Selenium Integrated Development Environment

David Corcoran

Table of Contents

Installation 3-4

Creating A Project 5-6

Saving a Project 7

Manipulating Automated Testing Files

- **Creating Automated Tests** 8-9
- **Renaming Automated Tests** 10
- **Deleting Automated Tests** 11
- **Test Suites** 12-13

Recording a Test 14-22

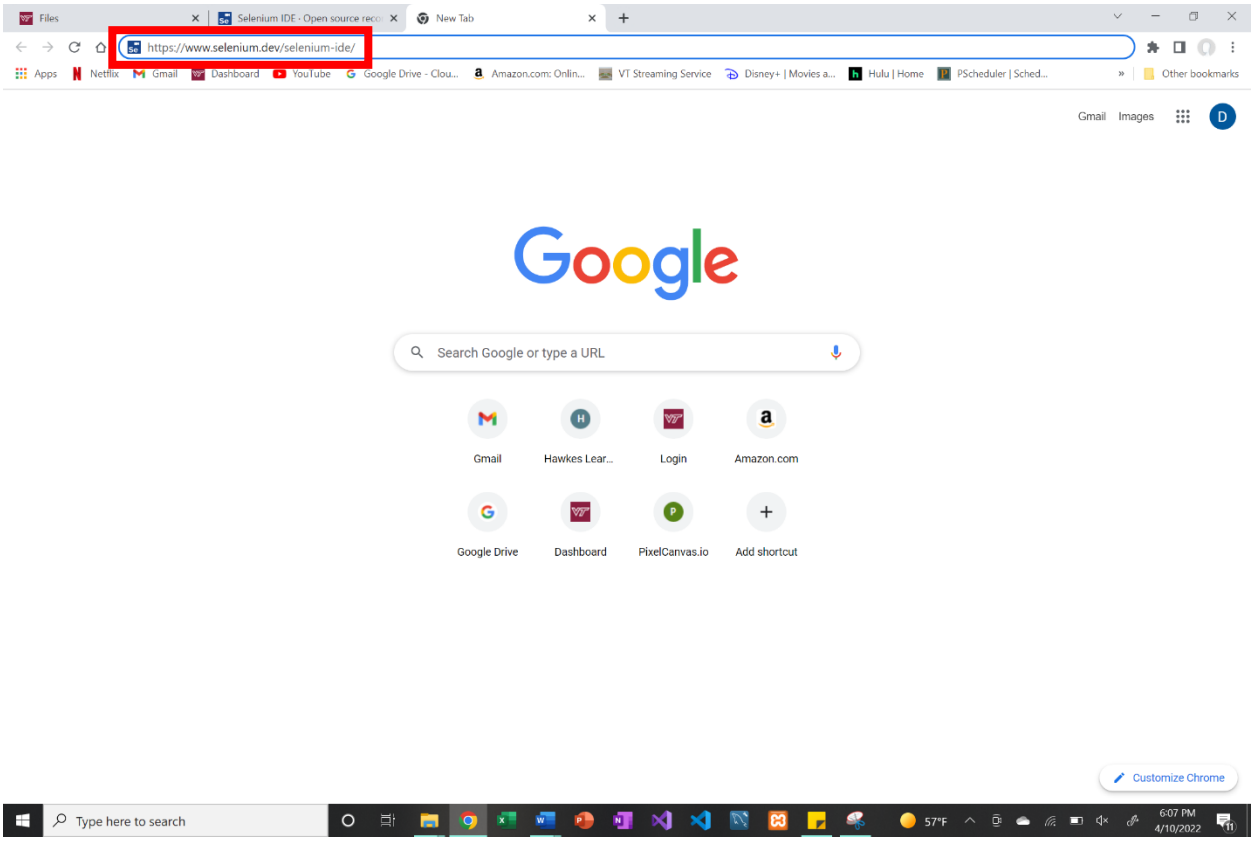
Test Playback 23-29

Conclusion 30

Installation

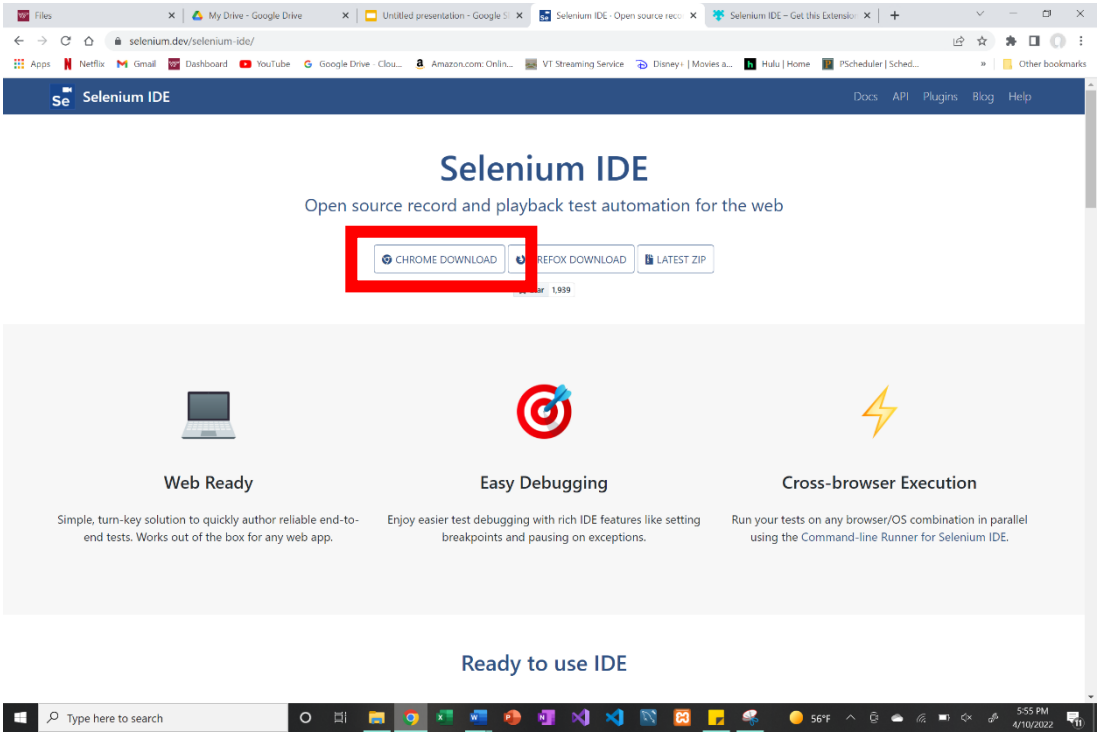
Selenium Integrated Development Environment (IDE) is an open source, record and playback automated testing software. It is implemented as a browser extension, available only on Mozilla Firefox and Google Chrome. For this demo, I will be using Chrome as my web browser. Selenium IDE allows its users to input and debug unique requirements tests on a pre-selected website. Steps for installation are as follows:

1. Open Google Chrome.
2. Navigate to https://www.selenium.dev/selenium-ide/ in your selected browser.



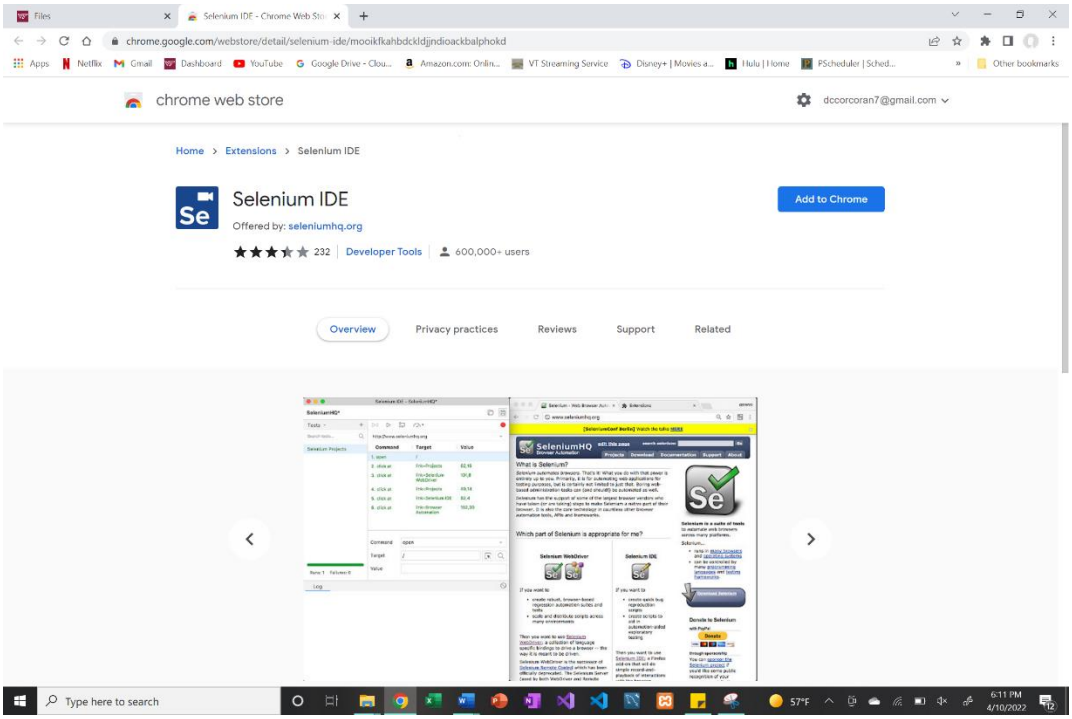
https://www.selenium.dev/selenium-ide/

3. Click the “Chrome Download” button to be redirected to the Google Chrome Web Store.



https://www.selenium.dev/selenium-ide/

4. Press the “Add to Chrome” button and confirm the action, allowing for the software to become an extension on your browser.

