# Energy Monitor Testing Plan and Test Execution

Contents

[Energy Monitor Testing Plan and Test Execution 1](#_Toc105972367)

[Test Statistics 3](#_Toc105972368)

[Test Inconclusive 3](#_Toc105972369)

[Test Failures 3](#_Toc105972370)

[Validation Testing 4](#_Toc105972371)

[Test V.H.Home.html 4](#_Toc105972372)

[Test V.H.Usage.html 4](#_Toc105972373)

[Test V.H.Panels.html 4](#_Toc105972374)

[Test V.H.Hourly.html 4](#_Toc105972375)

[Test V.H.Daily.html 4](#_Toc105972376)

[Test V.H.Help.html 4](#_Toc105972377)

[Test V.H.Faq.html 4](#_Toc105972378)

[Test V.H.Contact.html 4](#_Toc105972379)

[Test V.H.Support.html 4](#_Toc105972380)

[Test V.C.Style.css 5](#_Toc105972381)

[Test V.C.Views.css 5](#_Toc105972382)

[Test V.C.Map.css 5](#_Toc105972383)

[Test V.C.Form.css 5](#_Toc105972384)

[Test V.C.Input.css 5](#_Toc105972385)

[Unit Tests 6](#_Toc105972386)

[Test I.A – API test suite passes all tests (5 tests) 6](#_Toc105972387)

[Test I.S – SolarArrayModel and Intensity Calculator pass all tests (7 tests) 6](#_Toc105972388)

[Usage Tests 7](#_Toc105972389)

[Test U.Home.a – Navigating to home.html, the map selection screen loads with no console errors 7](#_Toc105972390)

[Test U.Home.b – Pressing on a location redirects you to usage.html 7](#_Toc105972391)

[Test U.Home.c – Pressing on a location saves the location to local storage 8](#_Toc105972392)

[Test U.Usage.a – Navigating to usage.html, the usage input screen loads with no console errors 8](#_Toc105972393)

[Test U.Usage.b – Numbers can be added to the usage input field using the on-screen numpad 9](#_Toc105972394)

[Test U.Usage.c – Pressing on the confirm button on the usage page redirects you to panels.html 9](#_Toc105972395)

[Test U.Usage.d – Pressing on the confirm button saves the daily usage to local storage 10](#_Toc105972396)

[Test U.Panels.a – Navigating to panels.html, the panel information input screen is loaded 10](#_Toc105972397)

[Test U.Panels.b – Pressing the confirm button changes the input field to the next one down 11](#_Toc105972398)

[Test U.Panels.c – Pressing the confirm button with direction selected redirects to daily.html 11](#_Toc105972399)

[Test U.Panels.d – Pressing the confirm button saves panel information to local storage 12](#_Toc105972400)

[Test U.Daily.a - Navigating to daily.html, the daily view screen is loaded 12](#_Toc105972401)

[Test U.Daily.b – Selecting a different day shows the appropriate output 13](#_Toc105972402)

[Test U.Daily.c – Clicking on the selected day opens the hourly view for the day 13](#_Toc105972403)

[Test U.Daily.d – Weather warnings are shown when there is a critical weather warning in the area 14](#_Toc105972404)

[Test U.Hourly.a - Navigating to hourly.html, the hourly view screen is loaded 14](#_Toc105972405)

[Test U.Hourly.b - Clicking on the selected hour opens the daily view 15](#_Toc105972406)

[Test U.Support.a - Navigating to support.html, the support page is loaded 15](#_Toc105972407)

[Test U.Support.b – A support request can be submitted 16](#_Toc105972408)

[Test U.Navigation.a – Clicking on the “Energy Monitor” header redirects to map screen 17](#_Toc105972409)

[Test U.Navigation.b – Clicking the hamburger icon toggles the navigation menu 17](#_Toc105972410)

[Test U.Navigation.c – Pressing the buttons on the navigation menu takes you to the appropriate pages 18](#_Toc105972411)

[Test U.Navigation.d – Pressing the “help” section on the navigation menu opens the help section 18](#_Toc105972412)

[Test U.Darkmode – Pressing the “dark mode” button on the navigation menu toggles the application between light and dark mode 19](#_Toc105972413)

[Accessibility Tests 20](#_Toc105972414)

[Test A.HighContrast – All application elements are visible in a high contrast mode 20](#_Toc105972415)

