

Test Specification

For CyberSmart (Senior QA Automation Engineer)

AUTHOR: SEAN PATERSON

DATE: 03-JUN-2021

# INTRODUCTION

This document is a software test specification and is to be used as part of the interview assessment process for the position of Senior QA Automation Engineer at [CyberSmart](https://apply.workable.com/cybersmart/j/C72DB24CE2/). This is a standalone document that describes the test approach and a sample of manual test cases for the target software under test – the shopping web application ‘Dat Shopping List’. Any manual test case that is automated will be highlighted alongside its manual test case accordingly.

# REFERENCES

Please see APPENDIX A – TECHNICAL CHALLENGE FOR SENIOR QA AUTOMATION ENGINEER[[1]](#footnote-1)

# RESOURCES

Hardware resources required for manual test execution:

* Test PC
* Test mobile phone (iPhone, Android, etc.)
* Tablet devices (including iPad, Kindle, Android, etc.)

Other resources for manual test execution include a version of any current mainstream browser (such as Mozilla Firefox), a live internet connection plus a copy of this test specification document.

For automated test execution, please access the GitHub repository at <https://github.com/SMKPaterson/CyberSmart>. Tools used during the creation of the automated test files:

* [Python](https://www.python.org/downloads/) v.3.9.4
* [PyCharm](https://www.jetbrains.com/pycharm/download/#section=windows) (Community Edition) IDE.
* Selenium Chrome [webdriver](https://chromedriver.chromium.org/downloads).[[2]](#footnote-2)
* [PyTest](https://docs.pytest.org/en/reorganize-docs/getting-started.html) testing framework and [HTML reporting](https://pypi.org/project/pytest-html/).
* Selenium with Python [documentation](https://selenium-python.readthedocs.io/).
* [ChroPath](https://chrome.google.com/webstore/detail/chropath/ljngjbnaijcbncmcnjfhigebomdlkcjo) plugin for Chrome.

The PyCharm IDE project settings were set to use the Python 3.9.4 exe as its Python interpreter. The “pip” and “selenium” packages were also installed as packages within the project’s settings. PyTest and the PyTest html reporter were installed using pip (pip install pytest / pip install pytest-html).

In developing test case coverage the mind map in APPENDIX F – Mind Map of Test Coverage was created using the freely available GitHub mind-mapping tool [gitmind](https://gitmind.com/app/doc/7bf86d8366cf4c572481f7cc6f95c28f).

# SOFTWARE UNDER TEST

The web application at the following URL is the subject of this test specification:

<https://dat-shopping.herokuapp.com/>

There will be no deployment of any code fixes for the scope of this test. Therefore, only one version – the current version - of the web application will be tested.

# ISSUE MANAGEMENT

Issues will not be formally raised since this is a demo project.

# TEST CASES

This section covers a sample of manual test cases, both positive and negative. A fuller list of test case names and objectives is given in APPENDIX G – TEST CASE MATRIX. This can be amplified to create a fuller set of manual test cases, if required. APPENDIX G – TEST CASE MATRIX also includes a column indicating whether a test case has been automated. Details for executing the automated test case will be given at the end of the manual test case.

### Home Page & Enter New Shopping List Page Content and Defaults Verification

Objective: Verifies the home page format and layout. Then checks that a new page displays when the user chooses to create a new shopping list and that the default state of the *Submit* button on the new page is disabled (i.e., bars null entry).

|  |  |  |  |
| --- | --- | --- | --- |
| # | Test Step | Test Data | Expected Result |
| 1 | Open browser at home page. | <https://dat-shopping.herokuapp.com/> | Dat Shopping List web application’s home page displays. |
| 2 | Check home page’s content, including header, main body and footer. |  | All screen content should render as shown in screenshot in APPENDIX B – Home Page Screen. |
| 3 | Click “Create a new Shopping List” button. |  | Web page URL navigates to <https://dat-shopping.herokuapp.com/create>. |
| 4 | Check the screen content |  | Layout and format should be as shown in APPENDIX C – Enter Name for New Shopping List Screen.  In the main body of the screen, only one field, labelled “Enter a name for your new shopping list”, will be displayed.  Below the field a Submit button should render and be disabled. |
| 5 | Click on the Submit button. |  | The action should **not** be respected. The screen should remain as is. |
| 6 | Teardown: Click on the “Dat Shopping List” logo in the top left of the screen. |  | The page should return to <https://dat-shopping.herokuapp.com/>.  The screen content should still render as shown in screenshot in APPENDIX B – Home Page Screen. |

