**- In “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/backend/routes/productRoute.js” file, update the following code:**

import express from 'express';

import Product from '../models/productModel';

import { isAuth, isAdmin } from '../util';

const router = express.Router();

router.get("/", async (req, res) => {

    const products = await Product.find({});

    res.send(products);

});

router.get("/:id", async (req, res) => {

    const product = await Product.findOne({\_id: req.params.id});

    if(product) {

        res.send(product);

    } else {

        res.status(404).send({ message: "Product Not Found" });

    }

});

// Creating Product

router.post("/", isAuth, isAdmin, async (req, res) => {

    const product = new Product({

        name: req.body.name,

        price: req.body.price,

        image: req.body.image,

        brand: req.body.brand,

        category: req.body.category,

        countInStock: req.body.countInStock,

        description: req.body.description,

        rating: req.body.rating,

        numReviews: req.body.numReviews,

    });

    const newProduct = await product.save();

    if(newProduct) {

        return res.status(201).send({ message: 'New Product Created', data: newProduct });

    }

    return res.status(500).send({ message: 'Error in Creating Product.' });

})

// Editing Product

router.put("/:id", isAuth, isAdmin, async (req, res) => {

    const productId = req.params.id;

    const product = await Product.findById(productId);

    if(product){

        product.name = req.body.name;

        product.price = req.body.price;

        product.image = req.body.image;

        product.brand = req.body.brand;

        product.category = req.body.category;

        product.countInStock = req.body.countInStock;

        product.description = req.body.description;

        const updatedProduct = await product.save();

        if(updatedProduct) {

            return res.status(200).send({ message: 'Product Updated', data: updatedProduct });

        }

    }

    return res.status(500).send({ message: 'Error in Updating Product.' });

})

// Deleting Product

router.delete("/:id", isAuth, isAdmin, async (req, res) => {

    const deletedProduct = await Product.findById(req.params.id);

    if(deletedProduct) {

        await deletedProduct.remove();

        res.send({message: "Product Deleted"});

    } else {

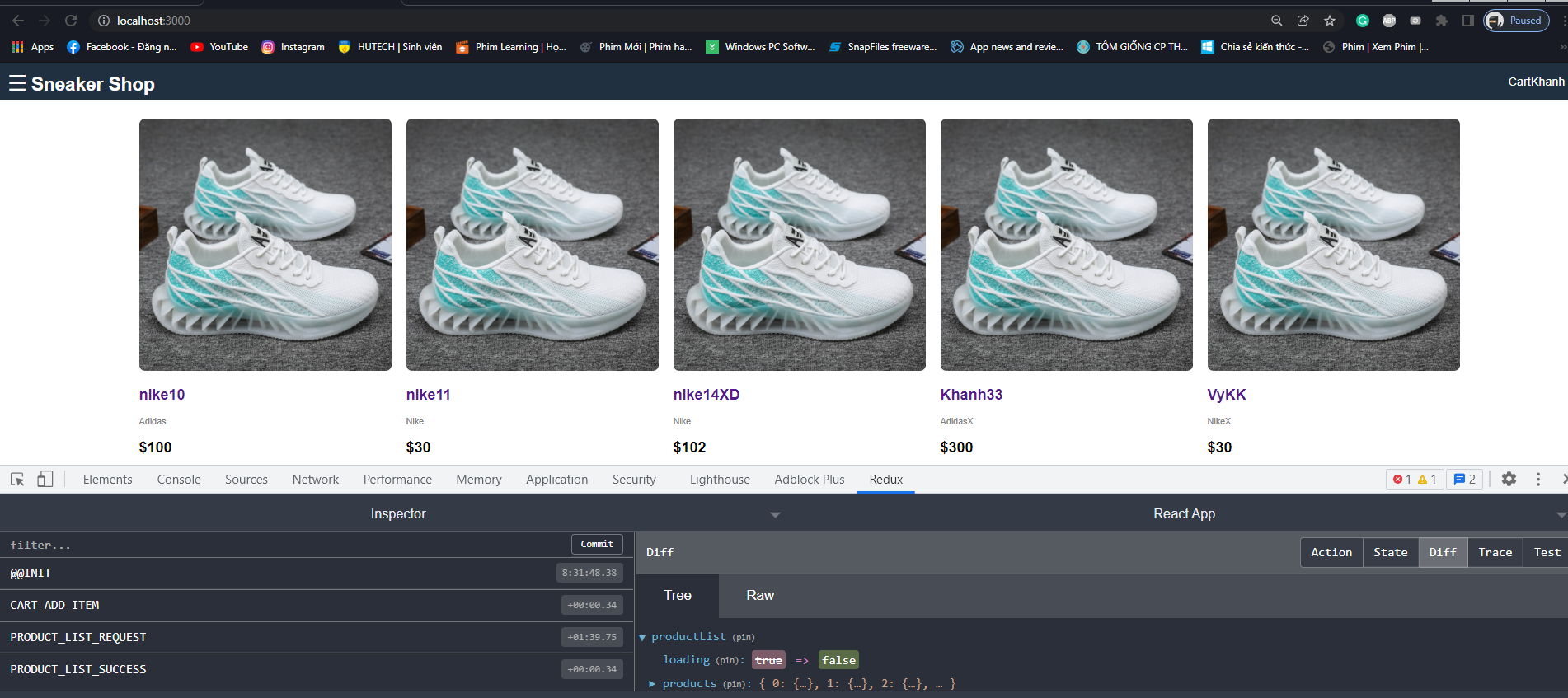
        res.send("Error in Deletion.");

    }

});

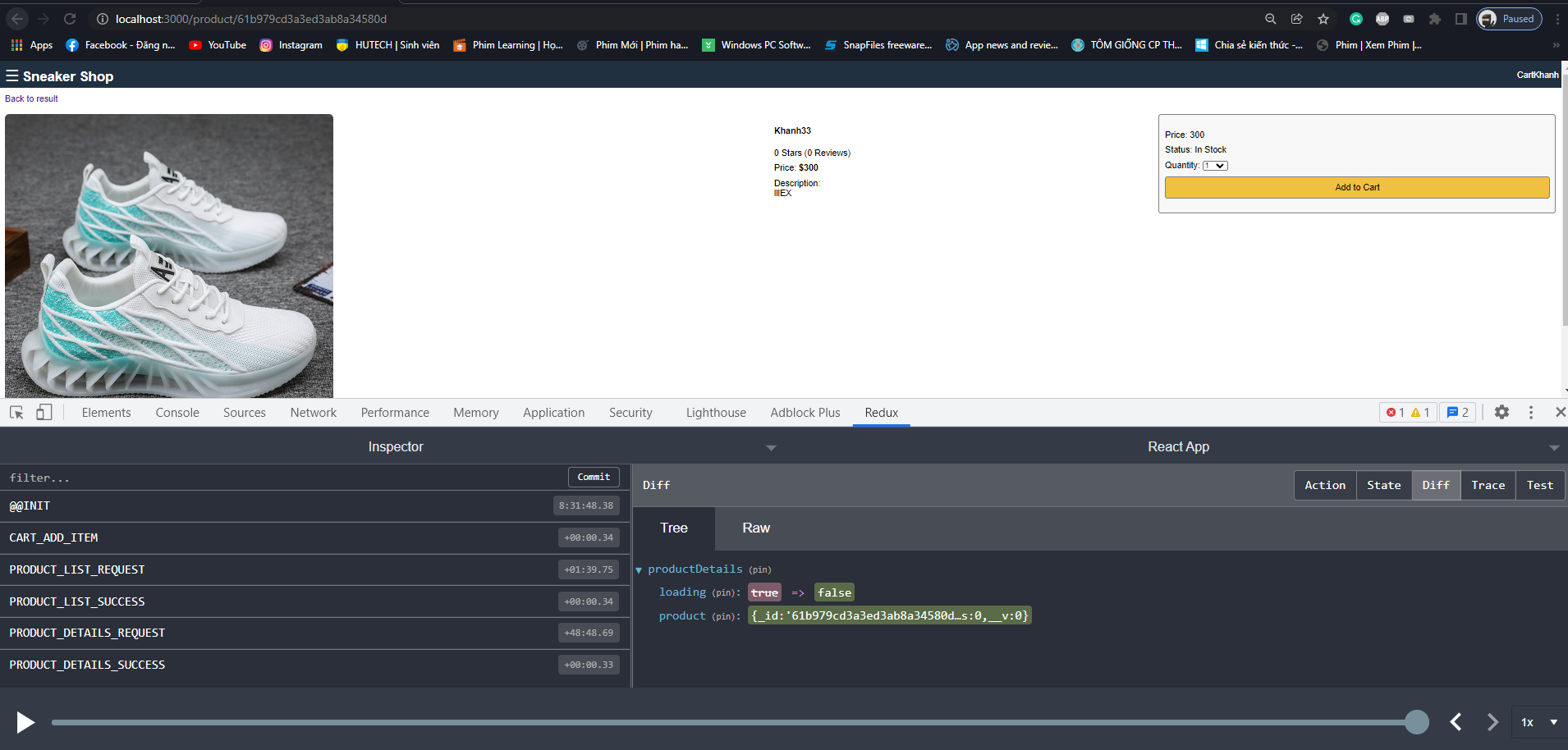
export default router;

