**Lesson 06 Demo 02**

**Implementing Framework Migration Using Generative AI**

**Objective:** To migrate the login functionality of a social media platform from Angular to Flask using a generative AI tool

**Tools required:** Visual Studio, Node JS and GitHub Copilot

**Prerequisites:** React JS knowledge

Steps to be followed:

1. Creating React JS Project for e-commerce application.
2. Debug the react js project

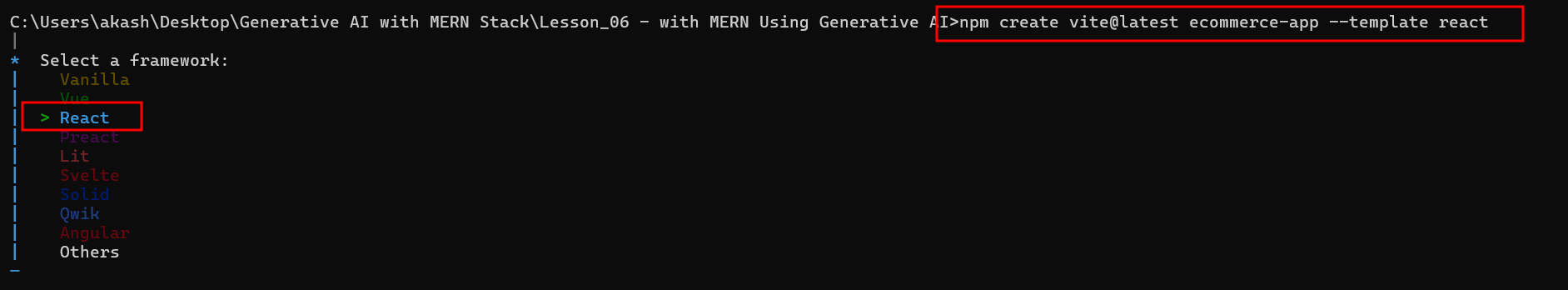
**Step 1: Creating the React JS project with Vite framework for e-commerce application**

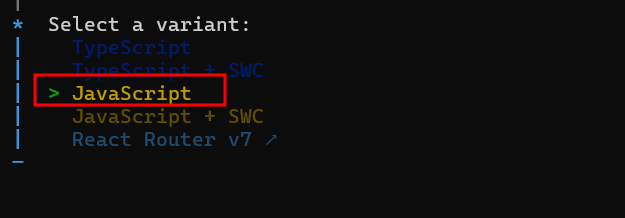
* 1. Step to create the project

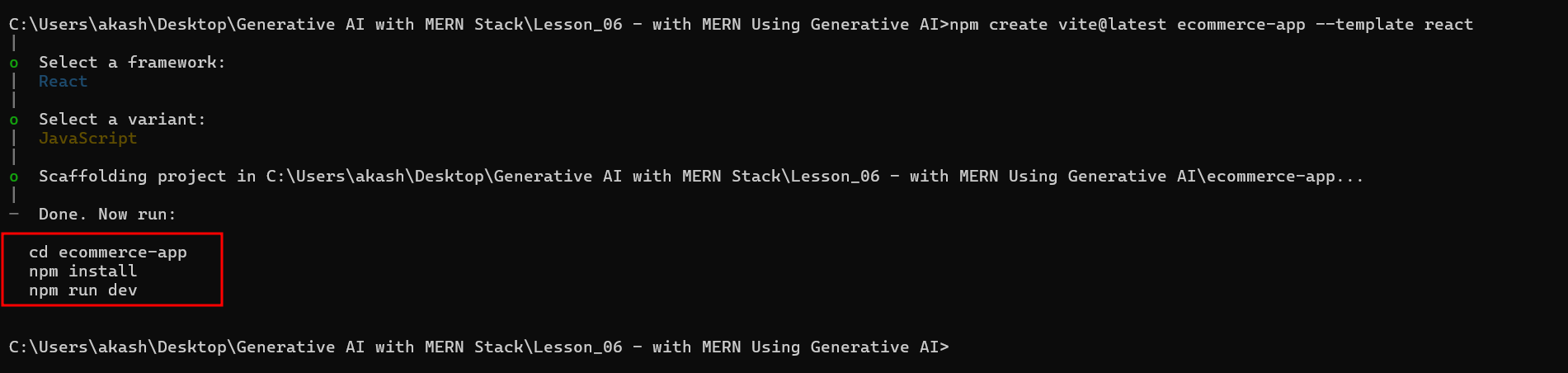
**npm create vite@latest ecommerce-app --template react**

**cd ecommerce-app**

**npm install**







* 1. Download the required dependencies

**cd ecommerce-app**

**npm install**

**npm install react-router-dom axios**

* 1. Open this project in VS Code
  2. Create the component base upon the below project structure



* 1. Create each component one by one

**These all component from components folder**

**Cart.tsx**

// src/components/Cart.js

import React from 'react';

const Cart = ({ cartItems, onRemoveFromCart }) => {

  return (

    <div className="cart">

      <h2>Your Cart</h2>

      {cartItems.length === 0 ? (

        <p>No items in the cart</p>

      ) : (

        cartItems.map((item, index) => (

          <div key={index}>

            <p>{item.name} - ${item.price}</p>

            <button onClick={() => onRemoveFromCart(item)}>Remove</button>

          </div>

        ))

      )}

    </div>

  );

};

export default Cart;

**Header.jsx**

// src/components/Header.js

import React from 'react';

import { Link } from 'react-router-dom';

const Header = () => {

  return (

    <header>

      <nav>

        <Link to="/">Home</Link>

        <Link to="/cart">Cart</Link>

      </nav>

    </header>

  );

};

export default Header;

**ProductCart.jsx**

// src/components/ProductCard.jsx

import React from 'react';

const ProductCard = ({ product, onAddToCart }) => {

  const { name, price, image } = product;

  return (

    <div className="product-card">

      <img className="product-image" src={image} alt={name} />

      <h3>{name}</h3>

      <p>${price}</p>

      <button onClick={() => onAddToCart(product)}>Add to Cart</button>

    </div>

  );

};

export default ProductCard;

**These all components from pages folder**

**CartPage.jsx**

// src/pages/CartPage.jsx

import React from 'react';

const CartPage = ({ cartItems, removeFromCart }) => {

  return (

    <div className="cart-page">

      <h2>Your Cart</h2>

      {cartItems.length === 0 ? (

        <p>No items in the cart</p>

      ) : (

        cartItems.map((item, index) => (

          <div key={index}>

            <p>{item.name} - ${item.price}</p>

            <button onClick={() => removeFromCart(item)}>Remove</button>

          </div>

        ))

      )}

    </div>

  );

};

export default CartPage;

**HomePage.jsx**

// src/pages/HomePage.jsx

import React, { useState, useEffect } from 'react';

import ProductCard from '../components/ProductCard';

const HomePage = ({ addToCart }) => {

  const [products, setProducts] = useState([]);

  useEffect(() => {

    const fetchProducts = async () => {

      const response = await fetch('https://fakestoreapi.com/products');

      const data = await response.json();

      setProducts(data);

    };

    fetchProducts();

  }, []);

  return (

    <div className="homepage">

      <h1>Our Products</h1>

      <div className="product-list">

        {products.map((product) => (

          <ProductCard

            key={product.id}

            product={product}

            onAddToCart={addToCart}

          />

        ))}

      </div>

    </div>

  );

};

export default HomePage;

**App.jsx** main component

// src/App.jsx

import React, { useState } from 'react';

import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';

import Header from './components/Header';

import HomePage from './pages/HomePage';

import CartPage from './pages/CartPage';

import './App.css';

const App = () => {

  const [cartItems, setCartItems] = useState([]);

  // Function to add products to the cart

  const addToCart = (product) => {

    setCartItems((prevCartItems) => [...prevCartItems, product]);

  };

  // Function to remove products from the cart

  const removeFromCart = (product) => {

    setCartItems((prevCartItems) => prevCartItems.filter(item => item.id !== product.id));

  };

  return (

    <Router>

      <Header />

      <Routes>

        <Route path="/" element={<HomePage addToCart={addToCart} />} />

        <Route path="/cart" element={<CartPage cartItems={cartItems} removeFromCart={removeFromCart} />} />

      </Routes>

    </Router>

  );

};

export default App;

