EXPERIMENT 3

Aim: Manage complex state with context API

Tools & Technologies:

- **Frontend Framework:** React.js
- **State Management:** Context API, React Hooks (`useContext`, `useReducer`)
- **Styling:** Tailwind CSS
- **Backend/Mock Data:** Sample API for fetching clothing data
- **Other Tools:** VS Code, Node.js, npm

Prerequisites:

- Knowledge of JavaScript and ES6+ features
- Basic understanding of React components and props
- Familiarity with React Hooks (`useState`, `useEffect`)
- Understanding of REST API integration
- Basic knowledge of Tailwind CSS for UI design

Theory:

In React, **state management** becomes complex when multiple components need to access and update the same data. Passing props down multiple levels leads to “prop drilling,” which makes the code difficult to maintain. To solve this, we use **Context API** (or Redux) for centralized state management. The Context API allows data (state) to be shared globally without explicitly passing props at every level. This improves maintainability, scalability, and performance in larger applications.

Application in Project

In this project, **Context API** has been used to manage **complex state** such as cart items, wishlist items, and checkout details. Instead of passing props across multiple components, a global context was created that allowed any component (Cart, Wishlist, Checkout) to directly access and update the shared state. For example, when an item is added to the cart, the global state updates instantly, and the changes reflect across all related components. This ensured smooth synchronization of data and avoided repetitive prop drilling.

Additional Work Beyond the Aim (Extra 30%)

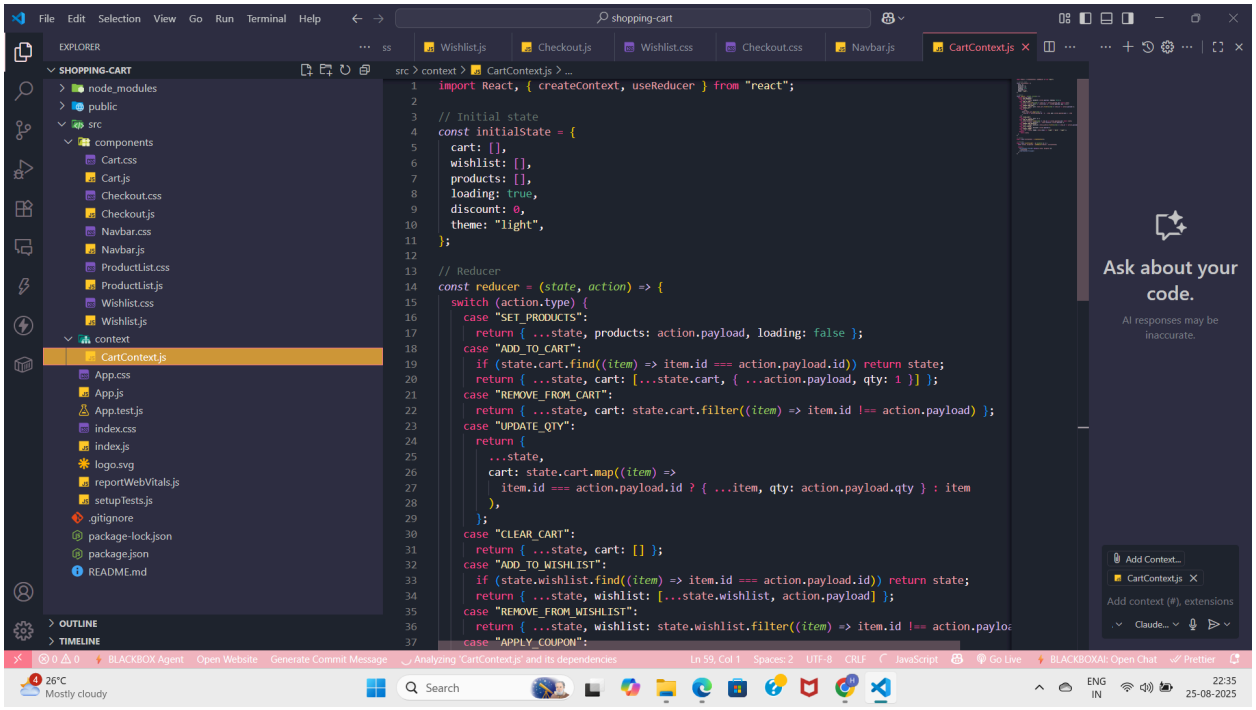
- **API Integration:** Fetched sample clothing/product data dynamically from an API instead of using static JSON.

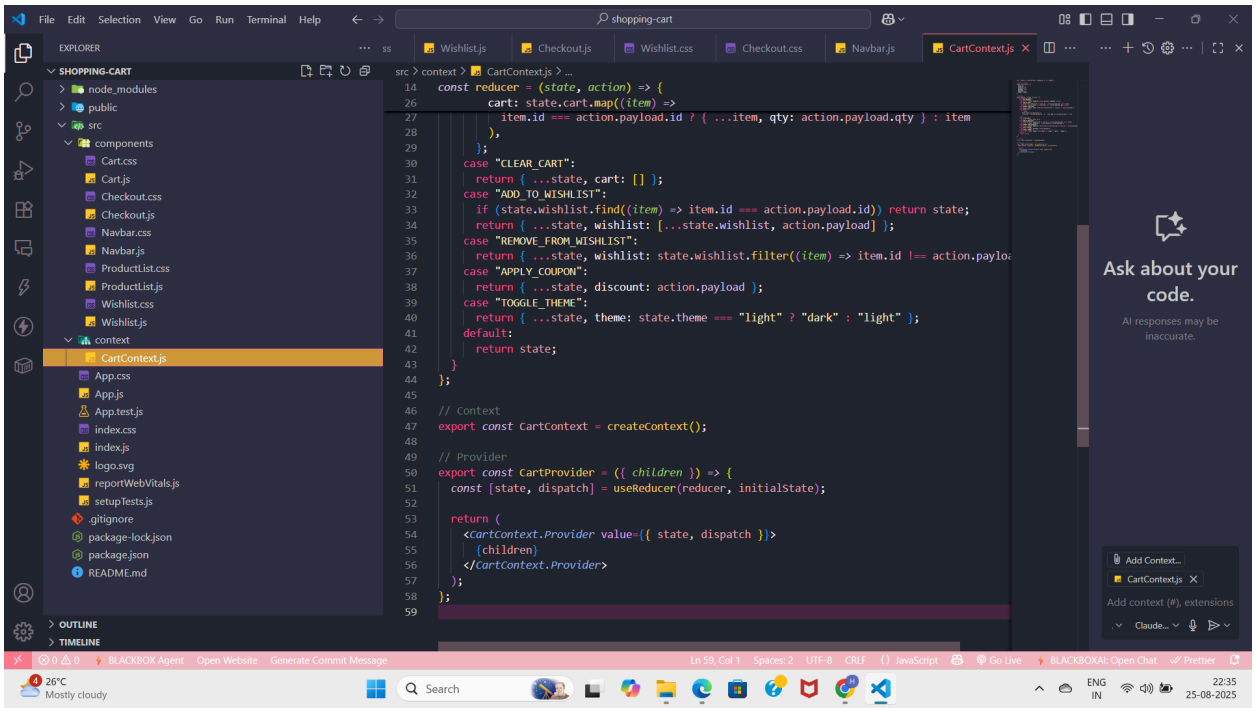
- **Responsive UI with Tailwind CSS:** Designed a clean, modern, and fully responsive user interface.