**- Result: to checking ... 4:40:50 (Signed)**



Click on any product

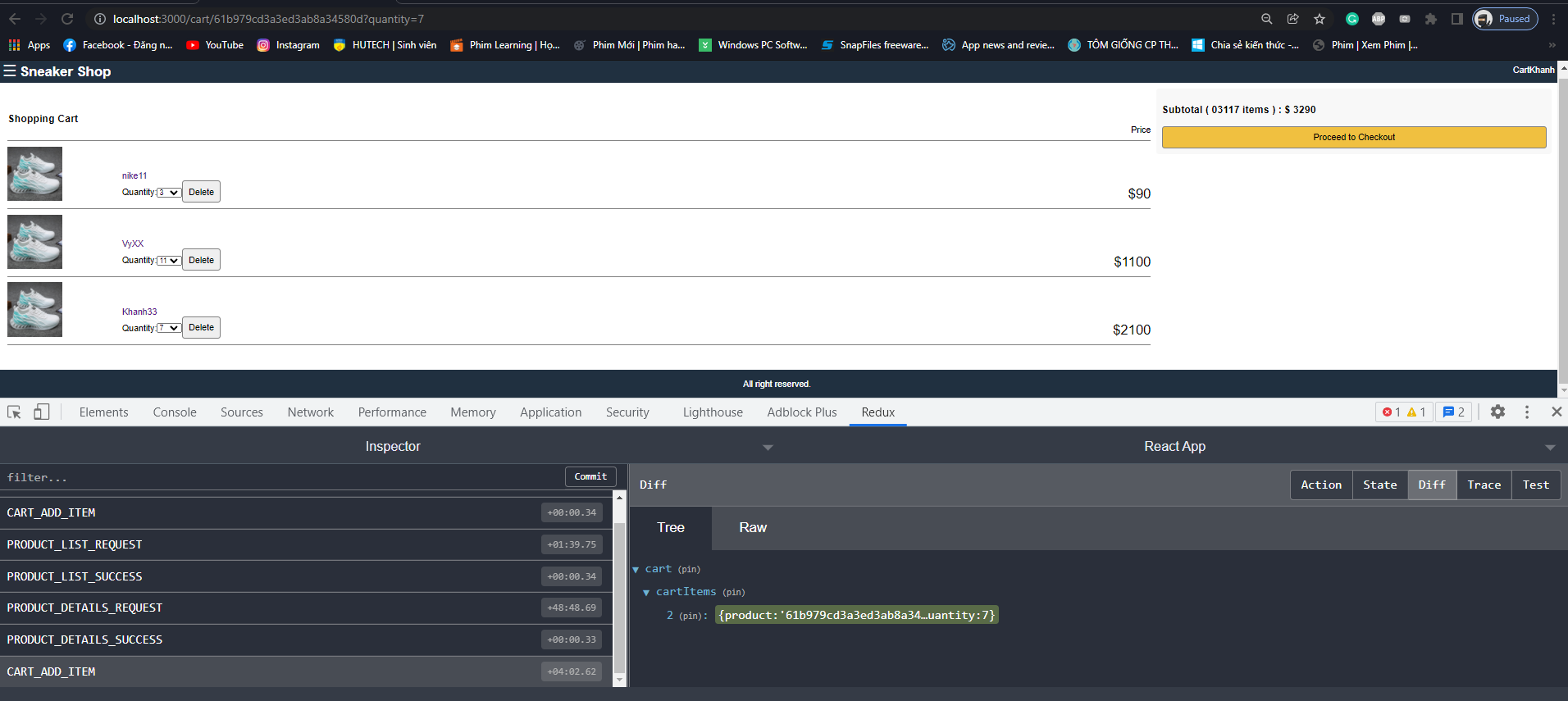
**Image 1: Step 1**



This is Product’s details

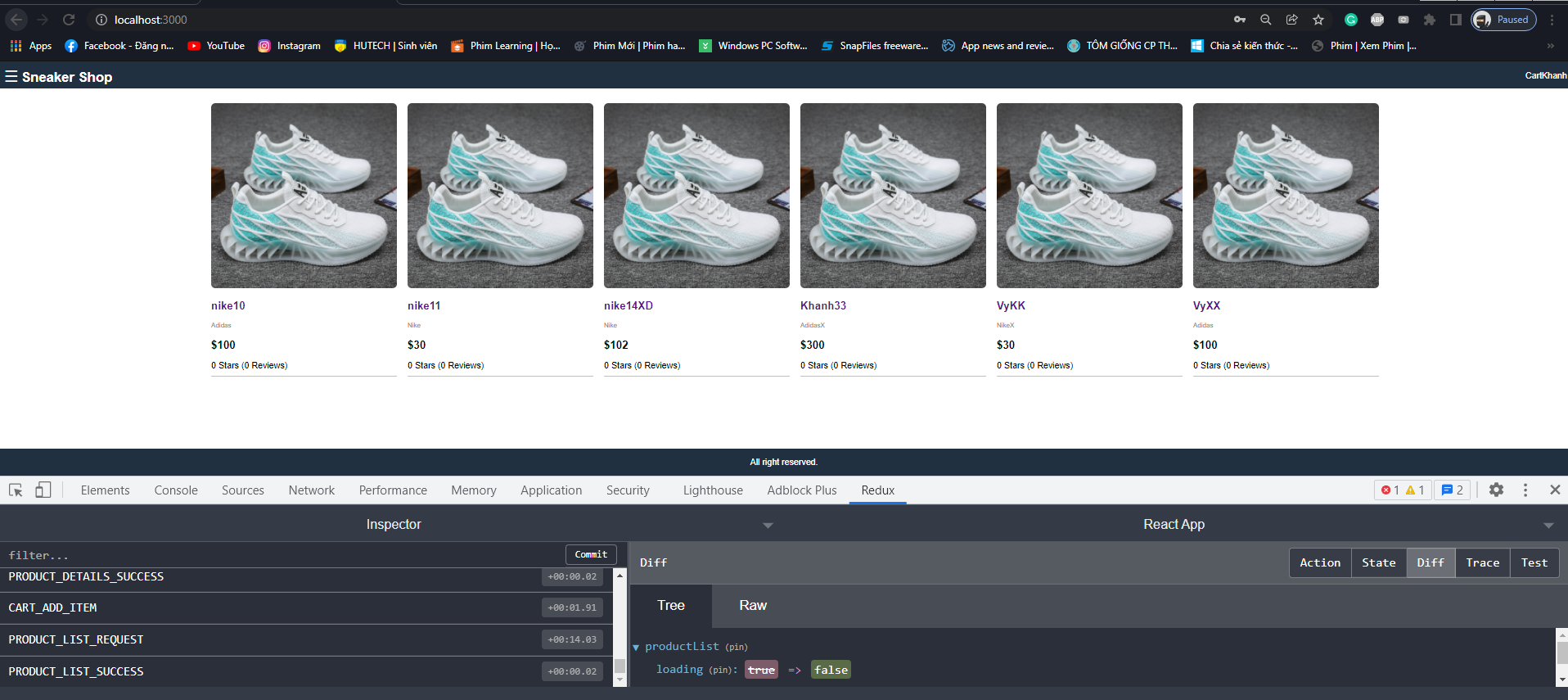
Select quantity and click “Add to Cart” button

