

# UNIT TEST FOR E-COMMERCE WEB APPLICATION

**PREPARED BY**

Michelle Nguyen

05/02/2024

# TABLE OF CONTENT

## 1. Overview

- 1.1 Overview
- 1.2 Warning: ReactDOM.render
- 1.3 Warning: Invalid DOM Property 'class'
- 1.4 Warning: window.scrollTo

## 2. Unit Test

- 2.1 Unit Test: account.test.js
- 2.2 Unit Test: adminDashboard.test.js
- 2.3 Unit Test: footer.test.js
- 2.4 Unit Test: home.test.js
- 2.5 Unit Test: inventory.test.js
- 2.6 Unit Test: navbar.test.js
- 2.7 Unit Test: productDetail.test.js
- 2.8 Unit Test: productListings.test.js
- 2.9 Unit Test: sales.test.js
- 2.10 Unit Test: search.test.js
- 2.11 Unit Test: store.test.js
- 2.12 Unit Test: userBar.test.js

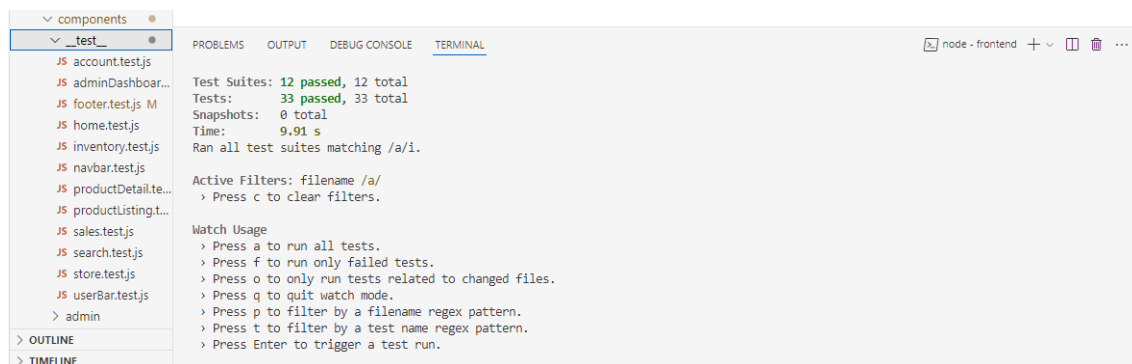
# 1. Overview

## 1.1 Overview

We're primarily focusing on rendering components to ensure they render properly. Due to our limited knowledge in unit testing, we haven't been able to perform more intricate tests that involve components relying on API calls from the backend, such as testing for proper data retrieval.

All tests were successfully executed and passed, with 33 tests passing out of 33 total, across 12 test suites.

During the test runs, there were warnings logged; however these warnings did not affect the execution or outcome of the unit tests, which all passed successfully.



## 1.2 Warning: ReactDOM.render

The error message indicates that the ReactDOM.render method, used for rendering components in React, is no longer supported in React 18. Instead, developers are advised to use the createRoot method. The warning message serves as a reminder that until the codebase is updated to use the new API (createRoot), the app will continue to behave as if it's running in React 17.

Despite this warning, the unit test in question still compiles and passes as expected, indicating that the test itself is not affected by this change.

```
console.error
Warning: ReactDOM.render is no longer supported in React 18. Use createRoot instead. Until you switch to the new API, your
app will behave as if it's running React 17. Learn more: https://reactjs.org/link/switch-to-createroot

    at printWarning (node_modules/react-dom/cjs/react-dom.development.js:86:30)
    at error (node_modules/react-dom/cjs/react-dom.development.js:60:7)
    at Object.render (node_modules/react-dom/cjs/react-dom.development.js:29670:5)
    at node_modules/@testing-library/react/dist/pure.js:99:25
    at act (node_modules/react/cjs/react.development.js:2512:16)
    at render (node_modules/@testing-library/react/dist/pure.js:95:26)
    at Object.<anonymous> (src/components/__test__/sales.test.js:6:31)
```

### 1.3 Warning: Invalid DOM Property `class`

This warning indicates that there is an attempt to use the DOM property class, which is not valid in React. React expects the use of className instead to define CSS classes for elements. The warning suggests that className should be used instead of class to avoid potential issues with rendering or styling components.

```
console.error
Warning: Invalid DOM property `class`. Did you mean `className`?
    at div
    at div
    at div
    at Sales

    at printWarning (node_modules/react-dom/cjs/react-dom.development.js:86:30)
    at error (node_modules/react-dom/cjs/react-dom.development.js:60:7)
    at validateProperty$1 (node_modules/react-dom/cjs/react-dom.development.js:3749:9)
    at warnUnknownProperties (node_modules/react-dom/cjs/react-dom.development.js:3803:21)
    at validateProperties$2 (node_modules/react-dom/cjs/react-dom.development.js:3827:3)
    at validatePropertiesInDevelopment (node_modules/react-dom/cjs/react-dom.development.js:9541:5)
    at setInitialProperties (node_modules/react-dom/cjs/react-dom.development.js:9830:5)
```

### 1.4 Warning: window.scrollTo

This warning indicates that the window.scrollTo method is not implemented in the current environment. This can occur when using JSDOM, a library used for testing React components in Node.js environments, which does not fully support all browser features. In this case, the window.scrollTo method, which is typically used to scroll the window to a specific position, is not available.

