Project Code Files

File: C:\Projects\skill lab\e-commerce\src\app\globals.css

```
@import "tailwindcss";



:root {
  --background: #ffffff;
  --foreground: #171717;
}

@theme inline {
  --color-background: var(--background);
  --color-foreground: var(--foreground);
  --font-sans: var(--font-geist-sans);
  --font-mono: var(--font-geist-mono);
}

@media (prefers-color-scheme: dark) {
  :root {
    --background: #0a0a0a;
    --foreground: #ededed;
  }
}

body {
  background: var(--background);
  color: var(--foreground);
  font-family: Arial, Helvetica, sans-serif;
}
```

File: C:\Projects\skill lab\e-commerce\src\app\layout.js

```
import { Geist, Geist_Mono } from "next/font/google";
import "./globals.css";
import { CartProvider } from "@/app/Context/CartContext"; // Import CartProvider
import { FavoriteProvider } from "@/app/Context/FavoriteContext"; // Import FavoriteProvider

const geistSans = Geist({
  variable: "--font-geist-sans",
  subsets: ["latin"],
});

const geistMono = Geist_Mono({
  variable: "--font-geist-mono",
  subsets: ["latin"],
});

export const metadata = {
  title: "Create Next App",
  description: "Generated by create next app",
};
```

Project Code Files

```
export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <body className={`${geistSans.variable} ${geistMono.variable} antialiased`}>
        {/* ? Wrap CartProvider and FavoriteProvider properly */}
        <CartProvider>
          <FavoriteProvider>
            {children} {/* Render children only once inside both providers */}
          </FavoriteProvider>
        </CartProvider>
      </body>
    </html>
  );
}
```

File: C:\Projects\skill lab\e-commerce\src\app\page.js

```
import Image from "next/image";

export default function Navbar() {
  return (
    <nav className="flex items-center justify-between px-2 py-4 shadow-md bg-white">
      <div>
        <Image src="/1.png" alt="Flipkart Logo" width={90} height={20} />
      </div>

      <div className="flex-grow mx-8 max-w-xl">
        <div className="flex items-center bg-gray-100 rounded-sm">
          <input
            type="text"
            placeholder="Search for Products, Brands and More"
                         className="w-full  px-4  py-2  bg-transparent  rounded-lg  outline-none  text-gray-700
placeholder-gray-500"
          />
          <button className="px-4 py-2 text-blue-600 hover:bg-gray-200 rounded-r-sm border-l border-red-300">
            Search
          </button>
        </div>
      </div>

      <div className="flex items-center space-x-4">
        <div className="flex space-x-2">
          <button className="px-4 py-1 text-white-200 bg-red-500 rounded-sm">
            Home
          </button>
          <a href="/Login">
            <button className="px-4 py-1 text-white-200 bg-green-500 rounded-sm">
              Login
            </button>
          </a>
          <button className="px-4 py-1 text-white bg-blue-200 bg-blue-500 rounded-sm">
            Sign Up
          </button>
        </div>
      </div>
    </nav>
  );
}
```

File: C:\Projects\skill lab\e-commerce\src\app\api\login\route.js

```
export async function POST(req) {
  try {
    const { email, password } = await req.json();
    if (email === "admin@example.com" && password === "password") {
      return new Response(JSON.stringify({ message: "Login successful" }), {
        status: 200,
        headers: { "Content-Type": "application/json" },
      });
    }
    return new Response(JSON.stringify({ error: "Invalid credentials" }), {
      status: 401,
      headers: { "Content-Type": "application/json" },
    });
  } catch (error) {
    return new Response(JSON.stringify({ error: "Server error" }), {
      status: 500,
      headers: { "Content-Type": "application/json" },
    });
  }
}
```