[application between light and dark mode 20](#_Toc105972416)

[Test A.NoKeyboard – The application can be used without a keyboard 20](#_Toc105972417)

[Test A.NoCursor – The application can be used without a pointer or cursor 20](#_Toc105972418)

[Test A.Captions – All notable application parts should have captioning to inform state 20](#_Toc105972419)

[Test A.Alt-text – All image based assets should have an alternate format 21](#_Toc105972420)

[Test A.Contrast – The application should have a high level of contrast between elements 21](#_Toc105972421)

[Test A.Font – The application uses an easily read and understood font-family 21](#_Toc105972422)

[Test A.Maintainability – The application is maintainable to be upgraded for use with new assistive technologies 21](#_Toc105972423)

## Test Statistics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Section | Validation Tests | Unit Tests | Usage Tests | Accessibility Tests |
| Total Tests | 14 | 12 | 24 | 8 |
| Test Passes | 14 | 12 | 22 | 7 |
| Test Inconclusive | 0 | 0 | 0 | 1 |
| Test Failures | 0 | 0 | 1 | 0 |
| Test Pass Percentage | 100% | 100% | 92% | 88% |

## Test Inconclusive

1. [A.Alt-text](#_Test_A.Alt-text_–)

## Test Failures

1. [U.Daily.d](#_Test_U.Daily.d_–)
2. [U.Support.b](#_Test_U.Support.b_–)

# Validation Testing

All HTML validation was done using the w3c validation service (<https://validator.w3.org/>)

All CSS validation was done using the w3c CSS validation service (<https://jigsaw.w3.org/css-validator/>)

## Test V.H.Home.html

Graphical user interface, text, application, chat or text message

Description automatically generated

## Test V.H.Usage.html

Graphical user interface, text, application

Description automatically generated

## Test V.H.Panels.html

Graphical user interface, text, application, chat or text message

Description automatically generated

## Test V.H.Hourly.html

Graphical user interface, text

Description automatically generated

## Test V.H.Daily.html

Graphical user interface, text, application, chat or text message

Description automatically generated

## Test V.H.Help.html

Graphical user interface, text, application, chat or text message

Description automatically generated

## Test V.H.Faq.html

Graphical user interface, text, application, chat or text message

Description automatically generated

## Test V.H.Contact.html

Graphical user interface, text, application, chat or text message

Description automatically generated

## Test V.H.Support.html

Graphical user interface, text, application, chat or text message

Description automatically generated

## Test V.C.Style.css

A picture containing graphical user interface

Description automatically generated

## Test V.C.Views.css

A picture containing graphical user interface

Description automatically generated

## Test V.C.Map.css

A picture containing graphical user interface

Description automatically generated

## Test V.C.Form.css

A picture containing diagram

Description automatically generated

## Test V.C.Input.css

A picture containing diagram

Description automatically generated

# Unit Tests

## Test I.A – API test suite passes all tests (5 tests)

Expected Result: The API test suite passes all 5 tests for functionality and request checking

Actual Result: The API passes all 5 tests

Evidence:

Text

Description automatically generated

## Test I.S – SolarArrayModel and Intensity Calculator pass all tests (7 tests)

Expected Result: The jest test suite passes all 7 tests for calculating values based off reference data

Actual result: The models pass all 7 tests

Evidence:

Text

Description automatically generated

# Usage Tests

## Test U.Home.a – Navigating to home.html, the map selection screen loads with no console errors

Expected Result: The map selection screen loads, and no errors are displayed in the web console

Actual Result: The map selection screen loads, there are no errors displayed

Evidence:

Graphical user interface, application, Word

Description automatically generated

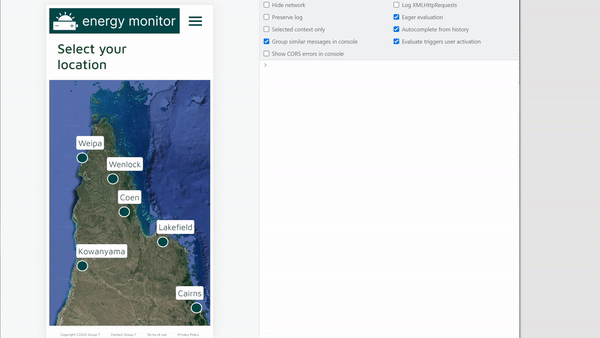
## Test U.Home.b – Pressing on a location redirects you to usage.html