This warning can be safely ignored in testing environments.

```
console.error
Error: Not implemented: window.scrollTo
    at module.exports (D:\mernabook\frontend\node_modules\jsdom\lib\jsdom\browser\not-implemented.js:9:17)
    at scrollTo (D:\mernabook\frontend\node_modules\jsdom\lib\jsdom\browser\Window.js:866:7)
    at D:\mernabook\frontend\src\components\general\navbar\navbar.js:22:12
    at commitHookEffectListMount (D:\mernabook\frontend\node_modules\react-dom\cjs\react-dom.development.js:23150:26)
    at commitPassiveMountOnFiber (D:\mernabook\frontend\node_modules\react-dom\cjs\react-dom.development.js:24931:11)
    at commitPassiveMountEffects_complete (D:\mernabook\frontend\node_modules\react-dom\cjs\react-dom.development.js:24891:9)
    at commitPassiveMountEffects_begin (D:\mernabook\frontend\node_modules\react-dom\cjs\react-dom.development.js:24878:7)
    at commitPassiveMountEffects (D:\mernabook\frontend\node_modules\react-dom\cjs\react-dom.development.js:24866:3)
    at flushPassiveEffectsImpl (D:\mernabook\frontend\node_modules\react-dom\cjs\react-dom.development.js:27039:3)
    at flushPassiveEffects (D:\mernabook\frontend\node_modules\react-dom\cjs\react-dom.development.js:26984:14) undefined
```

# 1. Unit Test

## 2.1 Unit Test: account.test.js

The test suite checks if the Account component renders correctly and if the email input field updates its value upon user interaction.

```
JS account.test.js ×
frontend > src > components > __test__ > JS account.test.js > ...
1 import React from 'react';
2 import { render, fireEvent } from '@testing-library/react';
3 import { BrowserRouter as Router } from 'react-router-dom';
4 import Account from '../admin/dashboard/Account';
5 import { UserContext } from '../user/UserContext';
6
7 const mockLogoutUser = jest.fn();
8
9 test('Account component renders #1', () => {
10   const { getByText } = render(
11     <Router>
12       <UserContext.Provider value={{ logoutUser: mockLogoutUser }}>
13         <Account />
14       </UserContext.Provider>
15     </Router>
16   );
17   const exportDataElement = getByText('Update Account');
18   expect(exportDataElement).toBeTruthy();
19 });
20
21 test('Account component renders #1', () => {
22   const { getByLabelText } = render(
23     <Router>
24       <UserContext.Provider value={{ logoutUser: mockLogoutUser }}>
25         <Account />
26       </UserContext.Provider>
27     </Router>
28   );
29
30   const emailInput = getByLabelText('Email');
31   expect(emailInput).toBeTruthy();
32
33   fireEvent.change(emailInput, { target: { value: 'test@example.com' } });
34
35   expect(emailInput.value).toBe('test@example.com');
36 });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

node - frontend + - [ ] [ ] ... ^ x

PASS: src/components/\_\_test\_\_/account.test.js  
✓ Account component renders #1 (259 ms)  
✓ Account component renders #1 (82 ms)

Test Suites: 1 passed, 1 total  
Tests: 2 passed, 2 total  
Snapshots: 0 total  
Time: 2.704 s  
Ran all test suites matching /account.test.js/i.

## 2.2 Unit Test: adminDashboard.test.js

The test suite verifies that the Admin UserBar component correctly renders navigation links for "Account Details," "Product," and "Orders," utilizing a mocked logOutUser function.

The screenshot shows a code editor with a file named `adminDashboard.test.js` and a terminal window displaying the test results.

**Code Editor Content:**

```

1  import React from 'react';
2  import { render } from '@testing-library/react';
3  import { BrowserRouter as Router } from 'react-router-dom';
4  import UserBar from '../admin/dashboard/AdminBar';
5  import { UserContext } from '../user/UserContext';
6
7  const mockLogOutUser = jest.fn();
8
9  test('Admin UserBar component renders #1', () => {
10   const { getByText } = render(
11     <Router>
12       <UserContext.Provider value={{ logOutUser: mockLogOutUser }}>
13         <UserBar />
14       </UserContext.Provider>
15     </Router>
16   );
17   const exportDataElement = getByText('Account Details');
18   expect(exportDataElement).toBeTruthy();
19 });
20
21
22 test('Admin UserBar component renders #2', () => {
23   const { getByText } = render(
24     <Router>
25       <UserContext.Provider value={{ logOutUser: mockLogOutUser }}>
26         <UserBar />
27       </UserContext.Provider>
28     </Router>
29   );
30   const exportDataElement = getByText('Product');
31   expect(exportDataElement).toBeTruthy();
32 });
33
34
35 test('Admin UserBar component renders #3', () => {
36   const { getByText } = render(
37     <Router>
38       <UserContext.Provider value={{ logOutUser: mockLogOutUser }}>
39         <UserBar />
40       </UserContext.Provider>
41     </Router>
42   );
43   const exportDataElement = getByText('Orders');
44   expect(exportDataElement).toBeTruthy();
45 });

```

**Terminal Output:**