File: C:\Projects\skill lab\e-commerce\src\app\api\products\route.js

```
export async function GET(request) {
  const products = [
    {
      id: 1,
      title: "Samsung Galaxy S21",
      desc: "Latest Samsung smartphone with 5G support",
      image:
        "https://pisces.bbystatic.com/image2/BestBuy_US/images/products/6441/6441108_sd.jpg",
      price: 69999,
    },
    {
      id: 2,
      title: "iPhone 13 Pro",
      desc: "Apple?s flagship with A15 Bionic chip and ProMotion display",
      image:
        "https://pisces.bbystatic.com/image2/BestBuy_US/images/products/6443/6443338_sd.jpg",
      price: 119999,
    },
    {
      id: 3,
      title: "OnePlus 9 Pro",
      desc: "Flagship OnePlus phone with Hasselblad camera",
      image:
        "https://pisces.bbystatic.com/image2/BestBuy_US/images/products/6457/6457028_sd.jpg",
      price: 64999,
    },
    {
      id: 4,
      title: "MacBook Air M2",
      desc: "Lightweight and powerful MacBook with M2 chip",
      image:
        "https://th.bing.com/th/id/OIP.qe3dE9T2pRXXxnDtghV9yAHaE7?rs=1&pid=ImgDetMain",
      price: 99999,
    },
    {
      id: 5,
      title: "Dell XPS 15",
      desc: "Premium Windows laptop with 4K display",
      image: "https://m.media-amazon.com/images/I/91WgL3IbNIL._AC_SL1500_.jpg",
      price: 149999,
    },
    {
      id: 6,
      title: "Sony WH-1000XM4",
      desc: "Noise-canceling headphones with 30-hour battery life",
      image:
        "https://th.bing.com/th/id/OIP.B4ZpRhAkszS_lq1SDFfA9gHaHa?rs=1&pid=ImgDetMain",
      price: 29999,
    },
    {
      id: 7,
      title: "JBL Flip 5",
      desc: "Portable Bluetooth speaker with deep bass",
      image:
        "https://brain-images-ssl.cdn.dixons.com/1/5/10198351/u_10198351.jpg",
      price: 8999,
    },
    {
      id: 8,
      title: "Samsung 55-inch QLED TV",
      desc: "Smart TV with Quantum Dot technology",
      image:

"https://images.samsung.com/is/image/samsung/p6pim/uk/qe55q75catxxu/gallery/uk-qled-q70c-qe55q75catxxu-53663858
4?$650_519_PNG$",
      price: 79999,
    },
    {
      id: 9,
      title: "Apple iPad Pro",
      desc: "M2-powered iPad with Liquid Retina display",
      image:
        "https://th.bing.com/th/id/OIP.IgpJm2qXRsUiOcj4bLT07wHaHa?rs=1&pid=ImgDetMain",
      price: 109999,
    },
    {
      id: 10,
      title: "Amazon Echo Dot (4th Gen)",
      desc: "Smart speaker with Alexa integration",
      image:
        "https://th.bing.com/th/id/OIP.lWcl6ype4DfLkR8c7-3TigHaG0?rs=1&pid=ImgDetMain",
      price: 4499,
    },
  ];

  return new Response(JSON.stringify(products), {
    status: 200,
    headers: {
      "Content-Type": "application/json",
    },
  });
}
```