Expected Result: upon pressing a button, the page href is changed to usage.html

Test Data: Pressing of button for “Wenlock”

Actual Result: The usage page loads

Evidence:

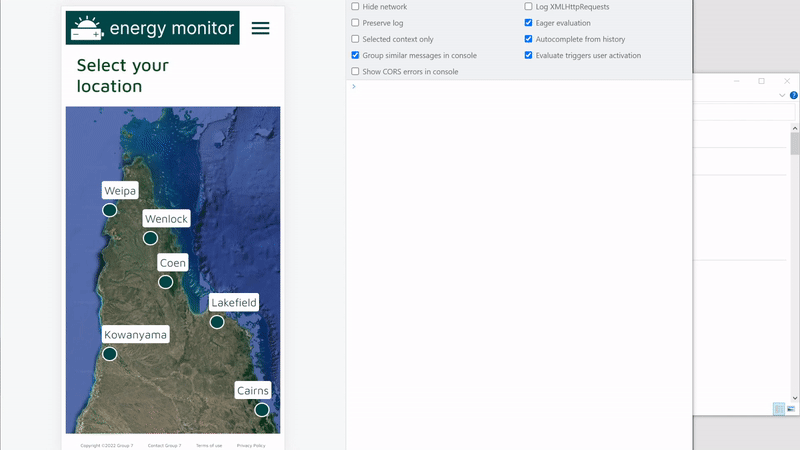


## Test U.Home.c – Pressing on a location saves the location to local storage

Expected result: after changing the location, we can get a log of the updated location with LocalDataManger.getCity() and LocalDataManager.getLocation()

Actual result: after changing the location, the new city name and coordinates can be logged

Evidence:



## Test U.Usage.a – Navigating to usage.html, the usage input screen loads with no console errors

Expected result: The usage input screen loads and there are no errors displayed in the web console

Actual Result: The usage screen loads, there are no errors displayed

Evidence:

Graphical user interface, application

Description automatically generated

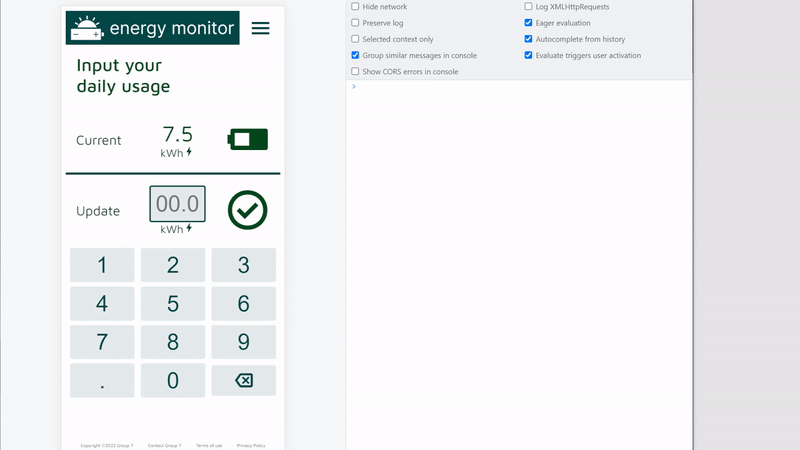
## Test U.Usage.b – Numbers can be added to the usage input field using the on-screen numpad

Expected result: upon pressing a number, it is added to the input field. Upon pressing the “.” Symbol, a decimal point is added and pressing the “x” button removes the last character in the field

Test Data: 2 + “.” + 7 + 9 + “x”

Actual Result: the input correctly registers the final input as 2.7

Evidence:

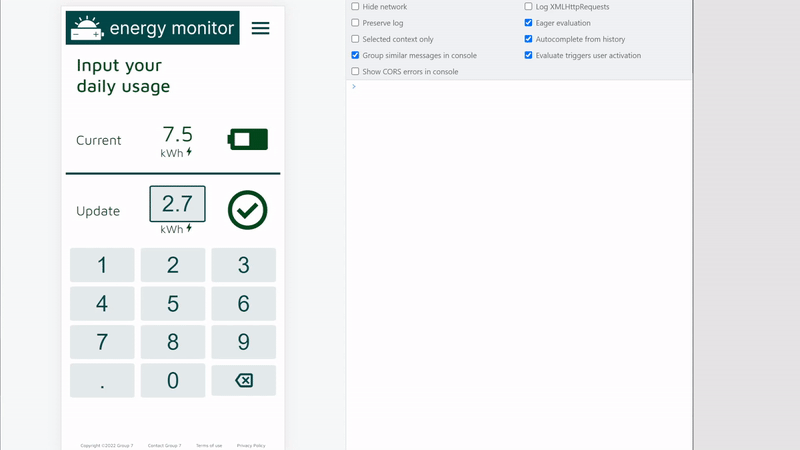


## Test U.Usage.c – Pressing on the confirm button on the usage page redirects you to panels.html

Expected result: upon pressing the confirm button, the page href is changed to panels.html

Actual result: the panels page loads

Evidence:



## Test U.Usage.d – Pressing on the confirm button saves the daily usage to local storage

Expected result: after changing the location, we can get a log of the updated location with LocalDataManager.getUsage()

Actual result: after changing the usage, the usage can be logged

Evidence:



## Test U.Panels.a – Navigating to panels.html, the panel information input screen is loaded

Expected result: The panel information input screen loads and there are no errors displayed in the web console

Actual Result: The panels screen loads, there are no errors displayed

Evidence:

Graphical user interface, application

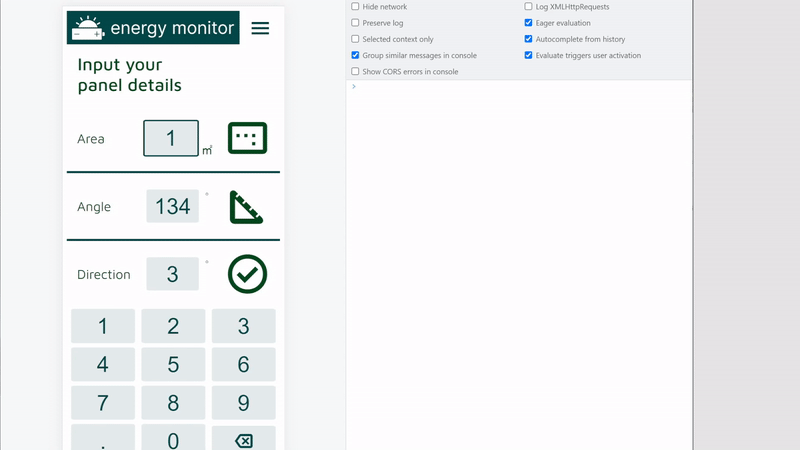
Description automatically generated

## Test U.Panels.b – Pressing the confirm button changes the input field to the next one down

Expected Result: The “selected” input field moves from area to angle, then to direction

Actual result: the selected input field changes from area to angle to direction

Evidence:

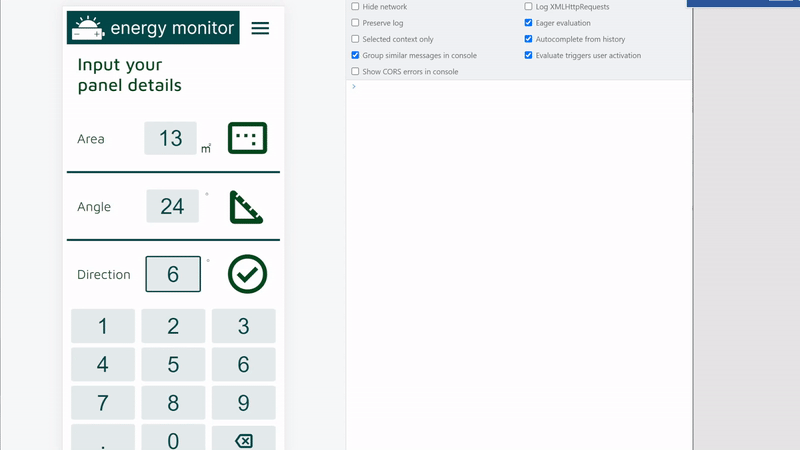


## Test U.Panels.c – Pressing the confirm button with direction selected redirects to daily.html

Expected result: upon pressing the confirm button, the page href is changed to daily.html

Actual result: the daily view page loads

Evidence:

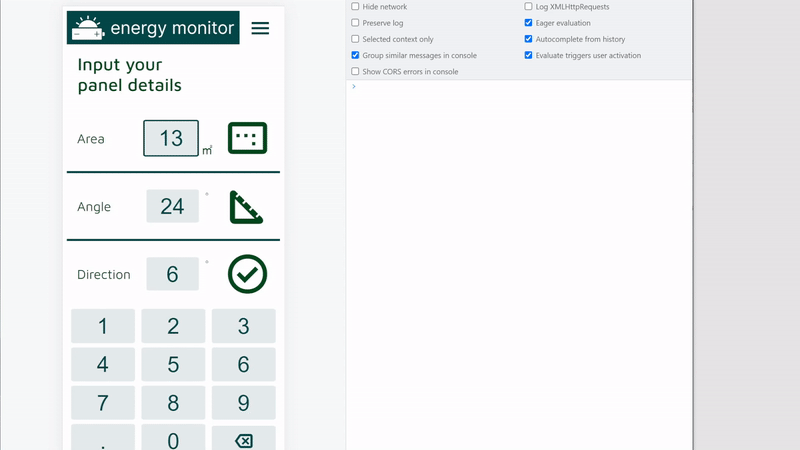


## Test U.Panels.d – Pressing the confirm button saves panel information to local storage

Expected result: after changing the panel info on panels.html,

Actual result: after changing the usage, we can get a log of the updated panel information with LocalDataManger.getPanelInfo()

Evidence:



## Test U.Daily.a - Navigating to daily.html, the daily view screen is loaded

Expected result: The daily view screen loads and there are no errors displayed in the web console

Actual Result: The daily view screen loads, there are no errors displayed

Evidence:

Graphical user interface, application

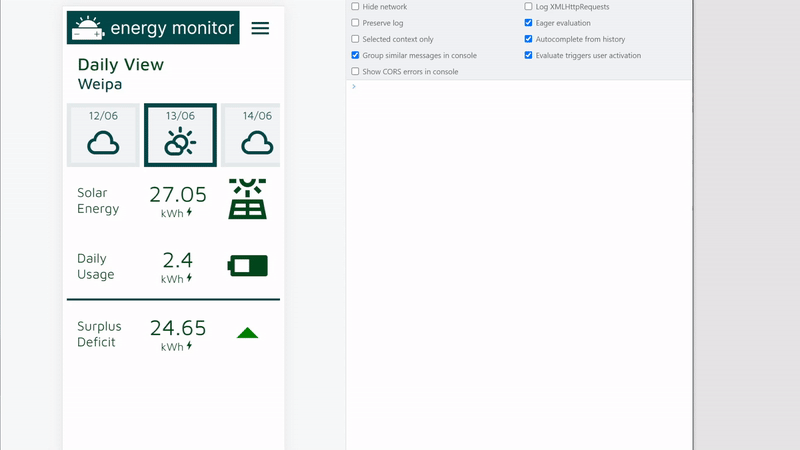
Description automatically generated

## Test U.Daily.b – Selecting a different day shows the appropriate output

Expected result: the selected days totals are shown, the selected box changes

Actual Result: the page reloads and the new input, output and surplus/deficit figures change, the new date box is highlighted

Evidence:

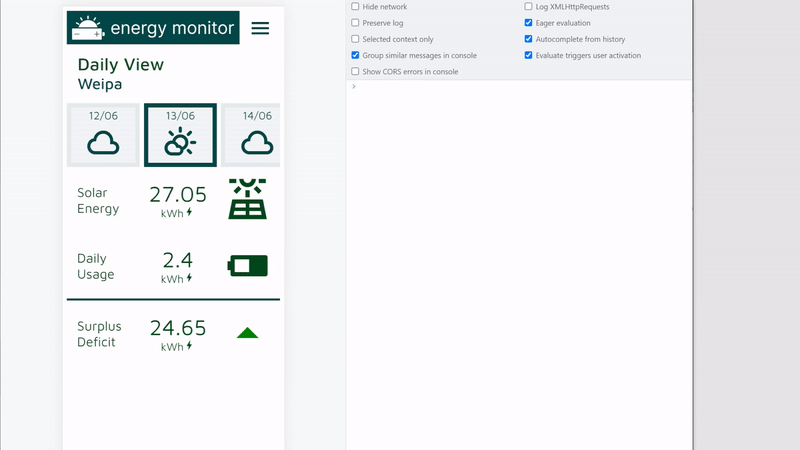


## Test U.Daily.c – Clicking on the selected day opens the hourly view for the day

Expected result: the hourly view is loaded for the date clicked

Actual Result: the page changes to hourly.html and the hourly view for the day is loaded