```

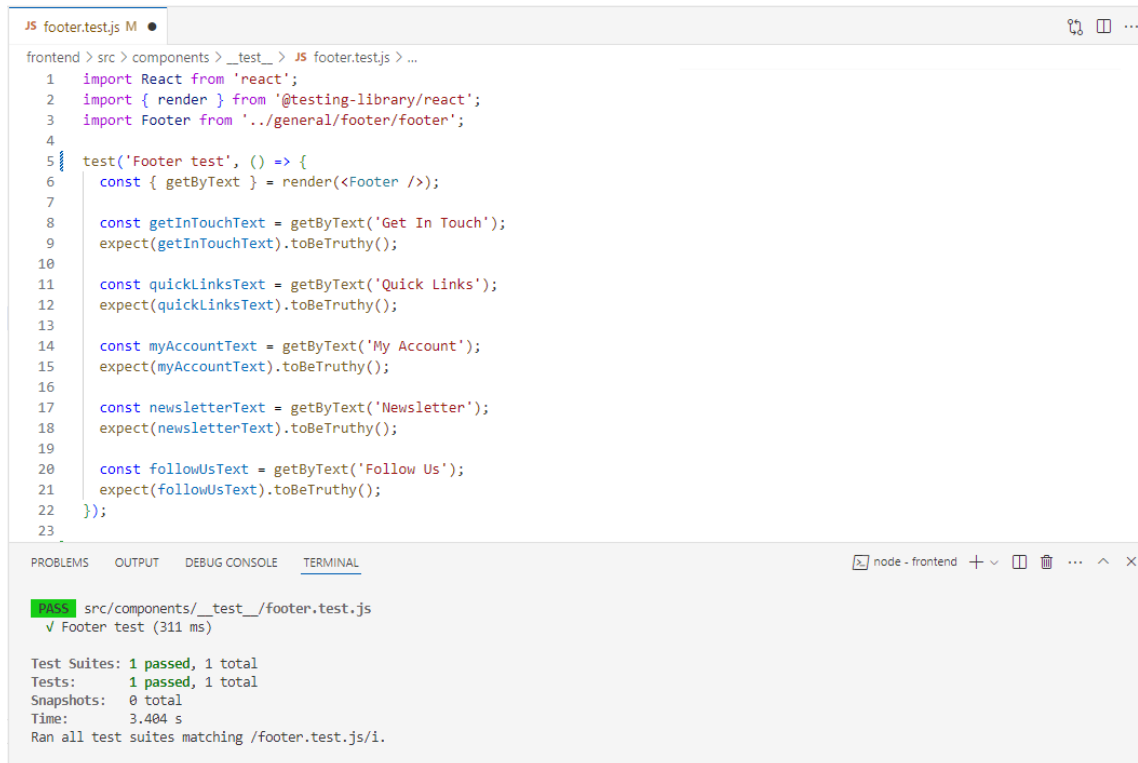
node - frontend
PASS src/components/_test_/adminDashboard.test.js
  ✓ Admin UserBar component renders #1 (183 ms)
  ✓ Admin UserBar component renders #2 (19 ms)
  ✓ Admin UserBar component renders #3 (28 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        3.085 s
Ran all test suites matching /adminDashboard.test.js/i.

```

## 2.3 Unit Test: footer.test.js

The test verifies that the Footer component renders sections for "Get In Touch," "Quick Links," "My Account," "Newsletter," and "Follow Us" successfully.



```
JS footer.test.js M
frontend > src > components > __test__ > JS footer.test.js > ...
1 import React from 'react';
2 import { render } from '@testing-library/react';
3 import Footer from '../general/footer/footer';
4
5 test('Footer test', () => {
6   const { getByText } = render(<Footer />);
7
8   const getInTouchText = getByText('Get In Touch');
9   expect(getInTouchText).toBeTruthy();
10
11   const quickLinksText = getByText('Quick Links');
12   expect(quickLinksText).toBeTruthy();
13
14   const myAccountText = getByText('My Account');
15   expect(myAccountText).toBeTruthy();
16
17   const newsletterText = getByText('Newsletter');
18   expect(newsletterText).toBeTruthy();
19
20   const followUsText = getByText('Follow Us');
21   expect(followUsText).toBeTruthy();
22 });
23
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

node - frontend + - [ ] [ ] ... ^ X

100% src/components/\_\_test\_\_/footer.test.js  
✓ Footer test (311 ms)

Test Suites: 1 passed, 1 total  
Tests: 1 passed, 1 total  
Snapshots: 0 total  
Time: 3.404 s  
Ran all test suites matching /footer.test.js/i.

## 2.4 Unit Test: home.test.js

The tests verify that the Home component renders sections for "Quality Product," "New Arrivals," "Newsletter," and "Privacy," as well as an image with a specified alt text successfully.

The screenshot shows a code editor with a file named `home.test.js` and a terminal window displaying the test results.

**Code Editor Content:**

```

1  import React from 'react';
2  import { render } from '@testing-library/react';
3  import Home from '../general/home/home';
4
5  test('Home component renders chekc #1', () => {
6    const { getByText } = render(<Home />);
7    const exportDataElement = getByText('Quality Product');
8    expect(exportDataElement).toBeTruthy();
9  });
10
11 test('Home component renders chekc #2', () => {
12   const { getByText } = render(<Home />);
13   const exportDataElement = getByText('New Arrivals');
14   expect(exportDataElement).toBeTruthy();
15 });
16
17 test('Home component renders chekc #3', () => {
18   const { getByText } = render(<Home />);
19   const exportDataElement = getByText('Newsletter');
20   expect(exportDataElement).toBeTruthy();
21 });
22
23 test('Home component renders chekc #4', () => {
24   const { getByText } = render(<Home />);
25   const exportDataElement = getByText('Privacy');
26   expect(exportDataElement).toBeTruthy();
27 });
28
29 test('Home component renders chekc #5', () => {
30   const { getByAltText } = render(<Home />);
31   const imageElement = getByAltText('Wiley');
32   expect(imageElement).toBeTruthy();
33 });
34

```

**Terminal Output:**

```

node - frontend
PASS src/components/__test__/home.test.js
  ✓ Home component renders chekc #1 (227 ms)
  ✓ Home component renders chekc #2 (50 ms)
  ✓ Home component renders chekc #3 (37 ms)
  ✓ Home component renders chekc #4 (35 ms)
  ✓ Home component renders chekc #5 (44 ms)

Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:  0 total
Time:        3.482 s