File: C:\Projects\skill lab\e-commerce\src\app\cart\page.js

```
"use client";
import { useCart } from "@/app/Context/CartContext";
import { useState, useEffect } from "react";

export default function CartPage() {
  const { cart, removeFromCart, updateQuantity } = useCart();
  const [totalPrice, setTotalPrice] = useState(0);

  useEffect(() => {
    const total = cart.reduce(
      (sum, item) => sum + item.price * (item.quantity || 1),
      0
    );
    setTotalPrice(total.toFixed(2));
  }, [cart]);

  return (
    <div className="container mx-auto py-8 px-4 max-w-6xl">
      <h1 className="text-3xl font-bold mb-6 border-b pb-4 text-black">
        Your Shopping Cart
      </h1>

      {cart.length === 0 ? (
        <div className="flex flex-col items-center justify-center py-12">
          <div className="text-6xl mb-4">?</div>
          <p className="text-xl text-black mb-6">Your cart is empty</p>
          <a
            href="/products"
            className="px-6 py-3 bg-blue-600 text-white rounded-lg hover:bg-blue-700 transition-colors"
          >
            Browse Products
          </a>
        </div>
      ) : (
        <div className="grid grid-cols-1 gap-8">
          <div className="grid grid-cols-1 gap-4">
            {cart.map((item) => (
              <div
                key={item.id}
                className="flex flex-col sm:flex-row border rounded-lg p-4 shadow-md bg-white hover:shadow-lg transition-shadow"
              >
                <div className="sm:w-1/4 mb-4 sm:mb-0 sm:mr-6 flex items-center justify-center">
                  <img
                    src={item.image}
                    alt={item.title}
                    className="max-w-full max-h-48 object-contain rounded-md"
                  />
                </div>

                <div className="flex-grow flex flex-col justify-between">
                  <div>
                    <h2 className="text-xl font-semibold text-black">
                      {item.title}
                    </h2>
                    <p className="text-black mt-2">{item.desc}</p>
                    <p className="font-bold text-lg mt-4 text-black">
                      ${item.price}
                    </p>
                  </div>

                  <div className="flex flex-col sm:flex-row items-start sm:items-center justify-between mt-4">
                    <div className="flex items-center mb-4 sm:mb-0">
                      <span className="mr-4 text-black">Quantity:</span>
                      <div className="flex items-center border rounded">
                        <button
                          className="px-3 py-1 border-r hover:bg-gray-100 text-black"
                          onClick={() =>
                            updateQuantity(
                              item.id,
                              Math.max(1, (item.quantity || 1) - 1)
                            )
                          }
                        >
```

```
                          -
                        </button>
                        <span className="px-4 py-1 text-black">
                          {item.quantity || 1}
                        </span>
                        <button
                          className="px-3 py-1 border-l hover:bg-gray-100 text-black"
                          onClick={() =>
                            updateQuantity(item.id, (item.quantity || 1) + 1)
                          }
                        >
                          +
                        </button>
                      </div>
                    </div>

                    <button
                      onClick={() => removeFromCart(item.id)}
                      className="px-4 py-2 bg-red-500 text-white rounded hover:bg-red-700 transition-colors"
                    >
                      Remove
                    </button>
                  </div>
                </div>
              </div>
            ))}
          </div>

          <div className="mt-6 border-t pt-6">
            <div className="flex justify-between items-center text-xl mb-6">
              <span className="font-semibold text-black">Total:</span>
              <span className="font-bold text-black">? {totalPrice}</span>
            </div>

            <div className="flex flex-col sm:flex-row justify-between gap-4">
              <a
                href="/products"
                        className="px-6 py-3 bg-gray-200 text-black rounded-lg text-center hover:bg-gray-300
transition-colors"
              >
                Continue Shopping
              </a>
                        <button className="px-6 py-3 bg-green-600 text-white rounded-lg hover:bg-green-700
transition-colors">
                Proceed to Checkout
              </button>
            </div>
          </div>
        </div>
      )}
    </div>
```

```
  );
}
```

File: C:\Projects\skill lab\e-commerce\src\app\Component\Navbar.js

```
"use client";

import Link from "next/link";
import Image from "next/image";
import { useCart } from "@/app/Context/CartContext";
import { useFavorites } from "@/app/Context/FavoriteContext";

export default function Navbar() {
  const { cart } = useCart();
  const { favorites } = useFavorites();

  return (
    <nav className="bg-gray p-3 text-white ">
      <div className="container mx-auto flex flex-col sm:flex-row justify-between items-center">
        <div className="flex items-center gap-2 mb-2 sm:mb-0">
          <Image
            src="/1.png"
            alt="Logo"
            width={80}
            height={80}

          />
        </div>

        <div className="flex items-center gap-6">
          <Link href="/favorites" className="flex items-center gap-1">
            <span>Favorites</span>
            <span className="bg-white text-black rounded-full px-2 py-0.5 text-xs">
              {favorites.length}
            </span>
          </Link>

          <Link href="/cart" className="flex items-center gap-1">
            <span>Cart</span>
            <span className="bg-white text-black rounded-full px-2 py-0.5 text-xs">
              {cart.length}
            </span>
          </Link>
        </div>
      </div>
    </nav>
  );
}
```

