PaycheckCalculator Component Test Results

1. Test Execution Summary

• Date of Execution: September 23, 2024

• Executed by: Chris Holmes

2. Test Environment

• Node.js version: v20.12.2

• React version: 18.3.1

• Jest version: 29.7.0

• React Testing Library version: 16.0.1

• Next.js version: 14.2.7

3. Test Execution Details

3.1 TC001: Render and Calculate Paycheck

Status: PASS

Steps Executed:

1. Rendered the PaycheckCalculator component

2. Verified presence of Start Date and End Date inputs

3. Verified presence of Calculate Paycheck button

4. Entered start date: 2023-01-01

5. Entered end date: 2023-01-15

6. Clicked Calculate Paycheck button

7. Verified loading state

8. Verified paycheck summary display

Actual Results:

• Component rendered without errors

• Start Date and End Date inputs were visible

• Calculate Paycheck button was visible

• After clicking Calculate Paycheck:

- Loading state ("Calculating...") was displayed

- Paycheck summary was displayed with correct information:

◦ Regular Hours: 40.00

◦ Overtime Hours: 5.00

◦ Regular Pay: $800.00

◦ Overtime Pay: $150.00

◦ Total Pay: $950.00

Execution Time: 151 ms

3.2 TC002: Handle API Error

Status: PASS

Steps Executed:

1. Rendered the PaycheckCalculator component

2. Entered start date: 2023-01-01

3. Entered end date: 2023-01-15

4. Mocked API error response

5. Clicked Calculate Paycheck button

6. Verified error message display

Actual Results:

• Component rendered without errors

• After clicking Calculate Paycheck:

- Error message "An error occurred while calculating your paycheck. Please try again." was displayed

Execution Time: 92 ms

4. Overall Test Execution Results

• Total Test Cases: 2

• Passed: 2

• Failed: 0

• Total Execution Time: 2.385 s

5. Test Output Screenshot

A screen shot of a computer

Description automatically generated

6. Conclusion

The PaycheckCalculator component has passed all test cases successfully. It correctly renders all required elements, handles user interactions, makes API calls with the correct parameters, and displays both successful results and error messages as expected.

7. Recommendations

1. Implement additional test cases for edge scenarios (e.g., invalid date inputs, different pay rates)

2. Add accessibility testing to ensure the component is usable by all users

3. Consider implementing end-to-end tests to verify integration with the backend API in a real environment

This test execution has verified the core functionality of the PaycheckCalculator component, and it is ready for integration into the Tyme Keepa application.