App.css file

/\* src/App.css \*/

.product-card {

  border: 1px solid #ddd;

  padding: 15px;

  margin: 10px;

  text-align: center;

}

.product-image {

  max-width: 150px; /\* Set the max width \*/

  max-height: 150px; /\* Set the max height \*/

  object-fit: contain; /\* Ensures the aspect ratio is maintained \*/

}

.product-list {

  display: flex;

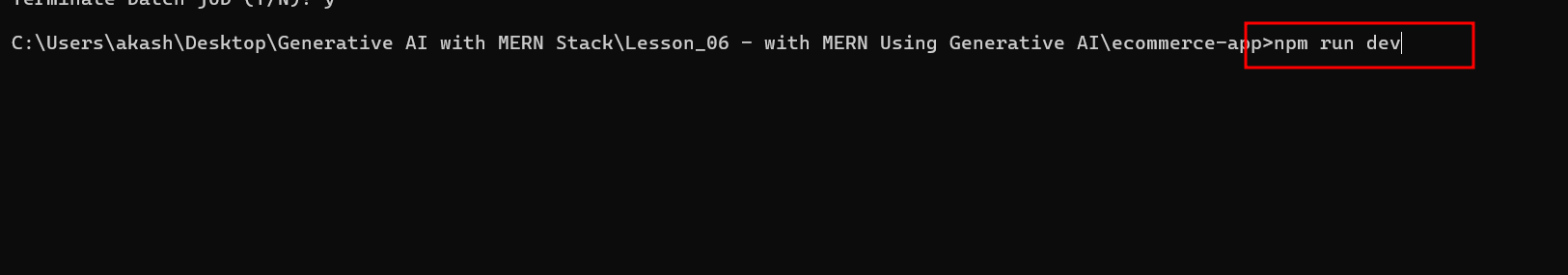
  flex-wrap: wrap;

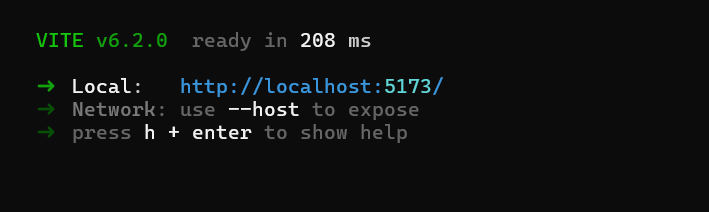
  justify-content: center;

}

Once you run the application on browser

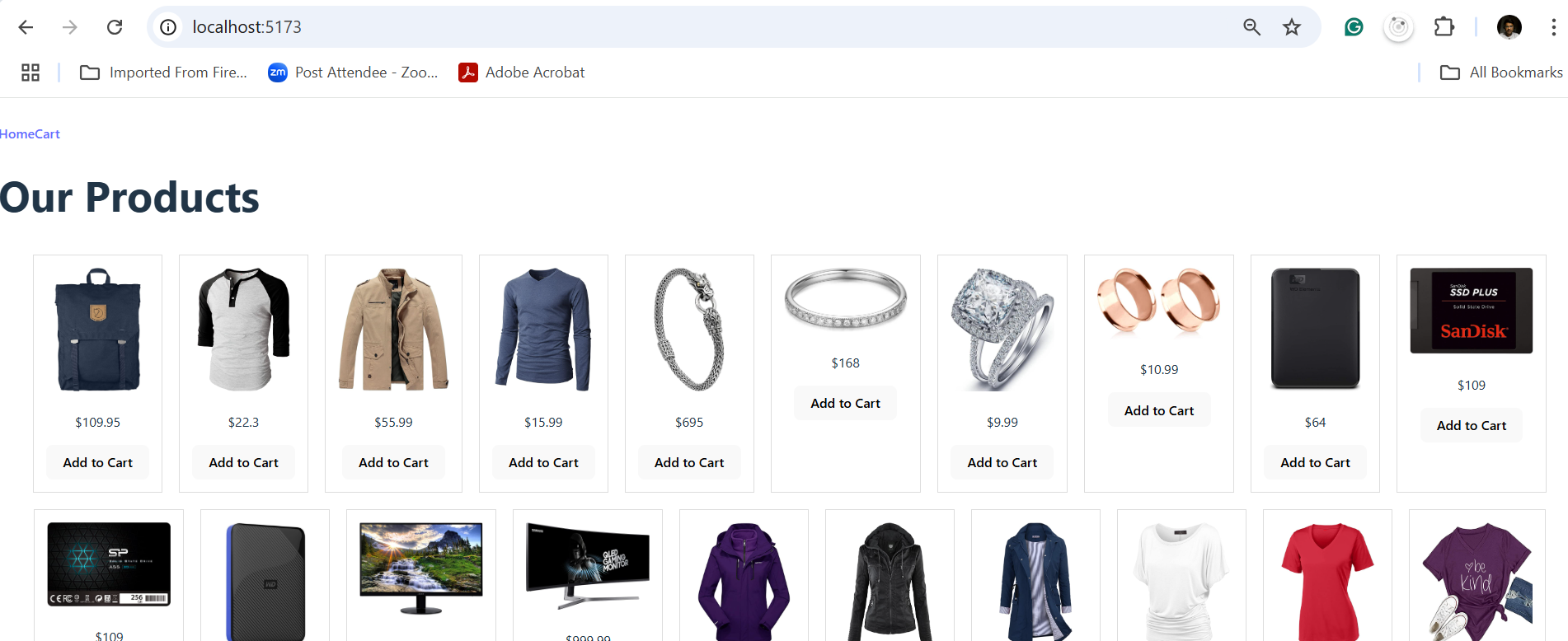
**npm run dev**





Open the application on browser

<http://localhost:5173>



You can add the product in cart

You can view the cart details after click on HomeCart link



**Step 2: Debug the application**

2.1 Check for Console Errors

Start by checking for any error messages in the browser console (right-click -> Inspect -> Console tab):

* Open the developer tools in your browser (F12 or Ctrl + Shift + I for Chrome).
* Look for any errors related to React, your state management, or missing files.
* If any errors are present, try to resolve them one by one.