File: C:\Projects\skill lab\e-commerce\src\app\Component\ProductCart.js

Project Code Files

```
"use client";

import { useCart } from "@/app/Context/CartContext";
import { useFavorites } from "@/app/Context/FavoriteContext";

export default function ProductCard({ product, showRemoveButton = false }) {
  const { cart, addToCart } = useCart();
  const { favorites, toggleFavorite, removeFavorite } = useFavorites();

  const isFavorite = favorites.some((item) => item.id === product.id);
  const isInCart = cart.some((item) => item.id === product.id);

  return (
    <div className="p-3 border border-gray-200 rounded-lg shadow-sm bg-white hover:shadow-md transition-shadow">
      <div className="relative overflow-hidden rounded-lg mb-2">
        <div className="aspect-w-1 aspect-h-1">
          <img
            src={product.image}
            alt={product.name}
            className="w-full h-40 object-contain"
          />
        </div>
      </div>

      <div className="space-y-1 text-center">
        <h2 className="text-sm font-semibold text-black truncate">
          {product.name}
        </h2>
        <p className="text-gray-800 font-medium">?{product.price}</p>
        <div className="pt-2 flex space-x-1">
          <button
            onClick={() => addToCart(product)}
            className={`flex-1 px-2 py-1 text-sm rounded-md text-white transition-colors ${
              isInCart
                ? "bg-black text-white"
                : "bg-black hover:bg-gray-800 text-white"
            }`}
          >
            {isInCart ? "? Added" : "Add To Cart"}
          </button>

          {showRemoveButton ? (
            <button
              onClick={() => removeFavorite(product.id)}
              className="flex-1 px-2 py-1 text-sm bg-white text-black border border-black rounded-md hover:bg-gray-100 transition-colors"
            >
              Remove
            </button>
          ) : (
            <button
              onClick={() => toggleFavorite(product)}
              className={`flex-1 px-2 py-1 text-sm rounded-md transition-colors ${
                isFavorite
                  ? "bg-white text-black border border-black"
                  : "bg-gray-200 text-black hover:bg-gray-300"
              }`}
            >
              {isFavorite ? "Favorited" : "?"}
            </button>
          )}
        </div>
      </div>
    </div>
  );
}
```

File: C:\Projects\skill lab\e-commerce\src\app\Context\CartContext.js

```
"use client";
import { createContext, useContext, useState, useEffect } from "react";

// Create Context
const CartContext = createContext();

export function CartProvider({ children }) {
  const [cart, setCart] = useState([]);

  // Load cart from localStorage when component mounts
  useEffect(() => {
    const storedCart = JSON.parse(localStorage.getItem("cart")) || [];
    setCart(storedCart);
  }, []);

  // Save cart to localStorage on changes
  useEffect(() => {
    localStorage.setItem("cart", JSON.stringify(cart));
  }, [cart]);

  // Add item to cart
  const addToCart = (product) => {
    setCart((prevCart) => [...prevCart, product]);
  };

  // Remove item from cart
  const removeFromCart = (id) => {
    setCart((prevCart) => prevCart.filter((item) => item.id !== id));
  };

  return (
    <CartContext.Provider value={{ cart, addToCart, removeFromCart }}>
```

```
      {children}
    </CartContext.Provider>
  );
}

// Hook to use cart
export function useCart() {
  return useContext(CartContext);
}
```