### Home Page – Paste Link

Objective: Verifies the operation of the 'Have a link? Paste it here.' button. Also checks for unacceptable input.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Test Step | Test Data | Expected Result |
| 1 | Open browser at home page. | <https://dat-shopping.herokuapp.com/> | Dat Shopping List web application’s home page displays. |
| 2 | Click on ‘Have Link? Paste it here.’ Button. |  | The Paste Link screen displays and content and format should be as shown in APPENDIX E – Paste Link Screen. |
| 3 | Click on data entry field. |  | It should be initially blank. Focus should be indicated by blinking caret. |
| 4 | Type in URL and click Submit button. | [www.bbc.co.uk](http://www.bbc.co.uk) | It should be refused with a popup warning stating “URL or key must contain a 64 character hex value”. |
| 5 | Click OK button on popup |  | Screen with URL from Step 4 persists. |
| 6 | Delete the entered URL and click Submit button again. |  | The button press will not be respected.  The form with the blank data entry field will persist.  There will be no visual cue that a null entry is unacceptable. |
| 7 | Teardown: Click on the “Dat Shopping List” logo in the top left of the screen. |  | The page should return to <https://dat-shopping.herokuapp.com/>.  The screen content should still render as shown in screenshot in APPENDIX B – Home Page Screen. |

#### Automated Test:

This test case has been automated. Please see APPENDIX H – AUTOMATION INSTRUCTIONS.

### Create New Shopping List Page – Submit Button Behaviour

Objective: Checks the default widget focus on the Enter Name for New Shopping List screen and operation of the Submit button when a valid name with non-alphanumeric character is entered.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Test Step | Test Data | Expected Result |
| 1 | Open browser at home page. | <https://dat-shopping.herokuapp.com/> | Dat Shopping List web application’s home page displays. |
| 2 | Click “Create a new Shopping List” button. |  | Web page URL navigates to <https://dat-shopping.herokuapp.com/create>. |
| 3 | Without clicking mouse anywhere, type the text string shown in Test Data. | My first new shopping list! | The data entry field should automatically have focus.  The entered text string should be echoed in the field. |
| 4 | Click Submit button. |  | Web page URL navigates to https://dat-shopping.herokuapp.com/doc/[64-char hex number], where the “[64-char hex number]” represents a random-generated 64 character hexadecimal string. |

### New Shopping List Page - Content and Defaults Verification

Objective: Verifies the New Shopping List page format and content in its default state. Expects no items to be in list since it is a new list.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Test Step | Test Data | Expected Result |
| 1 | Pre-requisite: Test “Create New Shopping List Page – Submit Button Behaviour” must have been run and passed. |  |  |
| 2 | Check general content and format |  | The screen content should render as shown in APPENDIX D – New Shopping List Screen. |
| 3 | Check screen content – heading text. |  | The main body of the web page should show name (see Test Data above) of the new shopping list in bold, at the top of the section. |
| 4 | Check 64-digit hex ‘local key’ is displayed. |  | The local key should be rendered in the format ######..##, right-justified, in smaller text, to the right of the main body heading text. |
| 5 | Check the content of the section immediately below the main body heading. |  | It should state “Authorized” in green text with the text string “(Expand to add writer”) following it. A small button with green label text Expand, with down-arrow should display.  Underneath this, there should be **no** shopping items on the list, with the text “No items” since this is a newly created list.  There **should** be an empty text-entry field with a green “Add” button to its side. |
| 6 | Check the page footer |  | This should be as footer shown in screenshot in APPENDIX D – New Shopping List Screen.  There should be a link on the left labelled “Home” and a link on the right labelled “Delete List”. |

### New Shopping List Page - Click on Home

Objective: Verifies clicking 'Home' on New Shopping List page takes the user back to the home page.

Test case steps TBD.

### New Shopping List Page – Click on Delete

Verifies clicking 'Delete' on New Shopping List page first prompts the user to confirm deletion before deleting the current shopping list.

Test case steps TBD.

### Enter New Shopping List Page - Negative Test: SQL Injection

Verifies New Shopping List page handles or refuses entry of 'SQL injection' text strings such as:

' OR 1=1;

Test case steps TBD.

PLEASE REFER TO APPENDIX G – TEST CASE MATRIX FOR A MORE COMPLETE LIST OF TEST OBJECTIVES FOR SUBSEQUENT ‘WORKING UP’.