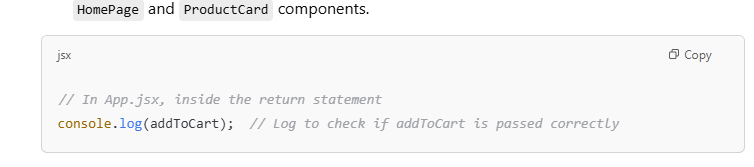


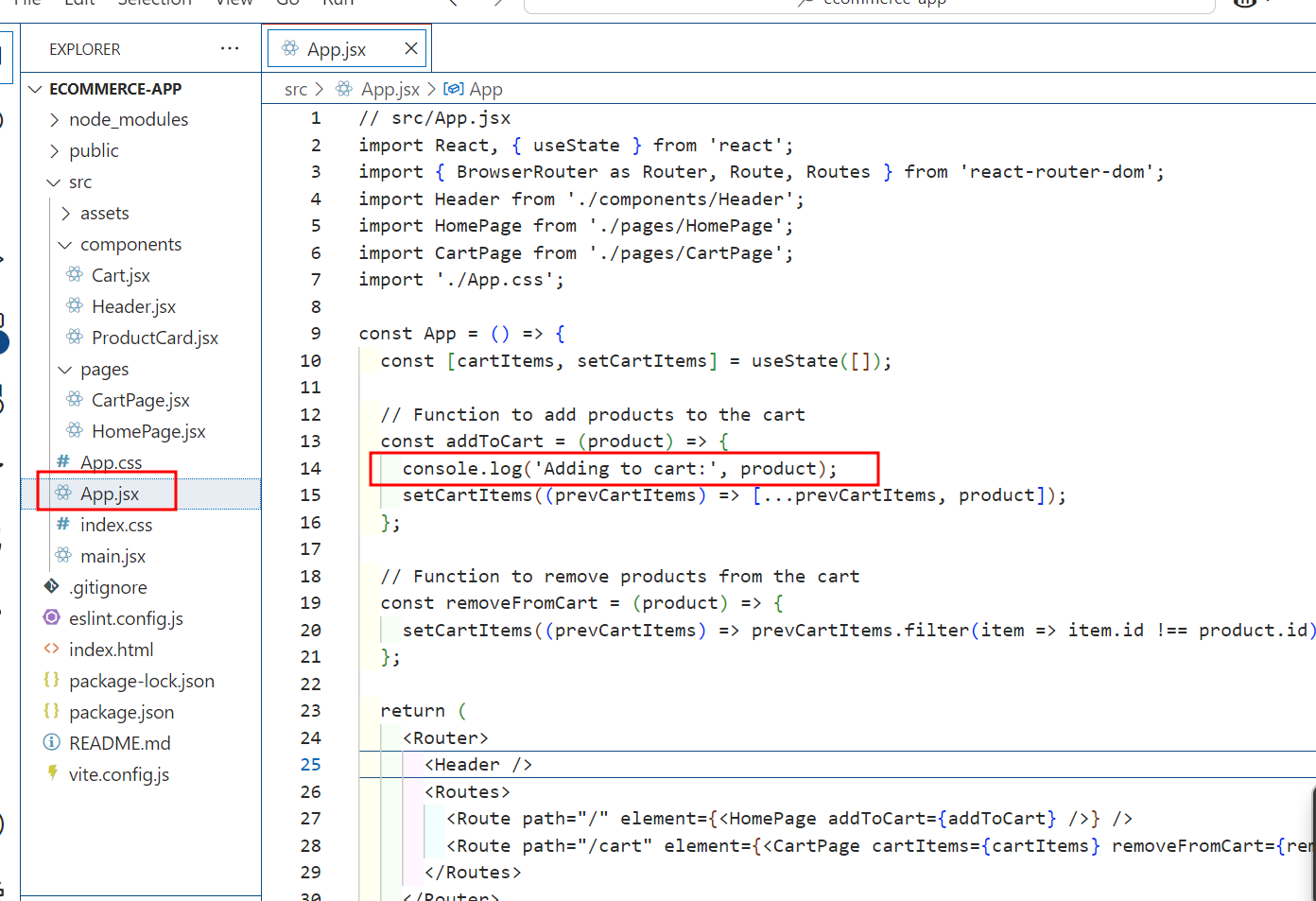
2.2: Check Component States and Props

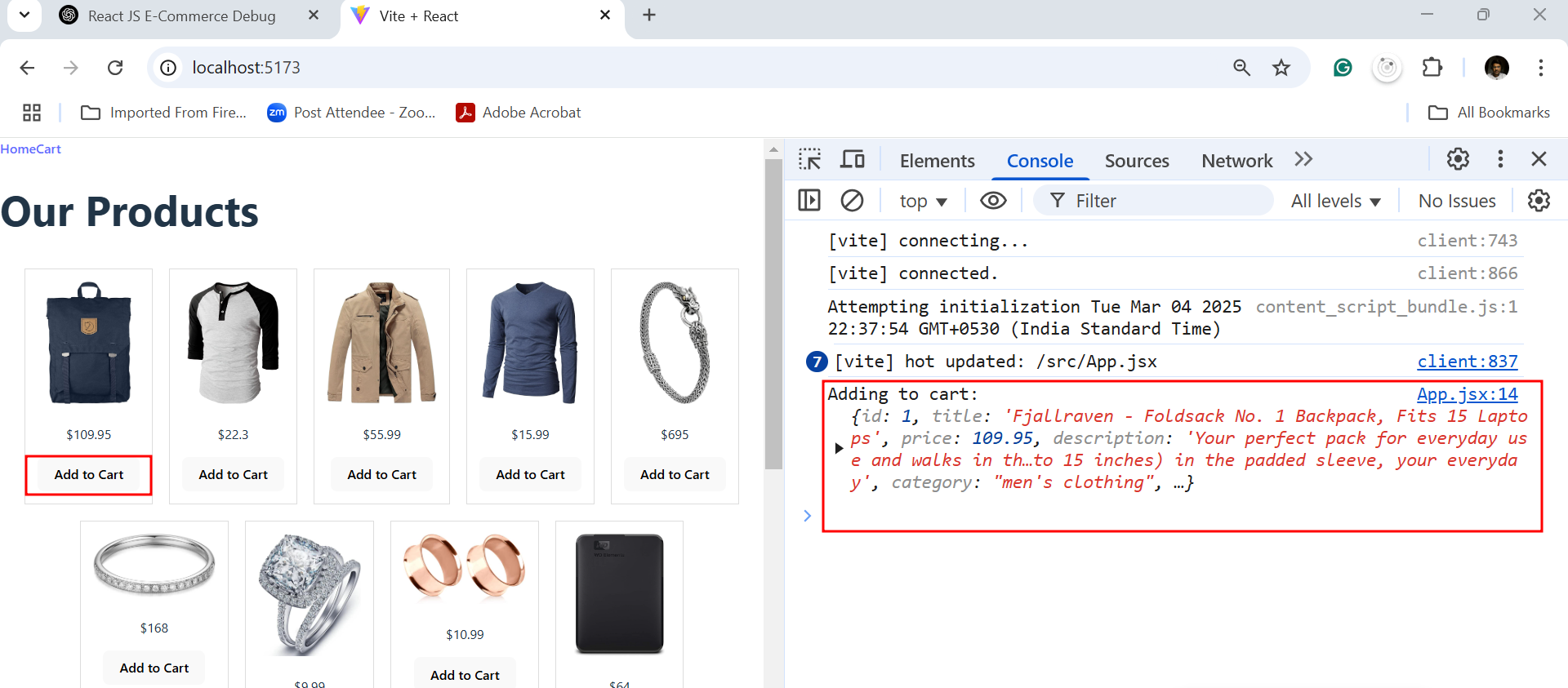
If your "Add to Cart" functionality is not working as expected, the issue could lie in how the state and props are managed.

**Debugging App.jsx:**

1. **Check if addToCart is being passed correctly:**
   * Use console.log to log the addToCart function to ensure it’s being passed as a prop to the HomePage and ProductCard components.



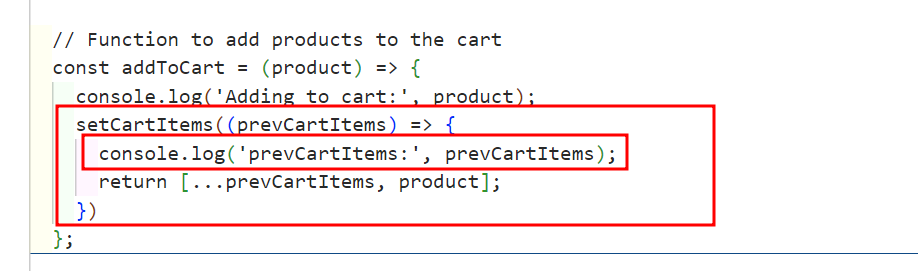




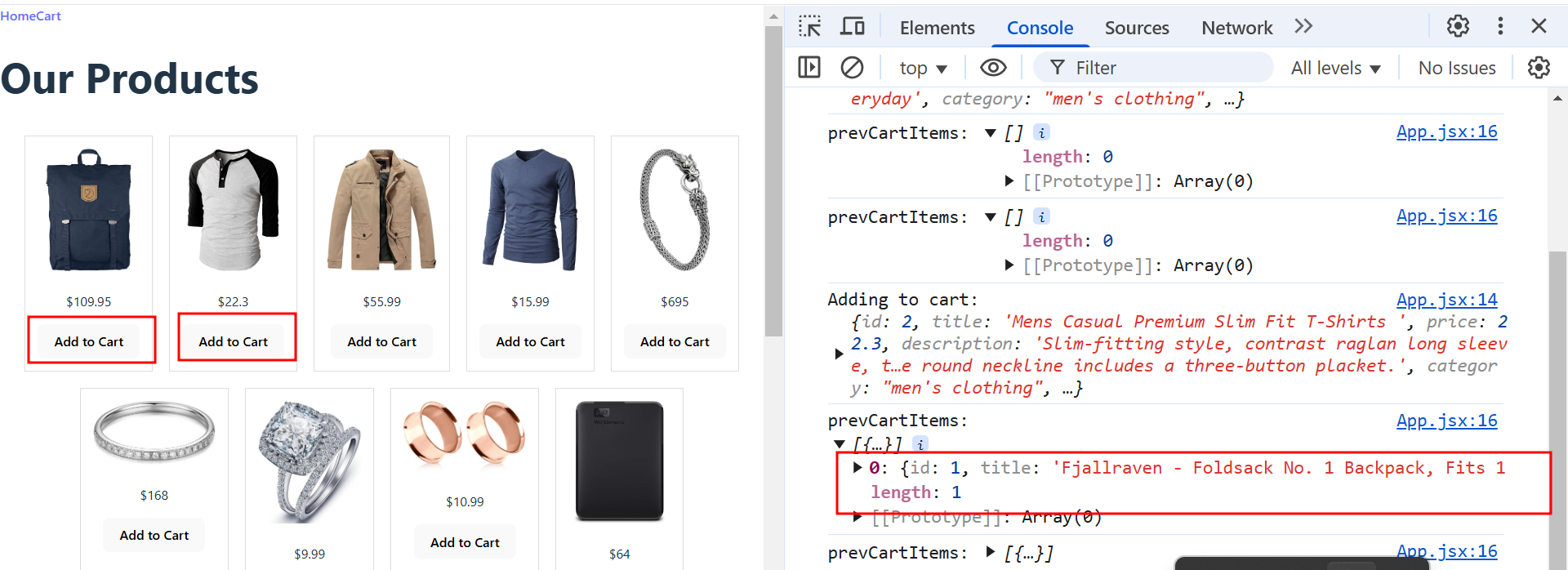
**Check if the cart state is updating properly:**