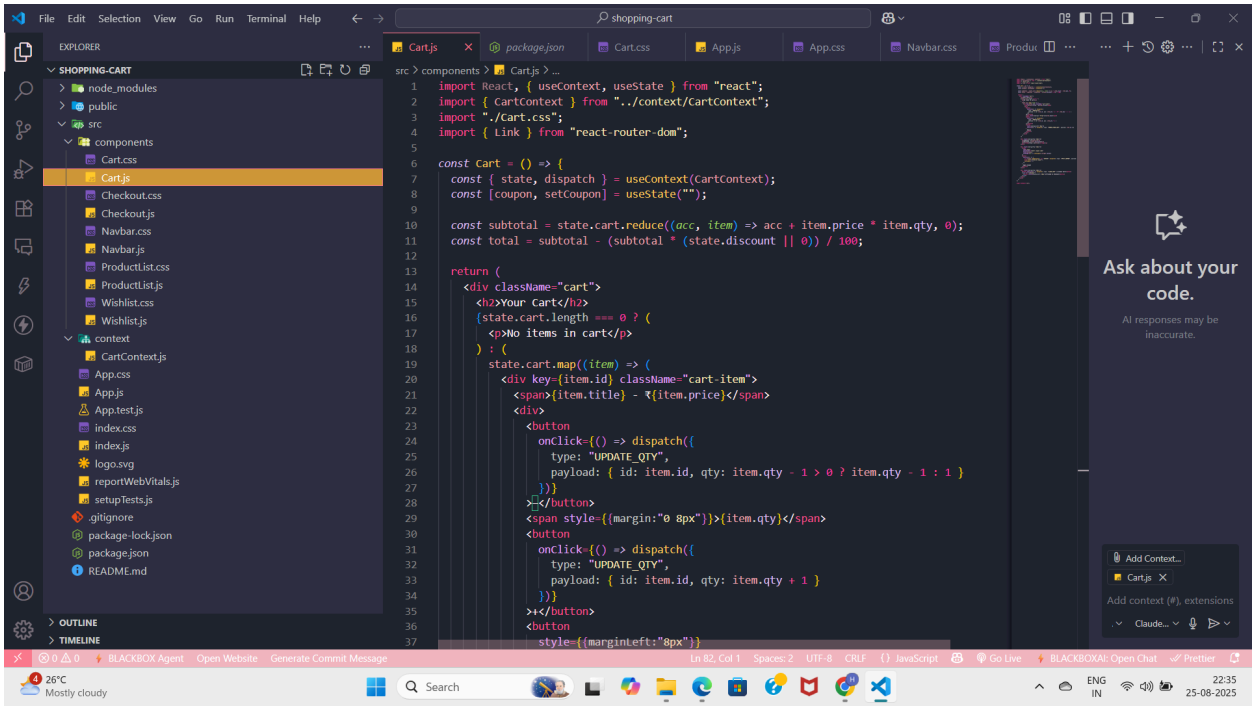
- **Routing:** Used React Router for navigation between pages like Home, Cart, Wishlist, and Checkout.

- **Enhanced User Experience:** Styled and improved UI buttons, added interactivity, and ensured better usability.

Code:


File Edit Selection View Go Run Terminal Help
shopping-cart
EXPLORER
SHOPPING-CART
node_modules
public
src
components
Cart.css
Cart.js
Checkout.css
Checkout.js
Navbar.css
Navbar.js
ProductList.css
ProductList.js
Wishlist.css
Wishlist.js
context
CartContext.js
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
Wishlist.js
Checkout.js
Wishlist.css
Checkout.css
Navbar.js
CartContext.js
src > context > CartContext.js > ...
import React, { createContext, useReducer } from "react";
// Initial state
const initialState = {
cart: [],
wishlist: [],
products: [],
loading: true,
discount: 0,
theme: "light",
};
// Reducer
const reducer = (state, action) => {
switch (action.type) {
case "SET_PRODUCTS":
return { ...state, products: action.payload, loading: false };
case "ADD_TO_CART":
if (state.cart.find((item) => item.id === action.payload.id)) return state;
return { ...state, cart: [...state.cart, { ...action.payload, qty: 1 }] };
case "REMOVE_FROM_CART":
return { ...state, cart: state.cart.filter((item) => item.id !== action.payload) };
case "UPDATE_QTY":
return {
...state,
cart: state.cart.map((item) =>
item.id === action.payload.id ? { ...item, qty: action.payload.qty } : item
),
};
case "CLEAR_CART":
return { ...state, cart: [] };
case "ADD_TO_WISHLIST":
if (state.wishlist.find((item) => item.id === action.payload.id)) return state;
return { ...state, wishlist: [...state.wishlist, action.payload] };
case "REMOVE_FROM_WISHLIST":
return { ...state, wishlist: state.wishlist.filter((item) => item.id !== action.payloa
case "APPLY_COUPON":
Ask about your code.
AI responses may be inaccurate.
Add Context...
CartContext.js
Add context (#), extensions
Claude...
Ln 59, Col 1 Spaces: 2 UTF-8 CRLF JavaScript
26°C
Mostly cloudy
Search
ENG
IN
22:35
25-08-2025