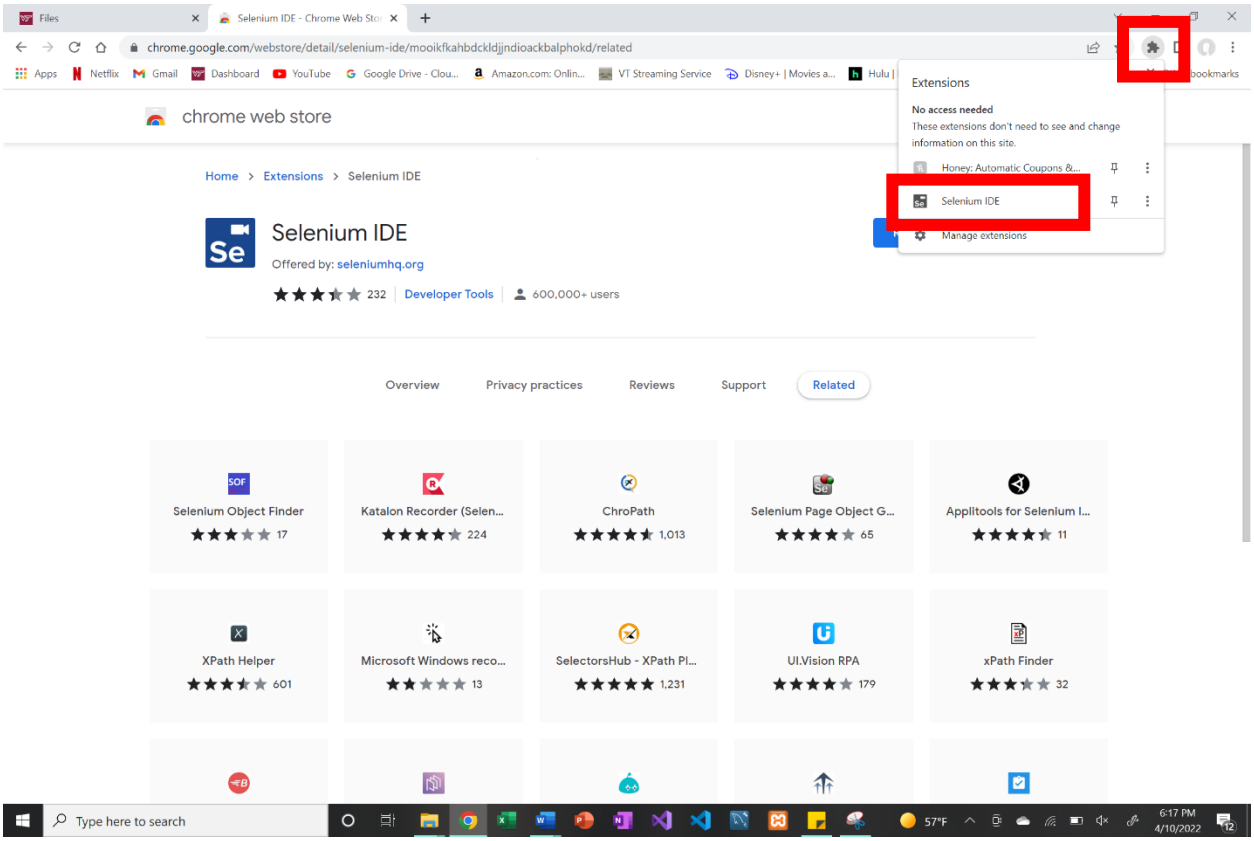


https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd

Creating A Project

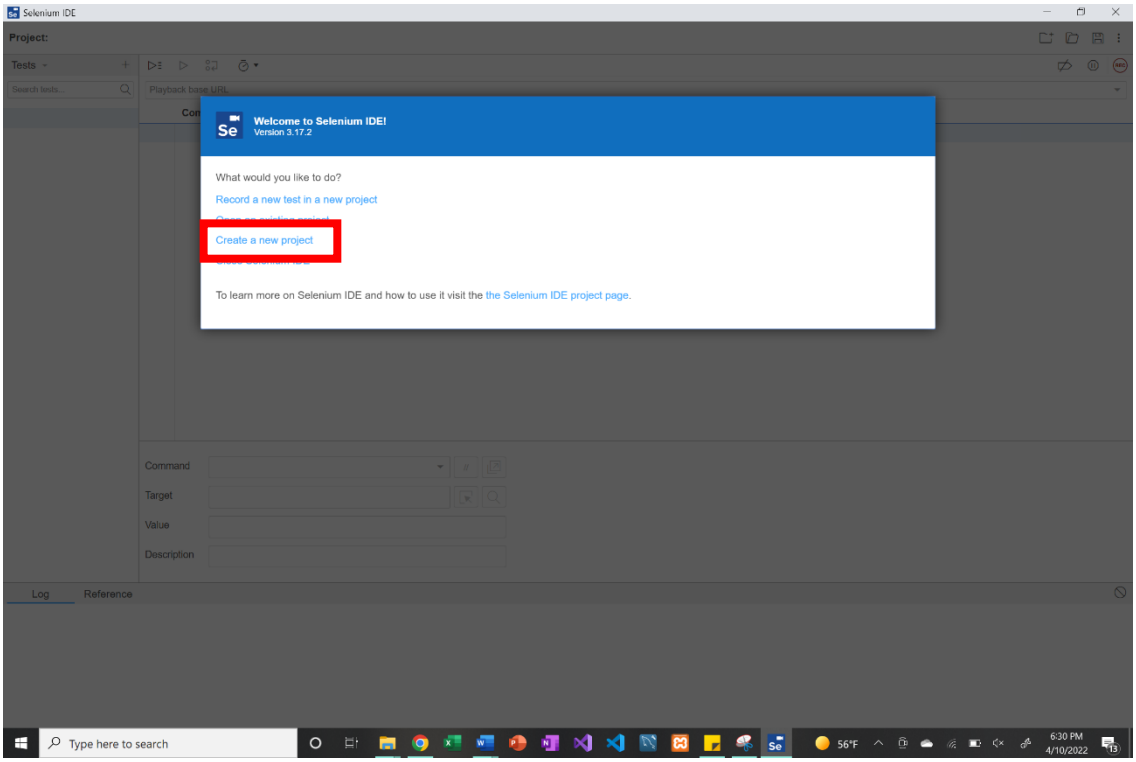
Next, I will instruct how to open the Selenium IDE application and create a project file. To efficiently use Selenium IDE, the user must first create project, and subsequent file, where all related automated tests can be stored. Creating different project files can help organize a user's requirements tests, separating non-related tests to decrease confusion. Creating a project can be done as follows:

1. Click the puzzle icon in the top right of your screen. This is the hub for all the chrome extensions downloaded to your computer.
2. Click the button titled "Selenium IDE." This will open the Selenium IDE application.

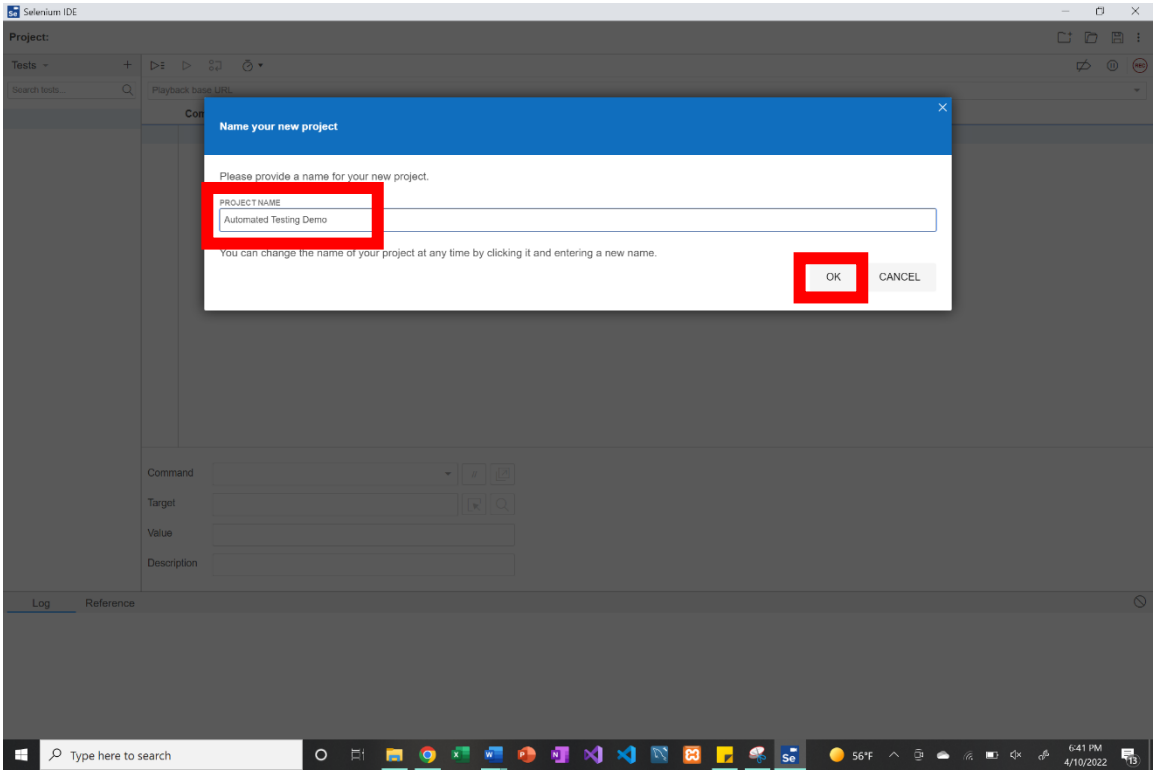


https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd

3. Click “Create a new project.” This will create a new project as a “.side” file. Because this program is completely new, you won’t have any existing project files to open.



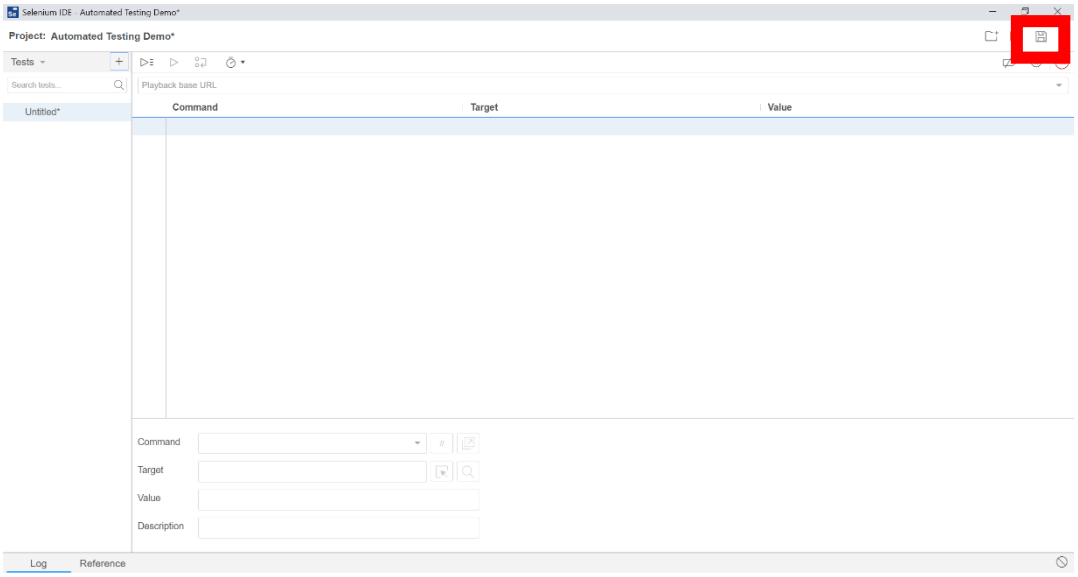
4. Title the new project “Automated Testing Demo” and click the “OK” button.



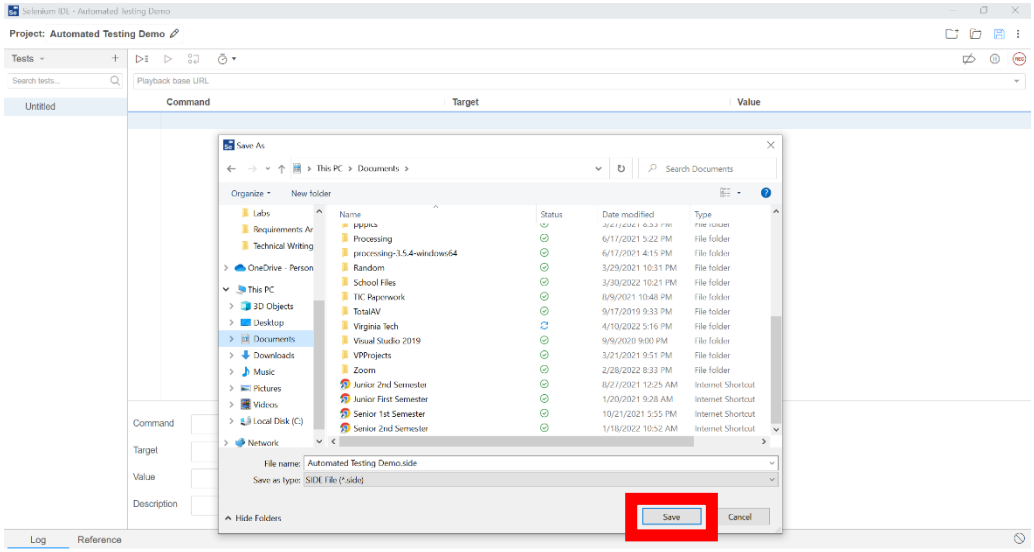
Saving A Project

Saving is important, as it allows the user to close the Selenium IDE application and return to work on it later. To save a project file, the instruction are as follows:

1. Click the save icon in the top right of Selenium IDE's application window



2. Save the project ".side" file anywhere you like in your computer files

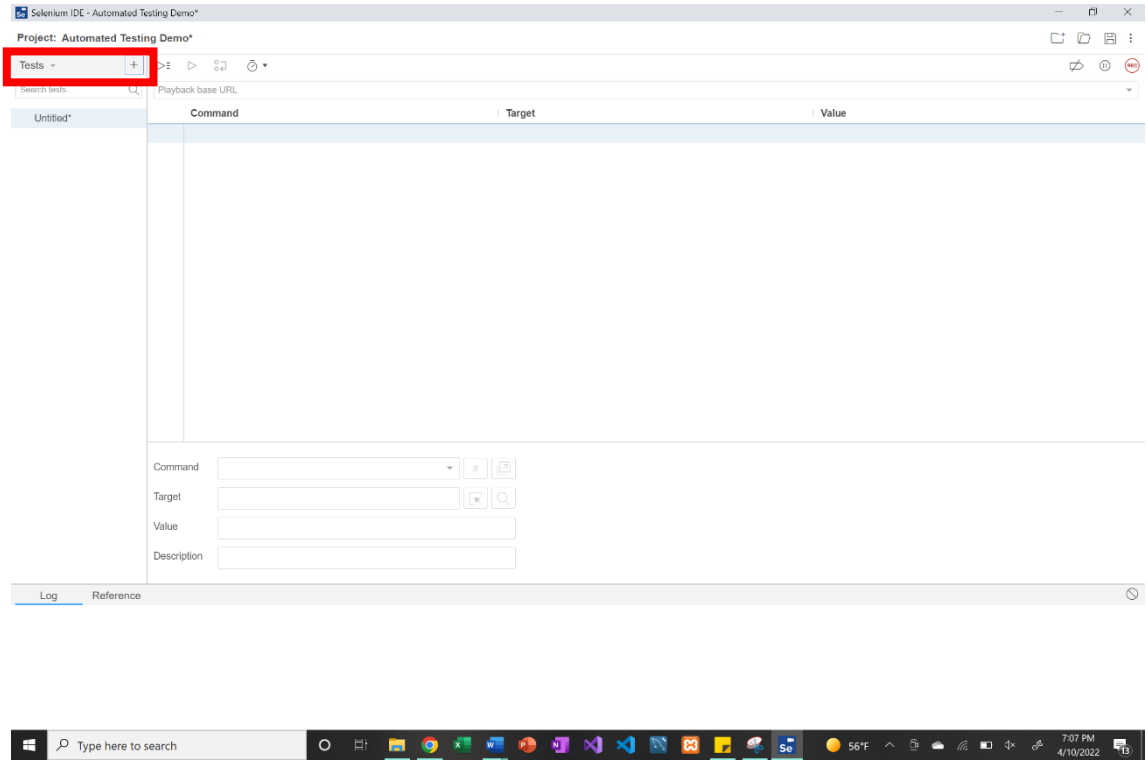


Manipulating Automated Testing Files

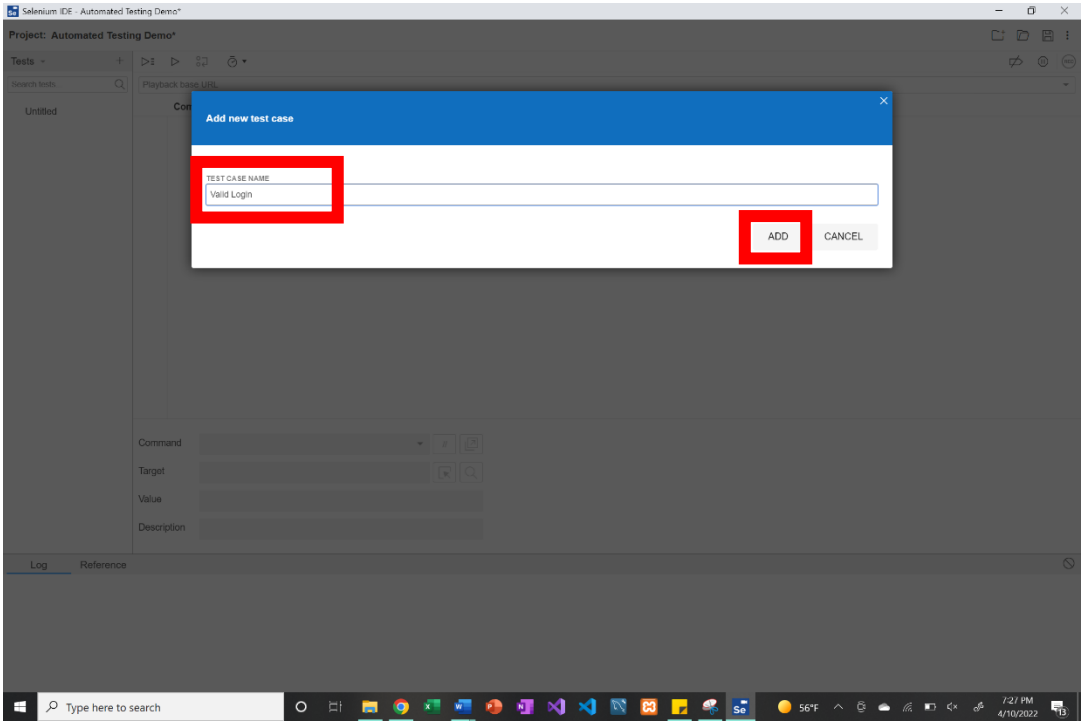
Next, I will provide instructions on how to create, rename, duplicate, delete, and save automated tests and test suites. A test suite is a collection of related automated test that the user has arranged in a collective group. The instructions on manipulating test files are as follows:

Creating Automated Tests

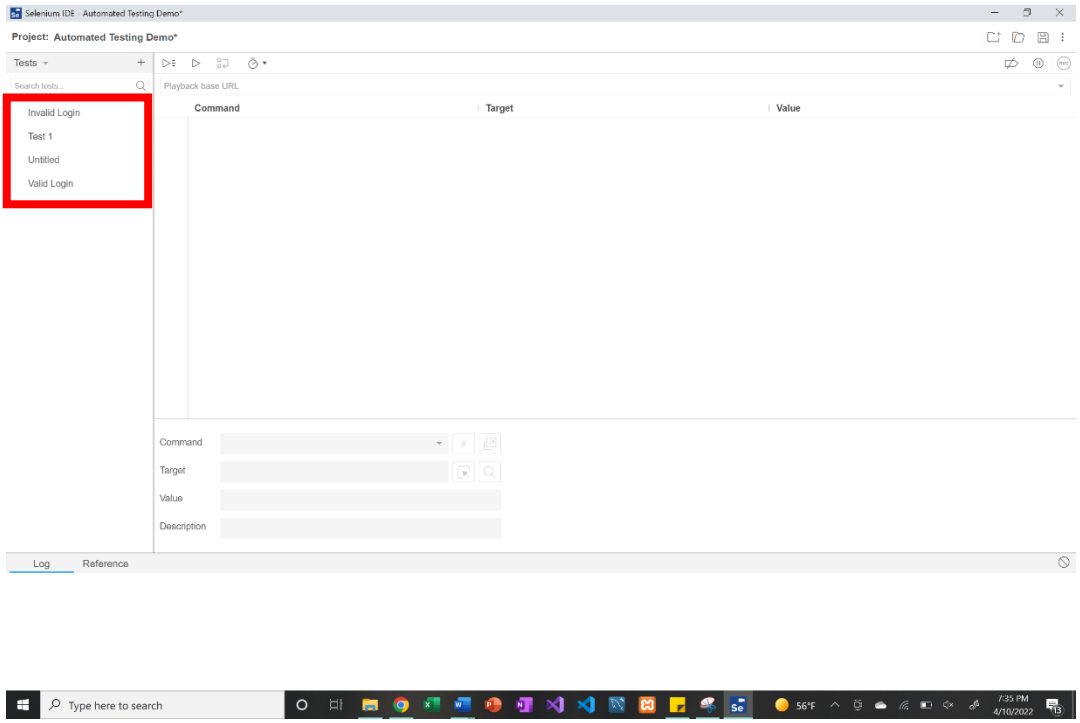
1. When creating a new project in Selenium IDE, an “Untitled” first test is automatically created. However, to create a new automated test, press the plus button at the top left of the window.



2. Name the new automated test “Valid Login” and press the “Add” button. This test will automate a correct username/password combination and successfully admit the user into the system

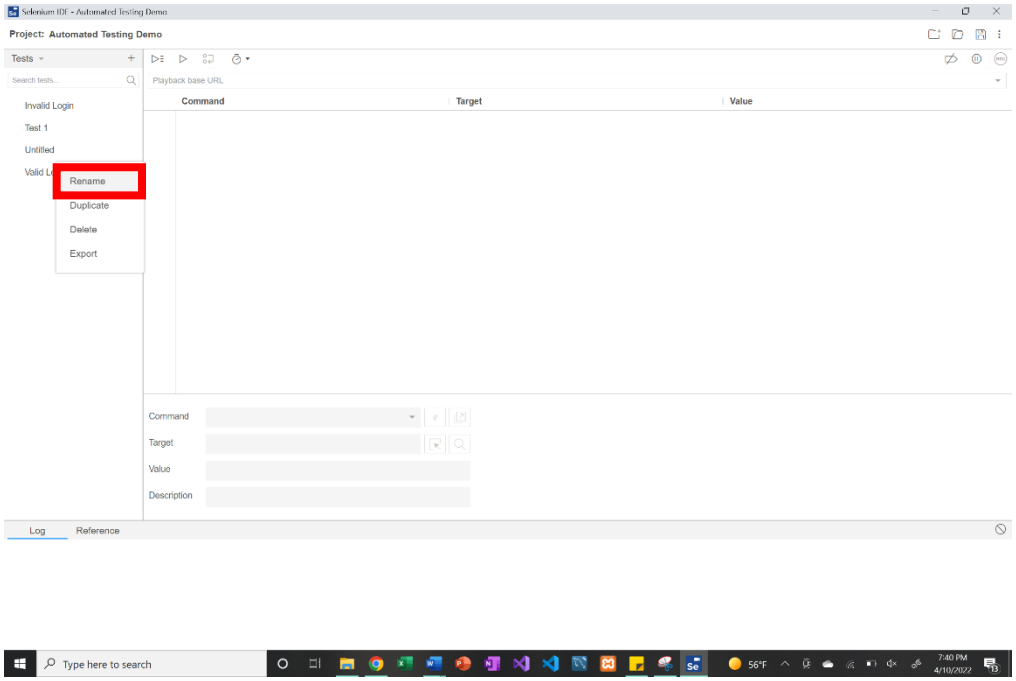


3. Create two more automated tests called “Invalid Login” and “Test 1.” “Invalid Login” will input incorrect username/password combinations, blocking the user from accessing the website. “Test 1” will be a throwaway test to demonstrate how to delete tests. New tests will appear in alphabetical order at the left side of the application window.

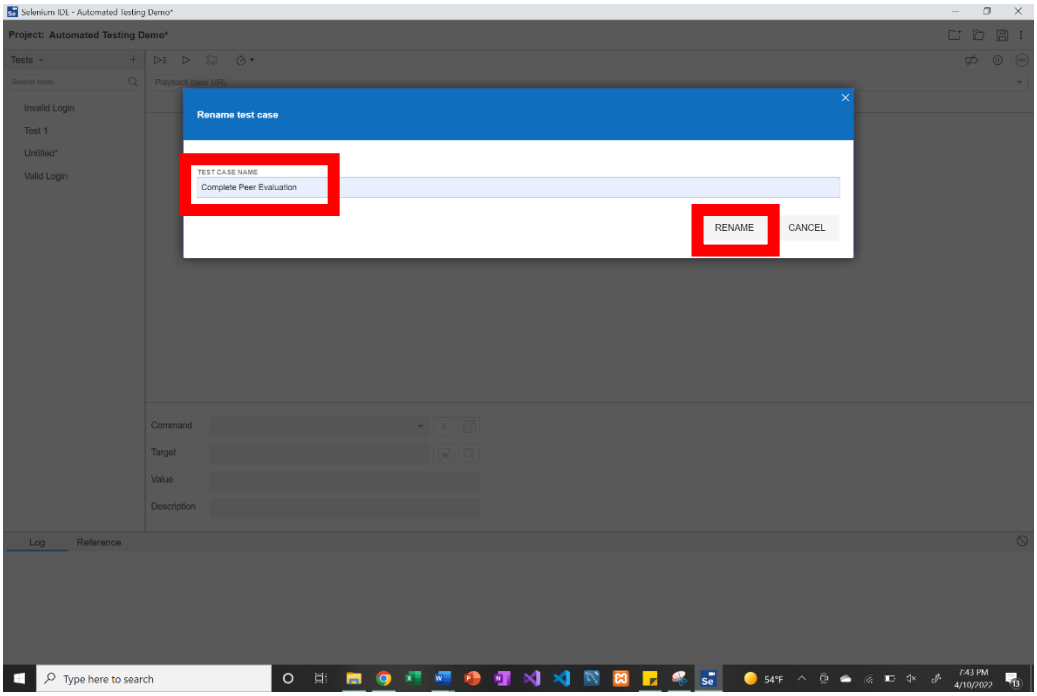


Renaming Automated Tests

1. Press the menu button for the “Untitled” test.
2. Press the “Rename” button on the dropdown to change the title of the test.