* Log the cartItems state to ensure that items are being added correctly.





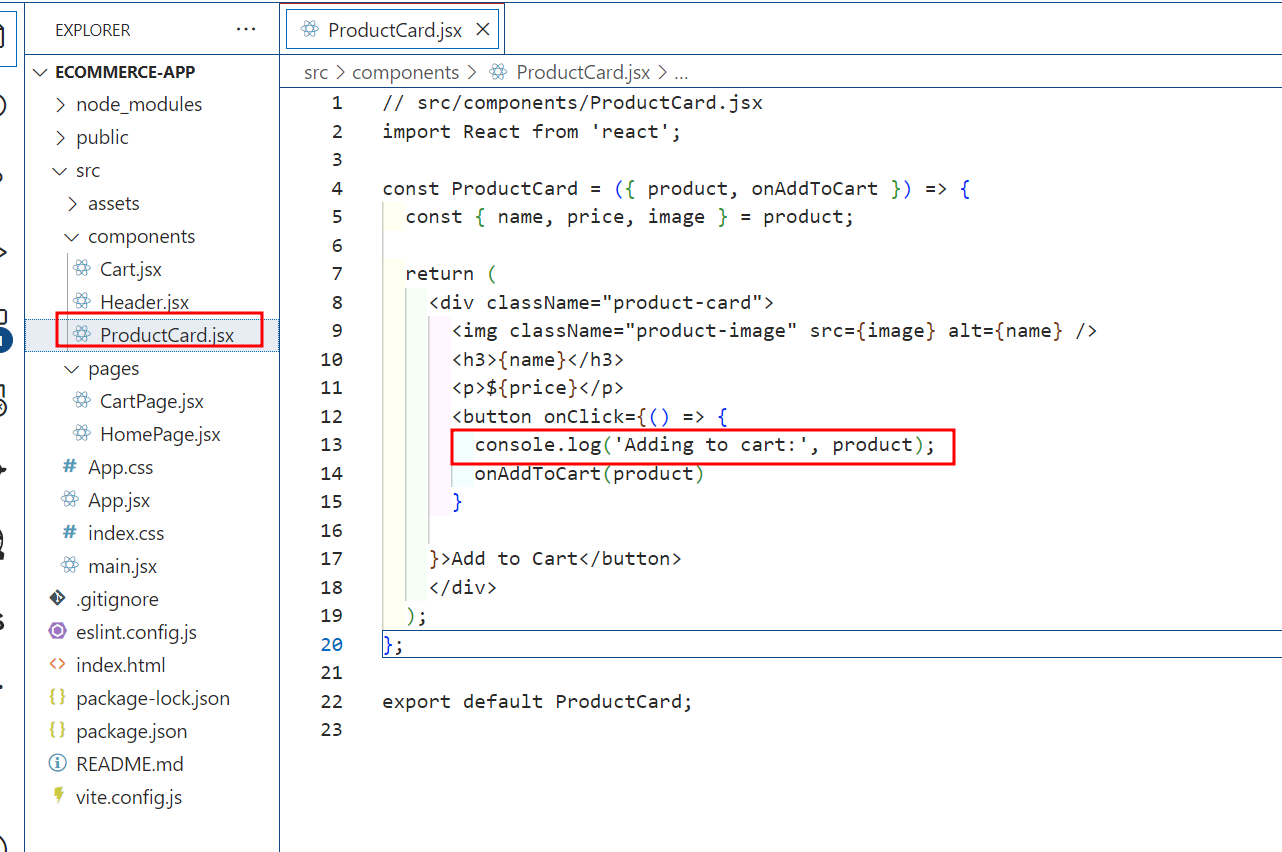
After added two product you need to check previous product present or not

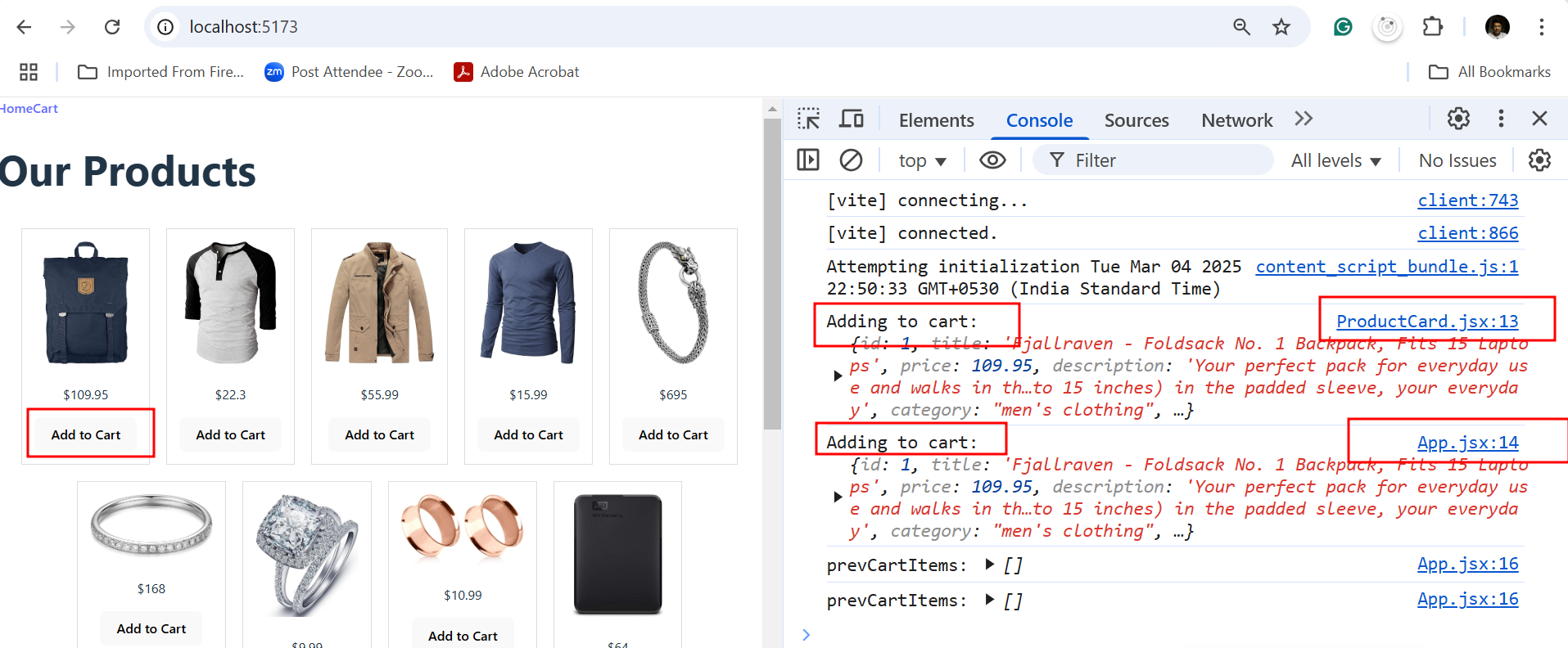


Debugging ProductCard.jsx:

1. Check if onAddToCart is being triggered:
   * In the ProductCard.jsx, add a console.log in the onClick event of the "Add to Cart" button to check if the function is being called.