```



## 2.5 Unit Test: inventory.test.js

The tests verify that the Inventory component renders elements such as "Export Data," "Inventory" heading, "Add new Product," "Sort By," "All Subjects" for filtering by subjects, and "All Prices" for filtering by prices successfully.

```

JS inventory.test.js x
frontend > src > components > __test__ > JS inventory.test.js > test("Inventory component renders Filter by Prices component") callback
1  import React from 'react';
2  import { render } from '@testing-library/react';
3  import Inventory from '../admin/dashboard/Inventory';
4
5  test('Inventory component renders', () => {
6    const { getByText } = render(<Inventory />);
7    const exportDataElement = getByText('Export Data');
8    expect(exportDataElement).toBeTruthy();
9  });
10
11 test('Inventory component renders Heading', () => {
12   const { getByText } = render(<Inventory />);
13   const exportDataElement = getByText('Inventory');
14   expect(exportDataElement).toBeTruthy();
15 });
16
17 test('Inventory component renders Add new Product component', () => {
18   const { getByText } = render(<Inventory />);
19   const exportDataElement = getByText('Add new Product');
20   expect(exportDataElement).toBeTruthy();
21 });
22
23 test('Inventory component renders Sort By component', () => {
24   const { getByText } = render(<Inventory />);
25   const exportDataElement = getByText('Sort By');
26   expect(exportDataElement).toBeTruthy();
27 });
28
29 test('Inventory component renders Filter by Subjects component', () => {
30   const { getByText } = render(<Inventory />);
31   const exportDataElement = getByText('All Subjects');
32   expect(exportDataElement).toBeTruthy();
33 });
34
35 test('Inventory component renders Filter by Prices component', () => {
36   const { getByText } = render(<Inventory />);
37   const exportDataElement = getByText('All Prices');
38   expect(exportDataElement).toBeTruthy();
39 });

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

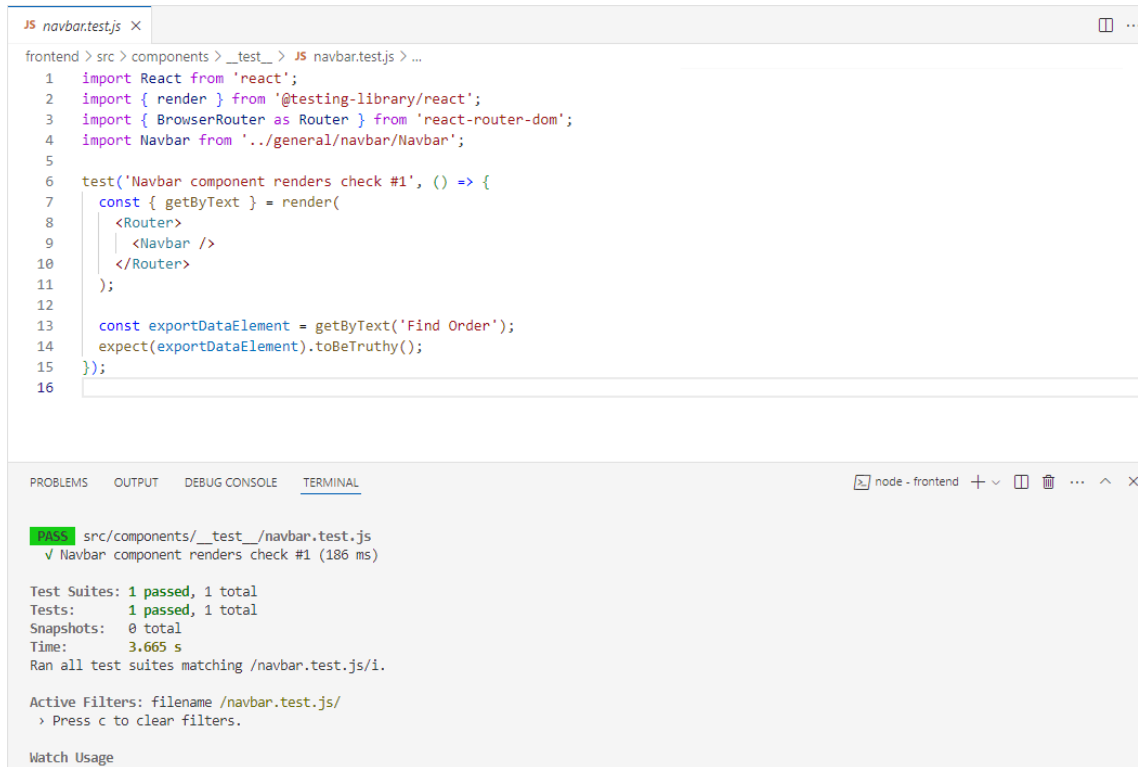
node - frontend + v [ ] [ ] ... ^ x
PASS: src/components/__test__/inventory.test.js
  ✓ Inventory component renders (354 ms)
  ✓ Inventory component renders Heading (74 ms)
  ✓ Inventory component renders Add new Product component (70 ms)
  ✓ Inventory component renders Sort By component (81 ms)
  ✓ Inventory component renders Filter by Subjects component (58 ms)
  ✓ Inventory component renders Filter by Prices component (101 ms)

Test Suites: 1 passed, 1 total
Tests: 6 passed, 6 total
Snapshots: 0 total
Time: 3.153 s
Ran all test suites matching /inventory.test/i.

