Kiran Chung

Exercycle Physical Activity Cash Back Incentive

Test Report

Test Objective

A scheme created by local governments, New Zealand power companies and sports/health advocates financially rewards households for generating power though physical activity on an exercycle. However, household points are not being calculated correctly. Defects in the application are to be found and reported.

Methodology

Although the main problem revolves around calculation errors, other parts of the web application such as form input were tested to identify possible unseen causes. For this project, Cypress end-to-end testing was used to test for defects.

Class names, ids, and other less general identifiers from the HTML file were used to identify elements rather than identifying them by details such as visible text. This was due to many elements (especially in the form page) having similar content which made it more time-consuming and harder to identify elements.

Each class/file is responsible for different parts of the application. For example, exercycle.cy.js only contains tests for the start page whereas exercycle-form-individual.js only contains tests for single member households.

Some tests were iterated over household member numbers as they were testing the same functionality. This ultimately reduced code duplication and also reduced file length so the code was more readable and consistent.

In addition to testing for incorrect functionality, some tests also check for incorrect visuals/incorrect output.

The approach taken was to traverse chronologically through the web application and identify errors while proceeding through the application. This is shown through the flow of commits on the GitHub repository[1]. The first tests produced are related to the home page. The next tests were relevant to inputs and 1 person calculations.

Findings

Total tests run: 41

Tests succeeded: 28

Tests failed: 13

See Figure 1 for a visual representation of the above.

Please see Appendix for test failures and other relevant test information

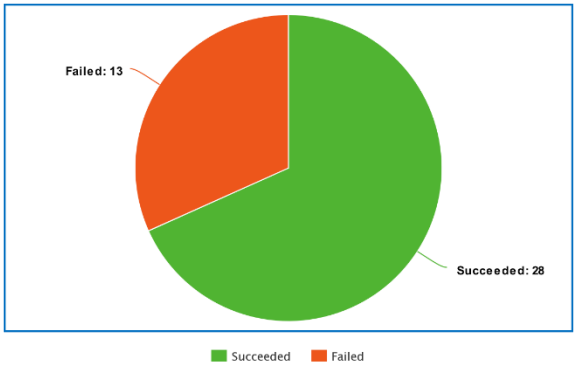


Figure 1: Pie chart of test passes and failures

It was found that some input types that should be invalid were allowed to be inputted and were involved in calculations:

- Negative numbers were allowed as input. Large negative numbers caused the final numbers in the calculation page to be in scientific form (eg. 1E+15)
- Scientific numbers were also allowed as input and were used in the calculations. However, it was unclear/unknown if this number format is valid. Regardless of its validity, test cases have been created for this format.

- Removing all numbers from the form and entering empty fields caused the table in the calculation page to not show any value. Although an empty input has been found to equal to 0 and calculate correctly, the value of 0 is not shown in the individual points table. This also occurs with multiple members; the individual and household calculations are correct but there is no value shown in the table for the respective members.

The form inputs allowed numbers greater than 7 and less than 0 to be inputted by typing in the number, using arrow keys or clicking the form arrow buttons. Although an error message in the calculation page appeared for numbers greater than 7, there was none for negative numbers.

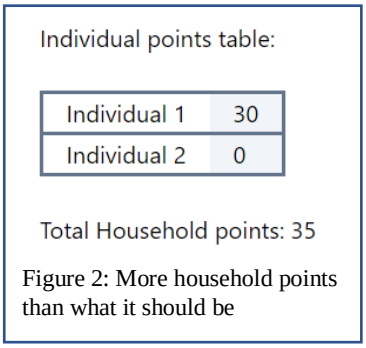


Figure 2: More household points than what it should be

It should also be noted that though manual testing, it has been found that the pop-up for when an input field contains a decimal informs the user of the nearest valid values. For example, if the user enters 5.5, it informs them that "the two nearest valid values are 5 and 6". It does the same thing for a negative decimal. For example, if the user enters -2.3, it tells them that "the two nearest **valid** values are -3 and -2". However, as negative numbers should not be valid, this tip is incorrect.