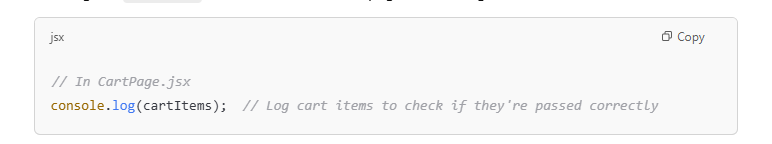


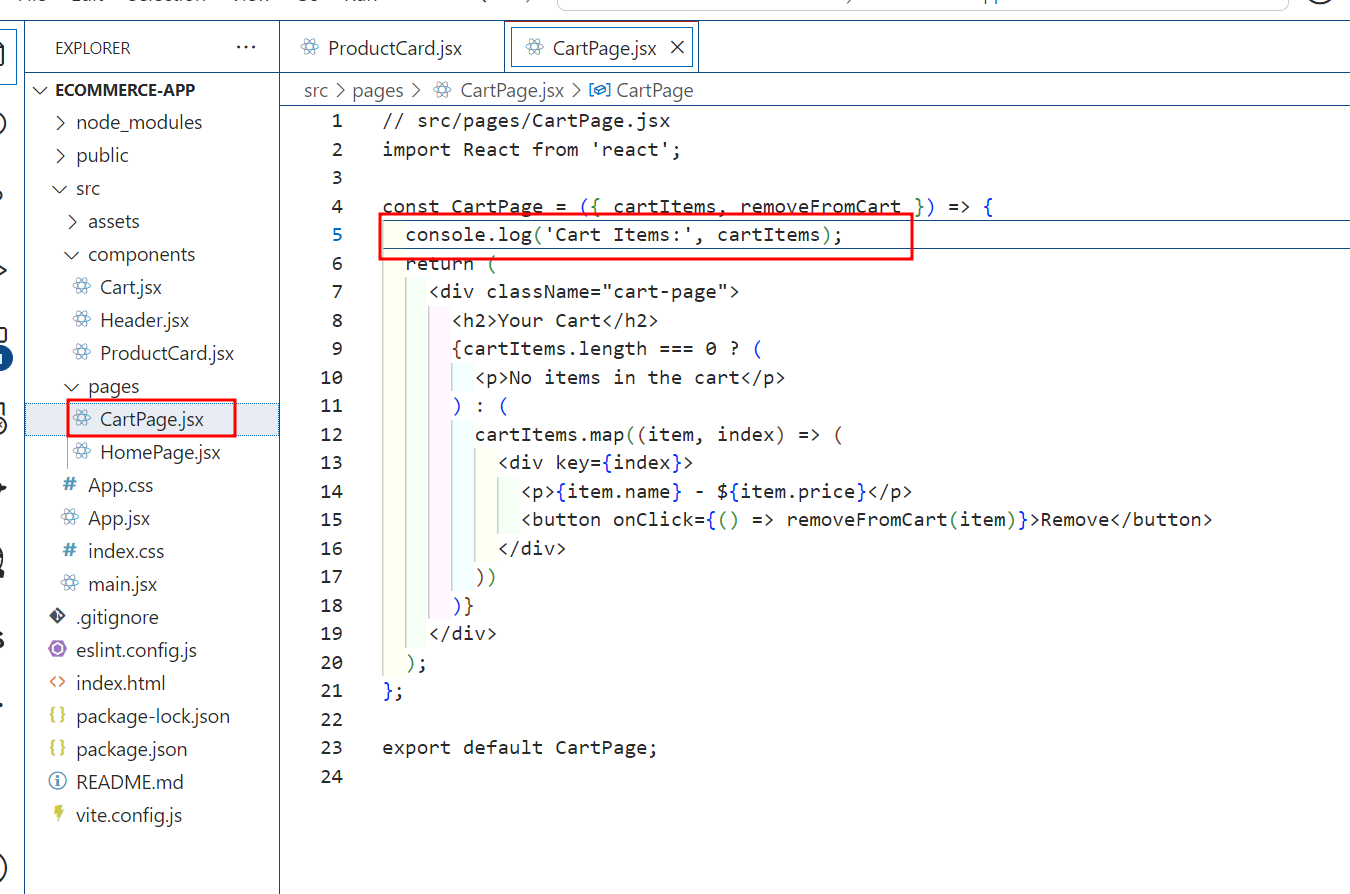


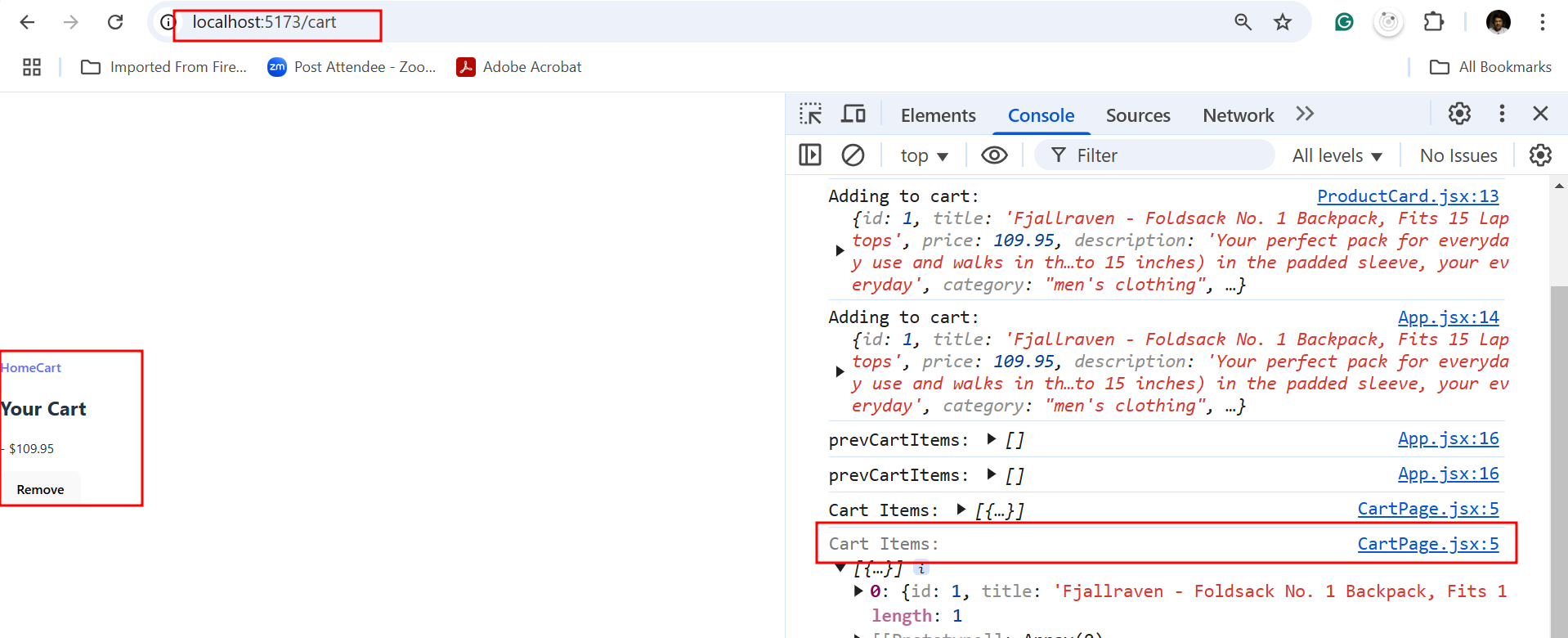


Debugging CartPage.jsx:

1. Check if cartItems is being passed correctly:
   * Log the cartItems to ensure that the cart page is receiving the correct data.