# ENTRY & EXIT CRITERIA

This section describes entry and exit criteria for the test phases.

## Manual Entry Criteria

* Dat Shopping application is deployed to test server location.
* Full set of manual test cases have been peer-reviewed by test and development team and updated, as necessary.
* All resources are available as listed in RESOURCES.

## Manual Exit Criteria

* All manual test cases have been run and passed.
* All test results are documented and recorded.
* All test results have been reviewed with the project management team.
* Any failed test cases must be reviewed and either a fix or formal workaround agreed.
* Sign off document approved by project test manager.
* Test Environment has been torn down and any resources archived for possible future re-use.

## Automated Entry Criteria

As above but also:

* All automation artefacts at <https://github.com/SMKPaterson/CyberSmart> have been downloaded and deployed to the test PC.

## Automated Exit Criteria

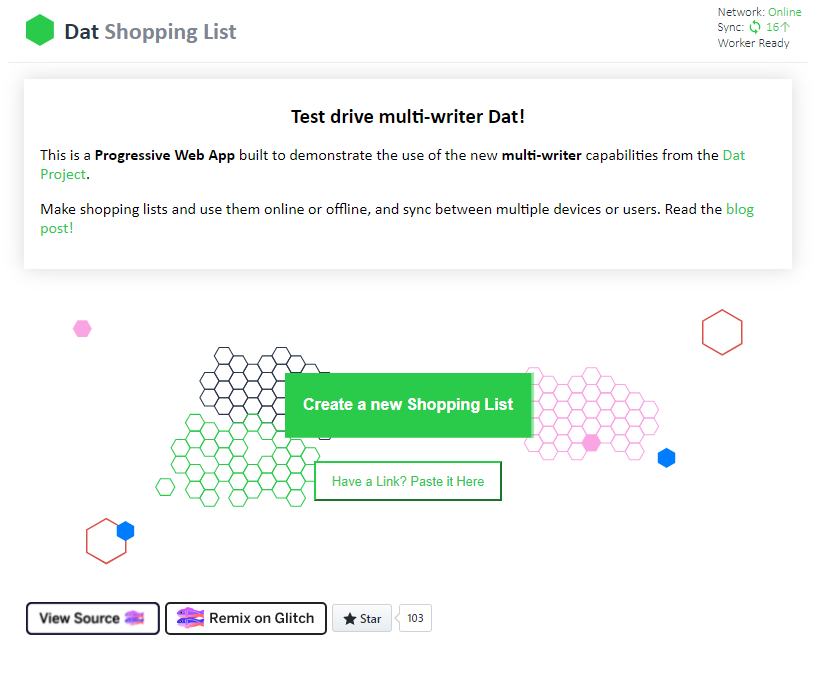
* All automated test cases have been run and passed.
* All test results are documented and recorded.
* All test results have been reviewed with the project management team.
* Any failed test cases must be reviewed and either a fix or formal workaround agreed.
* Sign off document approved by project test manager.
* Automated Test Environment has been torn down and any resources archived for possible future re-use.

# APPENDIX A – TECHNICAL CHALLENGE FOR SENIOR QA AUTOMATION ENGINEER

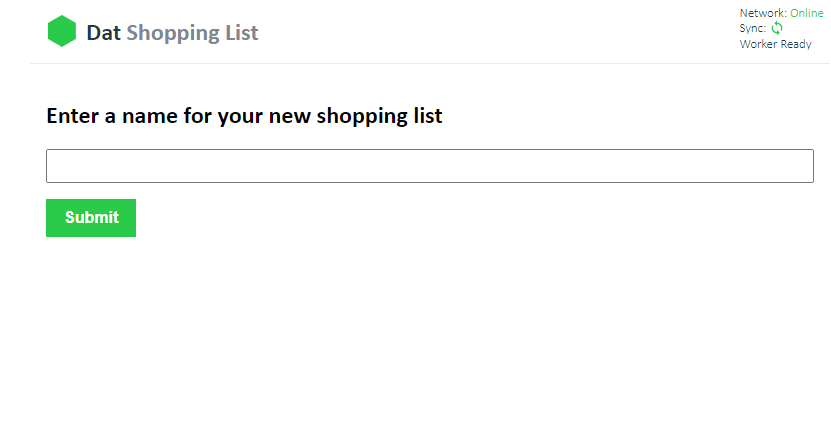
Please open embedded PDF document using Adobe Acrobat Reader or similar.