File: C:\Projects\skill lab\e-commerce\src\app\Context\FavoriteContext.js

```
"use client"; // Ensure this runs on the client side

import { createContext, useContext, useState, useEffect } from "react";

const FavoriteContext = createContext();

export function FavoriteProvider({ children }) {
  const [favorites, setFavorites] = useState([]);

  // Load favorites from localStorage when component mounts
  useEffect(() => {
    const storedFavorites = JSON.parse(localStorage.getItem("favorites")) || [];
    setFavorites(storedFavorites);
  }, []);

  // Save favorites to localStorage when it changes
  useEffect(() => {
    localStorage.setItem("favorites", JSON.stringify(favorites));
  }, [favorites]);

  // Function to add/remove favorite
  const toggleFavorite = (product) => {
    setFavorites((prevFavorites) => {
      if (prevFavorites.find((item) => item.id === product.id)) {
        return prevFavorites.filter((item) => item.id !== product.id);
      } else {
        return [...prevFavorites, product];
      }
    });
  };

  return (
    <FavoriteContext.Provider value={{ favorites, toggleFavorite }}>
      {children}
    </FavoriteContext.Provider>
  );
}

// Hook to use FavoriteContext
```

Project Code Files

```
export function useFavorites() {
  const context = useContext(FavoriteContext);
  if (!context) {
    throw new Error("useFavorites must be used within a FavoriteProvider");
  }
  return context;
}
```

File: C:\Projects\skill lab\e-commerce\src\app\Dashboard\page.js

```
"use client";
import { useState, useEffect } from "react";
import Navbar from "../../Component/Navbar";

export default function Dashboard() {
  const [products, setProducts] = useState([]);

  useEffect(() => {
    fetch("/api/products")
      .then((res) => res.json())
      .then((data) => setProducts(data));
  }, []);

  return (
    <div>
      <Navbar />
      <h1 className="text-2xl font-bold text-center mt-4">Dashboard</h1>
      <div className="container mx-auto p-4">
        {products.length > 0 ? (
          <ul>
            {products.map((product) => (
              <li key={product.id} className="border p-4 my-2 rounded-lg">
                {product.title} - ${product.price}
              </li>
            ))}
          </ul>
        ) : (
          <p>No products available</p>
        )}
      </div>
    </div>
  );
}
```

File: C:\Projects\skill lab\e-commerce\src\app\favorites\page.js

```
"use client"; // Ensure this runs on the client side

import { useFavorites } from "@/app/Context/FavoriteContext";
import { useCart } from "@/app/Context/CartContext";
```

Project Code Files

```
export default function FavoritesPage() {
  const { favorites, removeFromFavorites } = useFavorites();
  const { addToCart } = useCart();

  const handleAddToCart = (product) => {
    addToCart(product);
  };

  return (
    <div className="container mx-auto py-8 px-4 max-w-6xl">
      <h1 className="text-3xl font-bold mb-6 border-b pb-4 text-white">
        Favourite
      </h1>

      {favorites.length === 0 ? (
        <div className="flex flex-col items-center justify-center py-12">
          <div className="text-6xl mb-4">??</div>
          <p className="text-xl text-black mb-6">No favourite items yet.</p>
          <a
            href="/products"
            className="px-6 py-3 bg-blue-600 text-white rounded-lg hover:bg-blue-700 transition-colors"
          >
            Browse Products
          </a>
        </div>
      ) : (
        <div className="grid grid-cols-1 gap-4">
          {favorites.map((item) => (
            <div
              key={item.id}
              className="flex flex-col sm:flex-row border rounded-lg p-4 shadow-md bg-white hover:shadow-lg
transition-shadow"
            >
              <div className="sm:w-1/4 mb-4 sm:mb-0 sm:mr-6 flex items-center justify-center">
                <img
                  src={item.image}
                  alt={item.title}
                  className="max-w-full max-h-48 object-contain rounded-md"
                />
              </div>

              <div className="flex-grow flex flex-col justify-between">
                <div>
                  <h2 className="text-xl font-semibold text-black">
                    {item.title}
                  </h2>
                  <p className="text-black mt-2">{item.desc}</p>
                  <p className="font-bold text-lg mt-4 text-black">
                    ${item.price}
                  </p>
                </div>
                <div className="flex flex-col sm:flex-row items-start sm:items-center justify-between mt-4">
                  <div className="flex items-center gap-4 mb-4 sm:mb-0">
                    <button
                      onClick={() => handleAddToCart(item)}
                      className="px-4 py-2 bg-blue-600 text-white rounded hover:bg-blue-700 transition-colors"
                    >
                      Add to Cart
                    </button>
                  </div>

                  <button
                    onClick={() => removeFromFavorites(item.id)}
                    className="px-4 py-2 bg-red-500 text-white rounded hover:bg-red-700 transition-colors"
                  >
                    Remove
                  </button>
                </div>
              </div>
            </div>
          ))}
        </div>
      )}
    </div>
  );
}
```