**Image 2: Step 2**



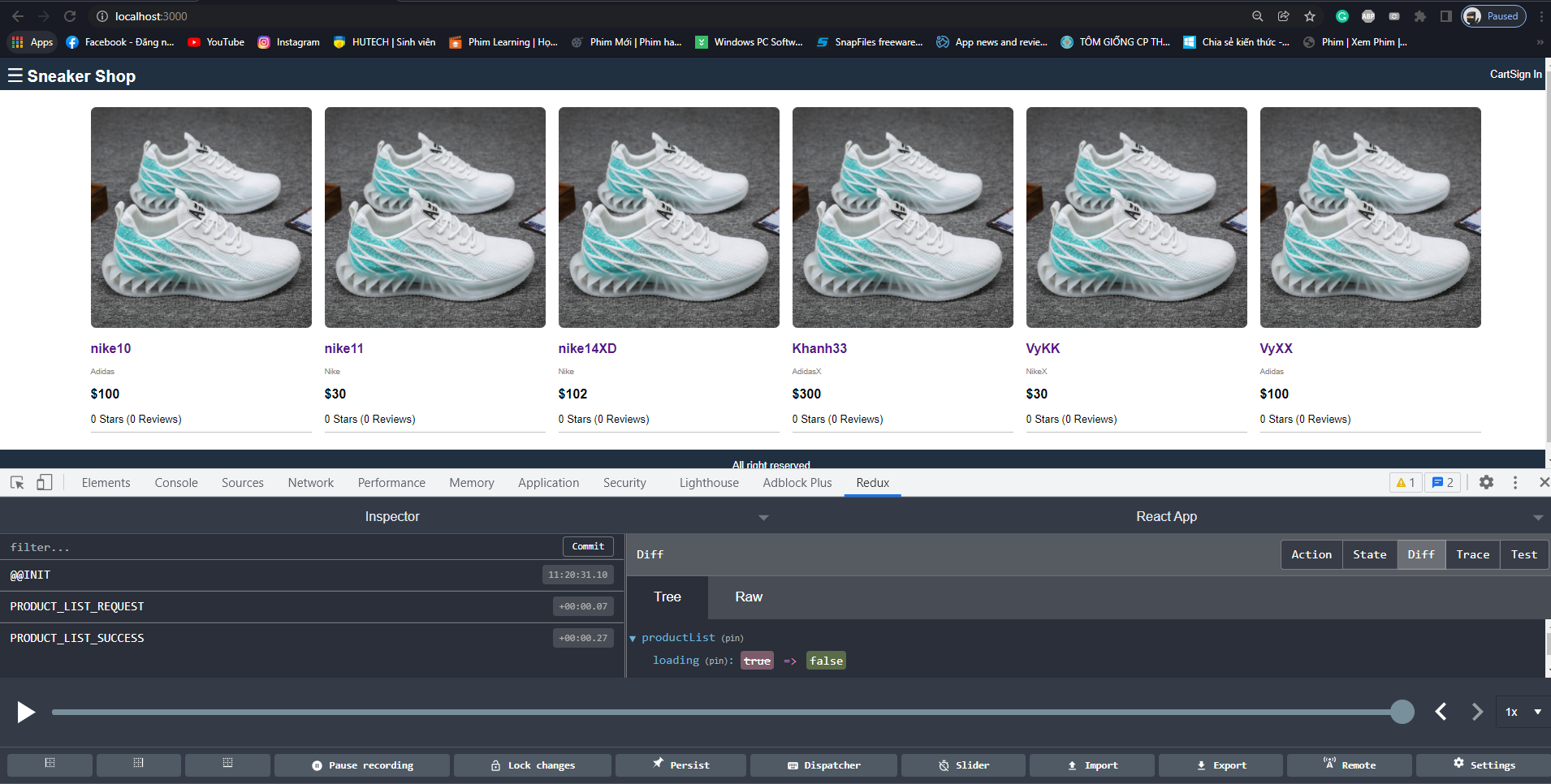
Press “Process to Checkout” button

**Image 3: Step 3**



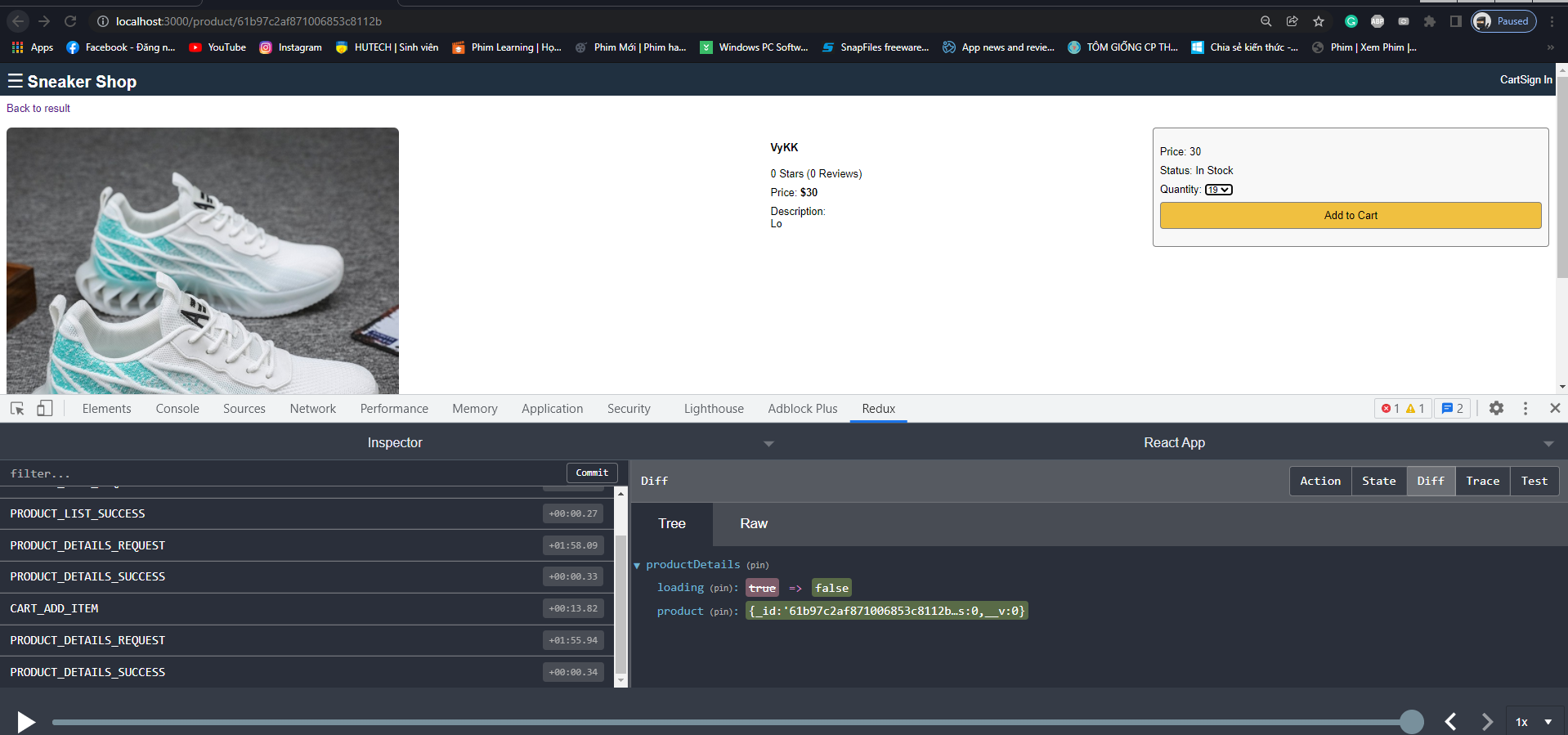
**Image 4: Result**

**- Result: to checking ... 4:40:50 (Not Signed in yet)**



Click on any product

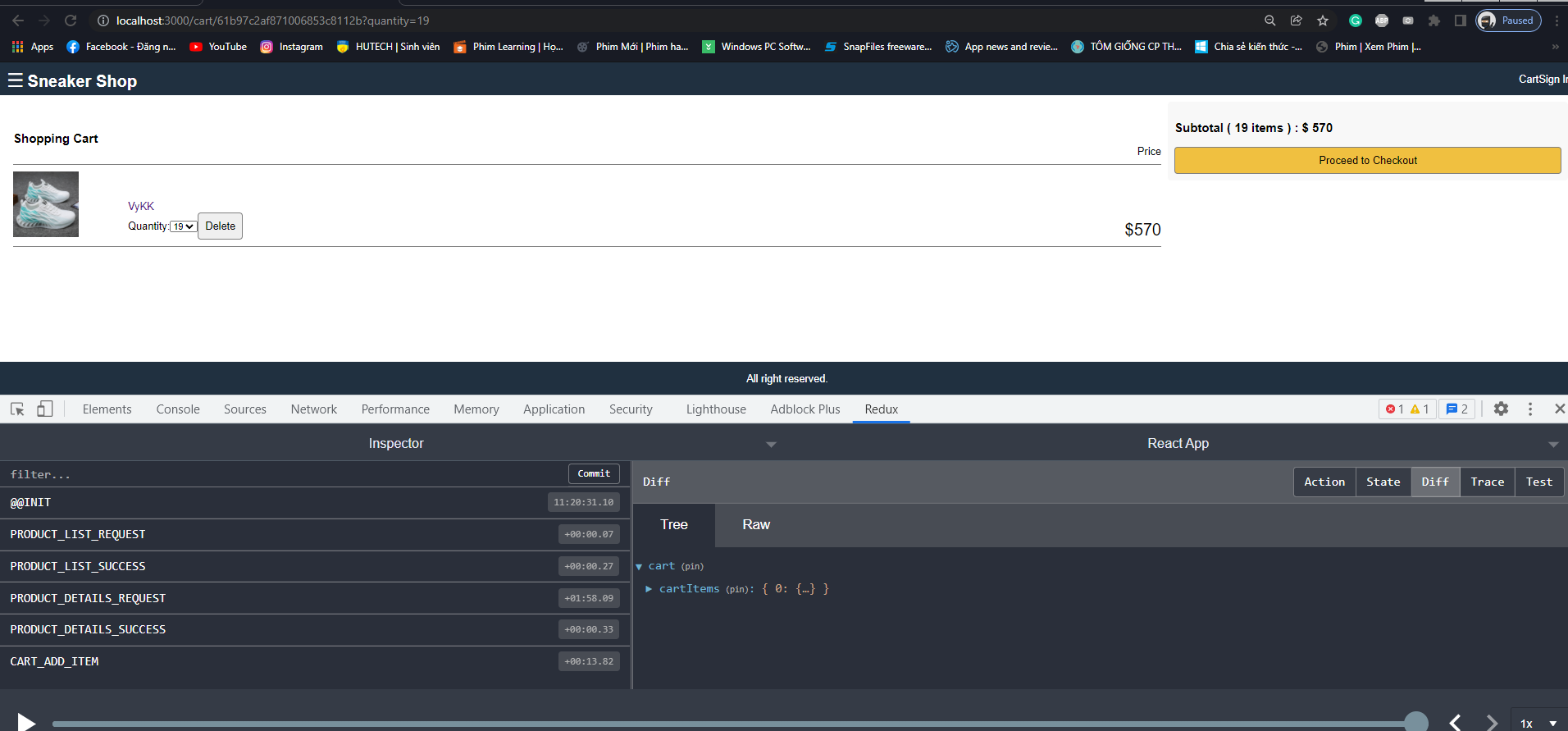
**Image 1: Step 1**



This is Product’s details

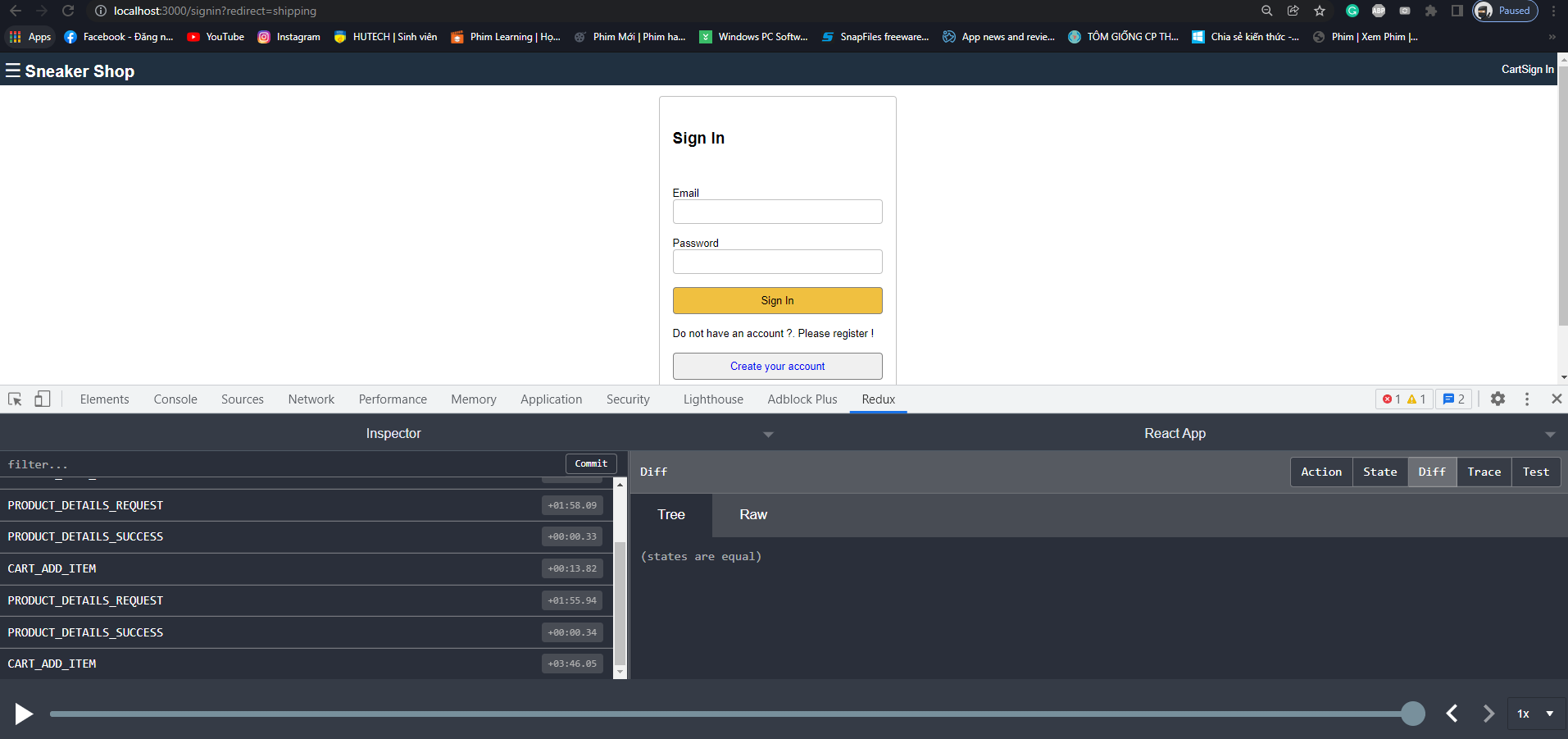