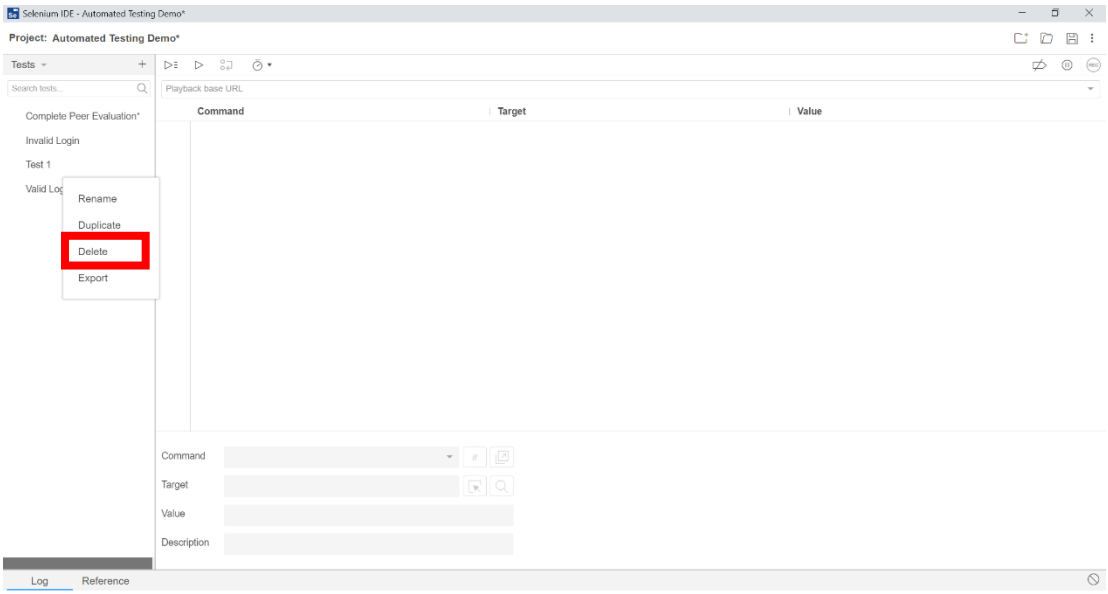


3. Rename the test to “Complete Peer Evaluation” and press the “Rename” button.

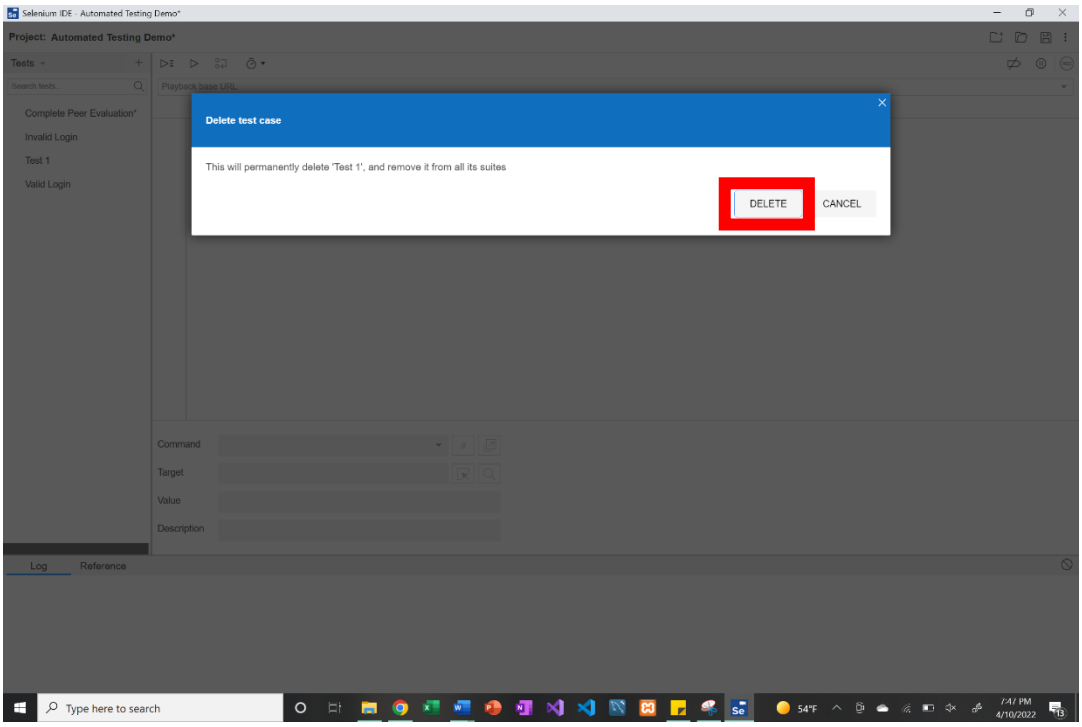


Deleting Automated Tests

1. Press the menu button for the “Test 1” test.
2. Press the “Delete” button on the dropdown to delete the test

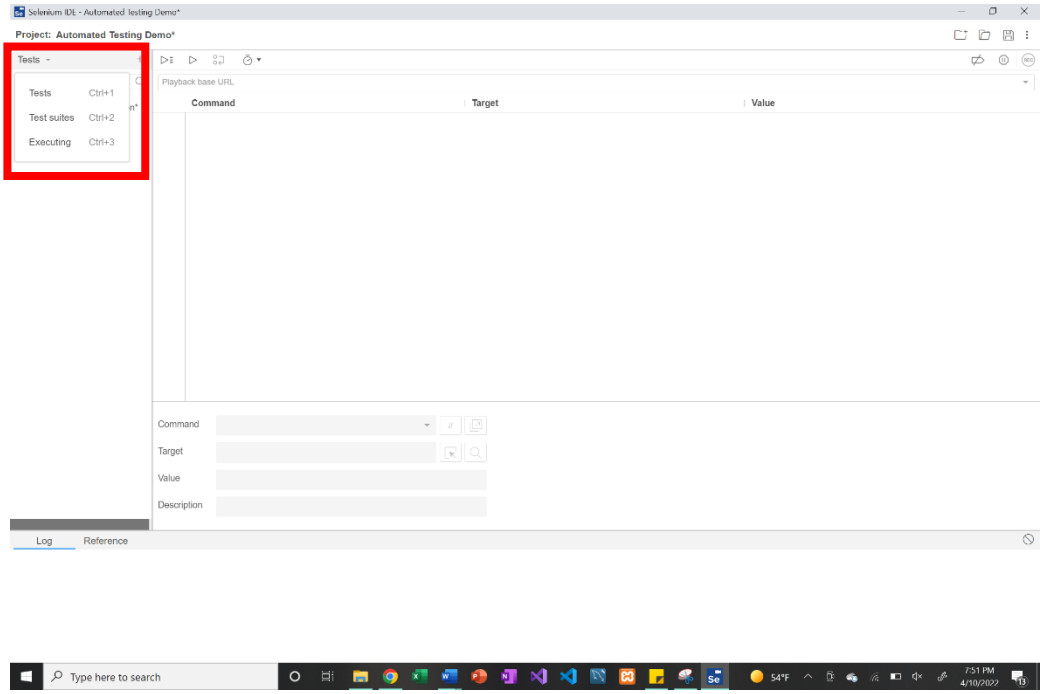


3. Press the “Delete” button on the verification window to permanently delete the automated test.

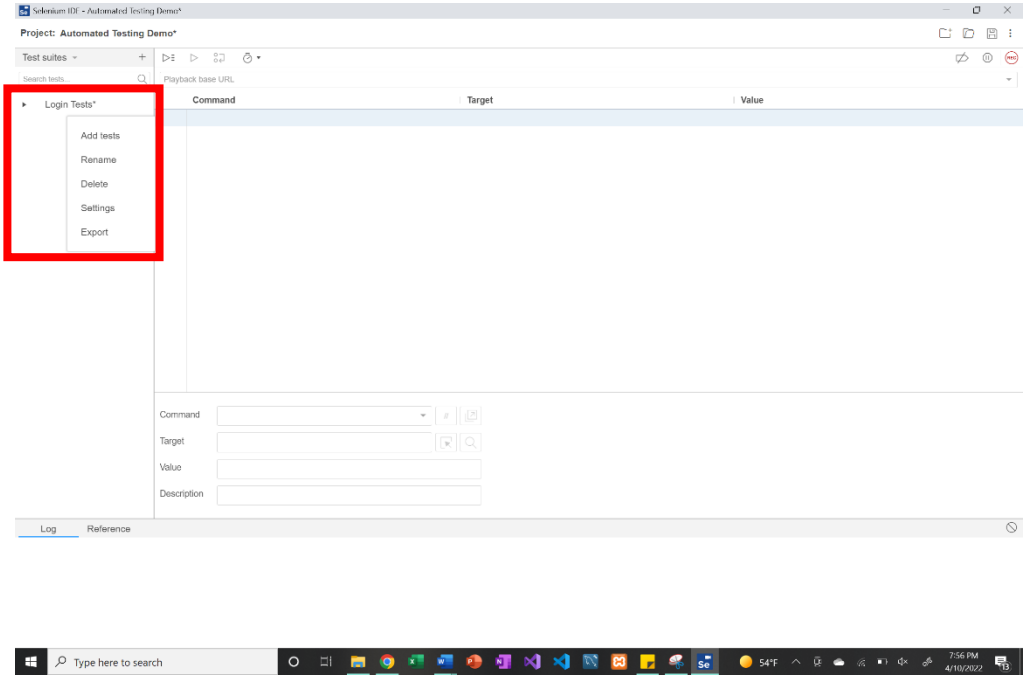


Test Suites

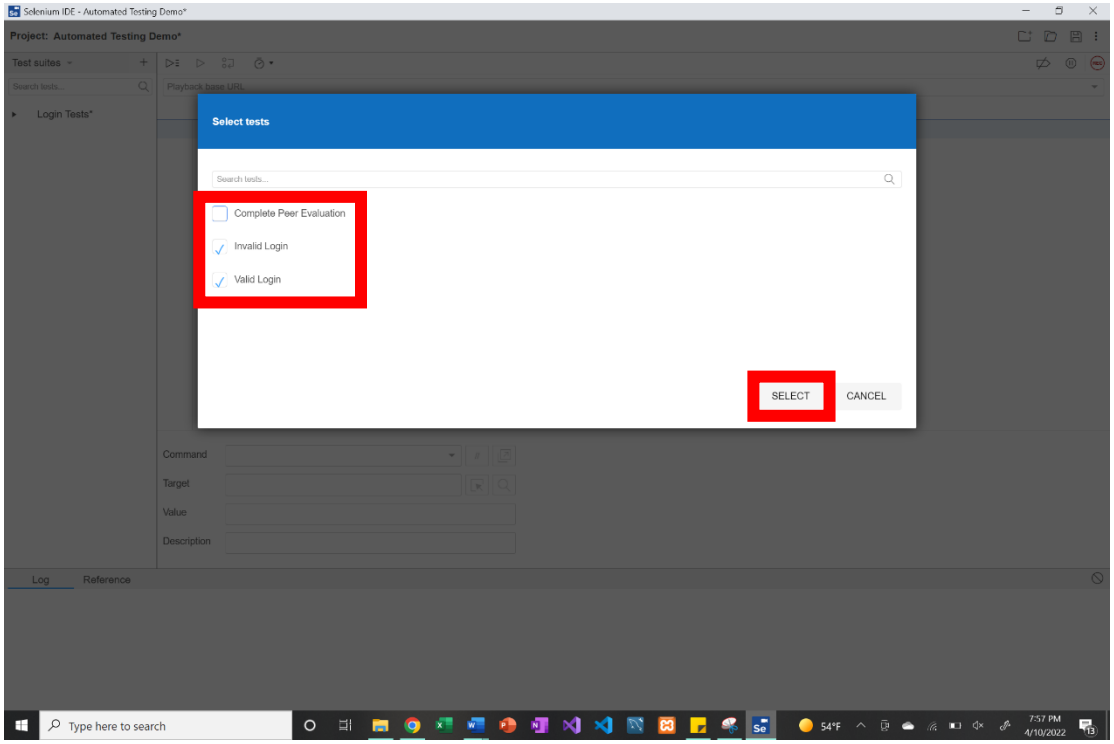
1. Click the dropdown arrow near the top left of the application window next to the word “Tests.” Right now, all test created in the project are shown in alphabetical order.
2. Press the “Test Suites” option int eh dropdown to display the current test suites, or test groupings.



