

# Software Testing Take Away Tasks

We are keen to see how you approach software testing of a website, including logical steps, evidence collection, reporting etc. Anything above and beyond the specified requirements is not necessary but will be recognised.

We understand that your time is precious to you, which is why we have estimated not longer than 3 hours on the following tasks. However, you may feel like spending a little more time on task 3, if this would help you complete the task.

## Before you begin

Please read each tasks instructions, requirements, considerations and restrictions carefully.

If you require further information to clarify any of the requirements, please contact [i.whalley@theidol.com](mailto:i.whalley@theidol.com), remember we're looking for quality rather than quantity.

## Important

The websites you will be testing are public facing and so we restrict the testing performed on them.

Please make sure that you:

- Do not submit any forms (unless stated in the task)
- Do not test outside of the domain provided in each task
- Do not security/penetration test the websites
- Do not attempt to load/volume test the websites

## Returning the Tasks

Once complete, please return to [i.whalley@theidol.com](mailto:i.whalley@theidol.com) and [iwanttowork@theidol.com](mailto:iwanttowork@theidol.com):

- All your files
- In one ZIP file/folder
- No larger than 5MB

You have 5 working days (or time period stated in the email) to complete the tasks. If you complete sooner please return to us, there's no penalty to getting it in earlier or on time!

## Task 1: Exploratory Testing & Reporting

We use exploratory testing, comparing multiple devices/browser, to evaluate a website as a user would navigate through the website journey.

Produce an exploratory testing report in Microsoft Word (.doc, .docs) or Excel (.xls, .xlsx) format on the website: <https://www.theidol.com>, documenting at least three issues that may hinder a user's experience on the website.

Please spend no longer than 1 hour on this task.

The report should include (but is not limited to):

- Steps to recreate the issue found
- Device(s) and browser(s) affected
- Date and time of occurrence
- Screen shot of issue (where appropriate)

Consider:

- User experience
- User interaction
- Accessibility standards
- Target audience

Restrictions:

- There's no need to test validation or submit any forms
- You don't need to test outside of the domain provided
- You don't need to test for security/penetration
- Please don't attempt to load test the website

## Task 2: Scripted Test Scenarios

We use many different types of testing processes at The Idol, including manual and automated scripted test cases. For both, we may use the same catalogue of data.

Produce a Microsoft Excel document (.xls, .xlsx) containing 10 or more rows of test case data which could be used in both manual and automated testing for the following website:

<https://www.comparecover.com/petcover/comparecover/?aff=220>.

The scenarios would be played against the form on the first page before submitting the form. An expected "Pass" would result in progression to the next page (results page). The column names and data from this spreadsheet would be used in the automated test script in task 3.

Please spend no longer than 1 hour on this task.

Your document should include:

- A column with unique test identification
- A column of your expected outcome of the test data
- A column for each field in the website
  - Header/ Top row should easily identify each unique field, this can be the html name or id of the field
  - Cells below the header row may contain the input value or selection
  - Cells can be left blank where the scenario would not access the specified field

Here is an example to help you start:

Case ID	Expected	petType	petName	petPrice	pedigree	breed
1	Pass	dog	Fenton	0	Mongrel	
2	Fail	cat	Rosie Lea	150	Pedigree	Doberman

Consider:

- Explore all fields as some values/choices show/hide different paths through the form
- Valid and invalid values

Restrictions:

- There should be no need to progress past the results page
- You don't need to test outside of the domain provided above
- Please don't test for security/penetration
- Please don't attempt to load/volume test the website
- There's no need to test your scenarios
- Address/Postcode to be used: Loqate, WR5 3DA
- Your scenarios should be restricted to one pet only (do not click "Add another pet")

### Task 3: Automated Test Script

If you have no previous knowledge of automation testing please do not worry, we will assess this task fairly based on your previous experience. We use a few types of automation software at The Idol. One of these technologies is TestCafe, implemented through Node JS that drives actions on selected web browsers and devices.

Produce a Microsoft Word (.doc, .docx) or JavaScript/Node (.js, .txt) format document.

Start by copying the content from the “Test Script” section into a document, completing the script with as many possible steps as you can find whilst investigating the website given.

The website to be used for testing is:

<https://www.comparecover.com/petcover/comparecover/?aff=220>

The code should include (but is not limited to):

- The missing steps between “/\* Start: Script \*/” and “/\* END: Script \*/”
- Similar code and formatting to the surrounding steps
- References to a fictitious set of data, like task 2, referenced as “r.” (e.g. r.petName)

Consider:

- Explore all fields as some values/choices show/hide different paths through the form
- Logical steps/decisions that would change the fields to be populated
- All fields on the form are included in the test script (e.g. pet size, pedigree, select address)

Restrictions:

- You don’t need to script outside of the first page form (pressing Get Quotes)
- There is no need to test outside of the domain provided
- Please don’t test for security/penetration
- Please don’t attempt to load/volume test the website
- There’s no need to run the tests with automation software to test your script
- Whilst investigating the form, use this address/postcode: Loqate, WR5 3DA
- Your code should be restricted to one pet only (do not click “Add another pet”)

Language used in this task

Please note, we do not require you to download/install TestCafe or Node JS or become proficient in either as part of this test. The below links are only for your reference should you wish to read in more detail the code we have used:

- <https://devexpress.github.io/testcafe/documentation/getting-started/>
- <https://www.w3schools.com/nodejs/>
- <https://www.w3schools.com/js/>

## Test Script:

```

test("Test Case: " + r.id, async t => { //START:test

    await t.maximizeWindow().click(".cookieconsent-gotit-button") //full screen and close popup

    if (r.petType === "cat") {
        await t.click("label[for='drpCoverType_cat'] ");
    } else if (r.petType === "dog") {
        await t.click("label[for='drpCoverType_dog']");
    }

    await t.typeText("input[name='nameofpet']", r.petName);

    await t.typeText("input[name='purchasePrice']", r.petPrice);

    /* START: Script */

    /* END: Script */

    await t.click("select[name='petbirth_month']")
        .click(Selector("select[name='petbirth_month']").find("option").withText(r.petMonth));

    await t.click("select[name='petbirth_year']")
        .click(Selector("select[name='petbirth_year']").find("option").withText(r.petYear));

    if (r.petGender === "male") {
        await t.click("label[for='petGender_m']");
    } else if (r.petGender === "female") {
        await t.click("label[for='petGender_f']");
    }

    if (r.petVaccinations === "yes") {
        await t.click("label[for='petVaccinations_1']");
    } else if (r.petVaccinations === "no") {
        await t.click("label[for='petVaccinations_0']");
    }

    /* START: Script */

    /* END: Script */

    await t.typeText("input[id='startdate']", r.policyStart);

    await t.click("button[class='generic_button nextBtn']");

    await t.expect(Selector("a[id='MoreInfoLink_1']").exists).ok("More link not found");

}); //END:test

```