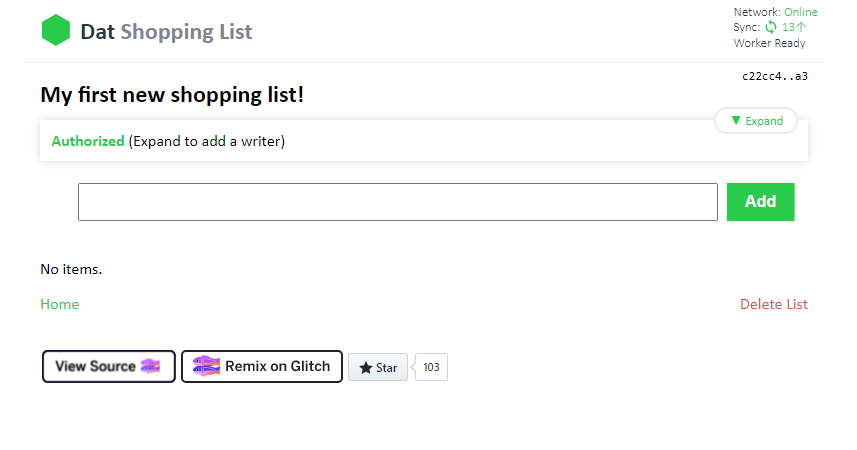
# APPENDIX B – Home Page Screen



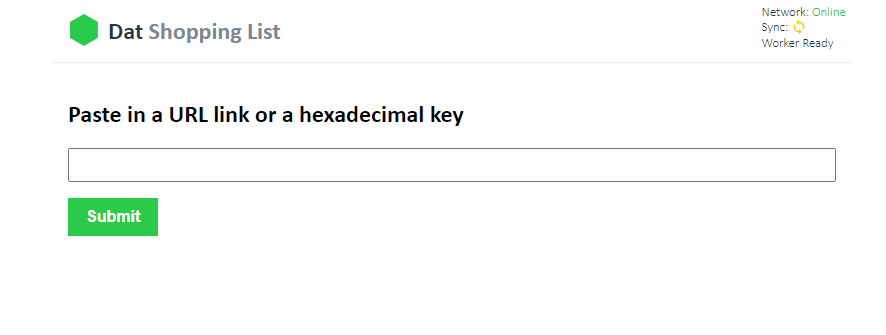
# APPENDIX C – Enter Name for New Shopping List Screen