```

## 2.6 Unit Test: navbar.test.js

The test verifies that the Navbar component renders the "Find Order" element successfully.



```
JS navbar.test.js X
frontend > src > components > __test__ JS navbar.test.js > ...
1 import React from 'react';
2 import { render } from '@testing-library/react';
3 import { BrowserRouter as Router } from 'react-router-dom';
4 import Navbar from '../general/navbar/Navbar';
5
6 test('Navbar component renders check #1', () => {
7   const { getByText } = render(
8     <Router>
9     | <Navbar />
10    </Router>
11  );
12
13   const exportDataElement = getByText('Find Order');
14   expect(exportDataElement).toBeTruthy();
15 });
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

node - frontend + - ... ^ X

186ms src/components/\_\_test\_\_/navbar.test.js  
✓ Navbar component renders check #1 (186 ms)

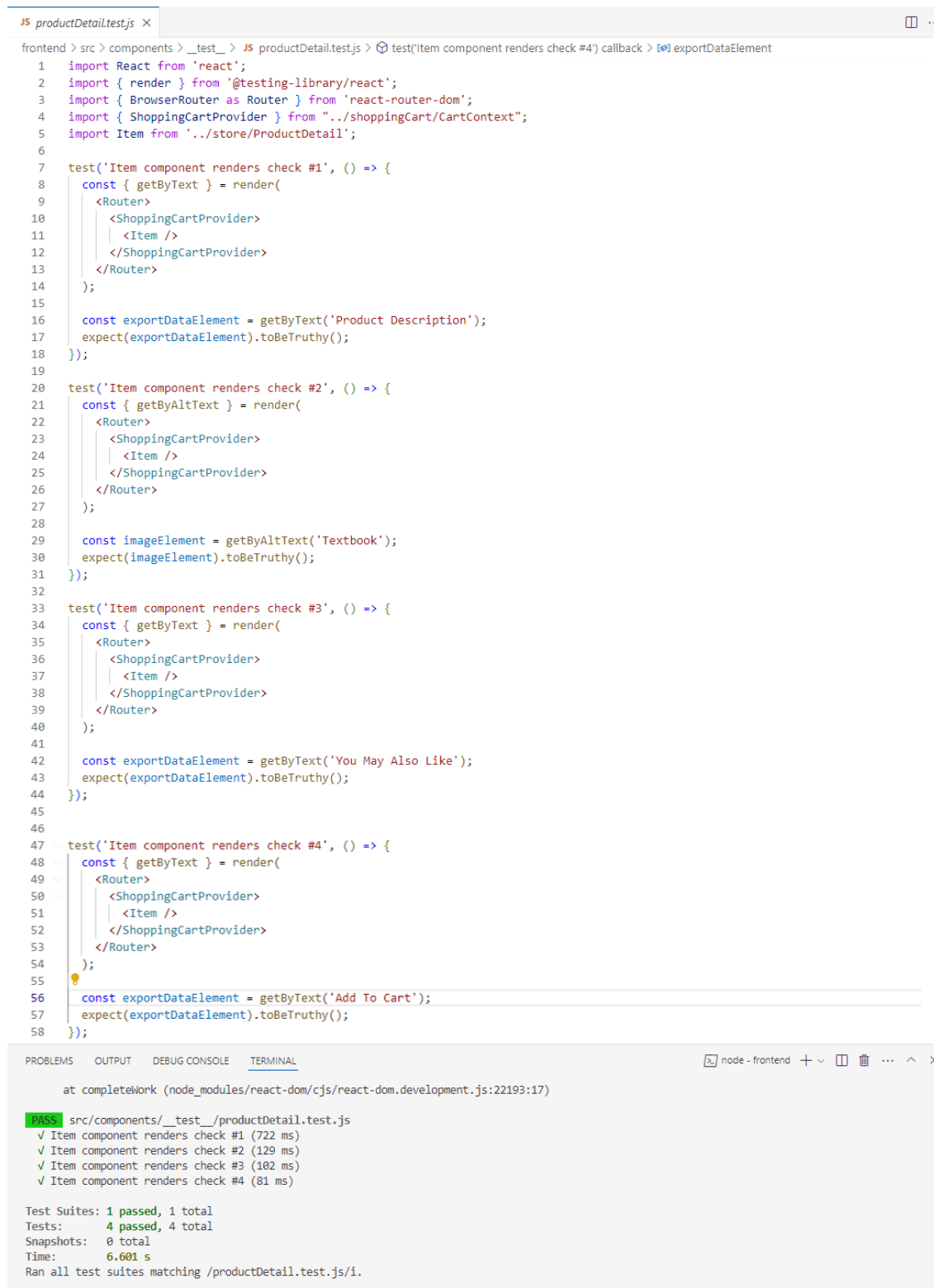
Test Suites: 1 passed, 1 total  
Tests: 1 passed, 1 total  
Snapshots: 0 total  
Time: 3.665 s  
Ran all test suites matching /navbar.test.js/i.

Active Filters: filename /navbar.test.js/  
› Press c to clear filters.

Watch Usage

## 2.7 Unit Test: productDetail.test.js

These tests verify that the Item component renders various elements correctly, including the product description, an image of the textbook, a section for related products, and an "Add To Cart" button.



```

JS productDetail.test.js x
frontend > src > components > __test__ > JS productDetail.test.js > test('Item component renders check #4') callback > exportDataElement

1  import React from 'react';
2  import { render } from '@testing-library/react';
3  import { BrowserRouter as Router } from 'react-router-dom';
4  import { ShoppingCartProvider } from '../shoppingCart/CartContext';
5  import Item from '../store/ProductDetail';
6
7  test('Item component renders check #1', () => {
8    const { getByText } = render(
9      <Router>
10     <ShoppingCartProvider>
11       <Item />
12     </ShoppingCartProvider>
13   </Router>
14 );
15
16   const exportDataElement = getByText('Product Description');
17   expect(exportDataElement).toBeTruthy();
18 });
19
20 test('Item component renders check #2', () => {
21   const { getByAltText } = render(
22     <Router>
23       <ShoppingCartProvider>
24         <Item />
25       </ShoppingCartProvider>
26     </Router>
27   );
28
29   const imageElement = getByAltText('Textbook');
30   expect(imageElement).toBeTruthy();
31 });
32
33 test('Item component renders check #3', () => {
34   const { getByText } = render(
35     <Router>
36       <ShoppingCartProvider>
37         <Item />
38       </ShoppingCartProvider>
39     </Router>
40   );
41
42   const exportDataElement = getByText('You May Also Like');
43   expect(exportDataElement).toBeTruthy();
44 });
45
46
47 test('Item component renders check #4', () => {
48   const { getByText } = render(
49     <Router>
50       <ShoppingCartProvider>
51         <Item />
52       </ShoppingCartProvider>
53     </Router>
54   );
55
56   const exportDataElement = getByText('Add To Cart');
57   expect(exportDataElement).toBeTruthy();
58 });

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
node - frontend + v [] [] ... ^ x

at completeWork (node_modules/react-dom/cjs/react-dom.development.js:22193:17)

PASS src/components/__test__/productDetail.test.js
  ✓ Item component renders check #1 (722 ms)
  ✓ Item component renders check #2 (129 ms)
  ✓ Item component renders check #3 (102 ms)
  ✓ Item component renders check #4 (81 ms)