Select quantity and click “Add to Cart” button

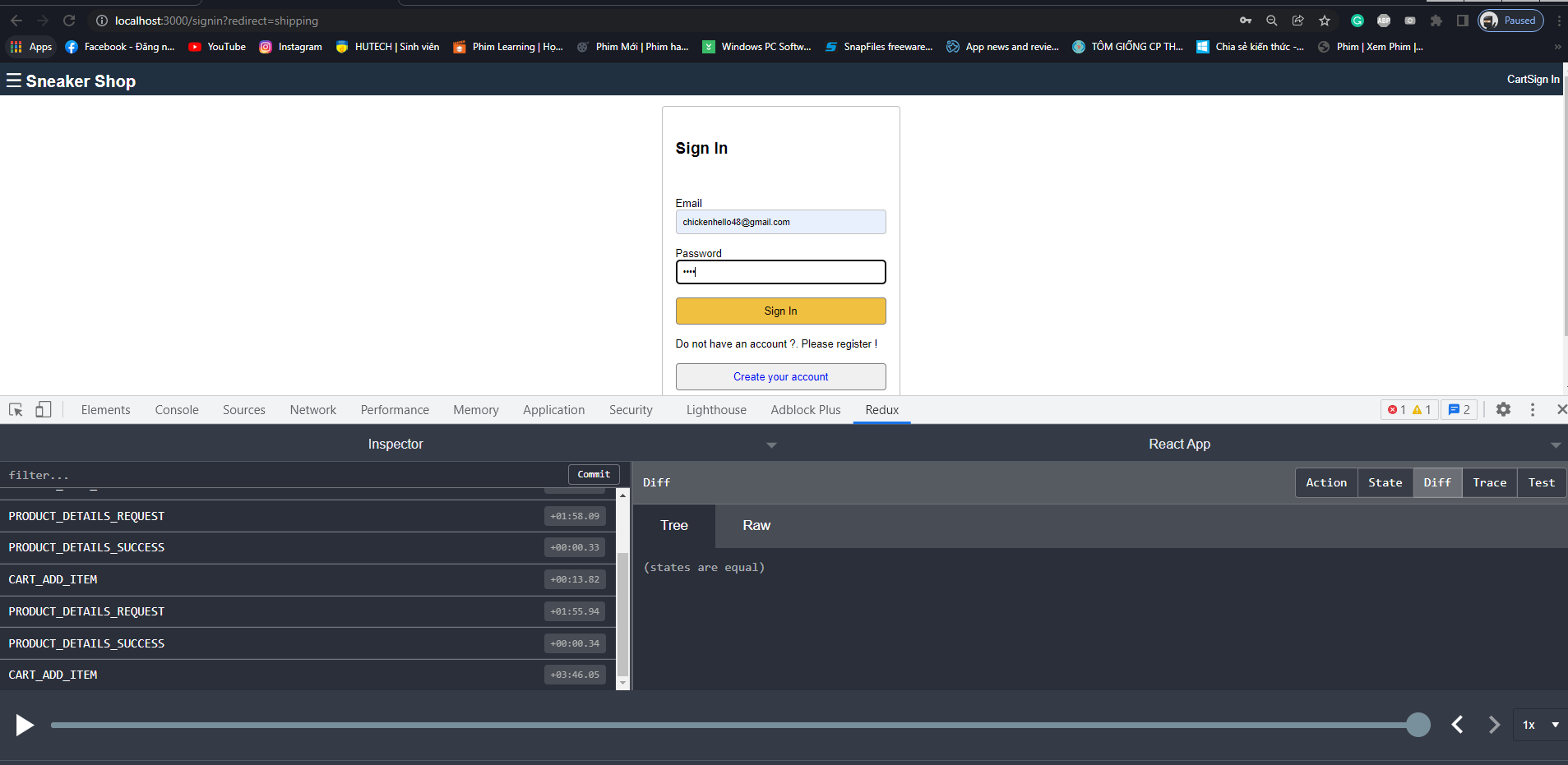
**Image 2: Step 2**



Press “Process to Checkout” button

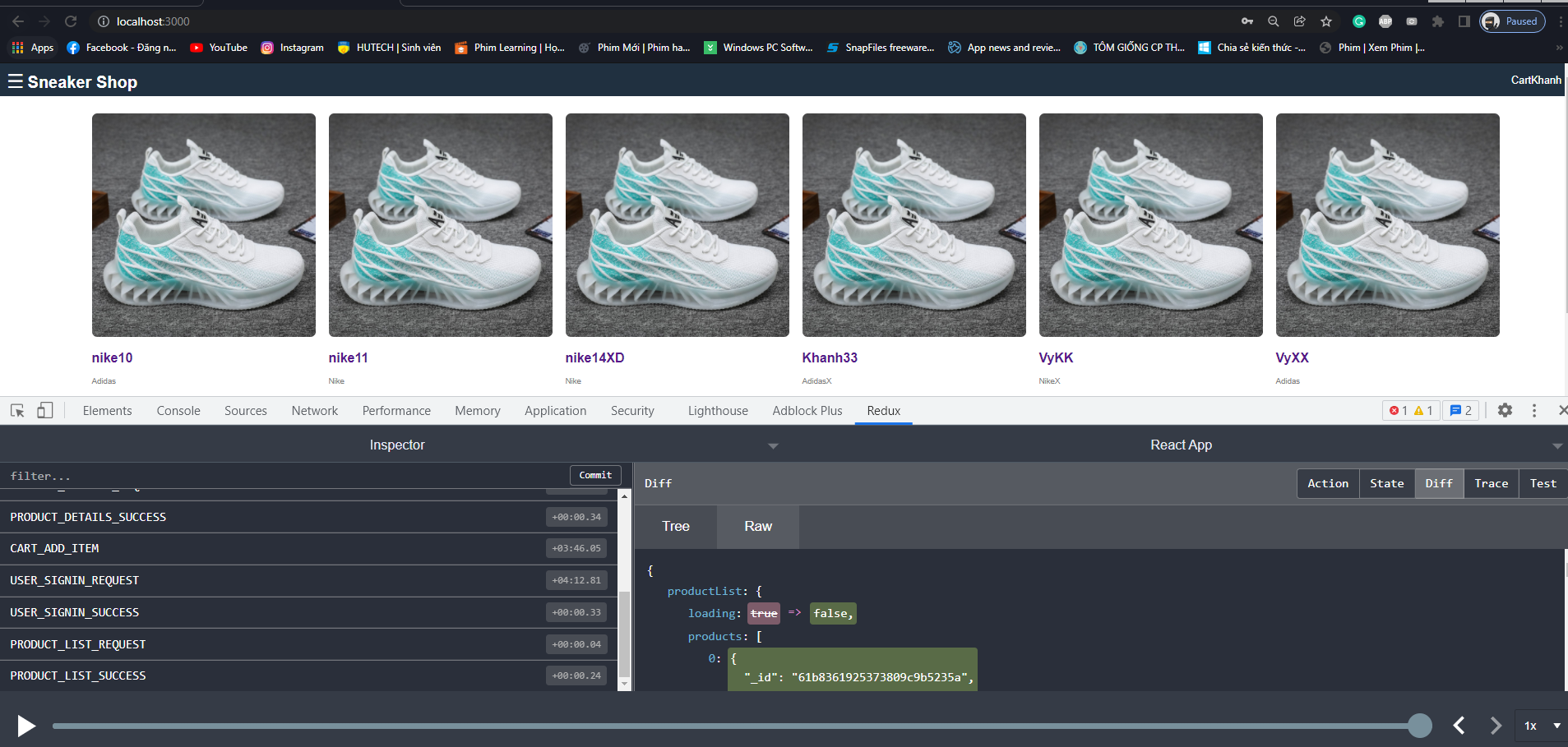
**Image 3: Step 3**





Filled in the “Sign In” form and press “Sign In” button

**Image 4,5: Step 4**



**Image 6: Result**

**- In “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/screens/SigninScreen.js” file, update the following code:**