Evidence:



## Test U.Daily.d – Weather warnings are shown when there is a critical weather warning in the area

Expected result: When a critical weather warning is in the area, a warning box is shown

Actual Result: As this feature is a non-functional prototype, it is shown in all situations

Evidence:

Graphical user interface, application, Teams

Description automatically generated

## Test U.Hourly.a - Navigating to hourly.html, the hourly view screen is loaded

Expected result: The hourly view screen loads given that a day and time is given in the URL

Test Data: hourly.html?day="06/13/22&time="01:00"#loaded

Actual Result: The hourly view screen loads, there are no errors displayed

Evidence:

Graphical user interface, application

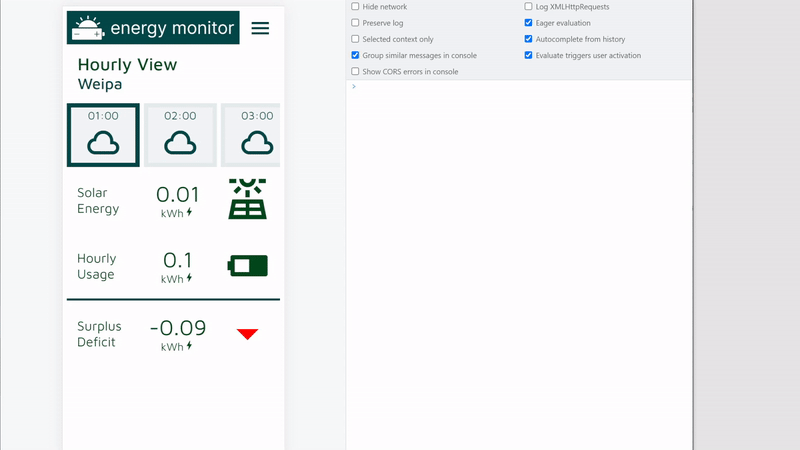
Description automatically generated

## Test U.Hourly.b - Clicking on the selected hour opens the daily view

Expected result: the daily view is loaded

Actual Result: the page changes to daily.html and the daily view for the next 5 days is loaded

Evidence:



## Test U.Support.a - Navigating to support.html, the support page is loaded

Expected result: The support page loads

Actual Result: The support screen loads, there are no errors displayed

Evidence:

Graphical user interface, text, application

Description automatically generated

## Test U.Support.b – A support request can be submitted

Expected result: After pressing the submit button a support ticket is opened

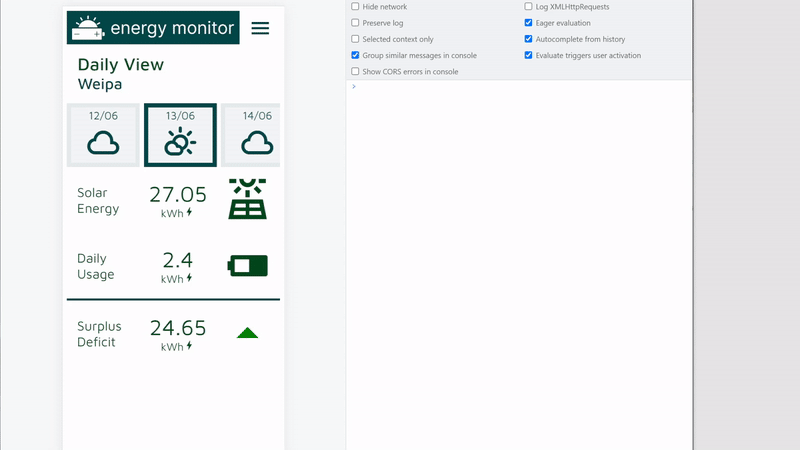
Actual Result: Nothing happens, the support ticket system was not built into this prototype

Test U.Navigation.a – Clicking on the “Energy Monitor” header redirects to map screen

Expected result: the location selection screen (home.html) is loaded

Actual Result: home.html is loaded

Evidence:

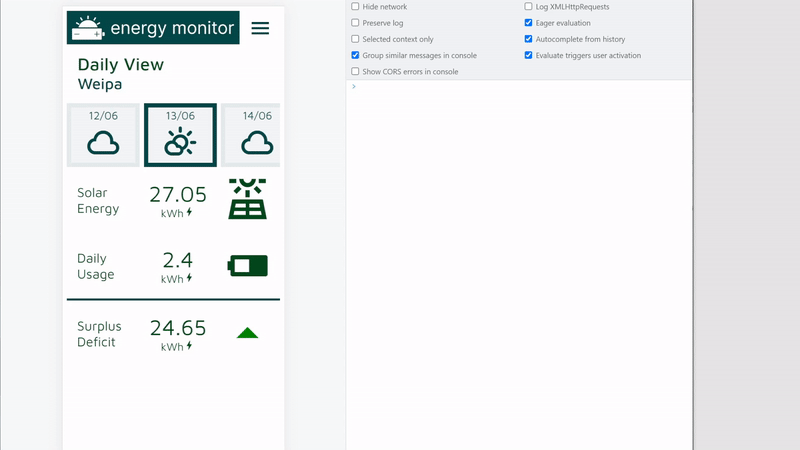


## Test U.Navigation.b – Clicking the hamburger icon toggles the navigation menu

Expected result: the navigation menu is toggled open when the menu is closed and toggled off when the menu is open

Actual Result: the navigation menu is toggleable

Evidence:

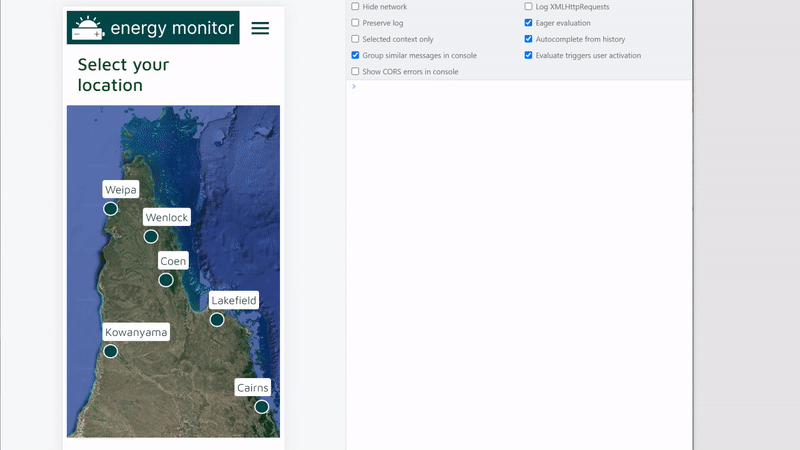


## Test U.Navigation.c – Pressing the buttons on the navigation menu takes you to the appropriate pages

Expected result: upon clicking the button there is a redirect to the relevant page

Actual Result: all buttons redirect to their relevant pages

Evidence:



## Test U.Navigation.d – Pressing the “help” section on the navigation menu opens the help section

Expected result: there is a redirect to help.html

Actual Result: the page redirects to help.html

Evidence:

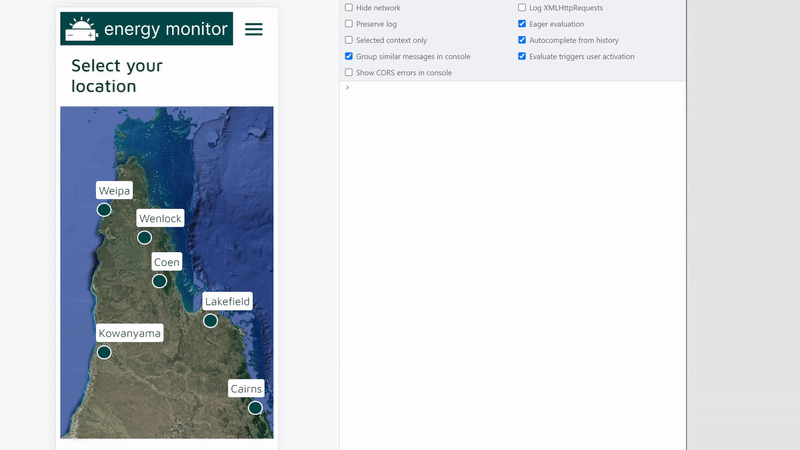


## Test U.Darkmode – Pressing the “dark mode” button on the navigation menu toggles the application between light and dark mode