The calculations for total household points were incorrect in the following ways:

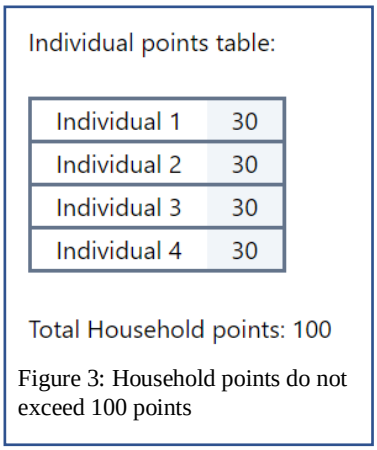


Figure 3: Household points do not exceed 100 points

Although the scheme rules state that individuals can only earn 30 points a week for their household, individuals' excess points – points above 30 – are included in the calculation. The final table shows the correct amount for the individual but the total household points is incorrect. In Figure 2, Individual 1 has earned 35 points, 30 after applying the weekly limit. The table correctly displays 30 but the message displays 35.

Total household points were limited at 100 and did not count any past this amount. This was only true for forms with 4 or 5 individuals as it is not possible for less members to reach 100 points (unless the calculation error above is considered. In this case, the 3-member form can reproduce this error as 3*49>100). In Figure 3, the total household points should be equal to 120. However, it displays the total as 100.

Recommendations

- Form inputs need to be checked and not allow the user to input the invalid number formats listed in the findings. It is recommended that a pop-up similar to the decimal pop-up is implemented.
- Numbers between 0-7 inclusive should only allowed to be inputted. Arrow keys/form buttons should not allow the value to go higher than 7 or lower than 0.
- Update the calculation so that:
    - Individual points over 30 are not counted towards the total household points.
    - Household points can exceed 100 points.
- UI: An instructions/help page to tell the user how to calculate/enter points will be helpful. This will also help families with more than 5 members as they may be unsure of what to input since the 5+ table only contains 5 individuals.

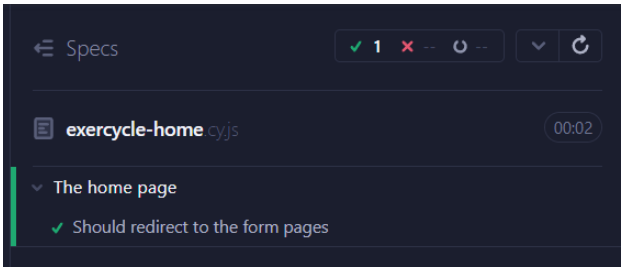
Other notes

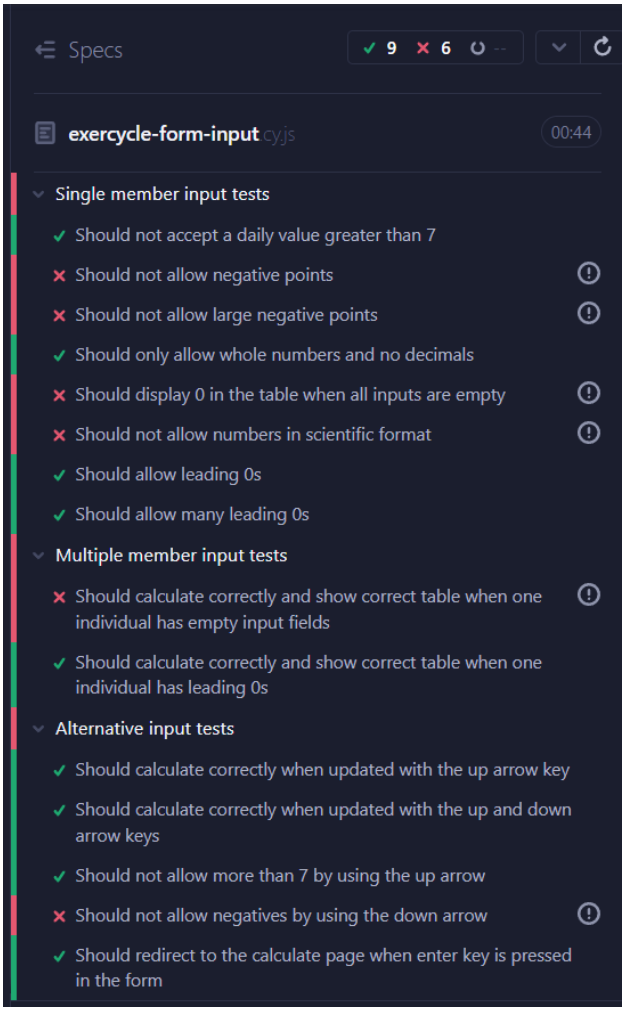
An error rarely occurred with test cases where Cypress would detect an element as "disabled" and mark the test as failed, even if the test usually passes. However, a solution to this was not implemented due to the solution allowing incorrect access to the input fields.