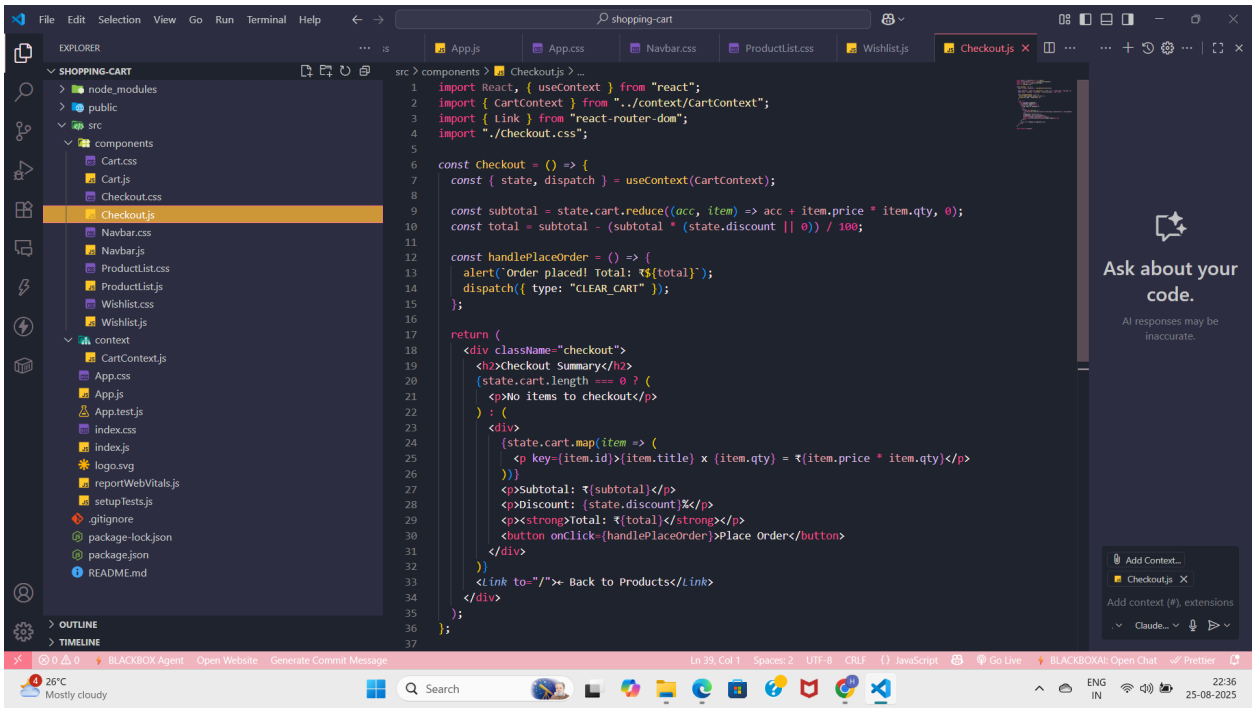

File Edit Selection View Go Run Terminal Help
shopping-cart
EXPLORER
SHOPPING-CART
node_modules
public
src
components
Cart.css
Cart.js
Checkout.css
Checkout.js
Navbar.css
Navbar.js
ProductList.css
ProductList.js
Wishlist.css
Wishlist.js
context
CartContext.js
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
Wishlist.js
Checkout.js
Wishlist.css
Checkout.css
Navbar.js
CartContext.js
src > context > CartContext.js > ...
const reducer = (state, action) => {
cart: state.cart.map((item) =>
item.id === action.payload.id ? { ...item, qty: action.payload.qty } : item
),
};
case "CLEAR_CART":
return { ...state, cart: [] };
case "ADD_TO_WISHLIST":
if (state.wishlist.find((item) => item.id === action.payload.id)) return state;
return { ...state, wishlist: [...state.wishlist, action.payload] };
case "REMOVE_FROM_WISHLIST":
return { ...state, wishlist: state.wishlist.filter((item) => item.id !== action.payloa
case "APPLY_COUPON":
return { ...state, discount: action.payload };
case "TOGGLE_THEME":
return { ...state, theme: state.theme === "light" ? "dark" : "light" };
default:
return state;
}
};
// Context
export const CartContext = createContext();
// Provider
export const CartProvider = ({ children }) => {
const [state, dispatch] = useReducer(reducer, initialState);
return (
<CartContext.Provider value={{ state, dispatch }}>
{children}
</CartContext.Provider>
);
};
Ask about your code.
AI responses may be inaccurate.
Add Context...
CartContext.js
Add context (#), extensions
Claude...
Ln 59, Col 1 Spaces: 2 UTF-8 CRLF JavaScript
26°C
Mostly cloudy
Search
ENG
IN
22:35
25-08-2025


File Edit Selection View Go Run Terminal Help
shopping-cart
EXPLORER
SHOPPING-CART
node_modules
public
src
components
Cart.css
Cart.js
Checkout.css
Checkout.js
Navbar.css
Navbar.js
ProductList.css
ProductList.js
Wishlist.css
Wishlist.js
context
CartContext.js
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
Cart.js
package.json
Cart.css
App.js
App.css
Navbar.css
src > components > Cart.js > ...
import React, { useContext, useState } from "react";
import { CartContext } from "../context/CartContext";
import "./Cart.css";
import { Link } from "react-router-dom";
const Cart = () => {
const { state, dispatch } = useContext(CartContext);
const [coupon, setCoupon] = useState("");
const subtotal = state.cart.reduce((acc, item) => acc + item.price * item.qty, 0);
const total = subtotal - (subtotal * (state.discount || 0)) / 100;
return (

<h2>Your Cart</h2>
{state.cart.length === 0 ? (
No items in cart
) : (
state.cart.map((item) => (

{item.title} - ₹{item.price}

<button
onClick={() => dispatch({
type: "UPDATE_QTY",
payload: { id: item.id, qty: item.qty - 1 > 0 ? item.qty - 1 : 1 }
})}
></button>
<span style={{margin:"0 8px"}}>{item.qty}
<button
onClick={() => dispatch({
type: "UPDATE_QTY",
payload: { id: item.id, qty: item.qty + 1 }
})}
>+</button>
<button
style={{marginLeft:"8px"}}
Ask about your code.
AI responses may be inaccurate.
Add Context...
Cart.js
Add context (#), extensions
Claude...
Ln 82, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live BLACKBOXAI: Open Chat Prettier
26°C
Mostly cloudy
Search
ENG
IN
22:35
25-08-2025