# APPENDIX D – New Shopping List Screen

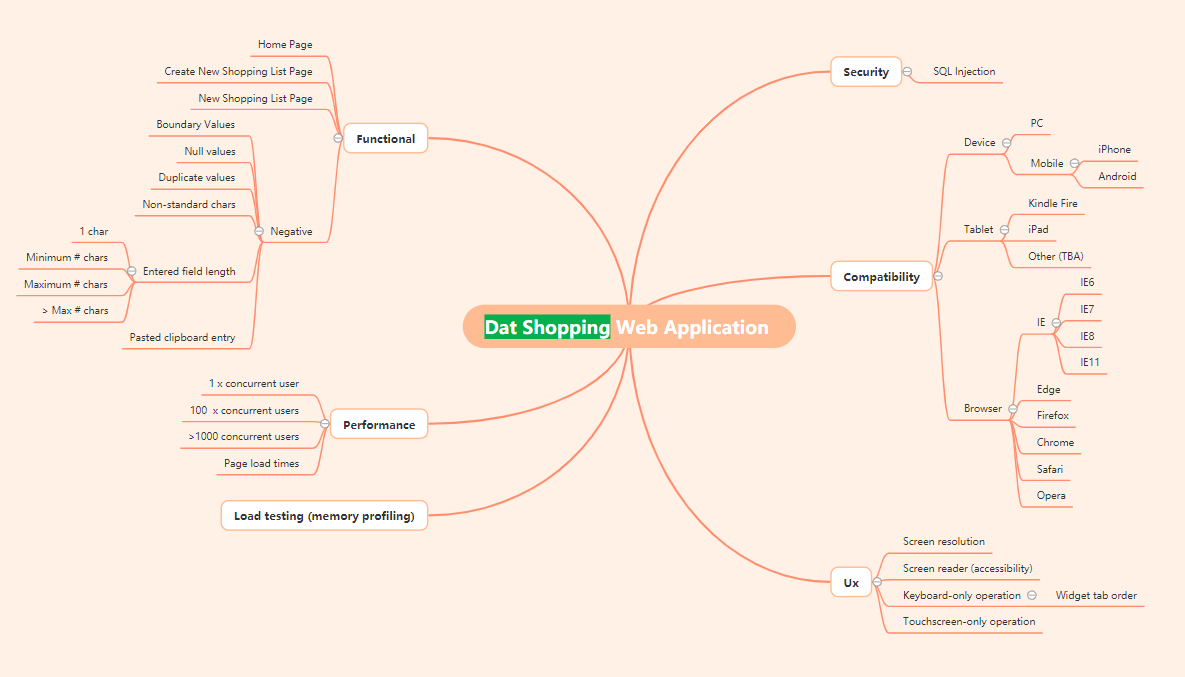


# APPENDIX E – Paste Link Screen



# APPENDIX F – Mind Map of Test Coverage

The following represents the current planned test coverage and is not exhaustive. Test cases can be created by referring along to the tip of each node:



# APPENDIX G – TEST CASE MATRIX

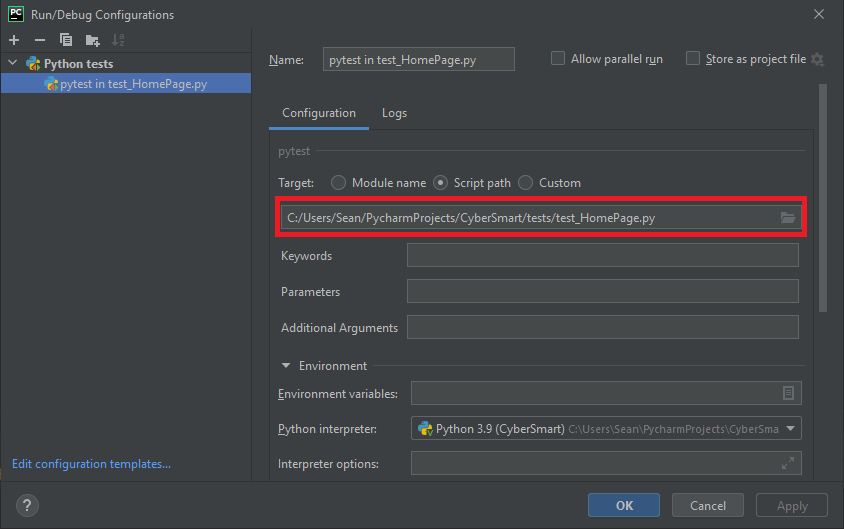
This represents an incomplete test matrix, amplifying the content of the mind map in APPENDIX F – Mind Map of Test Coverage.