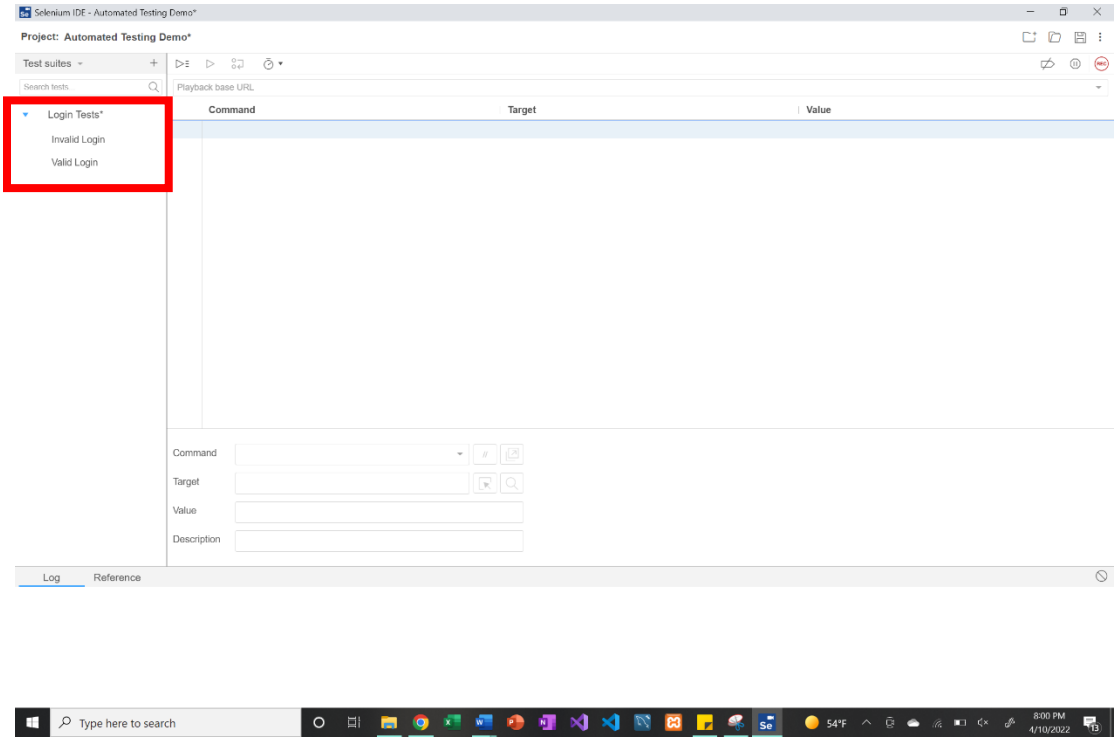
3. Press the menu button of “Default Suite,” and rename it “Login Tests” by using the same format used to rename the automated tests
4. Press the menu button of the “Login Tests” suite and click on the “Add Tests” option



5. Click the checkbox of both the “Invalid Login” and “Valid Login” tests. Then press “Select.” We are making sure to include only the tests that relate to one another. Both the “Valid Login” and Invalid Login” automated tests deal with logging into the system.



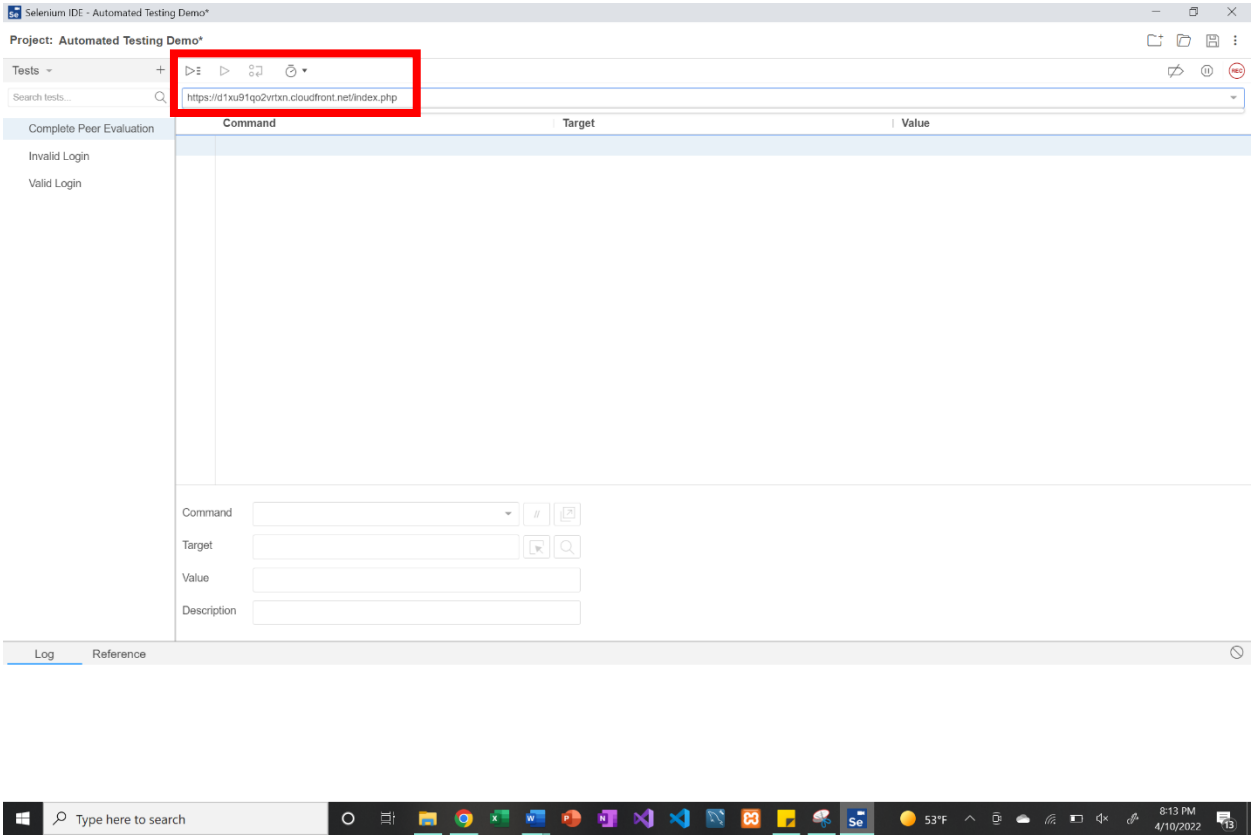
6. Press the dropdown arrow of the “Login Tests” suite to ensure that only the selected tests are assigned to that suite.



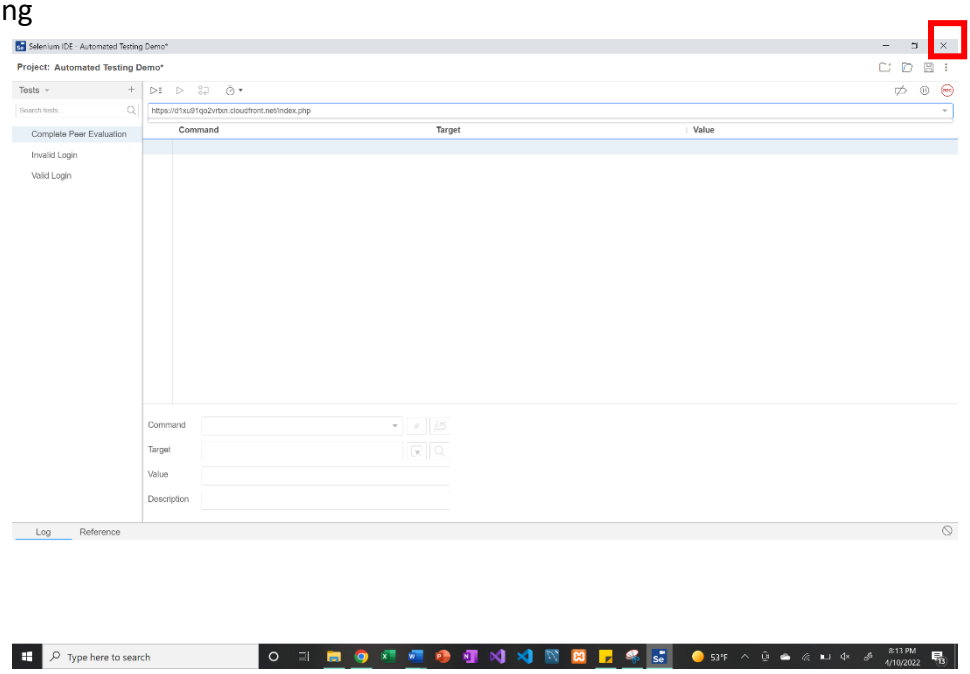
Recording a Test

Selenium IDE offers a feature in which the user can record themselves going through a selected website in real time. All actions done while in the live recording session will be automatically applied to the command line of the automated test. This eliminates the need to investigate the code of the website to locate the exact elements needed to test. Although there are a variety of testing commands, we will mostly be using “open,” “click,” “verify text,” “type,” and The instructions to record an automated test are as follows:

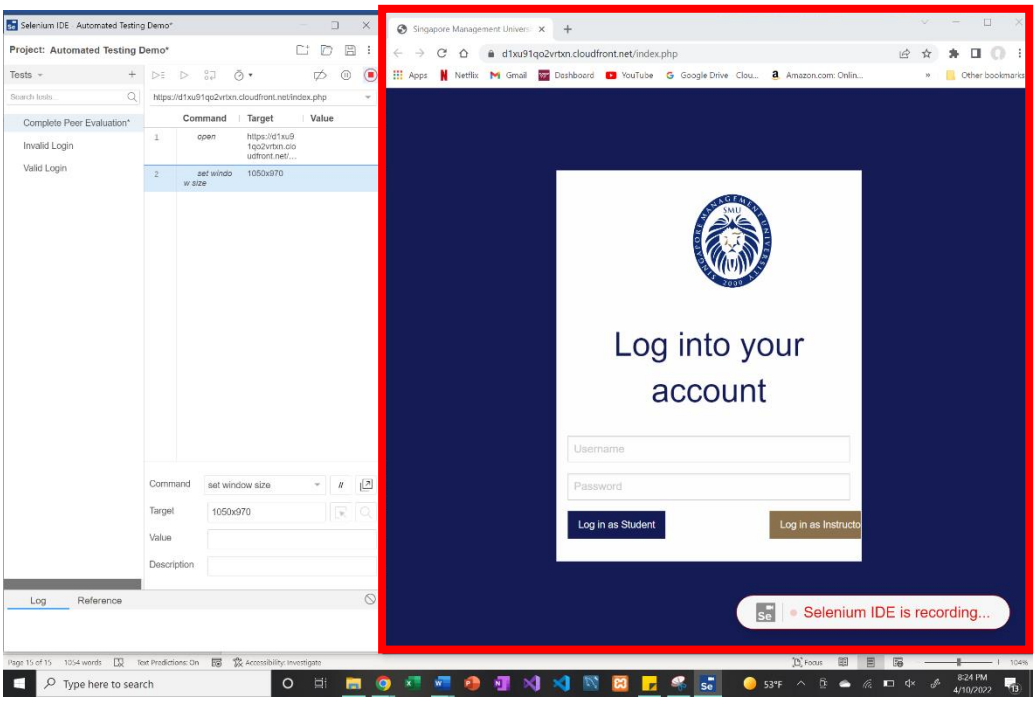
1. Return to the “Tests” display option and press the “Complete Peer Evaluation” automated test. We will be recording a test that will successfully submit a peer evaluation for a student in the SMU database.
2. Input the selected URL of the website that you are testing into the “Playback base URL” textbox at the top of Selenium IDE’s application window. For this demo, we will be using https://d1xu91qo2vrtxn.cloudfront.net/index.php, the URL for SMU Team 3’s peer review website.