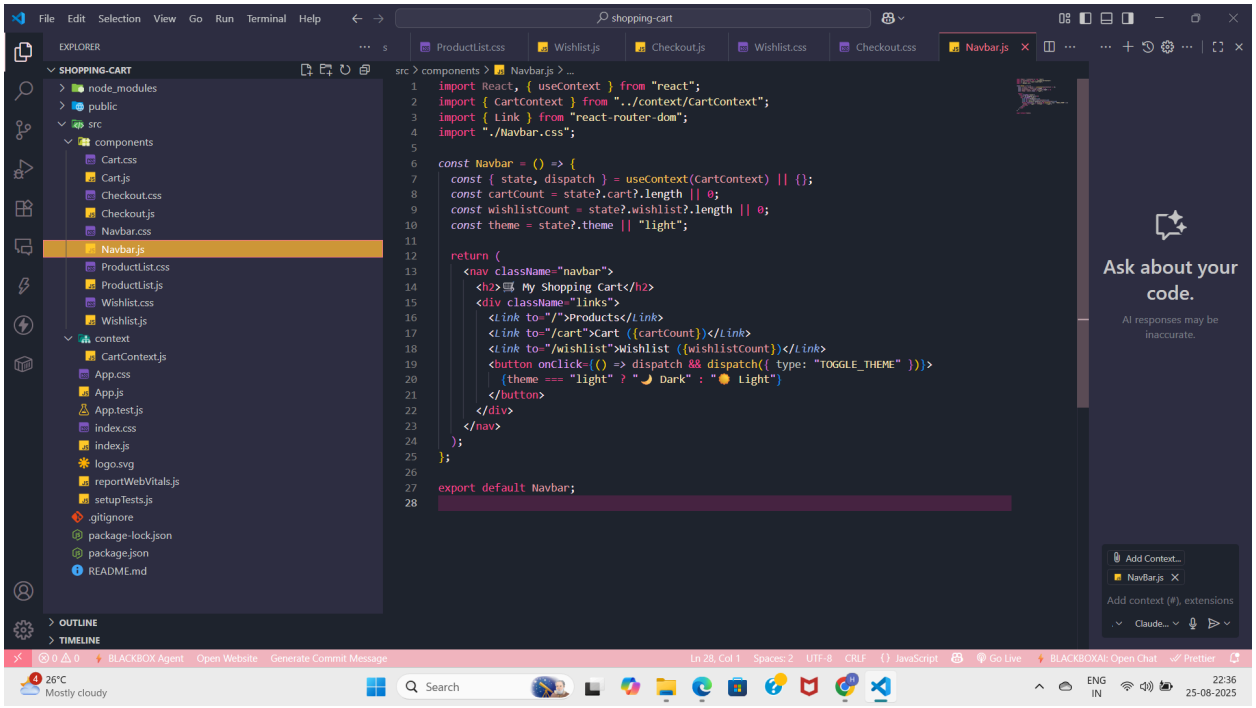

File Edit Selection View Go Run Terminal Help
shopping-cart
EXPLORER
SHOPPING-CART
node_modules
public
src
components
Cart.css
Cart.js
Checkout.css
Checkout.js
Navbar.css
Navbar.js
ProductList.css
ProductList.js
Wishlist.css
Wishlist.js
context
CartContext.js
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
App.js
App.css
Navbar.css
ProductList.css
Wishlist.js
Checkout.js
src > components > Checkout.js > ...
import React, { useContext } from "react";
import { CartContext } from "../context/CartContext";
import { Link } from "react-router-dom";
import "./Checkout.css";
const Checkout = () => {
const { state, dispatch } = useContext(CartContext);
const subtotal = state.cart.reduce((acc, item) => acc + item.price * item.qty, 0);
const total = subtotal - (subtotal * (state.discount || 0)) / 100;
const handlePlaceOrder = () => {
alert(`Order placed! Total: ₹${total}`);
dispatch({ type: "CLEAR_CART" });
};
return (

<h2>Checkout Summary</h2>
{state.cart.length === 0 ? (
No items to checkout
) : (

{state.cart.map(item => (
{item.title} x {item.qty} = ₹{item.price * item.qty}
))}
Subtotal: ₹{subtotal}
Discount: {state.discount}%
Total: ₹{total}
<button onClick={handlePlaceOrder}>Place Order</button>

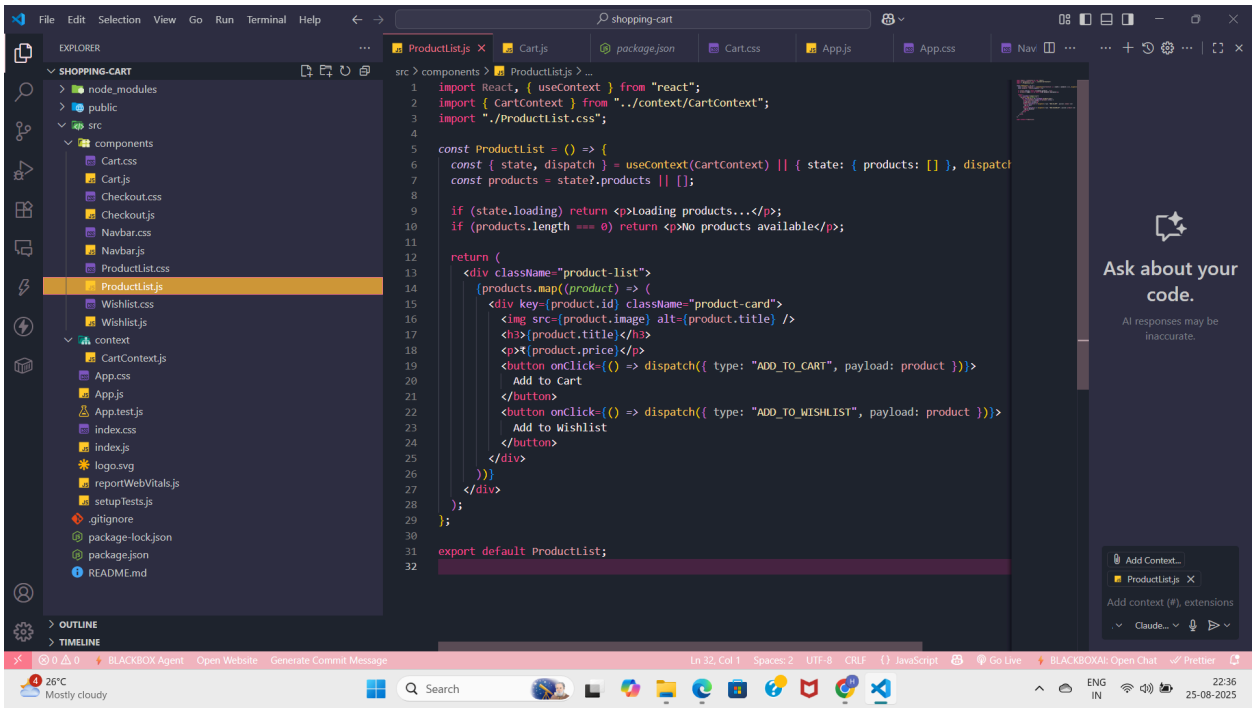
)}
<Link to="/">← Back to Products</Link>

);
};
Ask about your code.
AI responses may be inaccurate.
Add Context...
Checkout.js
Add context (#), extensions
Claude...
Ln 39, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live BLACKBOXAI: Open Chat Prettier
26°C
Mostly cloudy
Search
ENG
IN
22:36
25-08-2025


File Edit Selection View Go Run Terminal Help
shopping-cart
EXPLORER
SHOPPING-CART
node_modules
public
src
components
Cart.css
Cart.js
Checkout.css
Checkout.js
Navbar.css
Navbar.js
ProductList.css
ProductList.js
Wishlist.css
Wishlist.js
context
CartContext.js
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
ProductList.css
Wishlist.js
Checkout.js
Wishlist.css
Checkout.css
Navbar.js
src > components > Navbar.js > ...
import React, { useContext } from "react";
import { CartContext } from "../context/CartContext";
import { Link } from "react-router-dom";
import "./Navbar.css";
const Navbar = () => {
const { state, dispatch } = useContext(CartContext) || {};
const cartCount = state?.cart?.length || 0;
const wishlistCount = state?.wishlist?.length || 0;
const theme = state?.theme || "light";
return (
<nav className="navbar">
<h2>🛒 My Shopping Cart</h2>

<Link to="/">Products</Link>
<Link to="/cart">Cart ({cartCount})</Link>
<Link to="/wishlist">Wishlist ({wishlistCount})</Link>
<button onClick={() => dispatch && dispatch({ type: "TOGGLE_THEME" })}>
{theme === "light" ? "🌙 Dark" : "🌞 Light"}
</button>

</nav>
);
};
export default Navbar;
Ask about your code.
AI responses may be inaccurate.
Add Context...
Navbar.js
Add context (#), extensions
Claude...
BLACKBOX Agent
Open Website
Generate Commit Message
Ln 28, Col 1
Spaces: 2
UTF-8
CRLF
JavaScript
Go Live
BLACKBOXAI: Open Chat
Prettier
26°C
Mostly cloudy
Search
ENG
IN
22:36
25-08-2025