2.2 Check the Network Requests (API Fetch)

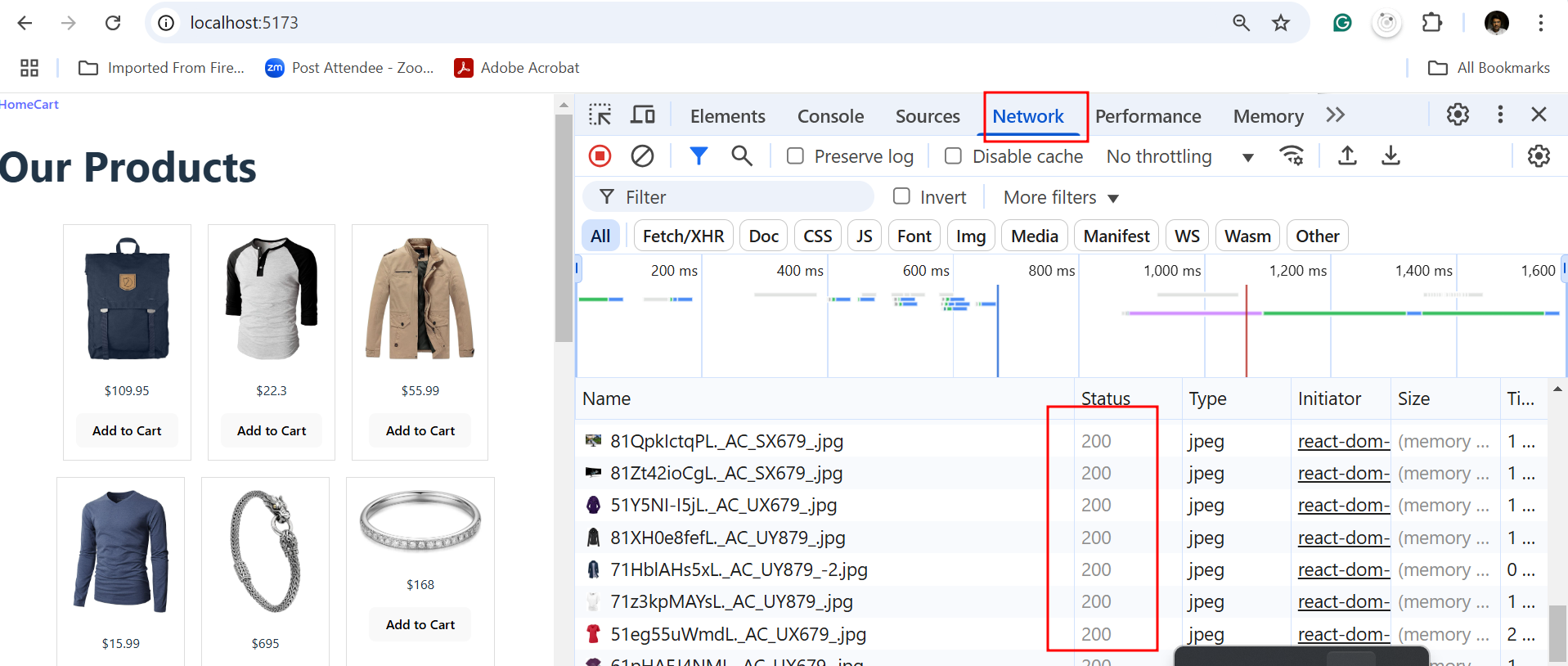
If your product data is not showing, or you're unable to see any products, check if your **API call** is successful:

* Open the **Network tab** in your browser's developer tools and check if the API request to fetch products (https://fakestoreapi.com/products) is successful.
* Ensure the request is returning valid product data.
* If the network request is failing, try logging any errors or checking for CORS issues (Cross-Origin Resource Sharing).

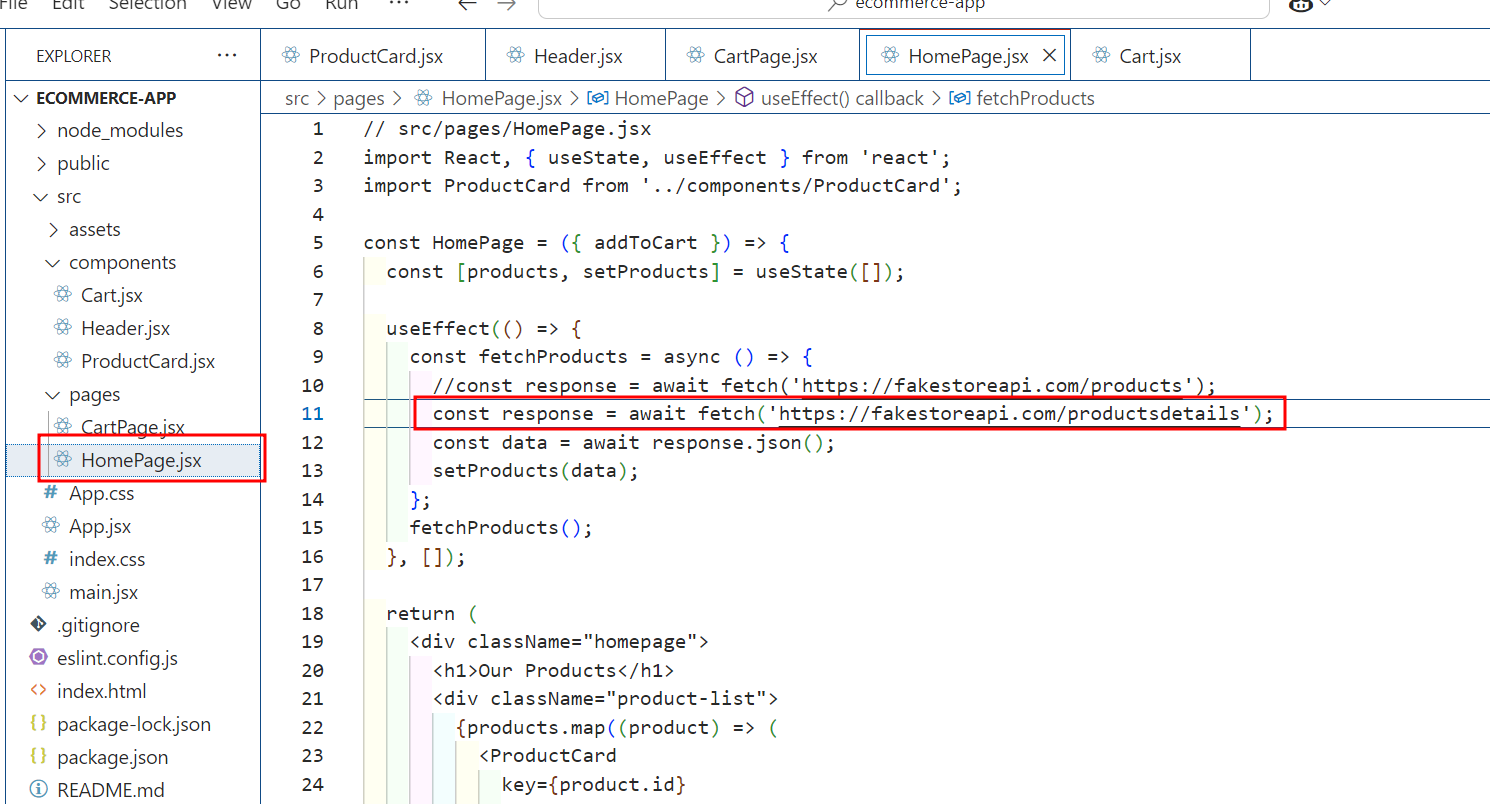
Example for checking the network request:

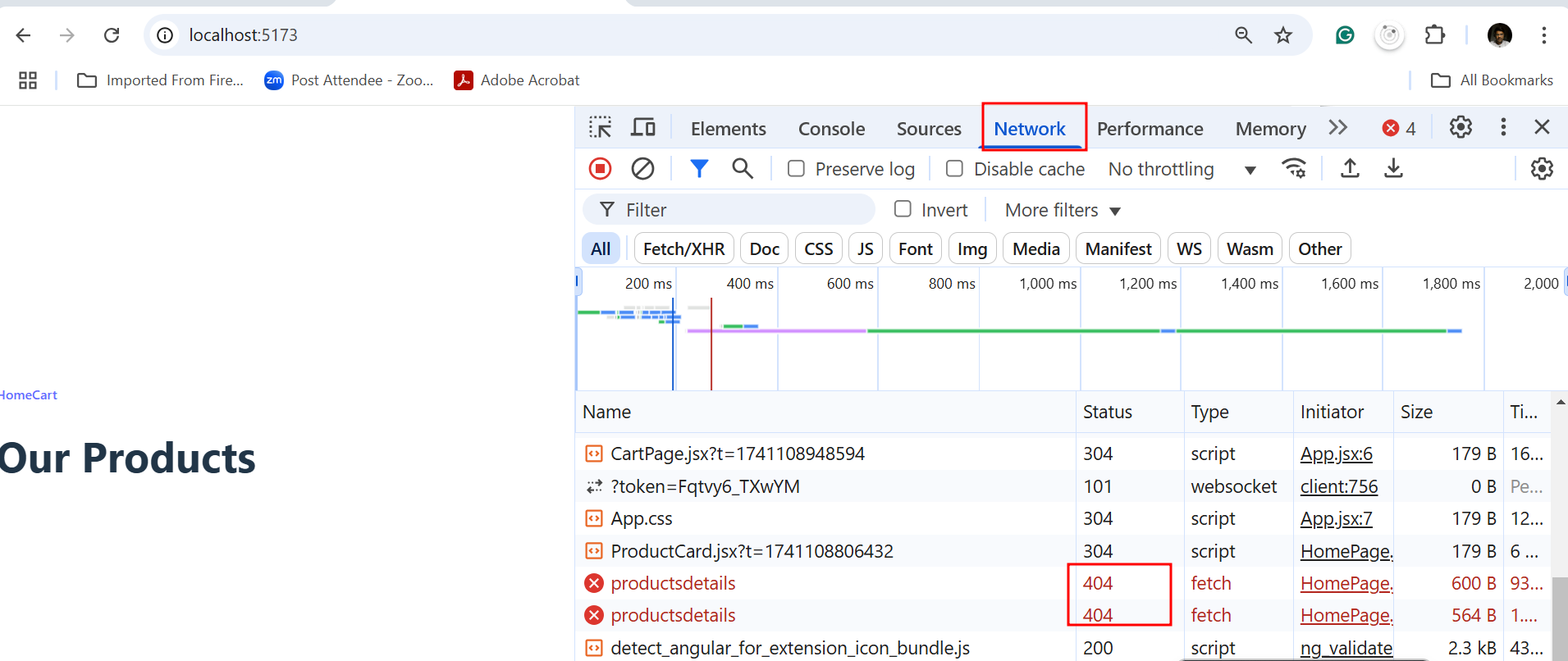
1. Open Developer Tools (F12).
2. Go to the **Network** tab.
3. Look for the fetch request for your product API.
4. Check the status code. If it's 200, the request succeeded, and if it's something like 404 or 500, there is an issue with the API endpoint.