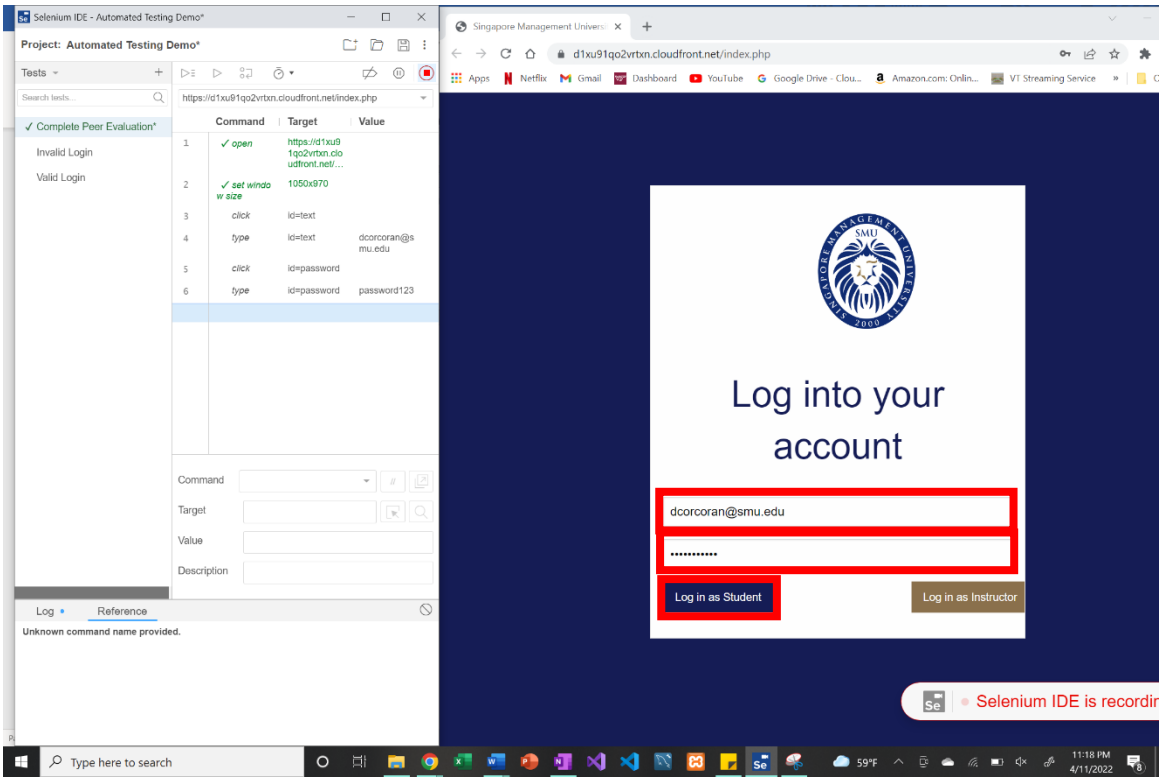
3. After inputting the target URL, press the “REC” button at the top right of the application window. This will open a new chrome browser with the selected website and begin the recording



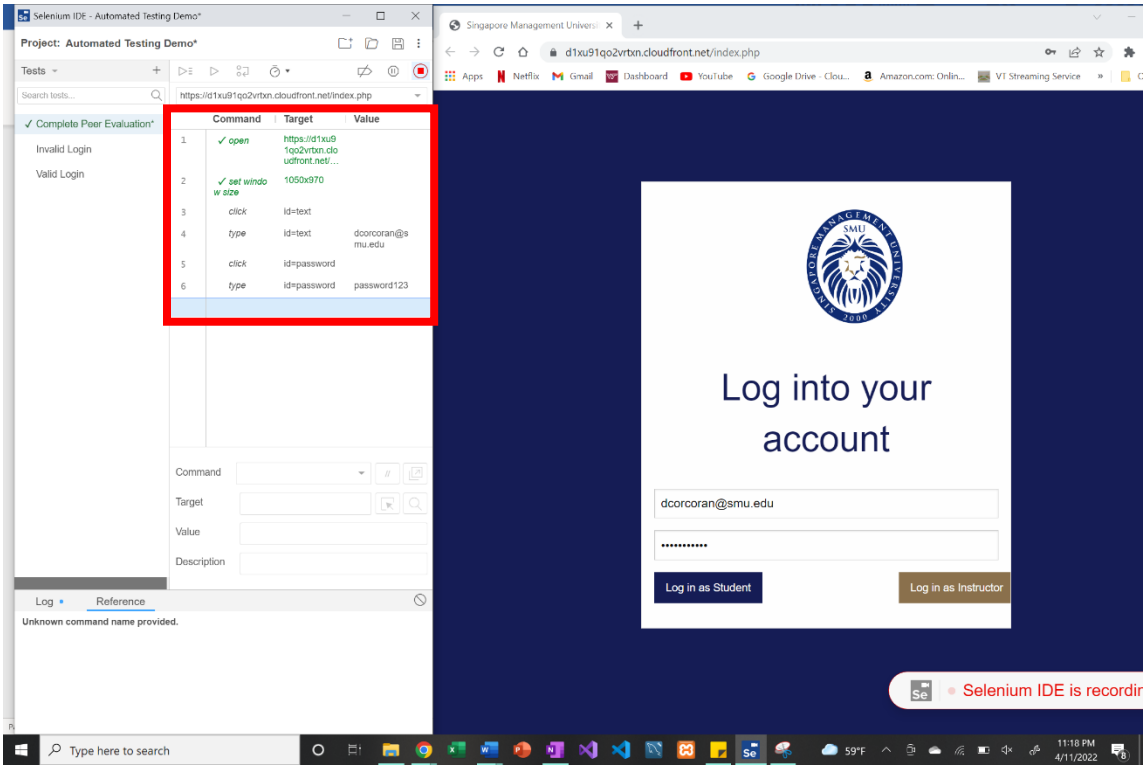
4. Once a chrome tab opens with the website in view, minimize the SMU website application to “1050x970.”
5. Resize the selenium IDE window to fit the remaining portion of your computer’s screen. This will allow you to perform actions on the SMU website and see the results as they come in on Selenium’s user interface.



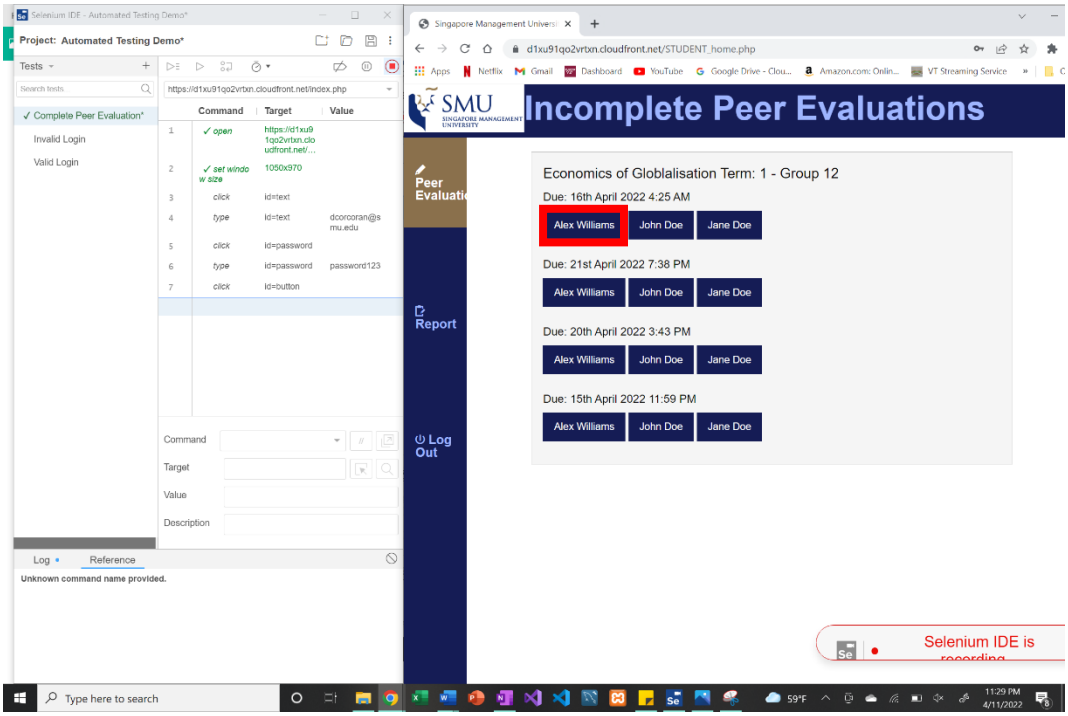
6. Type “dcorcoran@smu.edu” into the username textbox and “password123” into the password textbox.
7. Press the “Log in as Student” button



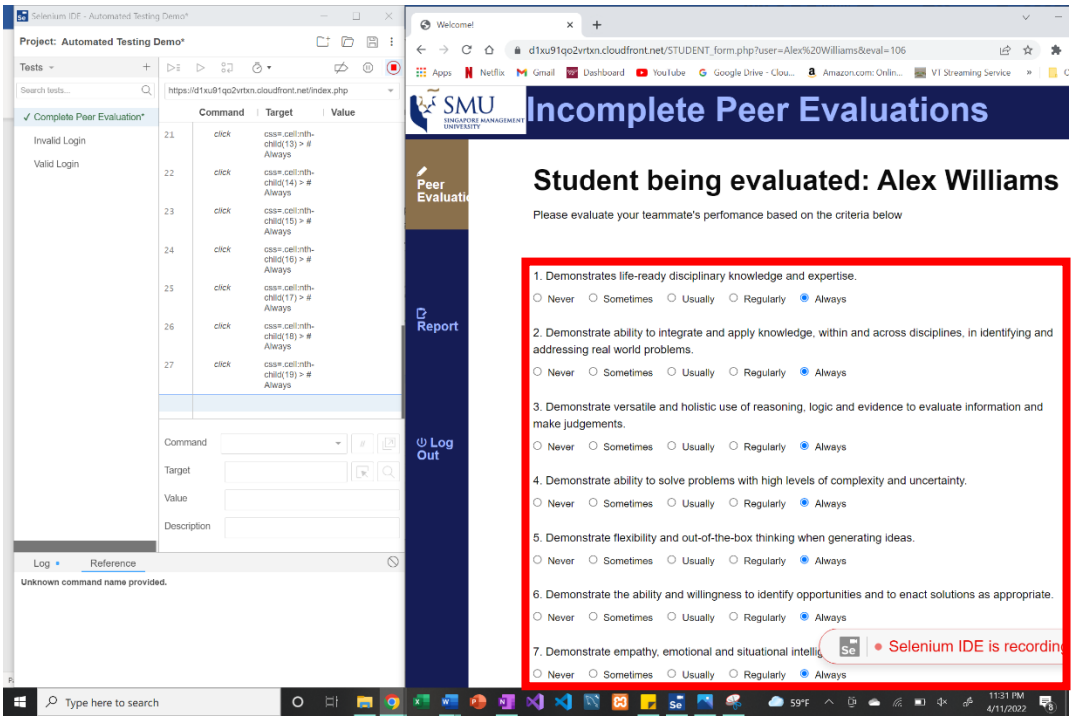
8. As an action is performed on the SMU website, the “Command”, “Target”, and “Value” items are automatically filled in on the Selenium IDE application window. This allows the user to complete the requirements for a test case much faster than manually inputted each item individually.



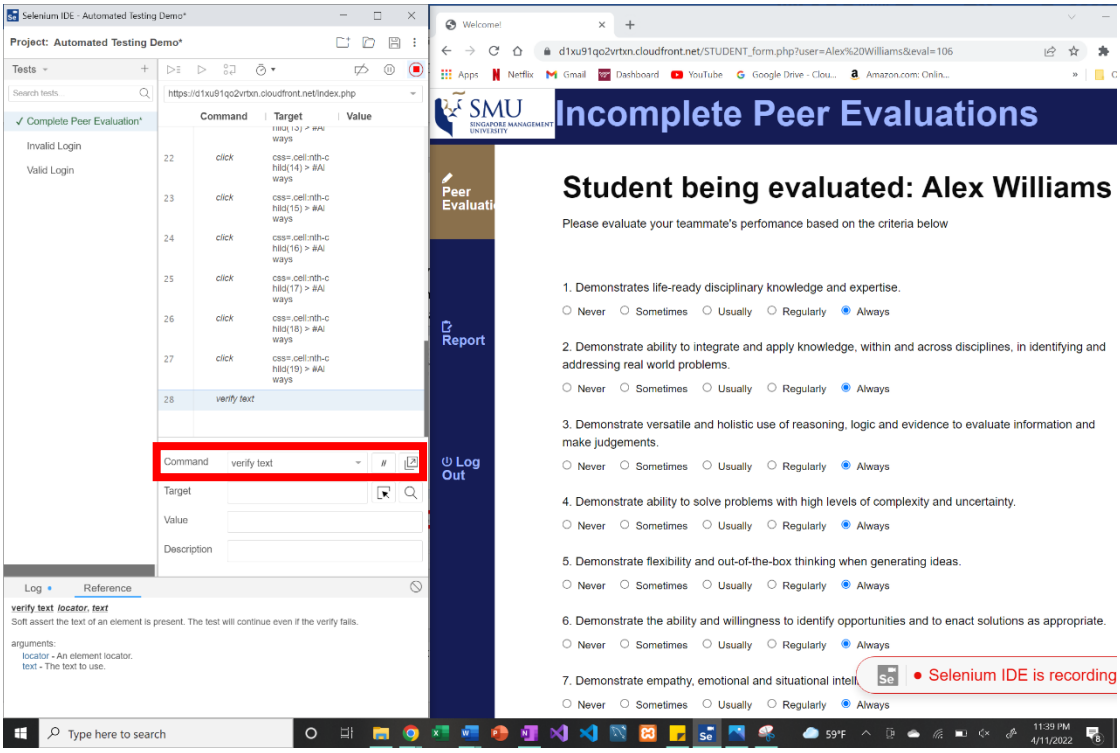
9. Press on the first peer evaluation available. Since only the incomplete peer reviews assigned for “David Corcoran” are displayed, the names of students for available evaluations will change over time.