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

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

import { useSelector, useDispatch } from 'react-redux';

import { signin } from '../actions/userActions';

function SigninScreen (props) {

    const [email, setEmail] = useState('');

    const [password, setPassword] = useState('');

    const userSignin = useSelector(state => state.userSignin);

    const { loading, userInfo, error } = userSignin;

    const dispatch = useDispatch();

    const redirect = props.location.search?props.location.search.split("=")[1]:'/';

    useEffect(() => {

        if(userInfo) {

            props.history.push(redirect);

        }

        return () => {

            //

        };

    }, [userInfo]);

    const submitHandler = (e) => {

        e.preventDefault();

        dispatch(signin(email, password));

    }

    return <div className="form">

        <form onSubmit={submitHandler}>

            <ul className="form-container">

                <li>

                    <h2>Sign In</h2>

                </li>

                <li>

                    {loading && <div>Loading...</div>}

                    {error && <div>{error}</div>}

                </li>

                <li>

                    <label htmlFor="email">Email</label>

                    <input type="email" name="email" id="email" onChange={(e) => setEmail(e.target.value)}>

                    </input>

                </li>

                <li>

                    <label htmlFor="password">Password</label>

                    <input type="password" name="password" id="password" onChange={(e) => setPassword(e.target.value)}>

                    </input>

                </li>

                <li>

                    <button type="submit" className="button primary">Sign In</button>

                </li>

                <li>

                    Do not have an account ?. Please register !

                </li>

                <li>

                    <Link to={redirect === "/" ? "register" : "register?redirect=" + redirect}

className="button secondary text-center">Create your account</Link>

                </li>

            </ul>

        </form>

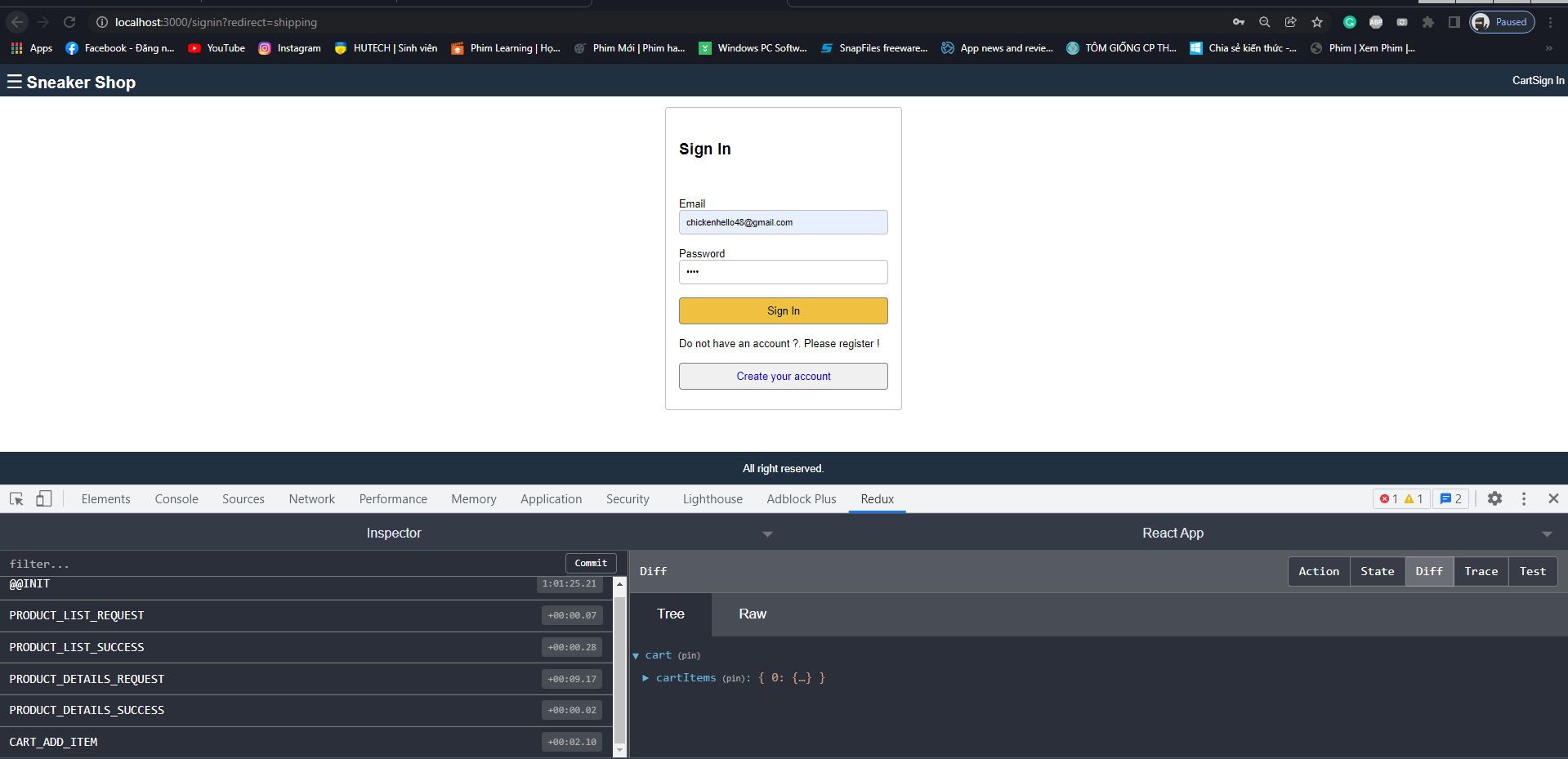
    </div>

}

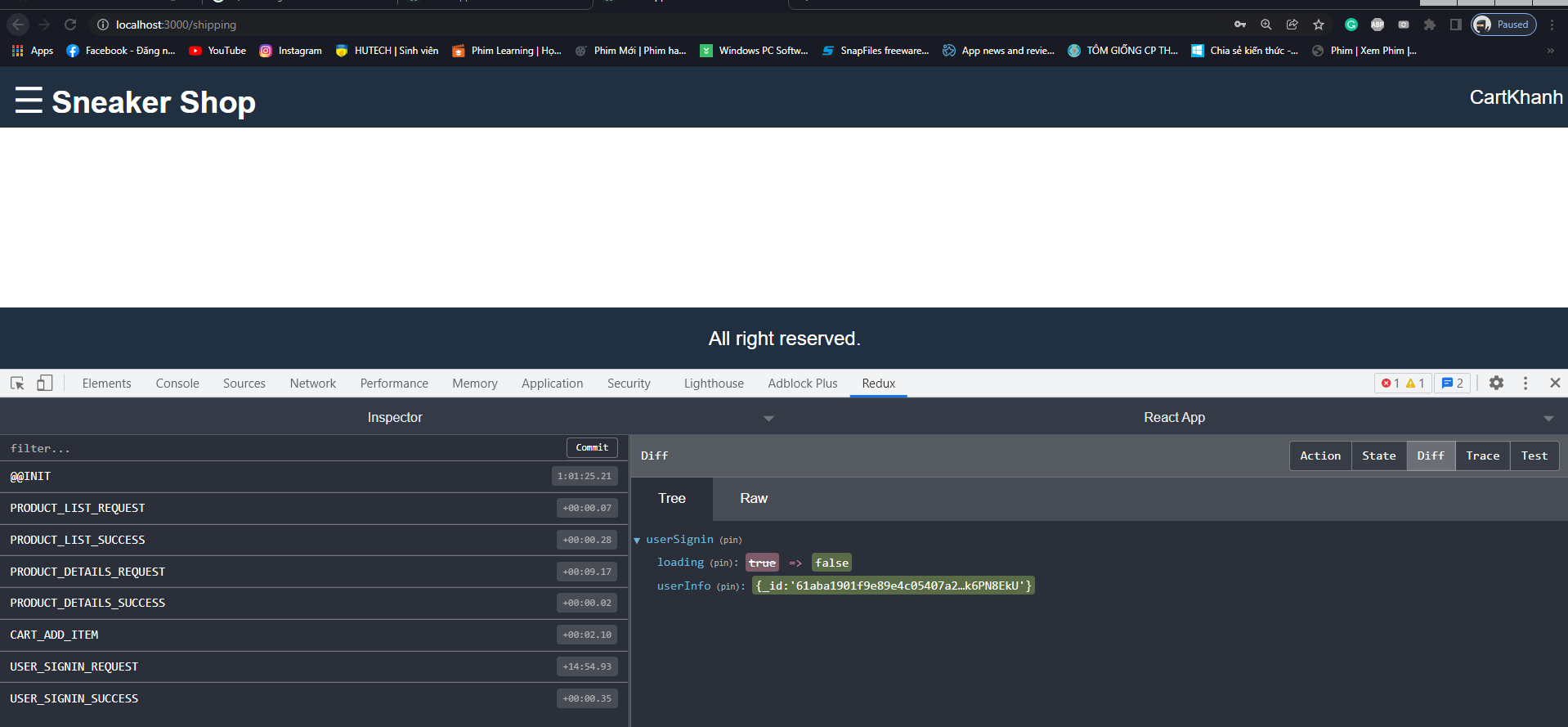
export default SigninScreen;

