

Test Assignments

The test assignments are in order of importance.

Test Automation assignment details:

Your task is to create a Selenium WebDriver test application using JUnit5 and Maven. The test **suite** should contain 3 tests. The first and third tests should pass meanwhile the second test should fail. The tests should be able to run separately. If you have any questions feel free to ask.

Please follow the test steps from below:

1. Test

Go to [google.com](https://www.google.com)

Search for demoqa.com

Hit the first result which will navigate to demoqa.com

Navigate to the Draggable element in Interactions section

Pick and drop the “Drag me” box into the “Drop here” box, print out the text from the “Drop here” box and take a screenshot.

Click on the ToolTips link on the left hand side in the Widgets section.

Move your cursor to the “Hover me to see” button input element and print out the text from the pop-up tooltip.

2. Test

Go to [google.com](https://www.google.com)

Search for cheese

Use JUnit assertion to compare the number of results to 777.

The test should fail during the comparison with an assertion message "There is too much cheese on the internet".

3. Test

Go to <https://orangehrm-demo-7x.orangehrmlive.com/>

Login with admin credentials (form is prefilled with the login data)

Wait for the dashboard page to load

Go to Recruitment page

Open the left side panel

Click on Candidates

Print out the number of candidates

Click on the green Add button

Fill the mandatory fields and vacancy. (Resume is also mandatory)

The candidate name should be QA Automation - <CurrentDate> (Please substitute the current date with the current date) Click Save

Use Junit Assert to check if the number of candidates is increased by 1.

Select the freshly created candidate

Click on the three dots in the left upper side of the table.

Delete the candidate

Use Junit assert to check if the number of candidates decreased by 1

Open the side menu

Click on the User icon

Click Logout

Important:

Use Java naming conventions! The packages, functions, variables should be named properly.

Use the **Page Object** pattern! Details about it can be found on the selenium website.

<https://www.pluralsight.com/guides/getting-started-with-page-object-pattern-for-your-selenium-tests>

You can use Firefox browser for a basic test run and please comment your code.

Use Maven for dependency management.

Publish your local git repo to any platform (preferably GitHub)

Please provide instructions for running the test cases.

Optional tasks for bonus points, this is the part where you can show off your skills:

- Use Chrome, Internet Explorer 11, Microsoft Edge and specify a way to make the browser changeable
- Tests should be able to run from any machine (there should be no local dependency)
- Add a logger framework and log the high level WebDriver actions, urls of pages when you arrive to a page
- Implement a generic screenshot creation module which will create a screenshot after each test failure
- Make the tests runnable via command line (browser type should be choosable via an argument)
- Setup jenkins for running tests (you can deploy it to somewhere)
- Use the functional aspects of Java where it is possible
- Utilize Selenium Grid hub - node architecture

What will you surely need:

1. Eclipse or IntelliJ
2. WebDriver
3. Patience

WebDriver setup instructions and tips can be found on their official site

<https://www.seleniumhq.org/docs/>

And remember google is your friend

Manual Testing assignment details:

There are 2 tasks for manual testing, one is for exploratory testing and the other is for test case writing.

1. Task - exploratory testing

Manual exploratory testing. Choose 2 sites from this list: <http://www.007museum.com/>, <http://www.arngren.net/> <http://www.gatesnfences.com/index.html>, <http://www.libertyvan.com/>

Important note: Do NOT submit any form from the pages

For testing use the latest Chrome or the latest Firefox or the latest Microsoft Edge - just clearly highlight in the report which browser was used.

How to report the issues:

- Report the issues in a text document (MS Word, LibreOffice Writer, Google Docs, etc.)
- Report 1 issue per page – all issues in 1 document
- All reported issues should have a screenshot or a screencast (video)
- Make sure that the title describes the issue very accurately
- Used device/browser should be listed in the issue
- Steps to reproduce have to be clear and clean
- Language: English

Example:

The right side of the checkout page is not displayed properly

Used Device/Browser:

- Chrome (Version 97.0.4692.99 (64-bit)) on Windows 10

Steps to reproduce:

- Open the XXXXXXXXXXXXXXXX page
- Click on the Get button
- The checkout page will be displayed
- Check the right side of the page
- It is not displayed correctly
- Some images and links are missing

Screenshot:

Add here the screenshot of the issue - if the issue is complex record a video instead of a screenshot, upload it and add the link here (you can use ShareX for screencast)

2. Task - Test case writing

Create test cases for the site: <https://www.walmartcontacts.com/lens/eureka-monthly>

You need to create the test cases for this page only - add test cases for scenarios that you think it is important to test on this page.

The test cases should be created in a spreadsheet (MS Excel, Google Sheet, LibreOffice Calc, etc.).

Language: English

Test case sample:

Section	Description	Steps	Expected Result
Main page	Check if the video is loading from the page or not	1 - Open the page xxxxxxx on Chrome browser 2 - Click on the play button 3 - Check if the video playback stated or not	The clicked video file should start playing in a popup