File: C:\Projects\skill lab\e-commerce\src\app\Login\page.js

```
"use client";
import { useState } from "react";
import { useRouter } from "next/navigation";
import Image from "next/image";

export default function Login() {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const router = useRouter();

  const handleSubmit = (event) => {
    event.preventDefault();
    if (email === "admin@example.com" && password === "password") {
      router.push("/Product");
    } else {
      alert("Invalid email or password");
    }
  };

  return (
    <div className="flex items-center justify-center h-screen bg-gray-100">
      <div className="bg-white p-6 shadow-md w-80 flex flex-col items-center">
        <Image
          src="/1.png"
          alt="Logo"
          width={80}
          height={30}
          className="mb-4"
        />
        <h2 className="text-xl font-bold mb-4 text-center text-gray-700">
          Login
        </h2>
        <form className="w-full" onSubmit={handleSubmit}>
          <div className="mb-3">
            <label htmlFor="email" className="block text-sm text-gray-700">
              Email:
            </label>
            <input
              type="email"
              id="email"
              className="w-full px-3 py-2 border border-black rounded"
              placeholder="Enter email"
              required
              value={email}
              onChange={(e) => setEmail(e.target.value)}
            />
          </div>

          <div className="mb-3">
            <label htmlFor="password" className="block text-sm text-gray-700">
              Password:
            </label>
            <input
              type="password"
              id="password"
              className="w-full px-3 py-2 border border-black rounded"
              placeholder="Enter password"
              required
              value={password}
              onChange={(e) => setPassword(e.target.value)}
            />
          </div>

          <button
            type="submit"
            className="w-full bg-blue-500 mt-4 text-white py-2 rounded hover:bg-blue-600"
          >
            Login
          </button>
        </form>
      </div>
    </div>
  );
```

```
}
```

File: C:\Projects\skill lab\e-commerce\src\app\Product\page.js

```
import Navbar from "@/app/Component/Navbar";
import ProductCard from "@/app/Component/ProductCart";

async function fetchProducts() {
  const response = await fetch("http://localhost:3000/api/products", {
    cache: "no-store",
  });

  if (!response.ok) {
    throw new Error("Failed to fetch products");
  }

  return response.json();
}

export default async function Home() {
  let products = [];

  try {
    products = await fetchProducts();
  } catch (error) {
    console.error("Error fetching products:", error);
  }

  return (
    <div>
      <Navbar />
      <div className="container mx-auto p-4 grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
        {products.length > 0 ? (
          products.map((product) => (
            <ProductCard key={product.id} product={product} />
          ))
        ) : (
          <p className="text-center text-gray-500">No products available.</p>
        )}
      </div>
    </div>
  );
}
```

File: C:\Projects\skill lab\e-commerce\src\app\styles\globals.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;

body {
```

Project Code Files

```
  font-family: 'Inter', sans-serif;
  background-color: #f9fafb;
}
```