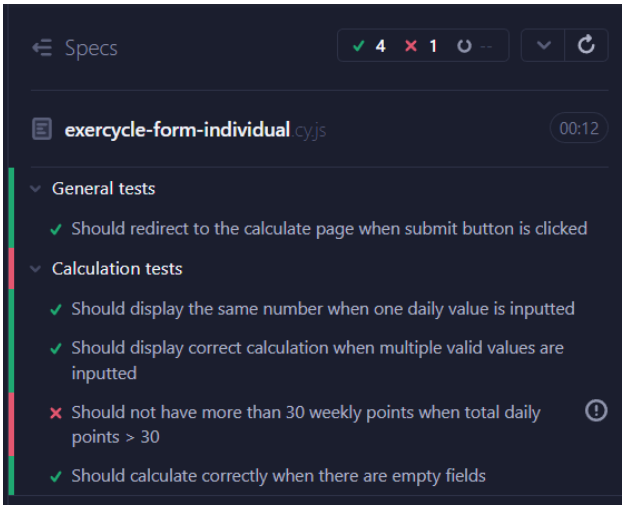
References

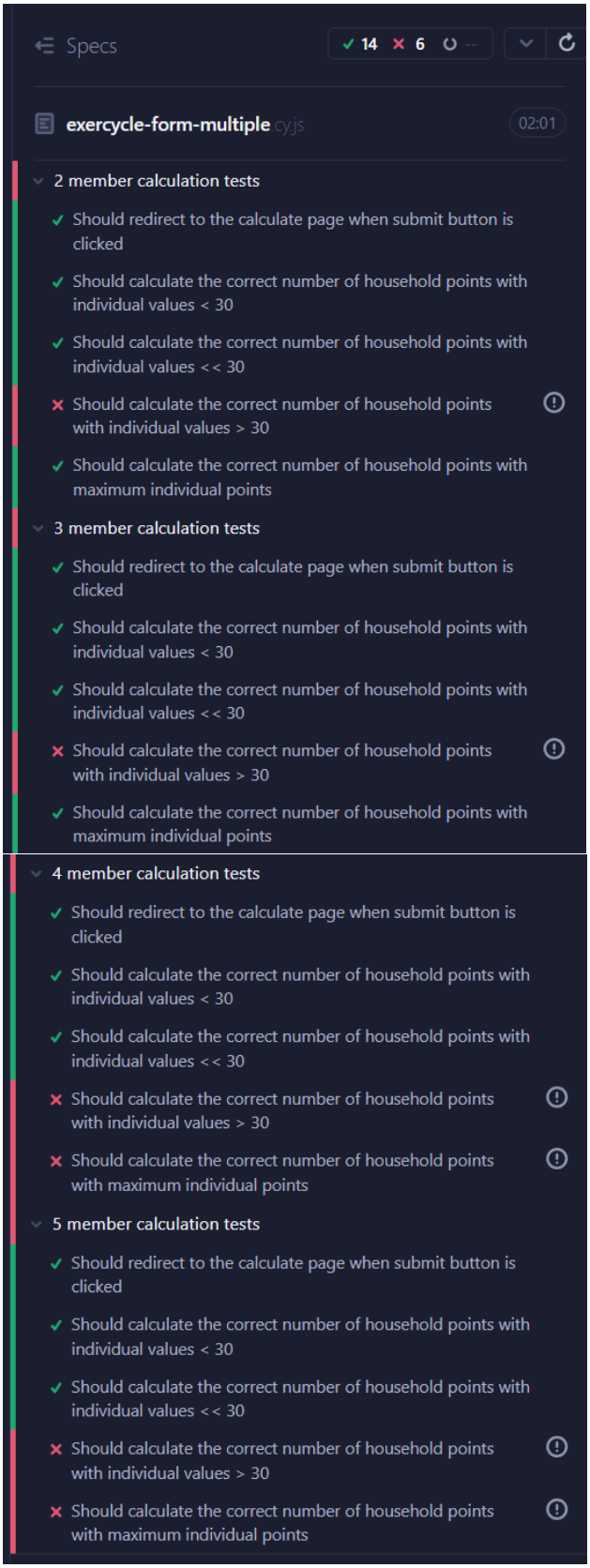
[1] https://github.com/k1dubv1/exercycle-testing