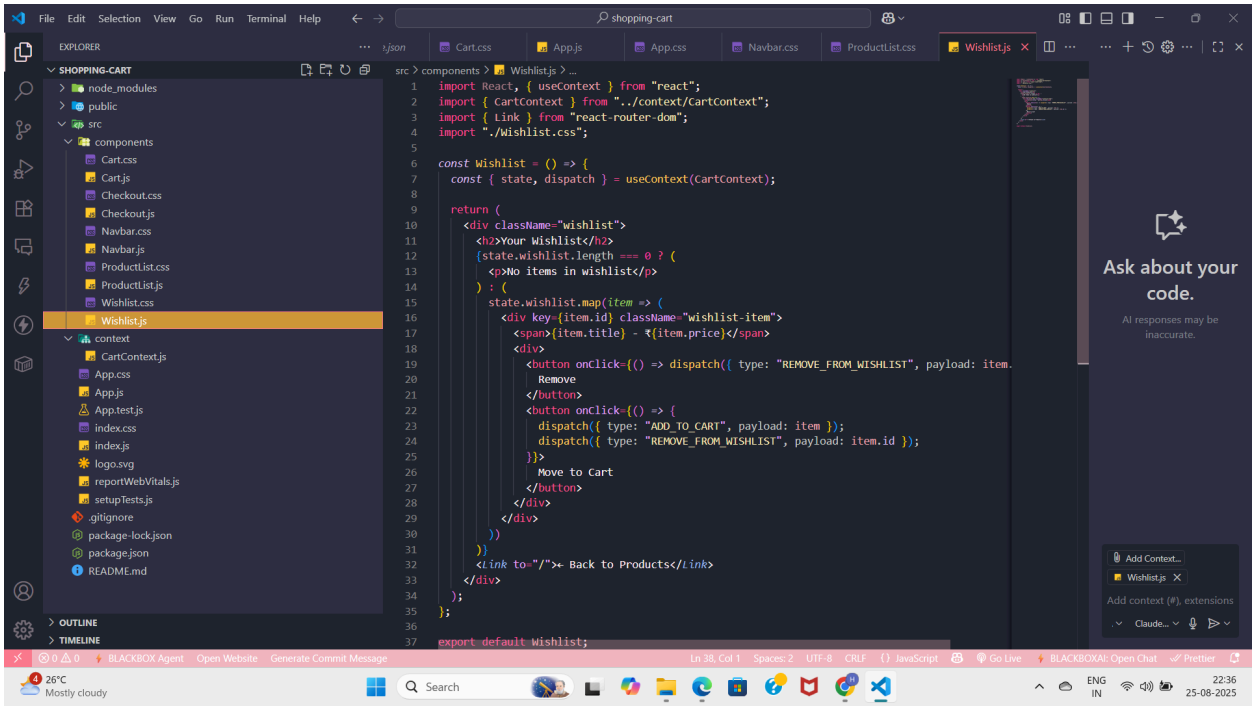

File Edit Selection View Go Run Terminal Help
shopping-cart
EXPLORER
SHOPPING-CART
node_modules
public
src
components
Cart.css
Cart.js
Checkout.css
Checkout.js
Navbar.css
Navbar.js
ProductList.css
ProductList.js
Wishlist.css
Wishlist.js
context
CartContext.js
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
ProductList.js
Cart.js
package.json
Cart.css
App.js
App.css
src > components > ProductList.js > ...
import React, { useContext } from "react";
import { CartContext } from "../context/CartContext";
import "./ProductList.css";
const ProductList = () => {
const { state, dispatch } = useContext(CartContext) || { state: { products: [] }, dispatch
const products = state?.products || [];
if (state.loading) return Loading products... ;
if (products.length === 0) return No products available ;
return (

{products.map((product) => (

<h3>{product.title}</h3>
₹{product.price}
<button onClick={() => dispatch({ type: "ADD_TO_CART", payload: product })}>
Add to Cart
</button>
<button onClick={() => dispatch({ type: "ADD_TO_WISHLIST", payload: product })}>
Add to Wishlist
</button>

))}

);
};
export default ProductList;
Ask about your code.
AI responses may be inaccurate.
Add Context...
ProductList.js
Add context (#), extensions
Claude...
BLACKBOX Agent
Open Website
Generate Commit Message
Ln 32, Col 1
Spaces: 2
UTF-8
CRLF
JavaScript
Go Live
BLACKBOXAI: Open Chat
Prettier
26°C
Mostly cloudy
Search
ENG
IN
22:36
25-08-2025


File Edit Selection View Go Run Terminal Help
shopping-cart
EXPLORER
SHOPPING-CART
node_modules
public
src
components
Cart.css
Cart.js
Checkout.css
Checkout.js
Navbar.css
Navbar.js
ProductList.css
ProductList.js
Wishlist.css
Wishlist.js
context
CartContext.js
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
src > components > Wishlist.js > ...
import React, { useContext } from "react";
import { CartContext } from "../context/CartContext";
import { Link } from "react-router-dom";
import "./Wishlist.css";
const Wishlist = () => {
const { state, dispatch } = useContext(CartContext);
return (

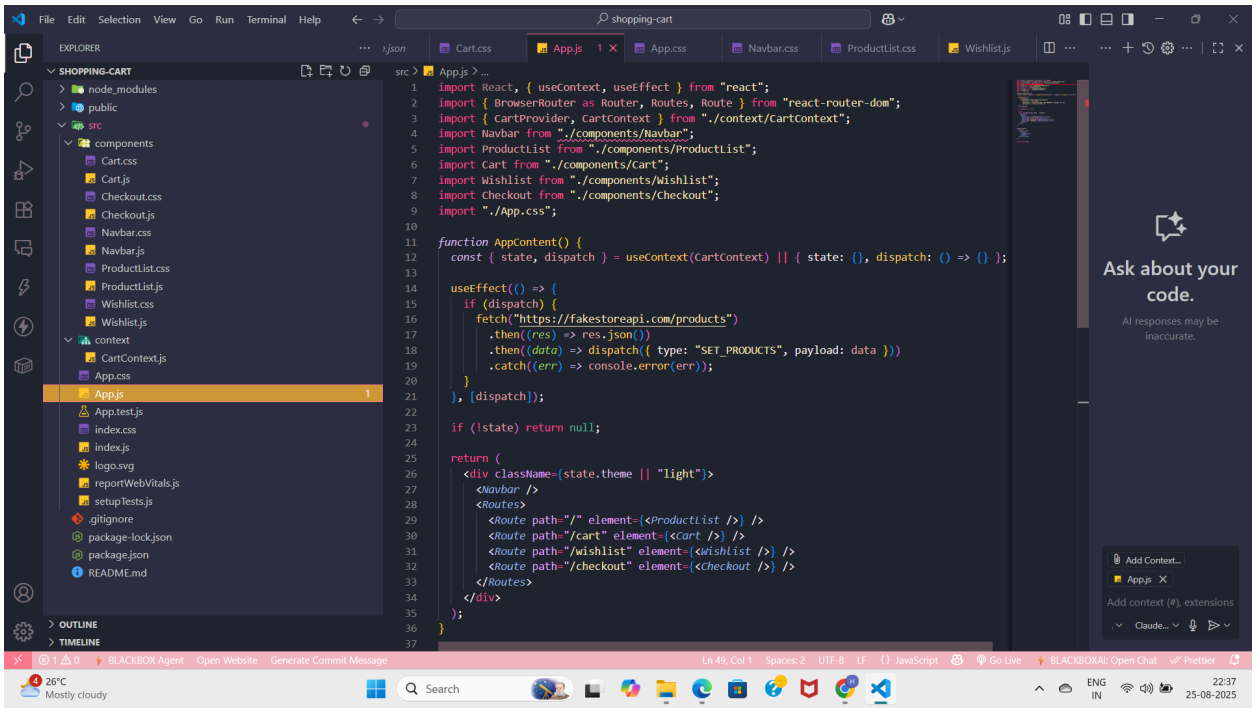
<h2>Your Wishlist</h2>
{state.wishlist.length === 0 ? (
No items in wishlist
) : (
state.wishlist.map(item => (