Test Suites: 1 passed, 1 total
Tests: 4 passed, 4 total
Snapshots: 0 total
Time: 6.601 s
Ran all test suites matching /productDetail.test.js/i.

```

## 2.8 Unit Test: productListing.test.js

These tests verify that the Products component renders elements such as a sort dropdown ("Sort by") and an option for alphabetical sorting ("Alphabetic A - Z").

The screenshot shows a code editor with a file named `productListing.test.js` and a terminal window below it.

**Code Editor:**

```

1  import React from 'react';
2  import { render } from '@testing-library/react';
3  import { BrowserRouter as Router } from 'react-router-dom';
4  import { ShoppingCartProvider } from "../shoppingCart/CartContext";
5  import Products from '../store/ProductListing';
6
7  test('Products component renders check #1', () => {
8    const { getByText } = render(
9      <Router>
10       <ShoppingCartProvider>
11         <Products />
12       </ShoppingCartProvider>
13     </Router>
14   );
15
16   const exportDataElement = getByText('Sort by');
17   expect(exportDataElement).toBeTruthy();
18 });
19
20 test('Products component renders check #2', () => {
21   const { getByText } = render(
22     <Router>
23       <ShoppingCartProvider>
24         <Products />
25       </ShoppingCartProvider>
26     </Router>
27   );
28
29   const exportDataElement = getByText('Alphabetic A - Z');
30   expect(exportDataElement).toBeTruthy();
31 });

```

**Terminal:**

```

at render (node_modules/@testing-library/react/dist/pure.js:95:26)
at Object.<anonymous> (src/components/__test__/productListing.test.js:21:31)

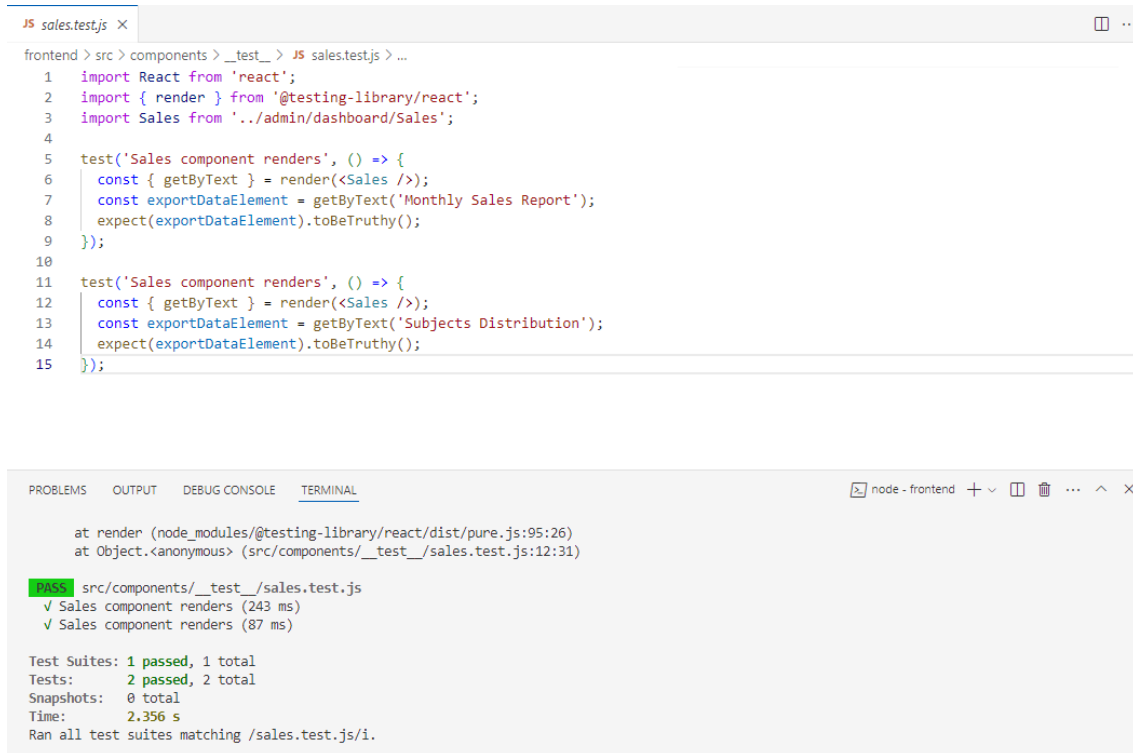
PASS src/components/__test__/productListing.test.js
  ✓ Products component renders check #1 (150 ms)
  ✓ Products component renders check #2 (23 ms)

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 2.179 s
Ran all test suites matching /productListing.test.js/i.

```

## 2.9 Unit Test: sales.test.js

These tests ensure that the Sales component renders elements such as "Monthly Sales Report" and "Subjects Distribution."



The screenshot displays the VS Code editor with a file named `sales.test.js` open. The file contains two Jest tests for the `Sales` component. The first test checks for the presence of the "Monthly Sales Report" text, and the second test checks for the presence of the "Subjects Distribution" text. Below the editor, the terminal window shows the output of running the tests, indicating that both tests passed successfully.

```
JS sales.test.js X
frontend > src > components > __test__ > JS sales.test.js > ...
1 import React from 'react';
2 import { render } from '@testing-library/react';
3 import Sales from '../admin/dashboard/Sales';
4
5 test('Sales component renders', () => {
6   const { getByText } = render(<Sales />);
7   const exportDataElement = getByText('Monthly Sales Report');
8   expect(exportDataElement).toBeTruthy();
9 });
10
11 test('Sales component renders', () => {
12   const { getByText } = render(<Sales />);
13   const exportDataElement = getByText('Subjects Distribution');
14   expect(exportDataElement).toBeTruthy();
15 });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