Appendix


Specs
✓ 1 ✕ -- ↻ --
exercycle-home.cy.js
00:02
The home page
✓ Should redirect to the form pages


Specs
✓ 9 ✕ 6 ↻ --
exercycle-form-input.cy.js
00:44
Single member input tests
✓ Should not accept a daily value greater than 7
✕ Should not allow negative points
✕ Should not allow large negative points
✓ Should only allow whole numbers and no decimals
✕ Should display 0 in the table when all inputs are empty
✕ Should not allow numbers in scientific format
✓ Should allow leading 0s
✓ Should allow many leading 0s
Multiple member input tests
✕ Should calculate correctly and show correct table when one individual has empty input fields
✓ Should calculate correctly and show correct table when one individual has leading 0s
Alternative input tests
✓ Should calculate correctly when updated with the up arrow key
✓ Should calculate correctly when updated with the up and down arrow keys
✓ Should not allow more than 7 by using the up arrow
✕ Should not allow negatives by using the down arrow
✓ Should redirect to the calculate page when enter key is pressed in the form


Specs
✓ 4 ✕ 1 ↻ --
exercycle-form-individual.cy.js
00:12
General tests
✓ Should redirect to the calculate page when submit button is clicked
Calculation tests
✓ Should display the same number when one daily value is inputted
✓ Should display correct calculation when multiple valid values are inputted
✕ Should not have more than 30 weekly points when total daily points > 30
✓ Should calculate correctly when there are empty fields


Specs
✓ 14 ✕ 6 ↻ --
exercycle-form-multiple.cy.js
02:01
2 member calculation tests
✓ Should redirect to the calculate page when submit button is clicked
✓ Should calculate the correct number of household points with individual values < 30
✓ Should calculate the correct number of household points with individual values << 30
✕ Should calculate the correct number of household points with individual values > 30
✓ Should calculate the correct number of household points with maximum individual points
3 member calculation tests
✓ Should redirect to the calculate page when submit button is clicked
✓ Should calculate the correct number of household points with individual values < 30
✓ Should calculate the correct number of household points with individual values << 30
✕ Should calculate the correct number of household points with individual values > 30
✓ Should calculate the correct number of household points with maximum individual points
4 member calculation tests
✓ Should redirect to the calculate page when submit button is clicked
✓ Should calculate the correct number of household points with individual values < 30
✓ Should calculate the correct number of household points with individual values << 30
✕ Should calculate the correct number of household points with individual values > 30
✕ Should calculate the correct number of household points with maximum individual points
5 member calculation tests
✓ Should redirect to the calculate page when submit button is clicked
✓ Should calculate the correct number of household points with individual values < 30
✓ Should calculate the correct number of household points with individual values << 30
✕ Should calculate the correct number of household points with individual values > 30
✕ Should calculate the correct number of household points with maximum individual points