{item.title} - ₹{item.price}

<button onClick={() => dispatch({ type: "REMOVE_FROM_WISHLIST", payload: item.
Remove
</button>
<button onClick={() => {
dispatch({ type: "ADD_TO_CART", payload: item });
dispatch({ type: "REMOVE_FROM_WISHLIST", payload: item.id });
}}>
Move to Cart
</button>

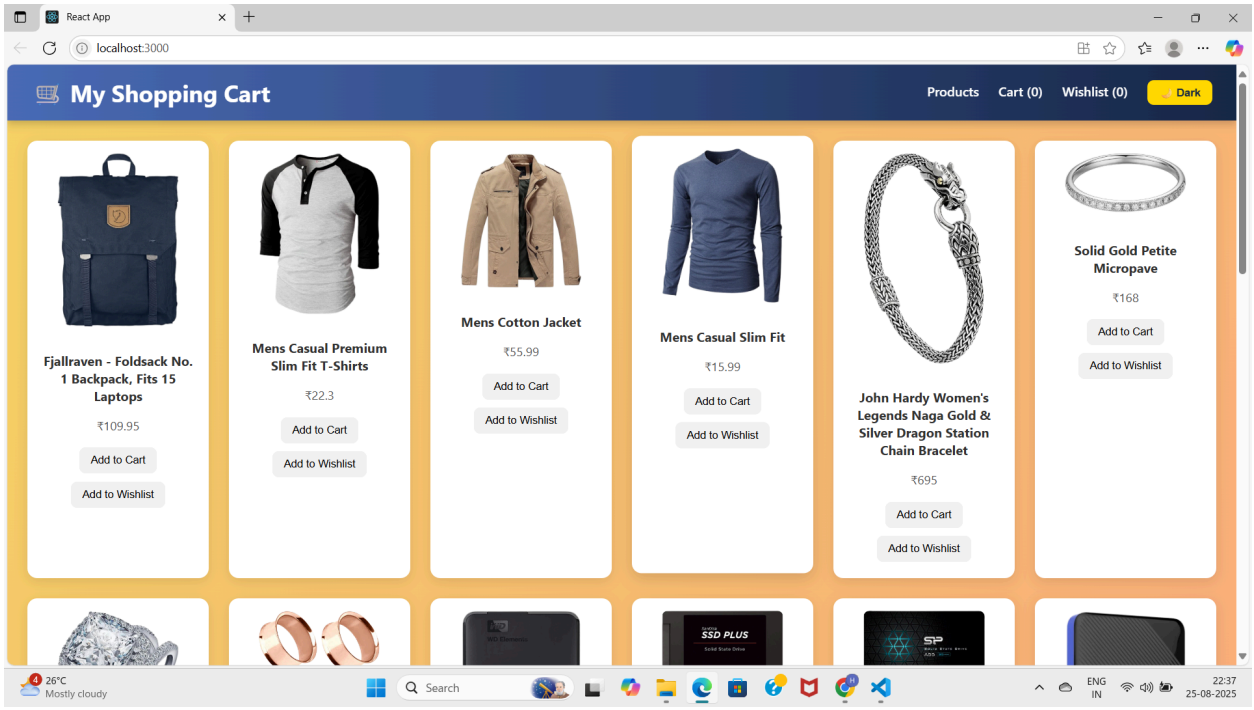
))
)}
<Link to="/">← Back to Products</Link>

);
};
export default Wishlist;
Ask about your code.
AI responses may be inaccurate.
Add Context...
Wishlist.js
Add context (#), extensions
Claude...
Ln 38, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live BLACKBOXAI: Open Chat Prettier
26°C
Mostly cloudy
Search
ENG
IN
22:36
25-08-2025

