Web App Instructions

Running a React.js and Node.js Project Locally

Introduction

This guide will walk you through the steps to run a React.js frontend and a Node.js backend project locally on both Windows and macOS systems. The project is structured with React.js code in the `client` folder and Node.js code in the `server` folder.

Prerequisites

Before we begin, make sure you have the following prerequisites installed on your system:

- **Node.js**: Download and install Node.js from nodejs.org.

Setting Up the Server (Node.js)

1. Open a terminal.

2. Navigate to the `server` directory in your project:

```
cd /path/to/your/project/server
```

3. Install the server dependencies using npm:

```
npm install
```

4. Start the server:

```
npm start
```

Setting Up the Client (React.js)

1. Open a new terminal window.

2. Navigate to the `client` directory in your project:

```
cd /path/to/your/project/client
```

3. Install the client dependencies using npm:

```
npm install
```

4. Start the React development server:

```bash
npm start
```

Accessing the Application

After following the above steps, your React.js app should be running on `http://localhost:3000`, and your Node.js server should be running on `http://localhost:8000`. You can access the application by opening a web browser and visiting `http://localhost:3000`.

Troubleshooting

If you encounter any issues, here are some common troubleshooting steps:

- Ensure that you are using the correct paths when navigating to the `client` and `server` directories.

- Check that you have correctly installed the required dependencies using `npm install`.

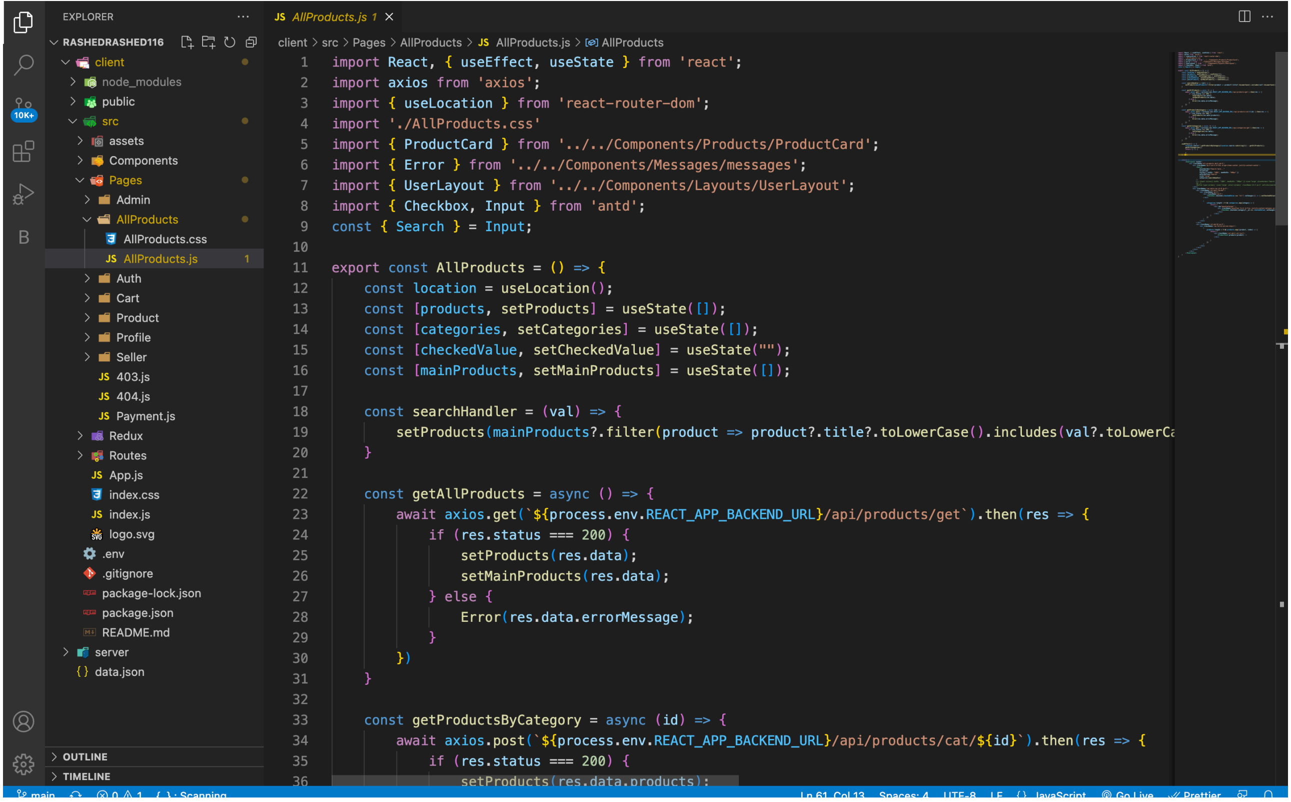
- Verify that no other processes are already using the required ports (3000 for React and 5000 for Node.js).

- Look for error messages in your terminal to identify the specific issue.

By following this guide, you should be able to run your React.js and Node.js project locally on both Windows and macOS systems.

If you want to make any changes:

I have made the project files structure according to the pages names. For example, All Products folder has the code for all products list page. Then each page has its own components in components folder. You can see their path at the top of that code file.