With correct rest api url



With wrong rest api url

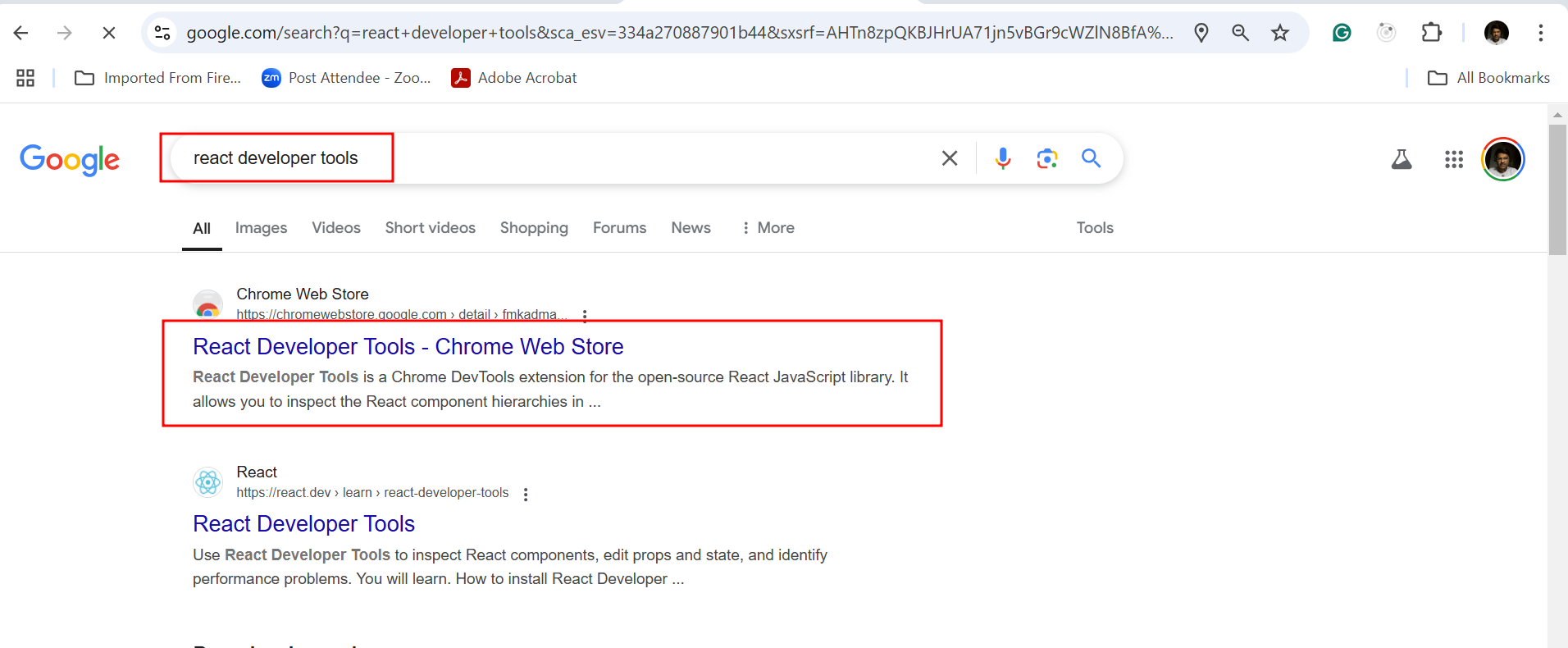




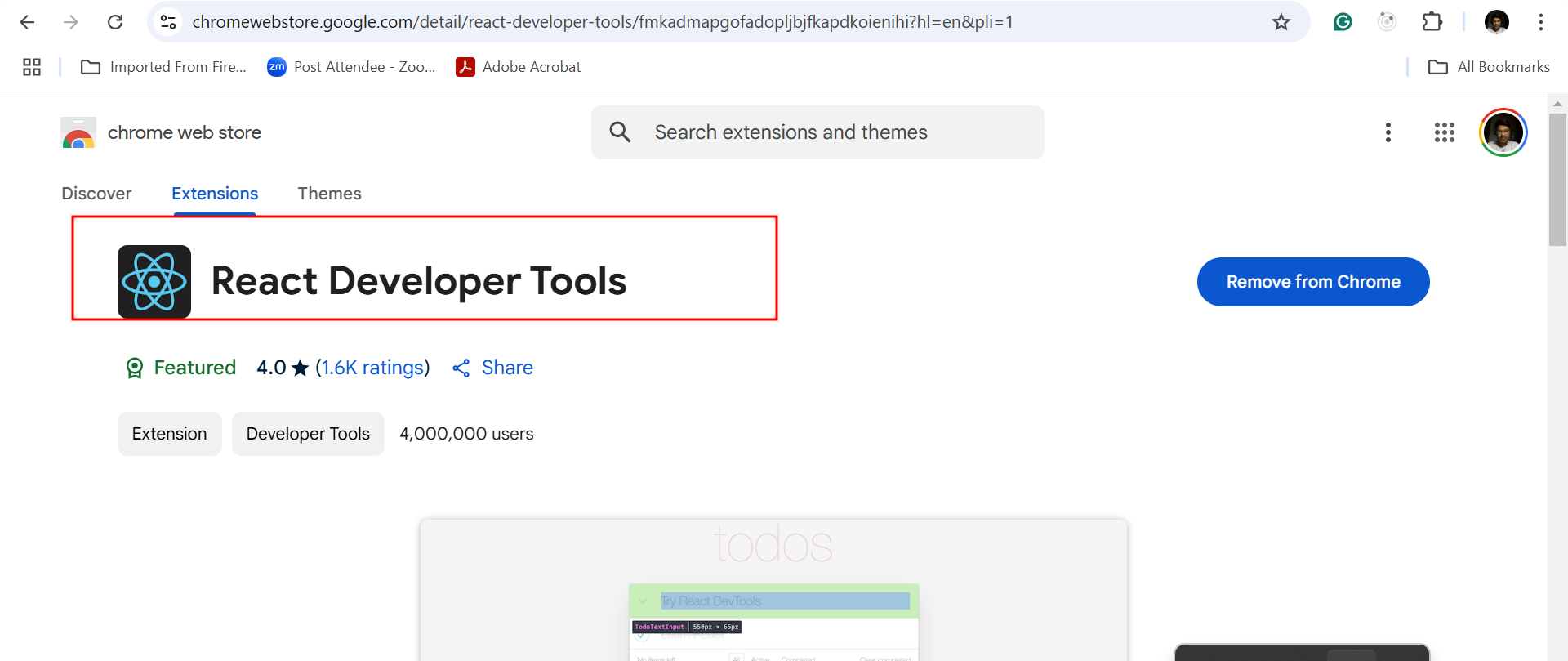
2.3 check React Developer Tools

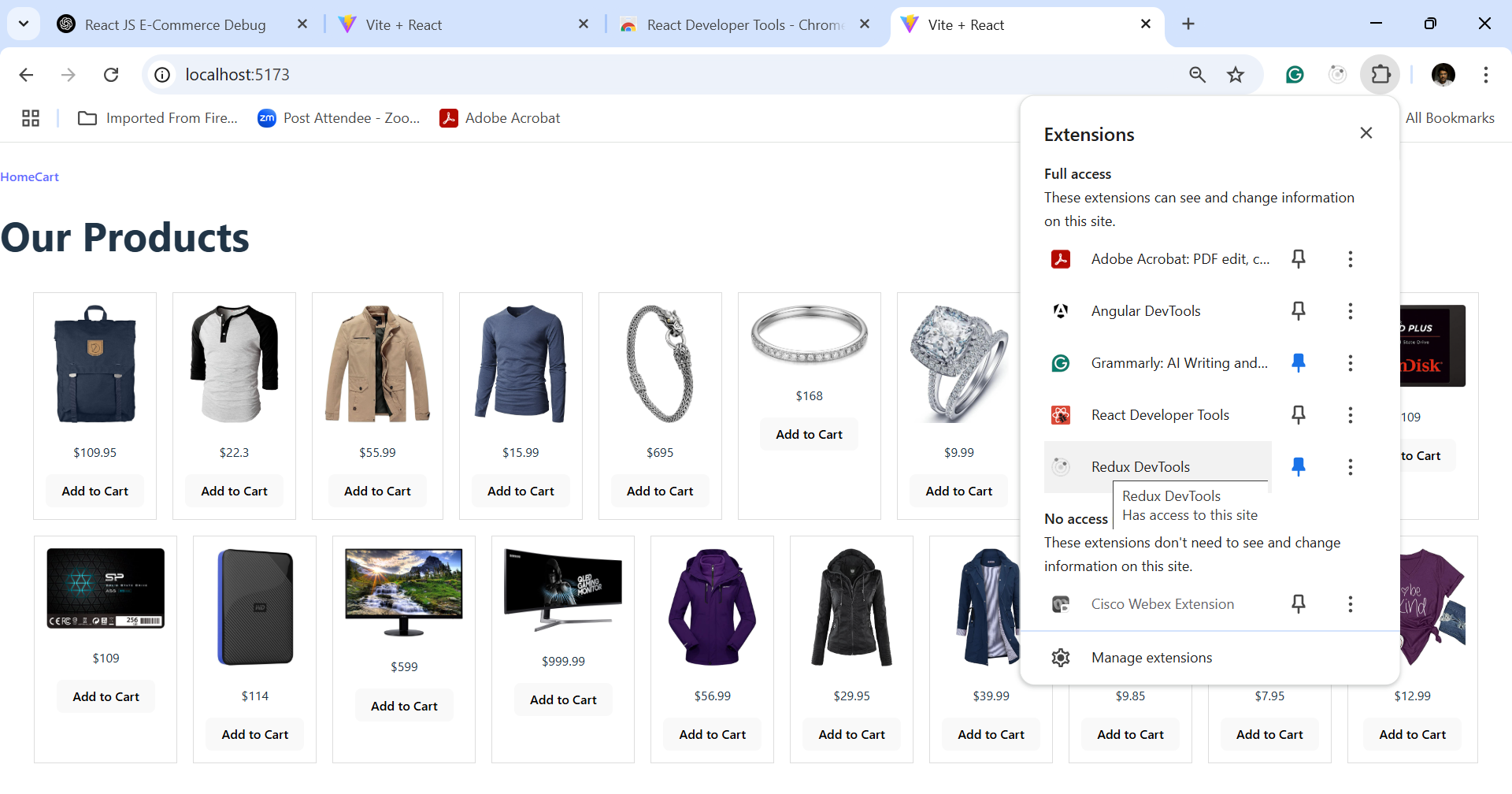
You can also use the **React Developer Tools** browser extension to inspect the component tree and see the current state and props of your components:

1. Install the **React Developer Tools** browser extension from Chrome Web Store or
2. Firefox Add-ons.
3. Open the Developer Tools in your browser (F12 or Ctrl + Shift + I).
4. Go to the **React** tab.
5. Inspect the component tree and ensure that props like addToCart are passed correctly.



You need to add from chrome

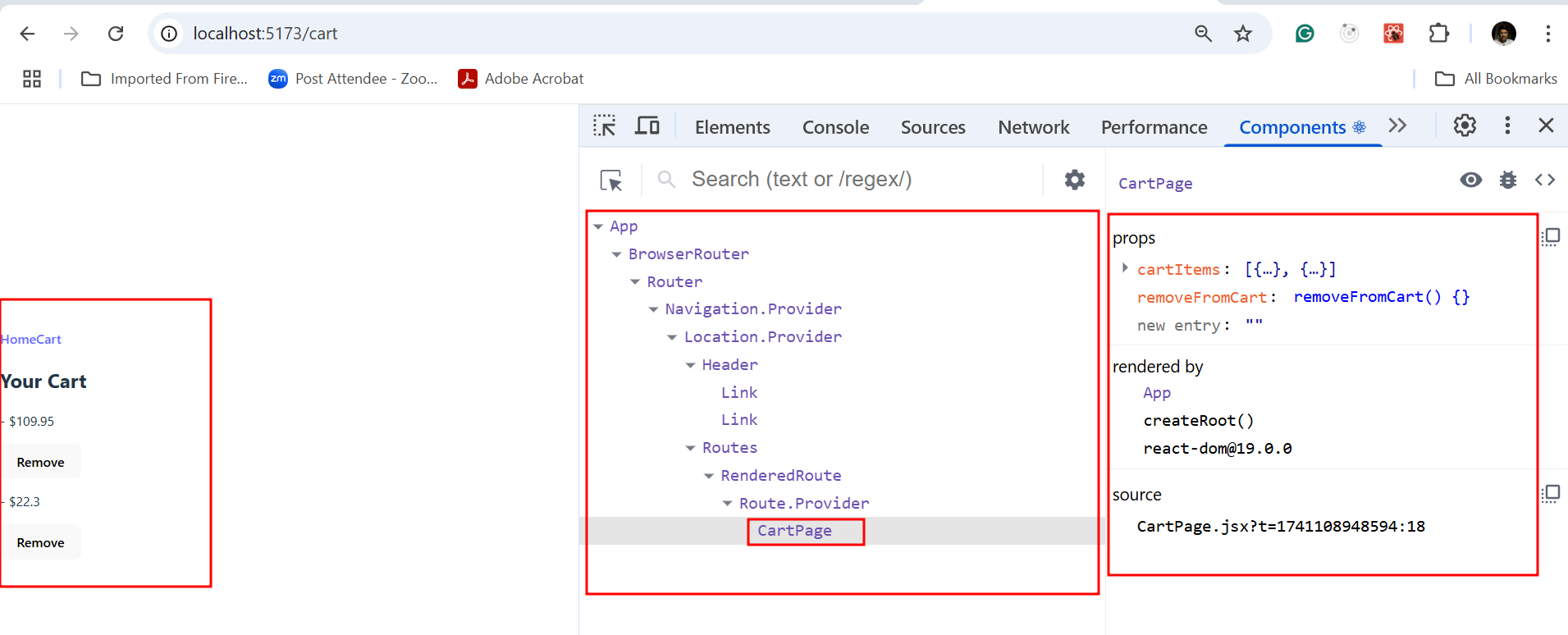




Here you can open the react develop tools options



After added the product in cart you can see the component tree structure



Click on profile option and click start button and do some operation and you can check the profile details