**- Result: to checking ... 4:41:53**

**Khi Users chưa Signed In thì ta vẫn chọn các Product và thêm vào Cart như bình thường, nhưng khi nhấn nút Checkout thì nó bắt Sign In, sau khi Signed thành công thì nó sẽ chuyển sang trang ShippingPage thay vì là HomePage như lúc trước:**



**Image 1: Steps**



**Image 2: Result**

**- In “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/screens/RegisterScreen.js” file, update the following code:**

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

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

import { useSelector, useDispatch } from 'react-redux';

import { register } from '../actions/userActions';

function RegisterScreen (props) {

    const [name, setName] = useState('');

    const [email, setEmail] = useState('');

    const [password, setPassword] = useState('');

    const [rePassword, setRePassword] = useState('');

    const userRegister = useSelector(state => state.userRegister);

    const { loading, userInfo, error } = userRegister;

    const dispatch = useDispatch();

    const redirect = props.location.search ? props.location.search.split("=")[1] : '/';

    useEffect(() => {

        if(userInfo) {

            props.history.push(redirect);

        }

        return () => {

            //

        };

    }, [userInfo]);

    const submitHandler = (e) => {

        e.preventDefault();

        dispatch(register(name, email, password));

    }

    return <div className="form">

        <form onSubmit={submitHandler}>

            <ul className="form-container">

                <li>

                    <h2>Create Account</h2>

                </li>

                <li>

                    {loading && <div>Loading...</div>}

                    {error && <div>{error}</div>}

                </li>

                <li>

                    <label htmlFor="name">Name</label>

                    <input type="name" name="name" id="name" onChange={(e) => setName(e.target.value)}>

                    </input>

                </li>

                <li>

                    <label htmlFor="email">Email</label>

                    <input type="email" name="email" id="email" onChange={(e) => setEmail(e.target.value)}>

                    </input>

                </li>

                <li>

                    <label htmlFor="password">Password</label>

                    <input type="password" name="password" id="password" onChange={(e) => setPassword(e.target.value)}>

                    </input>

                </li>

                <li>

                    <label htmlFor="rePassword">Re-Enter Password</label>

                    <input type="password" name="rePassword" id="rePassword" onChange={(e) => setRePassword(e.target.value)}>

                    </input>

                </li>

                <li>

                    <button type="submit" className="button primary">Register</button>

                </li>

                <li>

                    Already have an account ?

                    <Link to={redirect === "/" ? "signin" : "signin?redirect=" + redirect} className="button secondary text-center" >Create your amazona account</Link>

                </li>

            </ul>

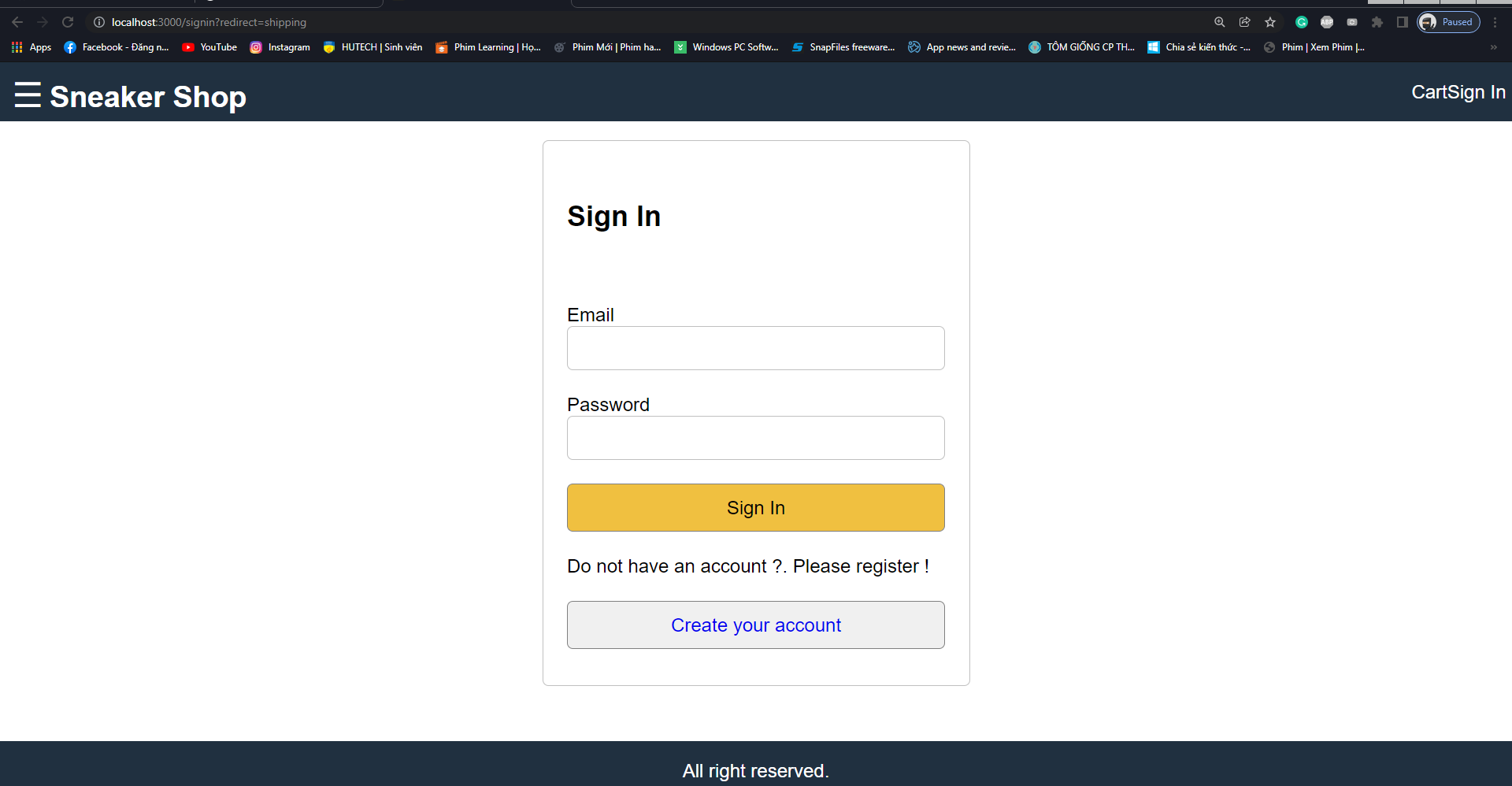
        </form>

    </div>

}

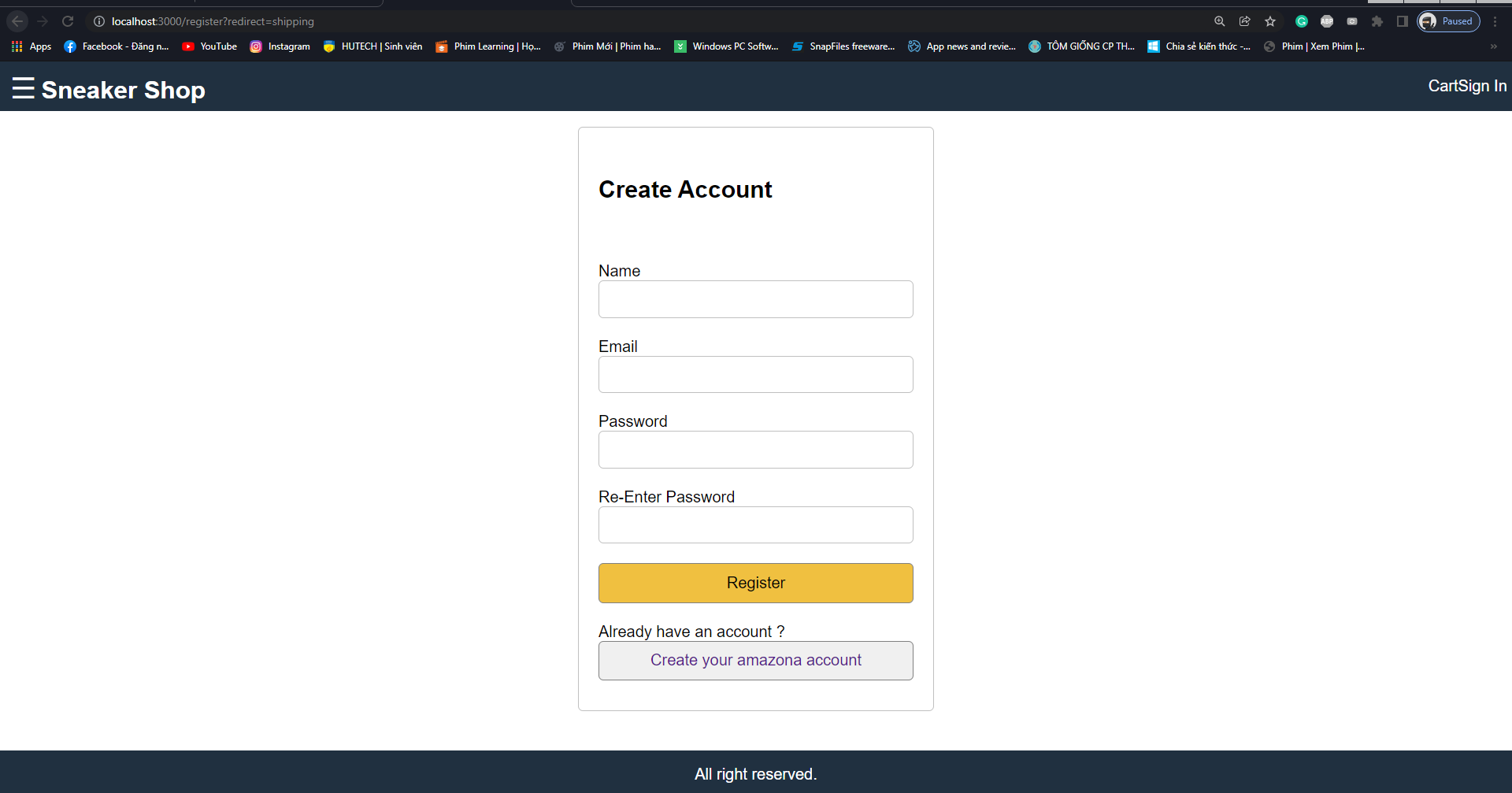
export default RegisterScreen;