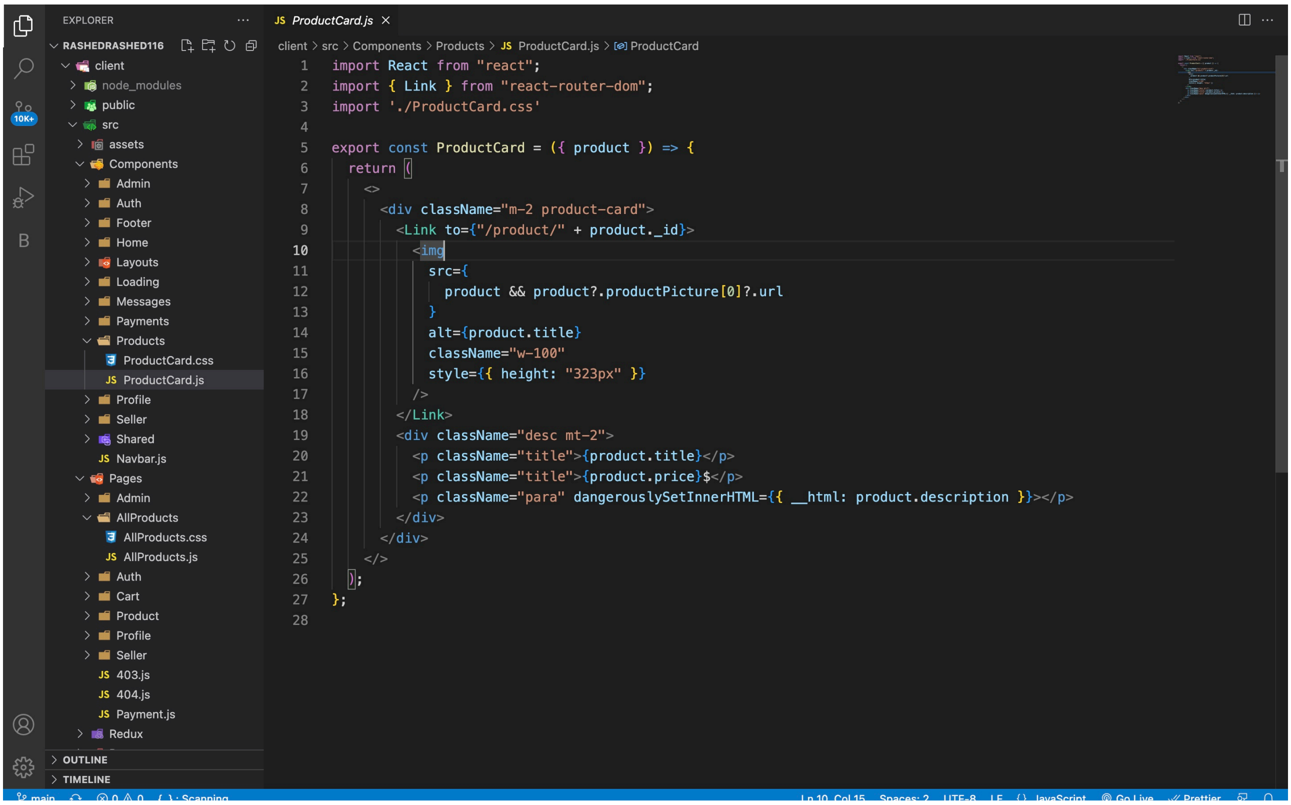
```
import React, { useEffect, useState } from 'react';
import axios from 'axios';
import { useLocation } from 'react-router-dom';
import './AllProducts.css'
import { ProductCard } from '../../Components/Products/ProductCard';
import { Error } from '../../Components/Messages/messages';
import { UserLayout } from '../../Components/Layouts/UserLayout';
import { Checkbox, Input } from 'antd';
const { Search } = Input;

export const AllProducts = () => {
    const location = useLocation();
    const [products, setProducts] = useState([]);
    const [categories, setCategories] = useState([]);
    const [checkedValue, setCheckedValue] = useState("");
    const [mainProducts, setMainProducts] = useState([]);

    const searchHandler = (val) => {
        setProducts(mainProducts?.filter(product => product?.title?.toLowerCase().includes(val?.toLowerCa
    }

    const getAllProducts = async () => {
        await axios.get(`${process.env.REACT_APP_BACKEND_URL}/api/products/get`).then(res => {
            if (res.status === 200) {
                setProducts(res.data);
                setMainProducts(res.data);
            } else {
                Error(res.data.errorMessage);
            }
        })
    }

    const getProductsByCategory = async (id) => {
        await axios.post(`${process.env.REACT_APP_BACKEND_URL}/api/products/cat/${id}`).then(res => {
            if (res.status === 200) {
                setProducts(res.data.products);
```

