

Test Assignments

Goals

The purpose of this test is 3 fold:

- To see what test scenarios you come up with
- Assess your approach to different segments of the QA engineering job
- Assess how comfortable you are with writing automation tests and the approach taken to accomplish it

Tech Stack

You're free to choose one from the following two paths:

- JavaScript based test automation
- Selenium based test automation

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

Please provide instructions for running the test cases.

JavaScript automation:

Your task is to create automated tests using a JavaScript automation framework. You're free to choose your tool, however, we prefer Cypress.io.

Make sure your code is clean, well organized, and readable. Naming conventions are used and, please, comment on your code.

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

- Use PageObject pattern
- Extract repeated code to reusable functions
- Use custom commands
- Provide details how to run separate tests directly from command line using headless browser
- Pass the value to the test runner through environment variables or command line arguments
- Use any Cypress plugin

Selenium automation:

Your task is to create a Selenium WebDriver test application. You're free to use any programming language with its supported software project management tool and unit testing framework. We prefer Java with Maven and JUnit5. The tests should be able to run separately. If you have any questions feel free to ask.

Use adequate 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 your preferred software project management tool for dependency management.

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

- Use Chrome, 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 the programming language where it is possible
- Utilize Selenium Grid hub - node architecture

WebDriver setup instructions and tips can be found on their official site <https://www.seleniumhq.org/docs/>

Automation test assignment details:

Task 1

Create end2end tests for the following web store <https://www.saucedemo.com/>. The required login credentials are displayed under the sign-in form. The test **suite** should contain at least 3 tests which are defined below. Any additional test cases you add will be treated as a plus.

If you have any questions feel free to ask.

Please, follow these test steps:

Go to <https://www.saucedemo.com/>

Log in to the site

Add an item from the list to the cart

Verify that the cart badge is updated correctly

Open another item's details page

Add the item to the cart

Open the cart

Verify that the correct items are present (2 different items)

Remove the first item from the cart

Verify that the correct item is present (1 item)

Continue to the Checkout page

Complete the checkout form

Complete the order

Verify that the order is completed successfully with the displayed message

Task 2

Using the testing site <https://gorest.co.in/>, write automation tests for the following endpoints:

1. Create a new user
2. Update user details
3. Delete user

The test **scenarios** are entirely up to you, and you may use additional endpoints if you feel they will provide better coverage and improve your coding solution. You will be evaluated on which test cases you create as well as on the overall quality and efficiency of your automation code

Manual Testing assignment details:

There are 2 tasks for manual testing, one is for exploratory testing, 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 Internet Explorer - just clearly highlight in the report which browser was used.

How to report the issues:

- Report the issues in a Word document
- Report 1 issue per page – all issues in 1 Word 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 recordit.co for screencast)

2. Task - Test case writing

Create test cases for the page: <https://www.walmartcontacts.com/Lens/614>

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 an Excel sheet.

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