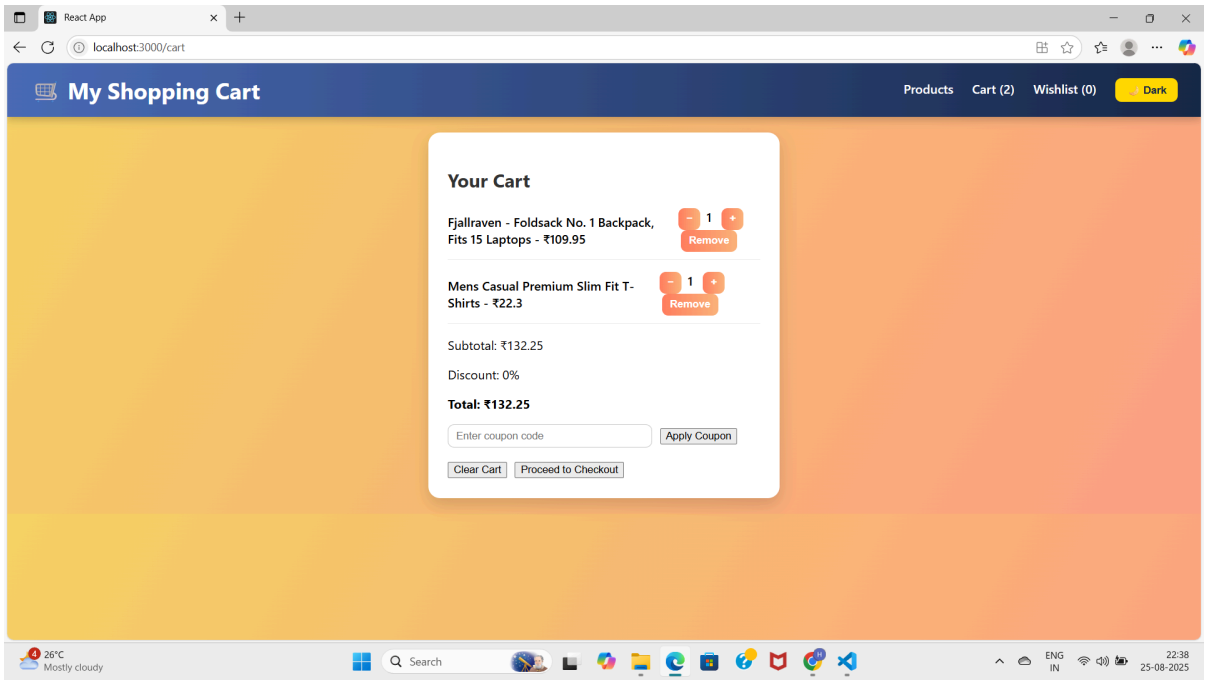

File Edit Selection View Go Run Terminal Help
shopping-cart
EXPLORER
SHOPPING-CART
node_modules
public
src
components
Cart.css
Cart.js
Checkout.css
Checkout.js
Navbar.css
Navbar.js
ProductList.css
ProductList.js
Wishlist.css
Wishlist.js
context
CartContext.js
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
src > App.js > ...
import React, { useContext, useEffect } from "react";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import { CartProvider, CartContext } from "./context/CartContext";
import Navbar from "./components/Navbar";
import ProductList from "./components/ProductList";
import Cart from "./components/Cart";
import Wishlist from "./components/Wishlist";
import Checkout from "./components/Checkout";
import "./App.css";
function AppContent() {
const { state, dispatch } = useContext(CartContext) || { state: {}, dispatch: () => {} };
useEffect(() => {
if (dispatch) {
fetch("https://fakestoreapi.com/products")
.then((res) => res.json())
.then((data) => dispatch({ type: "SET_PRODUCTS", payload: data }))
.catch((err) => console.error(err));
}
}, [dispatch]);
if (!state) return null;
return (
<div className={state.theme || "light"}>
<Navbar />
<Routes>
<Route path="/" element={<ProductList />} />
<Route path="/cart" element={<Cart />} />
<Route path="/wishlist" element={<Wishlist />} />
<Route path="/checkout" element={<Checkout />} />
</Routes>

);
}
Ask about your code.
AI responses may be inaccurate.
Add Context...
App.js
Add context (#), extensions
Claude...
Ln 49, Col 1 Spaces: 2 UTF-8 LF {} JavaScript Go Live BLACKBOXAI: Open Chat Prettier
26°C
Mostly cloudy
Search
ENG
IN
22:37
25-08-2025

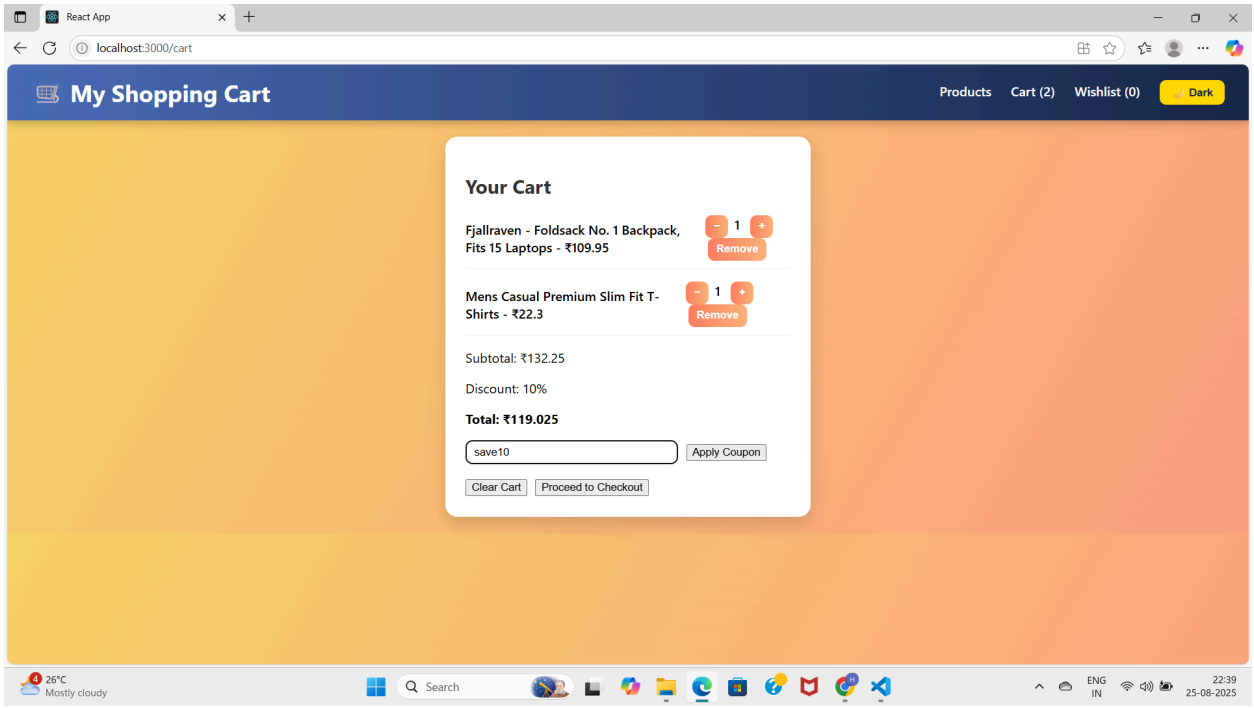
Output:



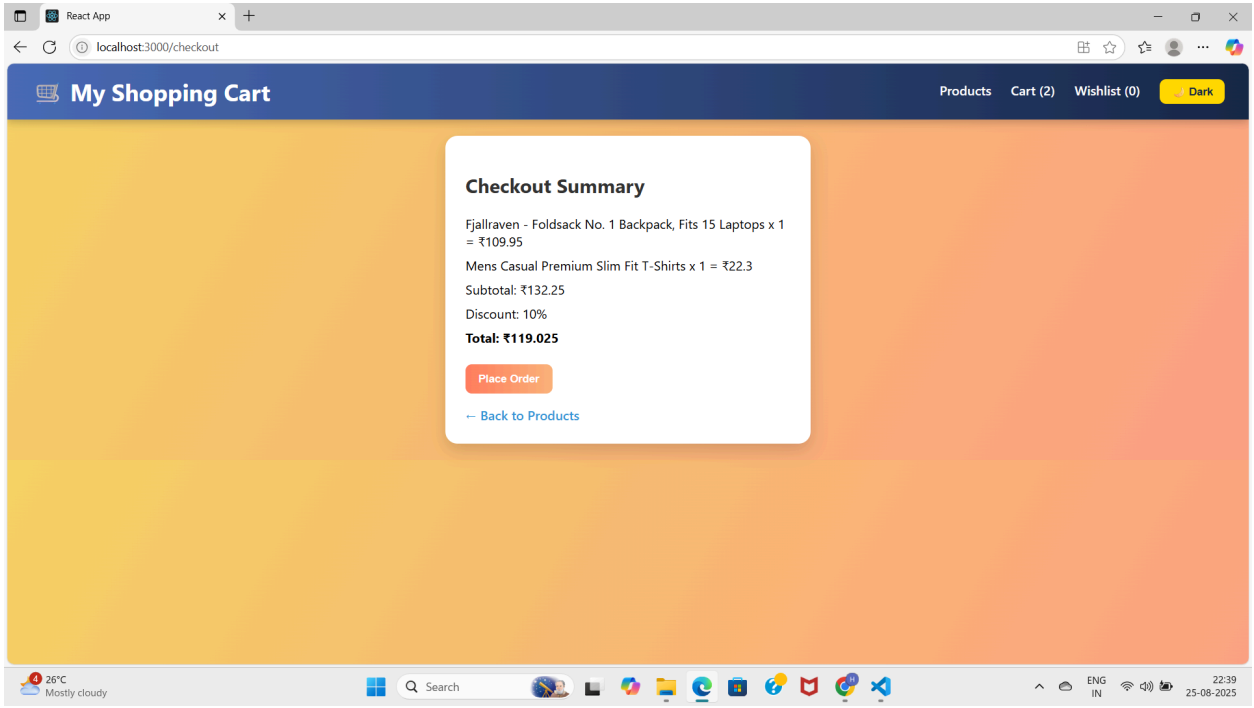
a. Landing page of websites showing products by fetching from API