Expected result: the application switches between light and dark mode as the setting is toggled

Actual Result: dark mode is applied when the setting is enabled and dark mode is disabled when the setting is disabled

Evidence:



# Accessibility Tests

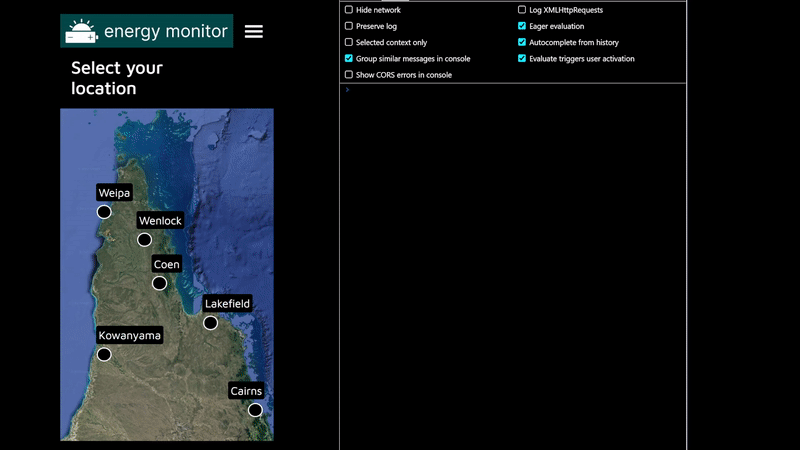
## Test A.HighContrast – All application elements are visible in a high contrast mode

## application between light and dark mode

Expected result: the application is both easily read and usable in a high contrast mode

Actual Result: the application is easily read and all elements and actions are distinguishable in high contrast mode

Evidence:



## Test A.NoKeyboard – The application can be used without a keyboard

Expected result: the application can be used using exclusively a pointer or cursor

Actual result: the application can be used using exclusively a cursor as it was designed to be a tap-only application.

## Test A.NoCursor – The application can be used without a pointer or cursor

Expected result: the application can be used exclusively with a keyboard or similar assistive technology

Actual result: due to several elements (notably on the location selection screen) the user is unable to use the application without a pointer or cursor

## Test A.Captions – All notable application parts should have captioning to inform state

Expected result: all essential and notable parts of the application have some description of state

Actual result: all pages have <h2> captioning to inform the user of the current page and what is expected to be inputted or displayed

## Test A.Alt-text – All image based assets should have an alternate format

Expected result: all image based assets should have an alt attribute with a reasonable description of their corresponding image

Actual result: the energy monitor logo has a reasonable alt attribute, the map has an alt attribute however it could be more descriptive

Evidence:





## Test A.Contrast – The application should have a high level of contrast between elements

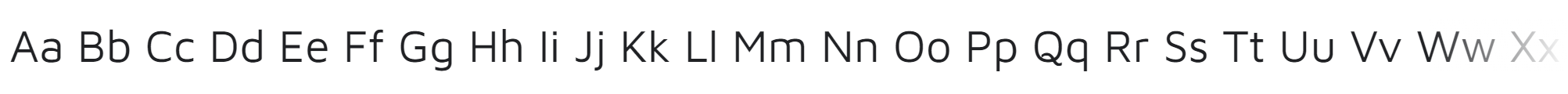
Expected result: the application has a high level of contrast between elements, allowing for easier viewing

Actual result: the application colour scheme was built for high glare environments and thus is designed around a high contrast concept of white on black and black on white

## Test A.Font – The application uses an easily read and understood font-family

Expected result: the application has a block font which is easily read and understood

Actual result: the application uses “Maven Pro” which is a sans-serif derivative block font



## Test A.Maintainability – The application is maintainable to be upgraded for use with new assistive technologies

Expected results: the application is maintainable to the point where implementing new assistive technologies would not require a complete redesign of the codebase

Actual result: The code is built on a node.js + HTML5 stack which lends itself to being maintainable as it is a widely known skillset, node.js can be easily expanded upon through packages and dependencies can be controlled using a manager such as NPM. As for the code it is inline with the code style requirements and upmost attention has been paid to sensible program flow, unit testing and utilizing non-ambiguous naming