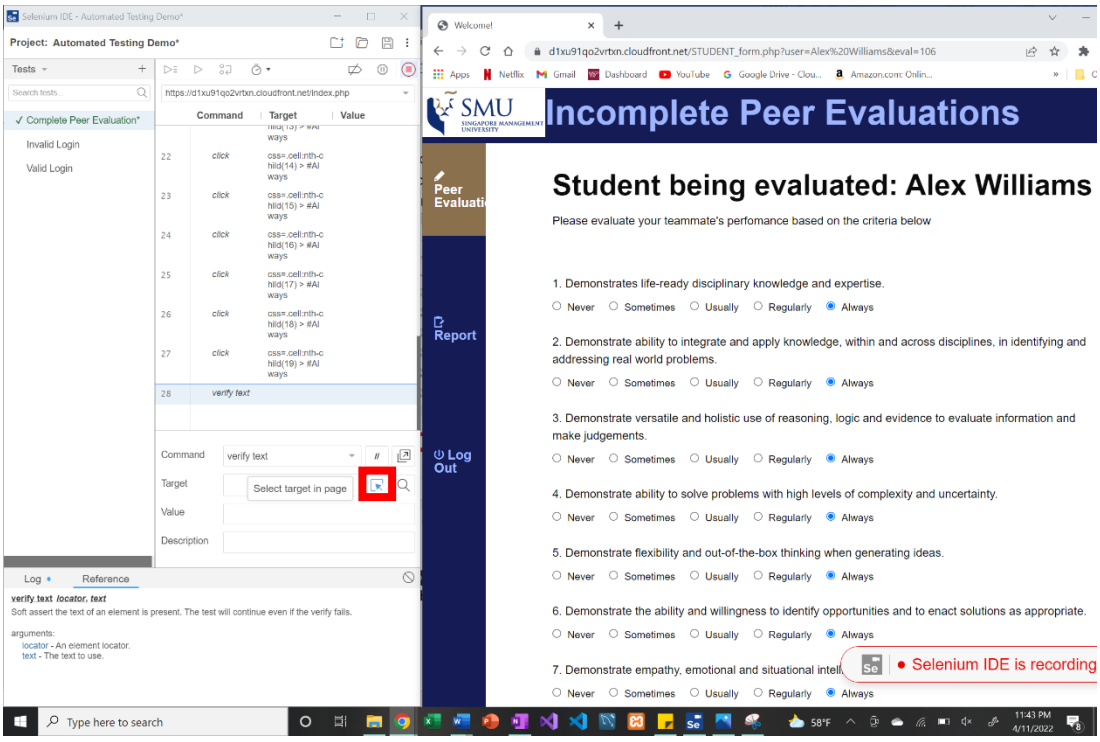
10. Click on a selection for all 19 the radio buttons for the student being evaluated. I will be clicking the fourth radio button, ”Always,” for all behavioral questions in the evaluation.



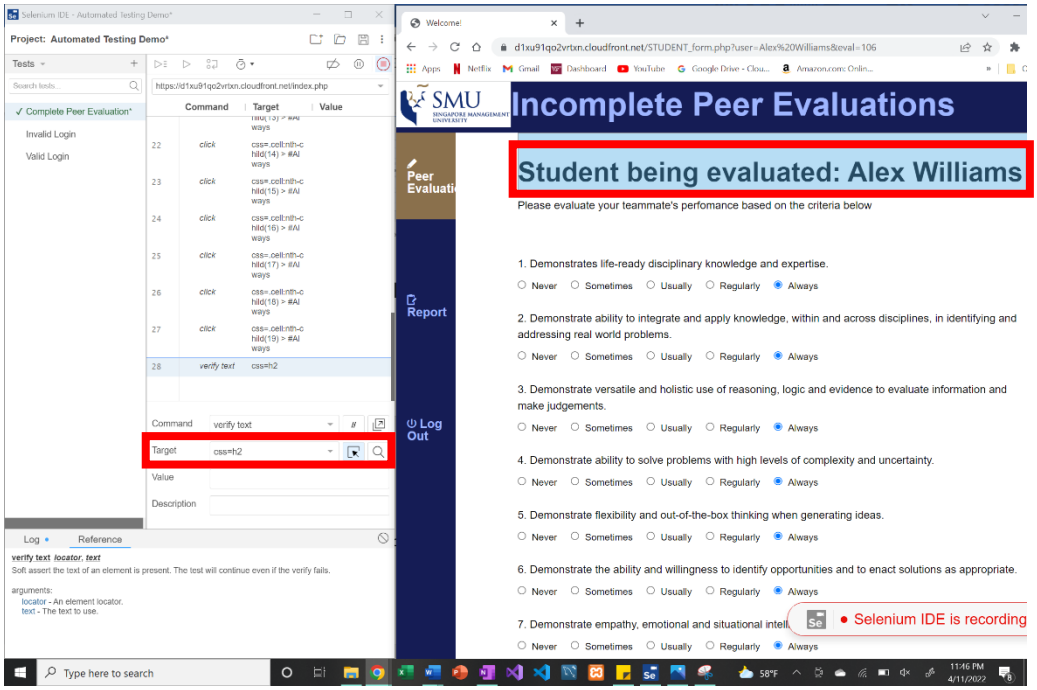
11. Click on the “Command” dropdown of the Selenium application window and type “Verify Text”. Because there is no mouse action that can perform the “verify text” command, we will have to input it in manually. Ensure that the words contain all lowercase letters.



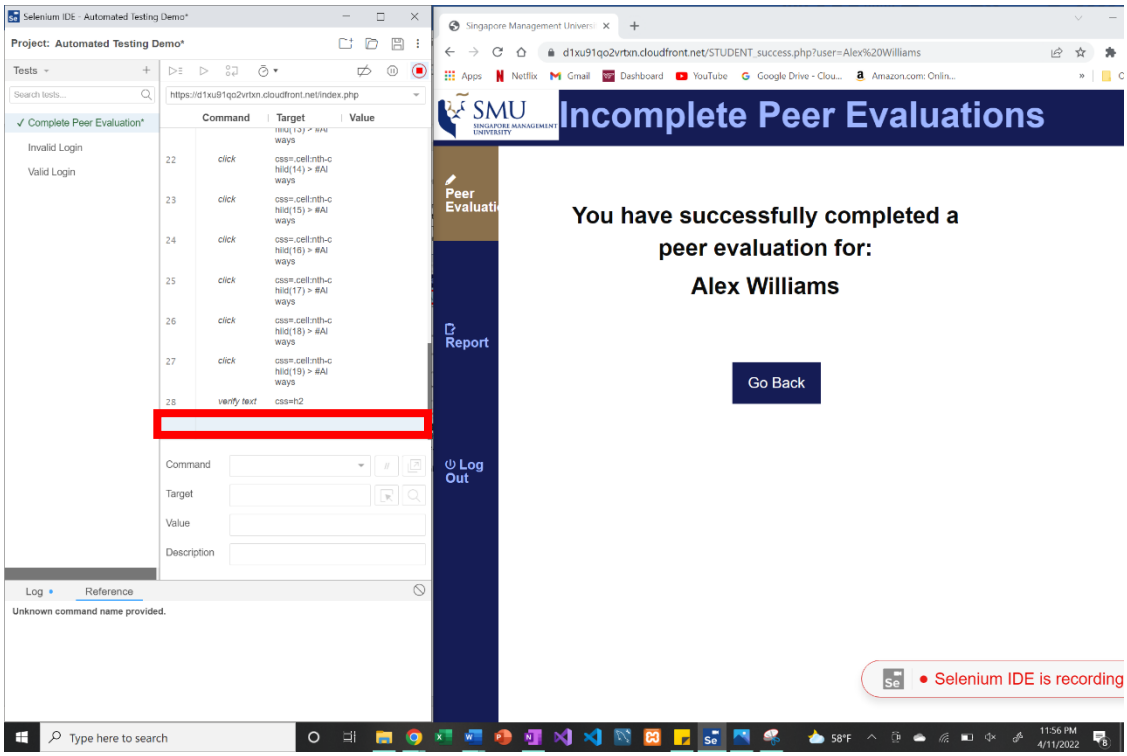
12. Click on the “Select target on page” button. This feature allows us to click on any element of the SMU website and assign that value to the “Target” textbox.



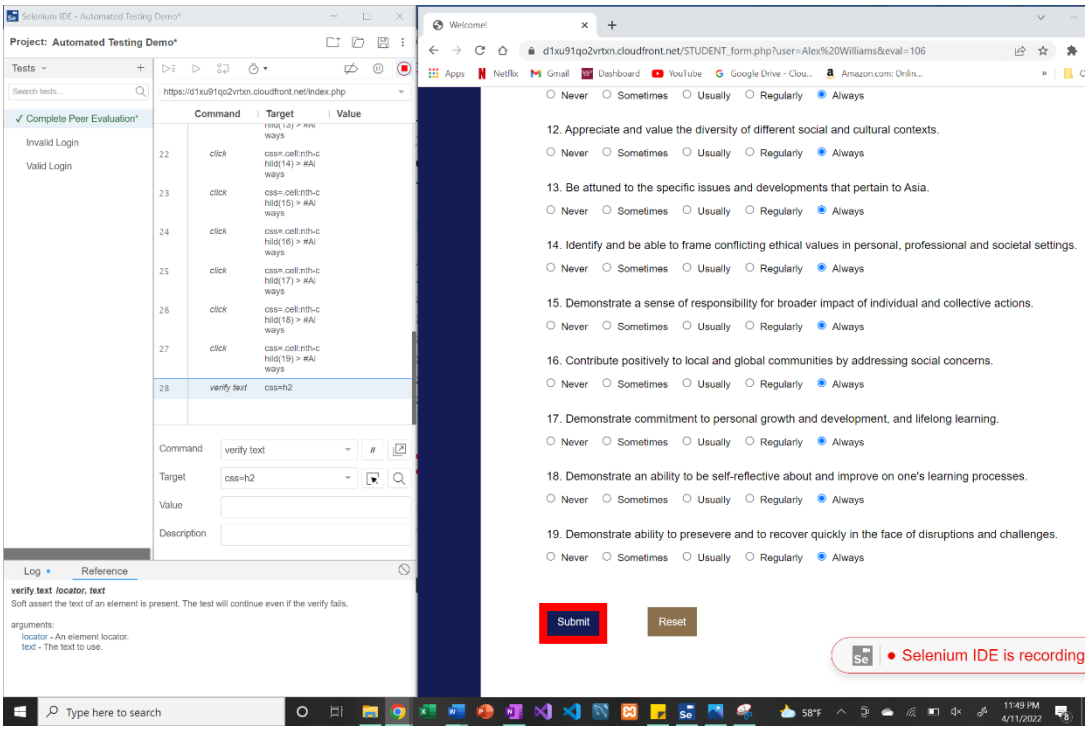
13. With the “Select target on page” button active, click the peer evaluation webpage’s title. This should say “Peer being evaluated: [Student’s Name].” By verifying that the label says the selected student’s name, we are checking that there is a connection to the student database. The element name will now appear in the “Target” textbox on the Selenium application window.