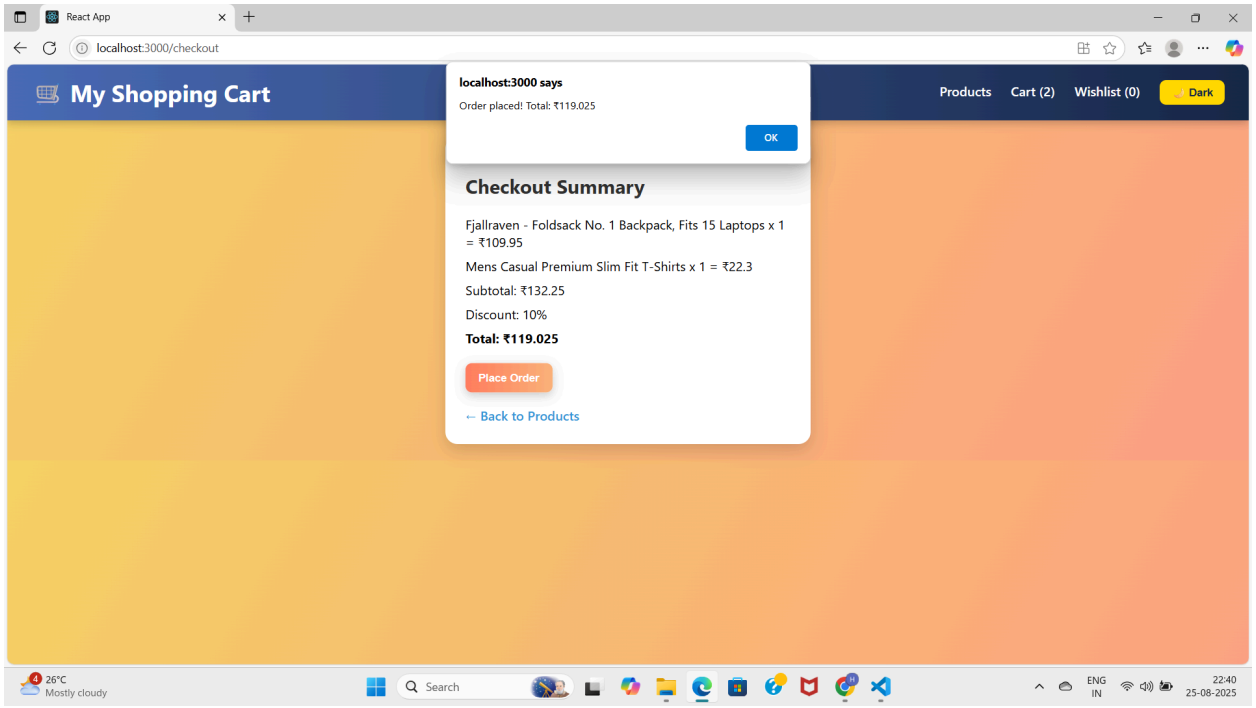
b. Cart page showing the updates



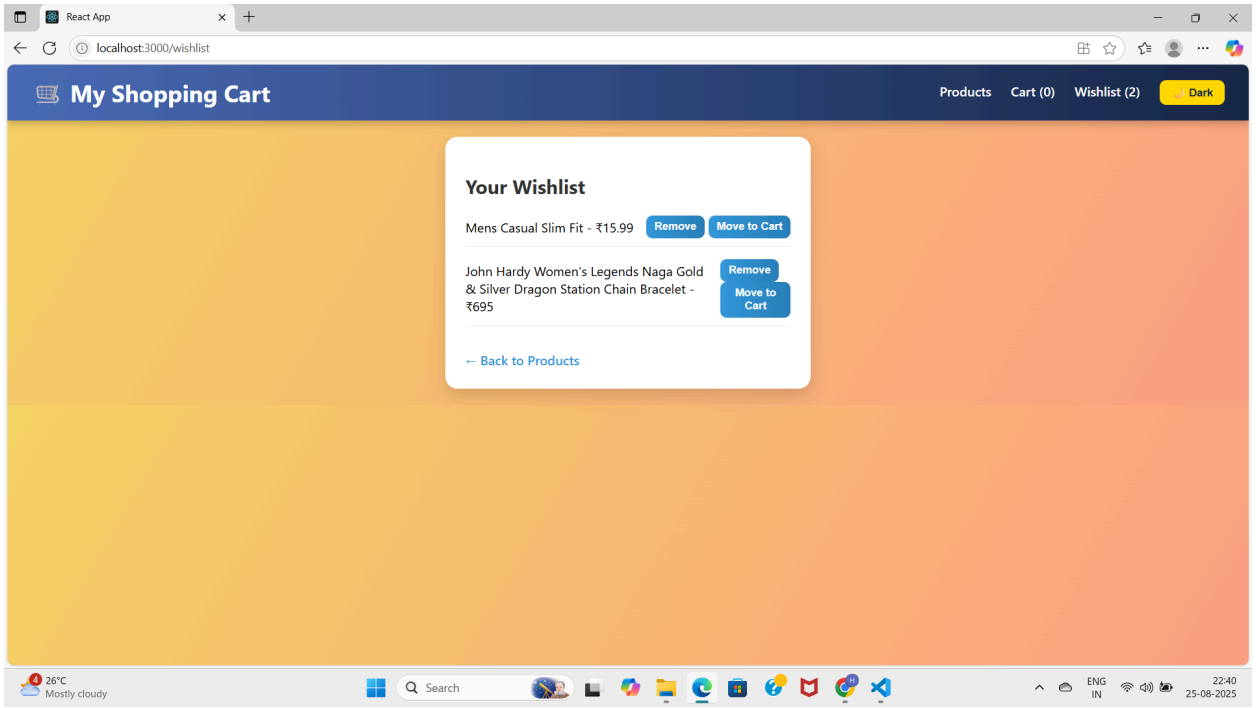
c. Coupon code applied



d. Checkout page



e. Order placed



f. Wishlisted items

Conclusion:

This project successfully demonstrated how **Context API** can be used to manage complex application state efficiently. By eliminating prop drilling and centralizing data, the app became more maintainable and scalable. Additionally, features like API integration, responsive UI, and routing extended the scope beyond the original aim, making the project more practical and user-friendly.