node - frontend + v [icon] [icon] ... ^ X

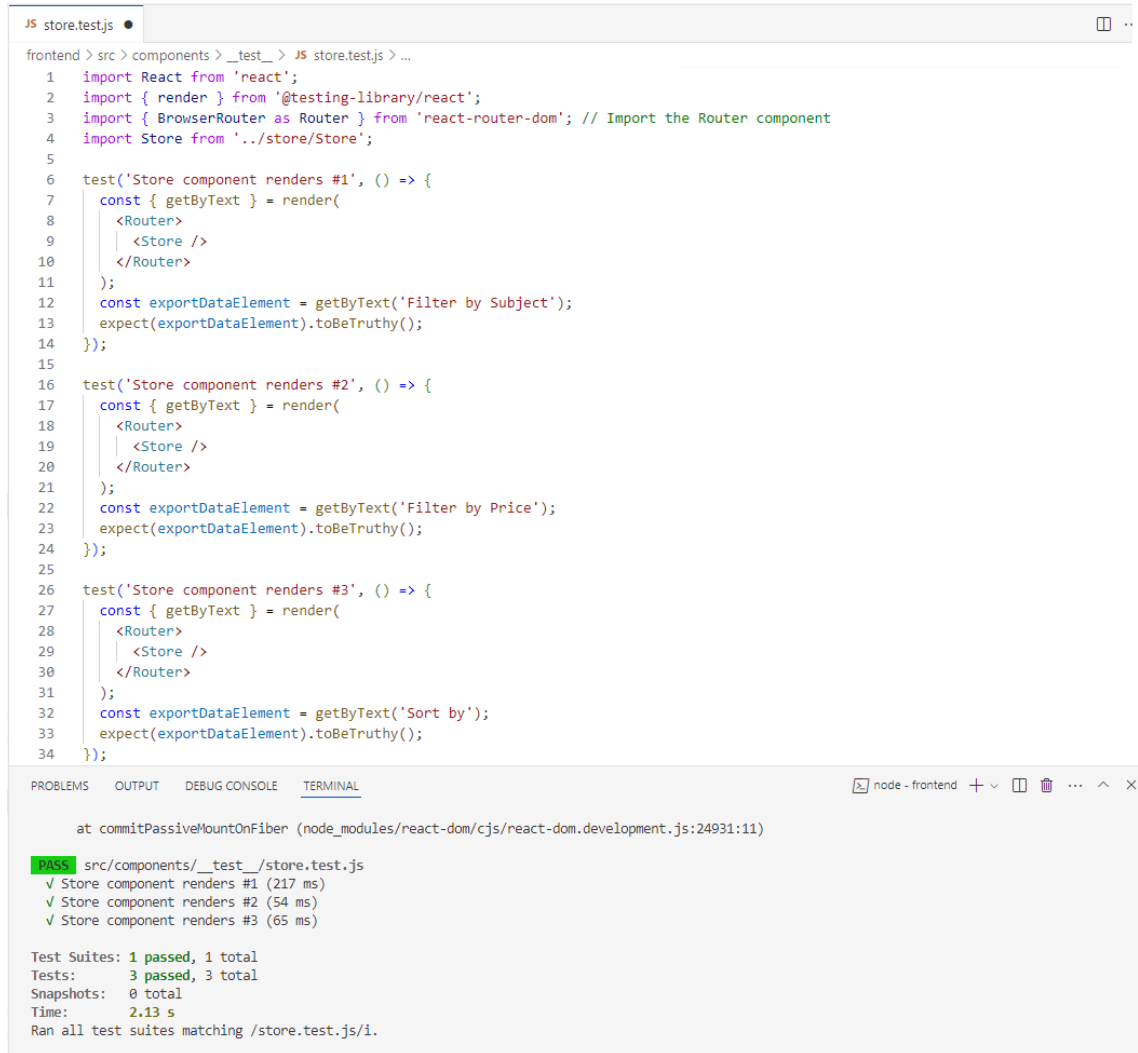
at render (node\_modules/@testing-library/react/dist/pure.js:95:26)  
at Object.<anonymous> (src/components/\_\_test\_\_/sales.test.js:12:31)

**PASS** src/components/\_\_test\_\_/sales.test.js  
✓ Sales component renders (243 ms)  
✓ Sales component renders (87 ms)

Test Suites: 1 passed, 1 total  
Tests: 2 passed, 2 total  
Snapshots: 0 total  
Time: 2.356 s  
Ran all test suites matching /sales.test.js/i.