| **#** | **TEST CASE NAME** | **TYPE** | **OBJECTIVE** | **Automated?** |
| --- | --- | --- | --- | --- |
| 1 | Home Page & Create New Shopping List Page Content and Defaults Verification | Functional | Verifies the home page format and layout. Then checks that a new page displays when the user chooses to create a new shopping list and that the default state of the Submit button on the new page is disabled. | N |
| 2 | Home Page - Paste Link | Functional | Verifies the operation of the 'Have a link? Paste it here.' button. Also checks for unacceptable input. | Y |
| 3 | Enter New Shopping List Page – Submit Button Behaviour | Functional | Checks the default widget focus on the Enter Name for New Shopping List screen and operation of the Submit button when a valid name with non-alphanumeric character is entered. | N |
| 4 | New Shopping List Page - Content and Defaults Verification | Functional | Verifies the New Shopping List page format and content in its default state. Expects no items to be in list since it is a new list. | N |
| 5 | New Shopping List Page - Click on Home | Functional | Verifies clicking 'Home' on New Shopping List page takes the user back to the home page. | N |
| 6 | New Shopping List Page - Click on Delete | Functional | Verifies clicking 'Delete' on New Shopping List page first prompts the user to confirm deletion before deleting the current shopping list. | N |
| 8 | Enter New Shopping List Page - Negative Test: SQL Injection | Security | Verifies New Shopping List page handles or refuses entry of 'SQL injection' text strings such as: ' OR 1=1; | N |
| 9 | Enter New Shopping List Page - Negative Test: >255 Chars Entered | Functional | Verifies New Shopping List page can handle a very long shopping list text string, > 255 chars. | N |
| 10 | Enter New Shopping List Page - Add A Second Shopping List | Functional | Verifies the user can enter a second shopping list | N |
| 11 | Enter New Shopping List Page - Add A Shopping List with a Duplicate Name | Functional | Verifies the user can enter a shopping list name that has already been created but that its 64-hex local key remains unique. | N |
| 12 | New Shopping List Page - Expand & Collapse the Authorized Section | Functional | Verifies expected operation of the ‘Expand’ / ‘Collapse’ toggle button. | N |
| 13 | New Shopping List Page – Copy URL & Share on Another Browser (same device) | Functional | Verifies New Shopping List page prompts the user to copy URL & share on another browser on same device. | N |
| 14 | New Shopping List page – Copy URL & Share on Another Device | Functional | Verifies New Shopping List page prompts the user to copy URL & can share on a completely different device such as Android phone. | N |
| 15 | New Shopping List Page – Add a Writer with Null Local Key Value | Functional | Verifies clicking ‘Authorize’ button on New Shopping List page without first entering a valid Local Key value prompts the user to enter a valid Local Key. | N |
| 16 | New Shopping List Page – Add a Writer & Authorize | Functional | Verifies the happy path when a user enters a valid Local Key and clicks Authorize button. | N |
| 17 | New Shopping List Page – Add a Null Item | Functional | Verifies that the user cannot enter a null shopping item to their shopping list. | N |
| 18 | New Shopping List Page – Add an Item | Functional | Verifies happy path that the user can add an item to their shopping list. | N |
| 19 | New Shopping List Page – Boundary Value Data Handling | Functional | Verifies various boundary value entries when a user adds an item to their shopping list, including numeric, non-alpha chars, long and very long strings and BLOBs from the clipboard. | N |
| 20 | New Shopping List Page – Adding Many Items | Functional | Verifies integrity and scrolling of shopping lists with >255 shopping items. | N |
| 21 | New Shopping List Page – Check an Item | Functional | Verifies the happy path when a user clicks a shopping item’s checkbox and its strikethrough format. | N |
| 22 | New Shopping List Page – Delete an Item | Functional | Verifies the happy path when a user clicks ‘X’ char for a shopping item and deletes it. | N |
| 23 | New Shopping List Page – Keyboard Driven Flow | Ux | Verify the operation of the GUI when a user only uses the keyboard to drive the user flow. | N |
| 24 | Dat Shopping Application – WCAG Adherence | Accessibility | Verify the operation of the whole Dat Shopping Web Application with respect to its adherence to W3C’s Web Content Accessibility Guidelines (WCAG). | N |
| 25 | Browser Compatibility | Compatibility | Verify the end-to-end operation of the Dat Shopping Web Application across popular browsers Chrome, Firefox, Edge, Safari, Opera and IE. | N |
| 26 | ETC | ETC | ETC |  |

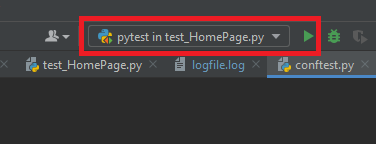
# APPENDIX H – AUTOMATION INSTRUCTIONS

Please set up a development /test PC with the software resources listed in RESOURCES and download the Selenium Python project from github <https://github.com/SMKPaterson/CyberSmart> using git pull origin master.

There are two options to run the test. Firstly, open an elevated command prompt and navigate to ..<your local download location\CyberSmart\tests> and type pytest test\_HomePage.py –browser\_name edge[[3]](#footnote-3). If you wish to see an HTML report output from the PyTest framework then type pytest test\_HomePage.py –html=<name\_of\_your\_report.html> where it will be saved to the *tests* folder. Alternatively, in PyCharm, open the *Run/Debug Configurations* dialog box and set the Script path to the test\_HomePage.py file:



Click Ok then click the green *play* arrow ensuring that pytest is pointing to the correct python test file:



A logfile, *logfile.log*, is created, and appended to, when the test is run and details of any assertion passes/fails are recorded.

1. Note: in the absence of a functional specification, assumptions have been made and reflected in the ‘expected results’ of the test cases in this document. [↑](#footnote-ref-1)
2. Webdriver exe should be installed to C:\Program Files (x86)\ for the automated tests to work [↑](#footnote-ref-2)
3. Use either “edge” or “chrome” at the command prompt to select browser type. Default is chrome if none specified. [↑](#footnote-ref-3)