**- Result: to checking ... 4:43:22**

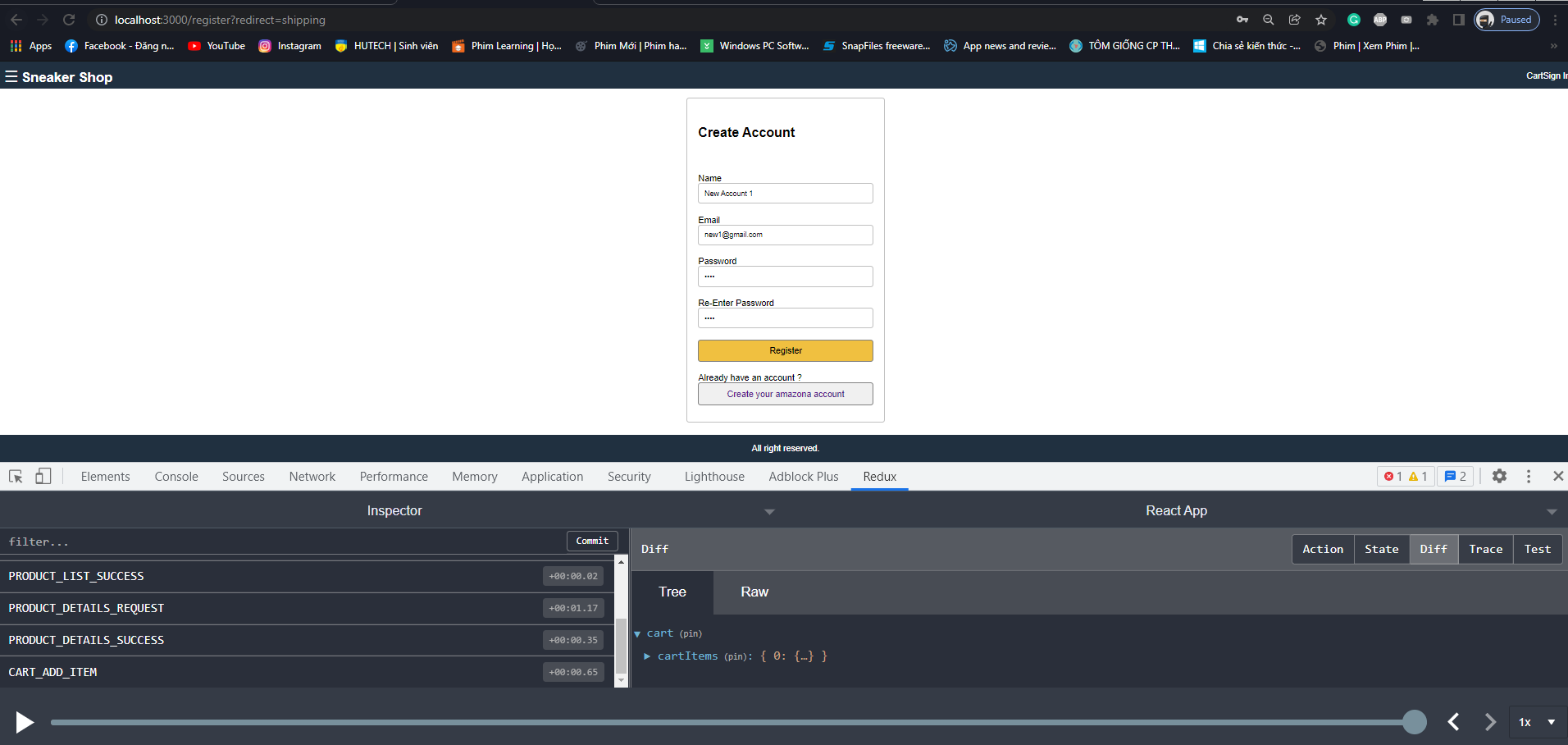


Press “Create your account” button

**Image 1: Step 1**

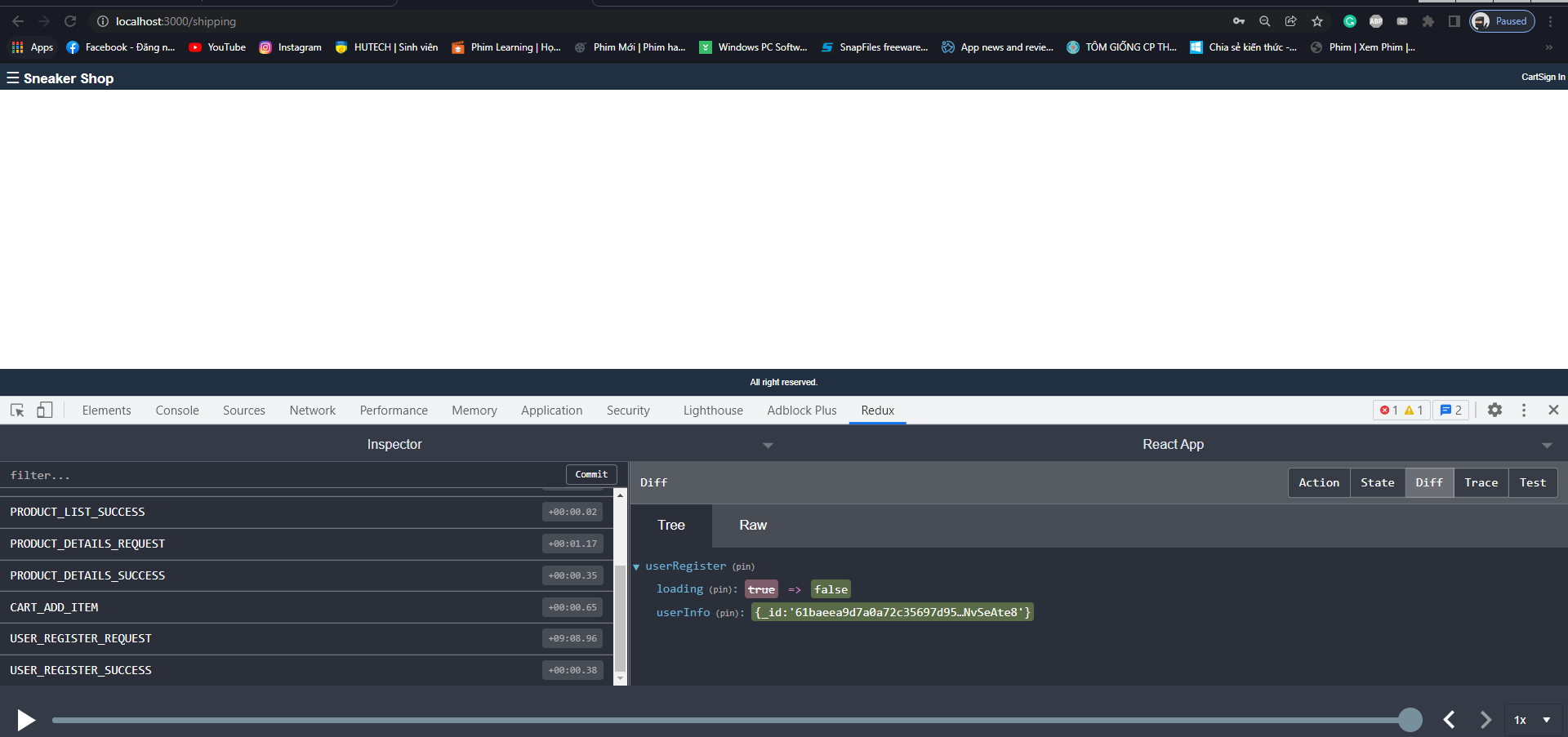


**Image 2: Step 2**



Filled the new account in Register Form and press “Register” button

**Image 3: Step 3**



**Image 4: Result**

**- Create “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/screens/ShippingScreen.js” file, add the following code:**

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

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

import { useSelector, useDispatch } from 'react-redux';

import { saveShipping } from '../actions/cartActions';

function ShippingScreen (props) {

    const [address, setAddress] = useState('');

    const [city, setCity] = useState('');

    const [postalCode, setPostalCode] = useState('');

    const [country, setCountry] = useState('');

    const dispatch = useDispatch();

    const submitHandler = (e) => {

        e.preventDefault();

        dispatch(saveShipping(address, city, postalCode, country));

    }

    return <div className="form">

        <form onSubmit={submitHandler}>

            <ul className="form-container">

                <li>

                    <h2>Shipping</h2>

                </li>

                <li>

                    <label htmlFor="address">Address</label>

                    <input type="text" name="address" id="address" onChange={(e) => setAddress(e.target.value)}>

                    </input>

                </li>

                <li>

                    <label htmlFor="city">City</label>

                    <input type="text" name="city" id="city" onChange={(e) => setCity(e.target.value)}>

                    </input>

                </li>

                <li>

                    <label htmlFor="postalCode">PostalCode</label>

                    <input type="text" name="postalCode" id="postalCode" onChange={(e) => setPostalCode(e.target.value)}>

                    </input>

                </li>

                <li>

                    <label htmlFor="country">Country</label>

                    <input type="text" name="country" id="country" onChange={(e) => setCountry(e.target.value)}>

                    </input>

                </li>

                <li>

                    <button type="submit" className="button primary">Continue</button>

                </li>

            </ul>

        </form>

    </div>

}

export default ShippingScreen;

**- In “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/actions/cartActions.js” file, update the following code:**

import Axios from "axios";

