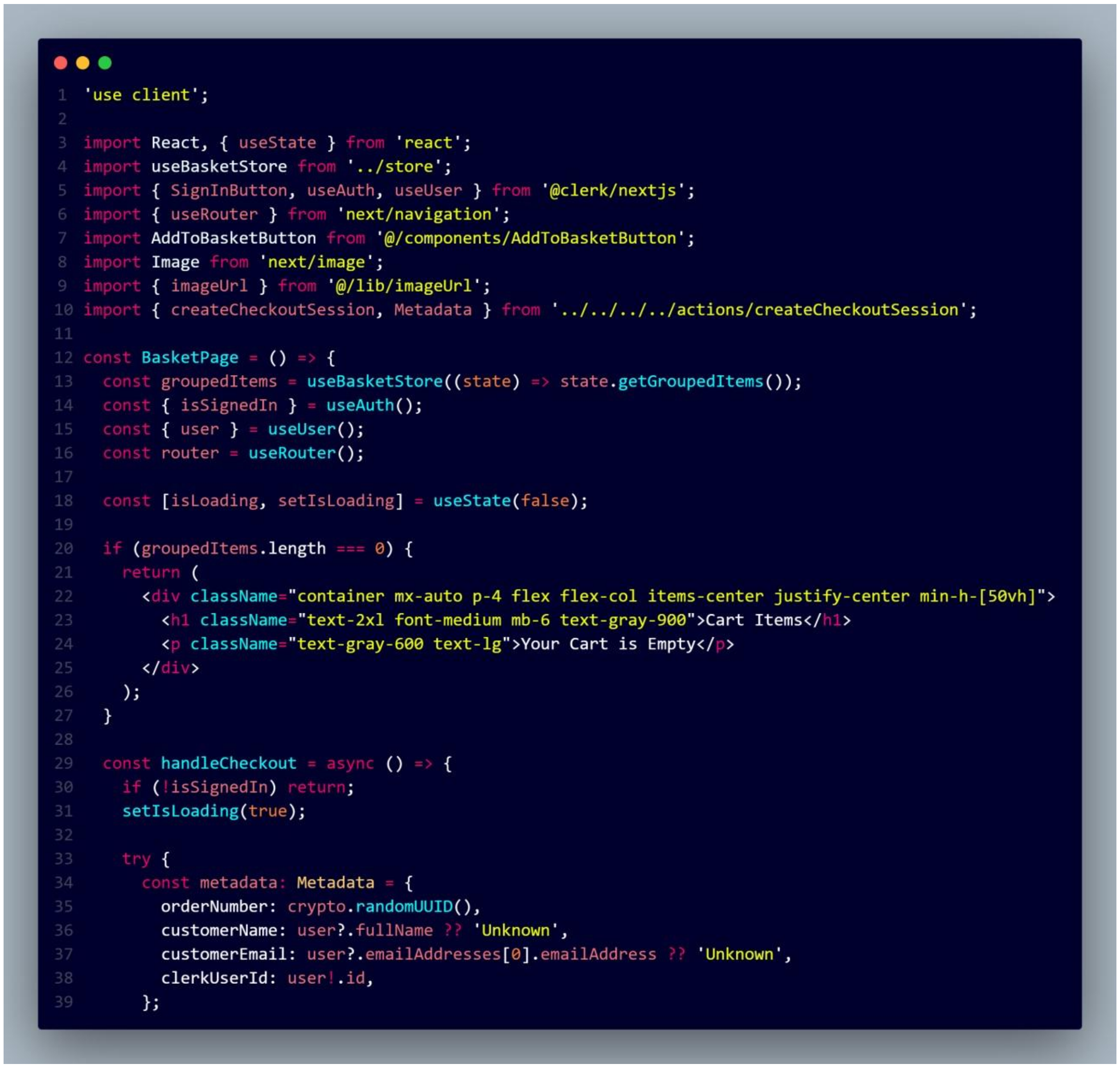
E-commerce Marketplace Website Day04

Reusable Components:

- Designed and implemented reusable components to maintain consistency and improve efficiency across the application.
  - **Header**: A navigation bar that includes the site logo, menu items, and a shopping cart icon. The header component is reusable across different pages of the website, providing easy access to navigation and user account features.
  - **Banner**: A dynamic promotional banner that displays marketing messages, special offers, or seasonal promotions. The content is easily configurable, ensuring flexibility in showcasing relevant information.
  - **Loading**: A loading spinner or animation that appears when content is being fetched from the server or during any asynchronous operations. This improves the user experience by providing feedback while waiting for content to load.
  - **Add to Basket**: A button component that allows users to add items to their shopping cart. This component is reusable in product listings and individual product detail pages, with dynamic text and functionality based on product availability.
  - **Product View**: A detailed view of a single product, which includes product images, descriptions, price, and other dynamic data. It is built to display data from a backend or API, allowing for easy updates and customization.

- **Filtering and Sorting Functionalities:**
  - Implemented advanced filtering and sorting options to enhance user experience while browsing through products.
    - **Filtering**: Users can filter products by categories (e.g., furniture, electronics) and by price ranges (e.g., $0-$50, $50-$100). This functionality ensures users can quickly find products that match their specific preferences.
    - **Sorting**: Users have the ability to sort products by attributes such as price (ascending/descending), rating, and popularity. This gives users control over how products are displayed according to their preferences, improving the overall shopping experience.
  -
- **Add to Cart and Delete Functionalities:**
  - Developed the **Add to Cart** functionality that allows users to add products to their shopping cart. The button updates the cart's visual indicator and stores the product details such as ID, name, price, and quantity in the cart object.
  - Implemented the **Delete** functionality within the cart, enabling users to remove products from their shopping cart. This updates the cart in real-time, ensuring the user can see the changes instantly. The cart's total price and item count are dynamically recalculated when an item is added or removed.

This documentation provides an overview of the key components and features implemented in the project, ensuring modularity, ease of use, and flexibility for future development.



```
'use client';

import React, { useState } from 'react';
import useBasketStore from '../store';
import { SignInButton, useAuth, useUser } from '@clerk/nextjs';
import { useRouter } from 'next/navigation';
import AddToBasketButton from '@/components/AddToBasketButton';
import Image from 'next/image';
import { imageUrl } from '@/lib/imageUrl';
import { createCheckoutSession, Metadata } from '../../../../actions/createCheckoutSession';

const BasketPage = () => {
  const groupedItems = useBasketStore((state) => state.getGroupedItems());
  const { isSignedIn } = useAuth();
  const { user } = useUser();
  const router = useRouter();

  const [isLoading, setIsLoading] = useState(false);

  if (groupedItems.length === 0) {
    return (
      <div className="container mx-auto p-4 flex flex-col items-center justify-center min-h-[50vh]">
        <h1 className="text-2xl font-medium mb-6 text-gray-900">Cart Items</h1>
        <p className="text-gray-600 text-lg">Your Cart is Empty</p>
      </div>
    );
  }

  const handleCheckout = async () => {
    if (!isSignedIn) return;
    setIsLoading(true);

    try {
      const metadata: Metadata = {
        orderNumber: crypto.randomUUID(),
        customerName: user?.fullName ?? 'Unknown',
        customerEmail: user?.emailAddresses[0].emailAddress ?? 'Unknown',
        clerkUserId: user!.id,
      };
```