## 2.10 Unit Test: store.test.js

These tests check that the Store component renders with specific elements: one for filtering by subject, one for filtering by price, and one for sorting.



```

JS store.test.js
frontend > src > components > __test__ > JS store.test.js > ...
1  import React from 'react';
2  import { render } from '@testing-library/react';
3  import { BrowserRouter as Router } from 'react-router-dom'; // Import the Router component
4  import Store from '../store/Store';
5
6  test('Store component renders #1', () => {
7    const { getByText } = render(
8      <Router>
9        <Store />
10     </Router>
11   );
12   const exportDataElement = getByText('Filter by Subject');
13   expect(exportDataElement).toBeTruthy();
14 });
15
16 test('Store component renders #2', () => {
17   const { getByText } = render(
18     <Router>
19       <Store />
20     </Router>
21   );
22   const exportDataElement = getByText('Filter by Price');
23   expect(exportDataElement).toBeTruthy();
24 });
25
26 test('Store component renders #3', () => {
27   const { getByText } = render(
28     <Router>
29       <Store />
30     </Router>
31   );
32   const exportDataElement = getByText('Sort by');
33   expect(exportDataElement).toBeTruthy();
34 });

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

node - frontend + v [ ] [ ] ... ^ X

at commitPassiveMountOnFiber (node\_modules/react-dom/cjs/react-dom.development.js:24931:11)

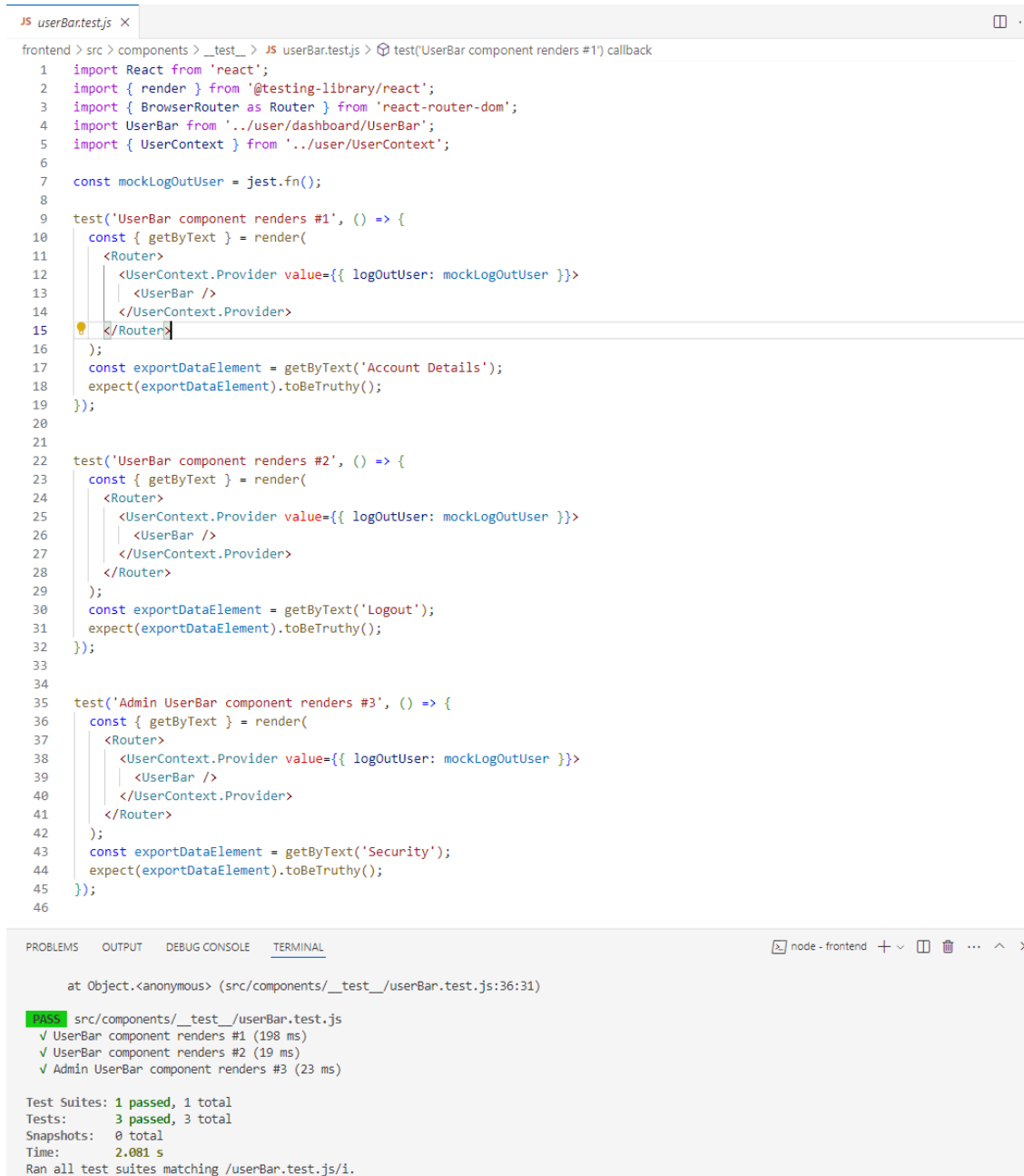
**PASS** src/components/\_\_test\_\_/store.test.js

- ✓ Store component renders #1 (217 ms)
- ✓ Store component renders #2 (54 ms)
- ✓ Store component renders #3 (65 ms)

Test Suites: 1 passed, 1 total  
 Tests: 3 passed, 3 total  
 Snapshots: 0 total  
 Time: 2.13 s  
 Ran all test suites matching /store.test.js/i.

## 2.11 Unit Test: userBar.test.js

These tests check that the UserBar component renders with specific elements: one for "Account Details," one for "Logout," and one for "Security."



```

JS userBar.test.js x
frontend > src > components > __test__ > JS userBar.test.js > test('UserBar component renders #1') callback
1 import React from 'react';
2 import { render } from '@testing-library/react';
3 import { BrowserRouter as Router } from 'react-router-dom';
4 import UserBar from '../user/dashboard/UserBar';
5 import { UserContext } from '../user/UserContext';
6
7 const mockLogoutUser = jest.fn();
8
9 test('UserBar component renders #1', () => {
10   const { getByText } = render(
11     <Router>
12       <UserContext.Provider value={{ logoutUser: mockLogoutUser }}>
13         <UserBar />
14       </UserContext.Provider>
15     </Router>
16   );
17   const exportDataElement = getByText('Account Details');
18   expect(exportDataElement).toBeTruthy();
19 });
20
21
22 test('UserBar component renders #2', () => {
23   const { getByText } = render(
24     <Router>
25       <UserContext.Provider value={{ logoutUser: mockLogoutUser }}>
26         <UserBar />
27       </UserContext.Provider>
28     </Router>
29   );
30   const exportDataElement = getByText('Logout');
31   expect(exportDataElement).toBeTruthy();
32 });
33
34
35 test('Admin UserBar component renders #3', () => {
36   const { getByText } = render(
37     <Router>
38       <UserContext.Provider value={{ logoutUser: mockLogoutUser }}>
39         <UserBar />
40       </UserContext.Provider>
41     </Router>
42   );
43   const exportDataElement = getByText('Security');
44   expect(exportDataElement).toBeTruthy();
45 });
46

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
node - frontend + v [ ] [ ] ... ^ x

at Object.<anonymous> (src/components/__test__/userBar.test.js:36:31)

PASS: src/components/__test__/userBar.test.js
✓ UserBar component renders #1 (198 ms)
✓ UserBar component renders #2 (19 ms)
✓ Admin UserBar component renders #3 (23 ms)

Test Suites: 1 passed, 1 total
Tests: 3 passed, 3 total
Snapshots: 0 total
Time: 2.081 s
Ran all test suites matching /userBar.test.js/i.

```