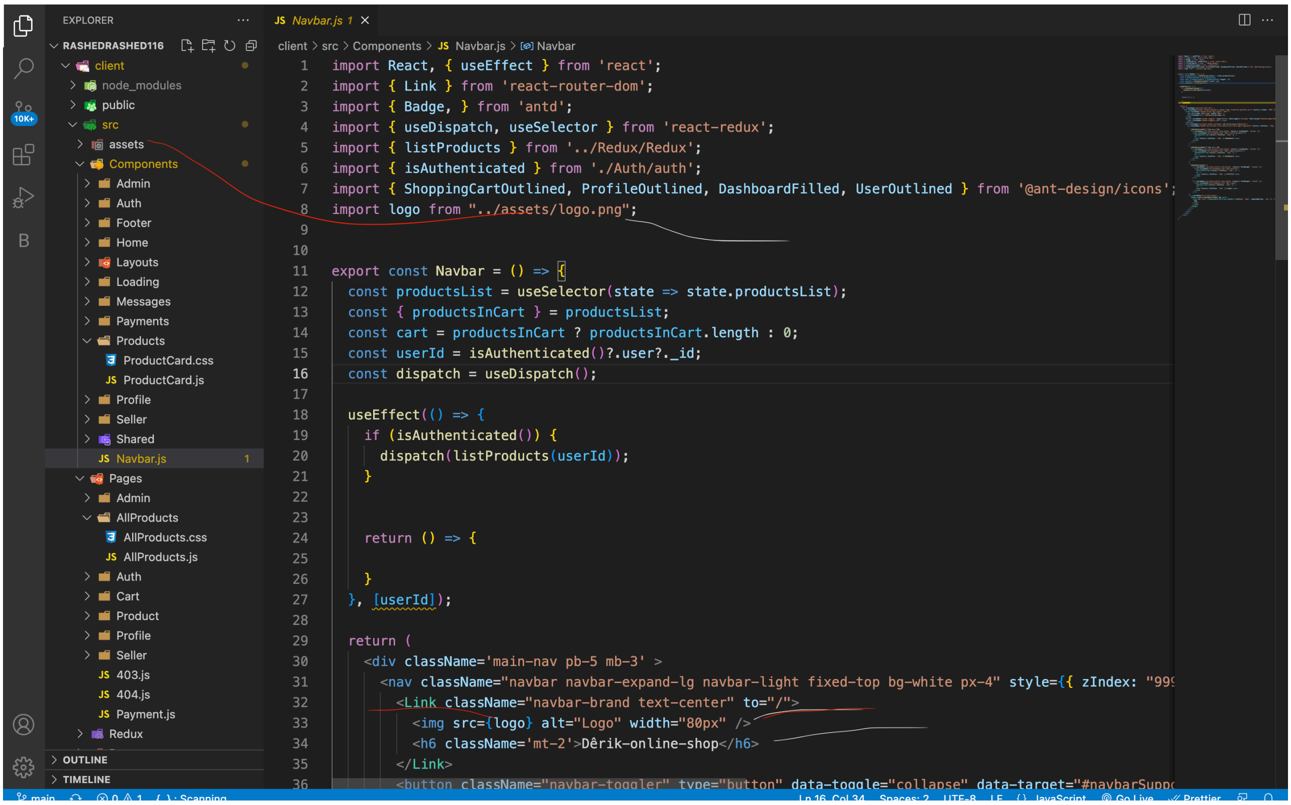


```
import React from "react";
import { Link } from "react-router-dom";
import './ProductCard.css'

export const ProductCard = ({ product }) => {
  return (
    <>
      <div className="m-2 product-card">
        <Link to={"/product/" + product._id}>
          <img
            src={
              product && product?.productPicture[0]?.url
            }
            alt={product.title}
            className="w-100"
            style={{ height: "323px" }}
          />
        </Link>
        <div className="desc mt-2">
          <p className="title">{product.title}</p>
          <p className="title">{product.price}$</p>
          <p className="para" dangerouslySetInnerHTML={{ __html: product.description }}></p>
        </div>
      </div>
    </>
  );
};

```

If you want to change logo and name:



```
import React, { useEffect } from 'react';
import { Link } from 'react-router-dom';
import { Badge, } from 'antd';
import { useDispatch, useSelector } from 'react-redux';
import { listProducts } from '../Redux/Redux';
import { isAuthenticated } from './Auth/auth';
import { ShoppingCartOutlined, ProfileOutlined, DashboardFilled, UserOutlined } from '@ant-design/icons';
import logo from "../assets/logo.png";


export const Navbar = () => {
  const productsList = useSelector(state => state.productsList);
  const { productsInCart } = productsList;
  const cart = productsInCart ? productsInCart.length : 0;
  const userId = isAuthenticated()?.user?._id;
  const dispatch = useDispatch();

  useEffect(() => {
    if (isAuthenticated()) {
      dispatch(listProducts(userId));
    }


    return () => {

    }
  }, [userId]);

  return (
    <div className='main-nav pb-5 mb-3' >
      <nav className="navbar navbar-expand-lg navbar-light fixed-top bg-white px-4" style={{ zIndex: "999
        <Link className="navbar-brand text-center" to="/">
          <img src={logo} alt="Logo" width="80px" />
          <h6 className='mt-2'>Dêrik-online-shop</h6>
        </Link>
        <button className="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSuppo
```

So this is basic info on how to update the website. To make it live you need to add code in server.