// import Cookies from "js-cookie";

import { CART\_ADD\_ITEM, CART\_REMOVE\_ITEM, CART\_SAVE\_SHIPPING } from "../constants/cartConstants";

// Nếu muốn lưu với Cookie (Server Side)

const Cookies = require('js-cookie');

// Add product into the Cart

const addToCart = (productId, quantity) => async (dispatch, getState) => {

    try {

        const { data } = await Axios.get("/api/products/" + productId);

        dispatch({

            type: CART\_ADD\_ITEM, payload: {

                product: data.\_id,

                name: data.name,

                image: data.image,

                price: data.price,

                countInStock: data.countInStock,

                quantity

            }

        });

        // Nếu muốn lưu với Cookie (Server Side)

        const { cart: { cartItems } } = getState();

        Cookies.set("cartItems", JSON.stringify(cartItems));

        // Nếu muốn lưu với Local Storage (Client Side)

        // localStorage.setItem('cartItems', JSON.stringify(getState().cart.cartItems));

    } catch (error) {

    }

}

// Remove product from cart

const removeFromCart = (productId) => (dispatch, getState) => {

    dispatch({type: CART\_REMOVE\_ITEM, payload: productId});

    // Nếu muốn lưu với Cookie (Server Side)

    const { cart: { cartItems } } = getState();

    Cookies.set("cartItems", JSON.stringify(cartItems));

    // Nếu muốn lưu với Local Storage (Client Side)

    // localStorage.setItem('cartItems', JSON.stringify(getState().cart.cartItems));

}

// Save shipping

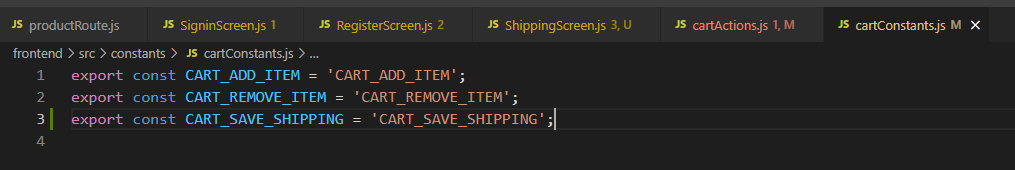
const saveShipping = (data) => (dispatch) => {

    dispatch({ type: CART\_SAVE\_SHIPPING, payload: data });

}

export { addToCart, removeFromCart, saveShipping }

**- In “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/constants/cartConstants.js” file, update the following code:**



**- In “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/reducers/cartReducers.js” file, update the following code:**

import { CART\_ADD\_ITEM, CART\_REMOVE\_ITEM, CART\_SAVE\_SHIPPING } from "../constants/cartConstants";

function cartReducer(state={cartItems:[]}, action) {

    switch(action.type) {

        case CART\_ADD\_ITEM:

            const item = action.payload;

            const product = state.cartItems.find(x=> x.product === item.product);

            if(product){

                return { cartItems:

                    state.cartItems.map(x=> x.product === product.product ? item : x)

                };

            }

            return { cartItems: [...state.cartItems, item] };

        case CART\_REMOVE\_ITEM:

            return{cartItems: state.cartItems.filter(x => x.product !== action.payload)}

        case CART\_SAVE\_SHIPPING:

            return {...state, shipping: action.payload}

        default:

            return state

    }

}

export { cartReducer }

**- In “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/App.js” file, update the following code:**

import React from 'react';

import { useSelector } from 'react-redux';

import {BrowserRouter, Route, Link} from 'react-router-dom';

import './App.css';

import CartScreen from './screens/CartScreen';

import HomeScreen from './screens/HomeScreen';

import ProductScreen from './screens/ProductScreen';

import ProductsScreen from './screens/ProductsScreen';

import RegisterScreen from './screens/RegisterScreen';

import ShippingScreen from './screens/ShippingScreen';

import SigninScreen from './screens/SigninScreen';

function App() {

  const userSignin = useSelector( state => state.userSignin);

  const { userInfo } = userSignin;

  const openMenu = () => {

    document.querySelector(".sidebar").classList.add("open");

  }

  const closeMenu = () => {

    document.querySelector(".sidebar").classList.remove("open");

  }

  return (

    <BrowserRouter>

    <div className="grid-container">

        <header className="header">

            <div className="brand">

                <button onClick={openMenu}>

                    &#9776;

                </button>

                <Link to="/" >Sneaker Shop</Link>

            </div>

            <div className="header-links">

                <a href="cart.html">Cart</a>

                {

                    userInfo ? <Link to="/profile">{userInfo.name}</Link> : <Link to="/signin">Sign In</Link>

                }

            </div>

        </header>

        <aside className="sidebar">

            <h3>Shopping Categories</h3>

            <button className="sidebar-close-button" onClick={closeMenu}>x</button>

            <ul>

                <li>

                    <a href="index.html">Man</a>

                </li>

                <li>

                    <a href="index.html">Woman</a>

                </li>

            </ul>

        </aside>

        <main className="main">

            <div className="content">

                <Route path="/products" component={ProductsScreen} />

                <Route path="/shipping" component={ShippingScreen} />

                <Route path="/signin" component={SigninScreen} />

                <Route path="/register" component={RegisterScreen} />

                <Route path="/product/:id" component={ProductScreen} />

                <Route path="/cart/:id?" component={CartScreen} />

                <Route path="/" exact={true} component={HomeScreen} />

            </div>

        </main>

        <footer className="footer">

            All right reserved.

        </footer>

    </div>

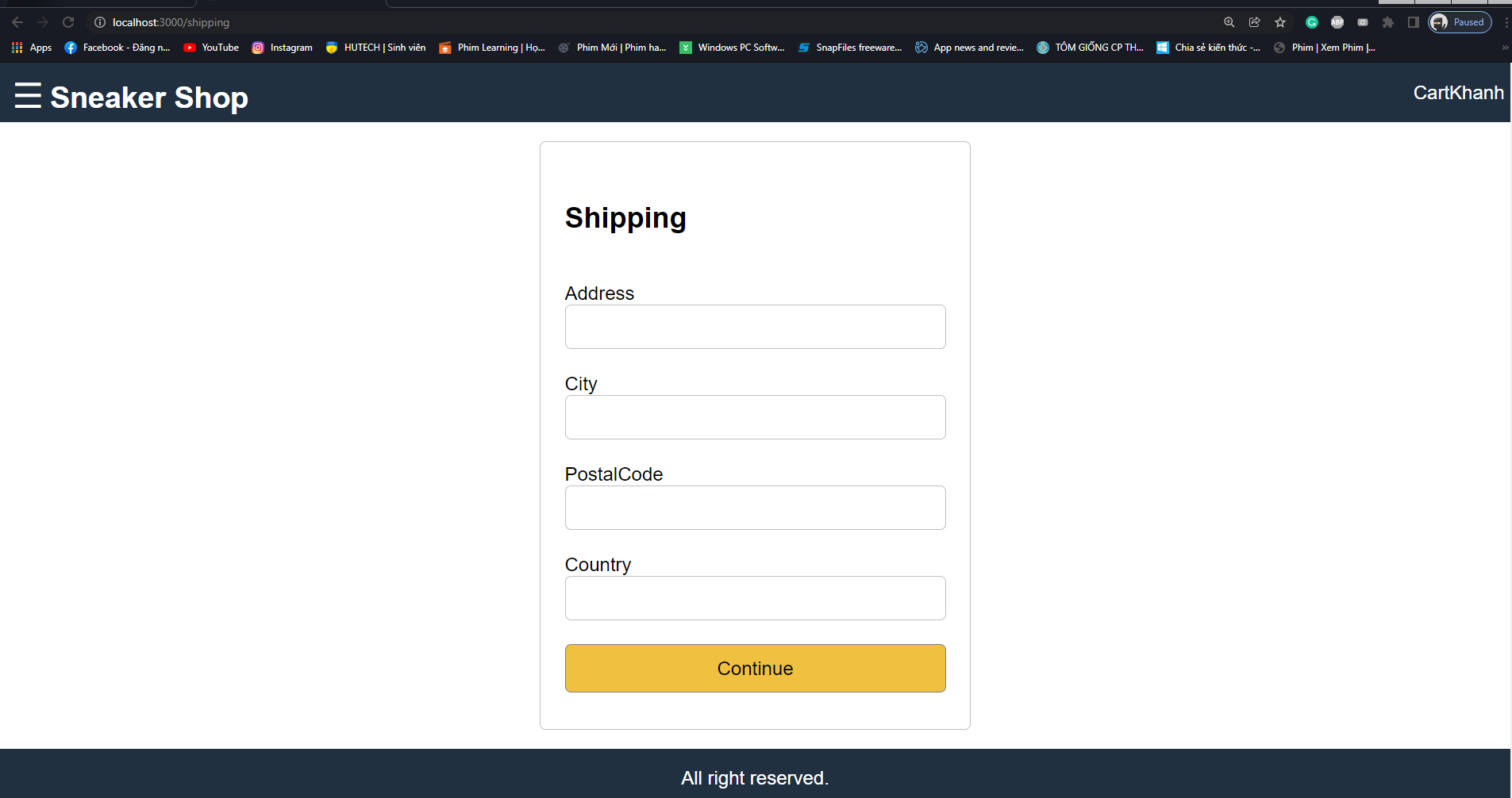
    </BrowserRouter>

  );

}

export default App;

**- Result: to checking ... 4:48:00**



**- Create “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/actions/userActions.js” file, add the following code:**

**- Create “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/actions/userActions.js” file, add the following code:**

**(Error 3 ... 3:55:26) Nếu làm như trong video mà tại lúc test đăng ký, khi nhập hết các field và nhấn nút đăng ký mà nó bắn ra lỗi 500.**

**(Solution (Error 3)):**

**- Result: to checking ... 3:55:26**

2