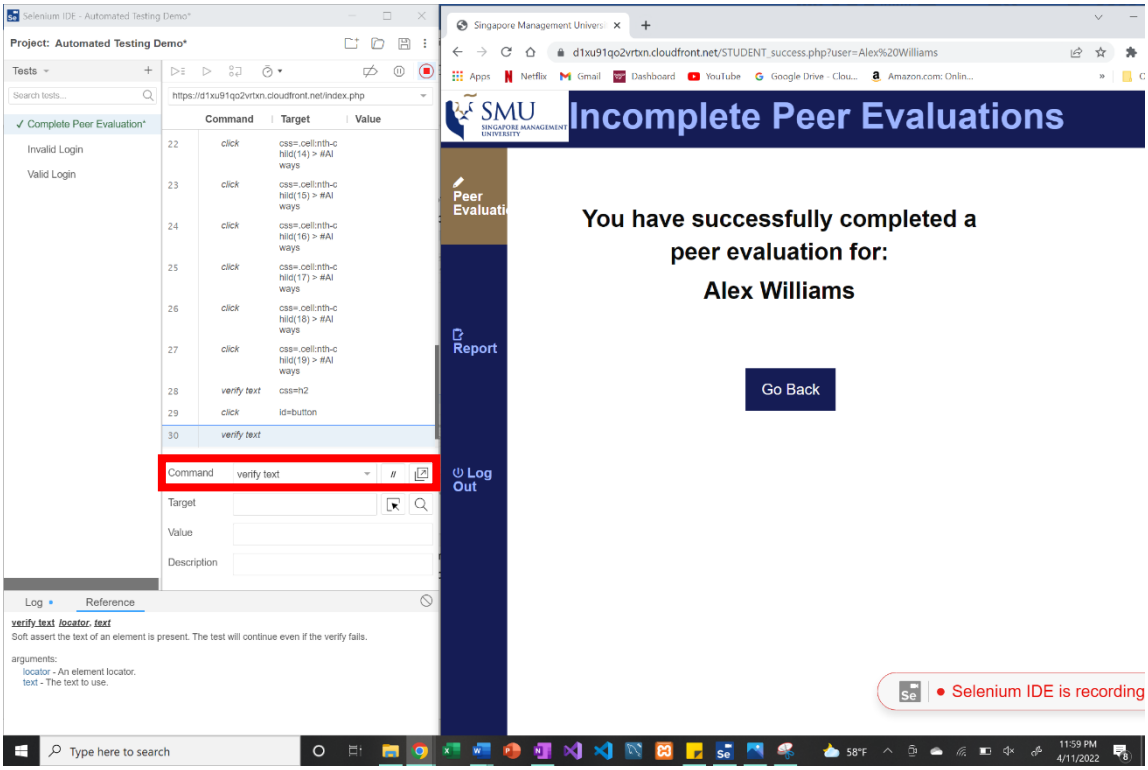
14. Click on the next blank row after the “verify text” command. By doing this, the recording will write commands after the “verify text” command instead of before.



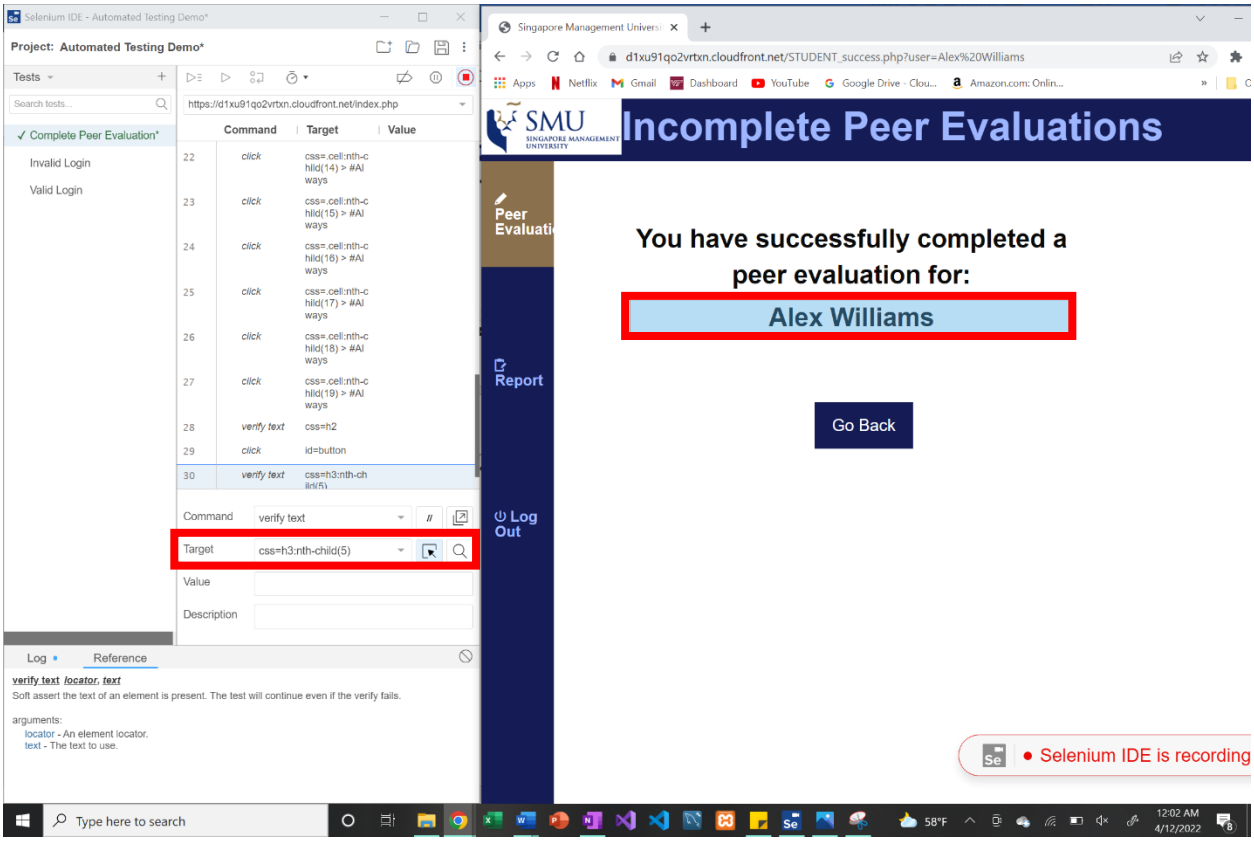
15. Scroll to the bottom of the webpage and press the “Submit” button. This will submit the use form with the selected peer evaluation answers.



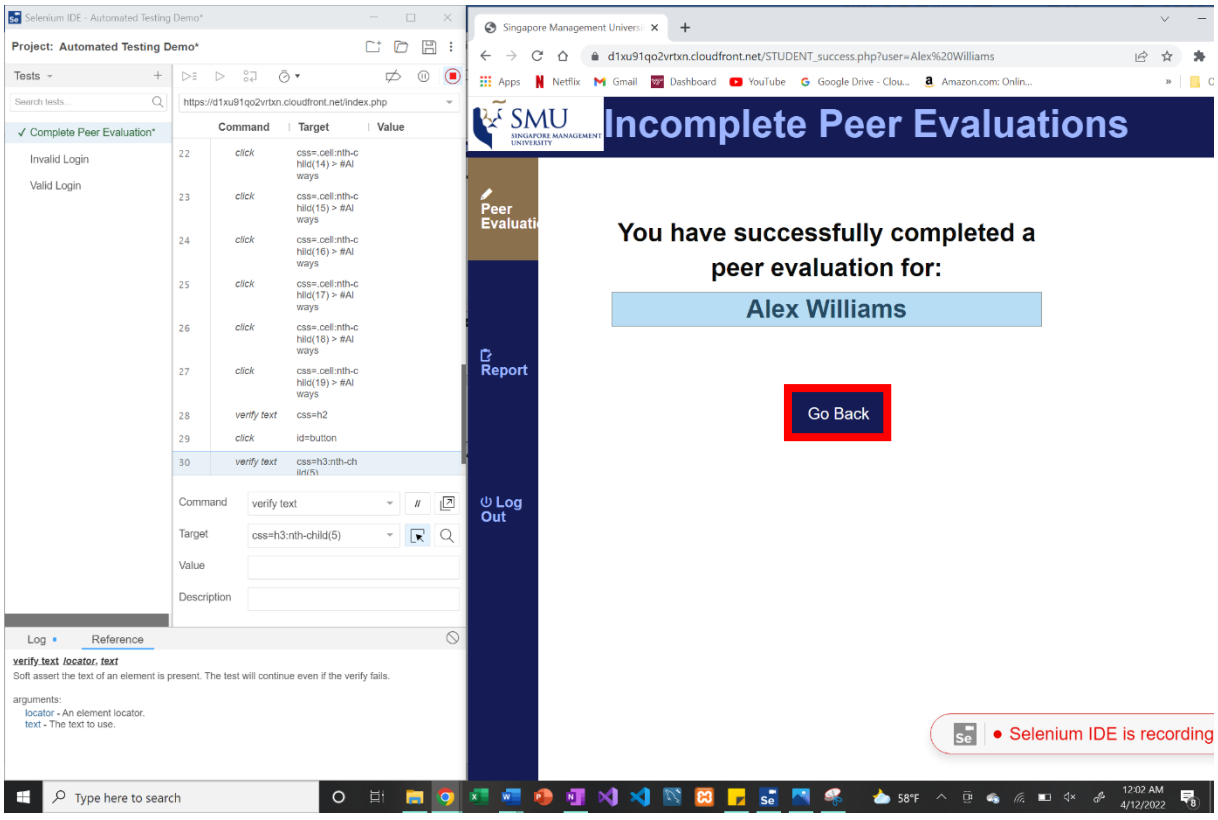
16. Verify that the confirmation text has a connection to the database by writing “verify text” into the “Command” dropdown of the Selenium IDE application window.



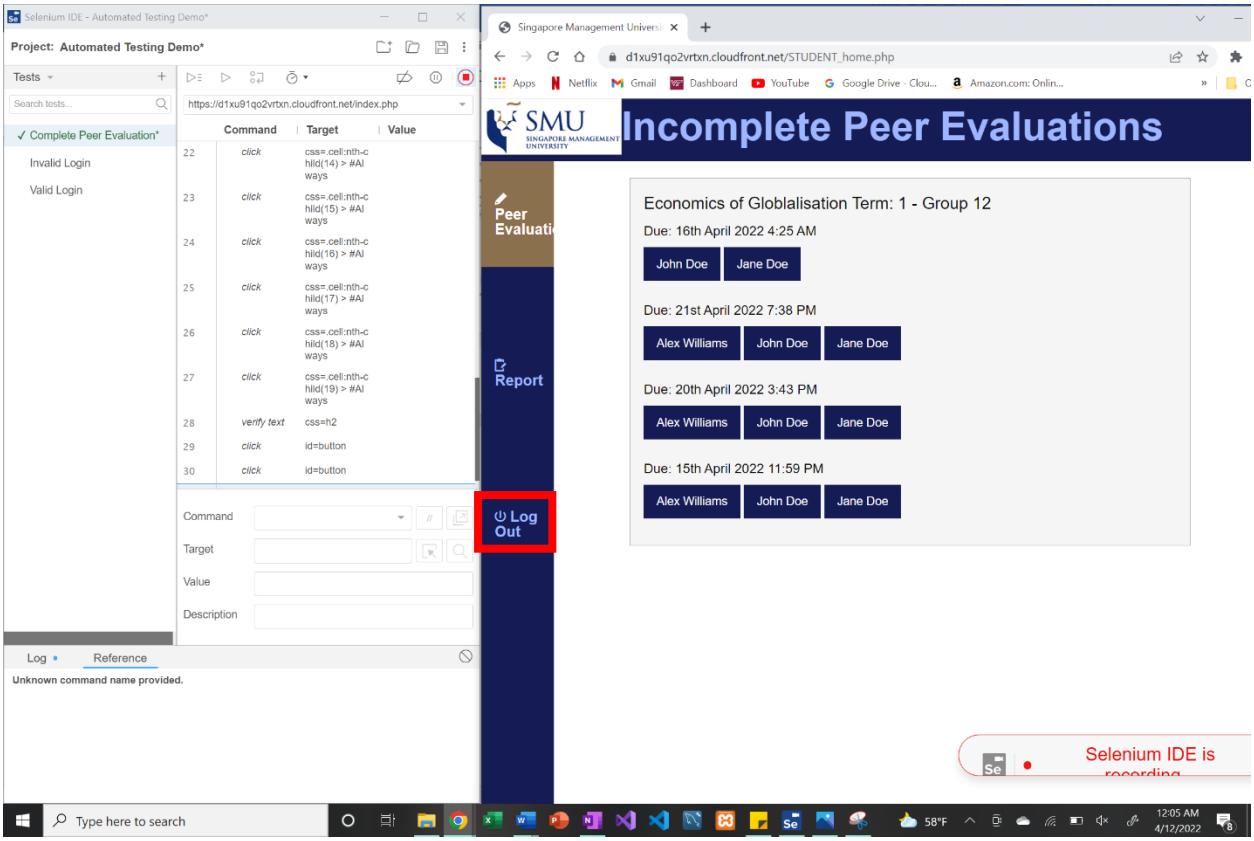
17. Click on the “Select target on page” button and select the [Student’s Name] element.



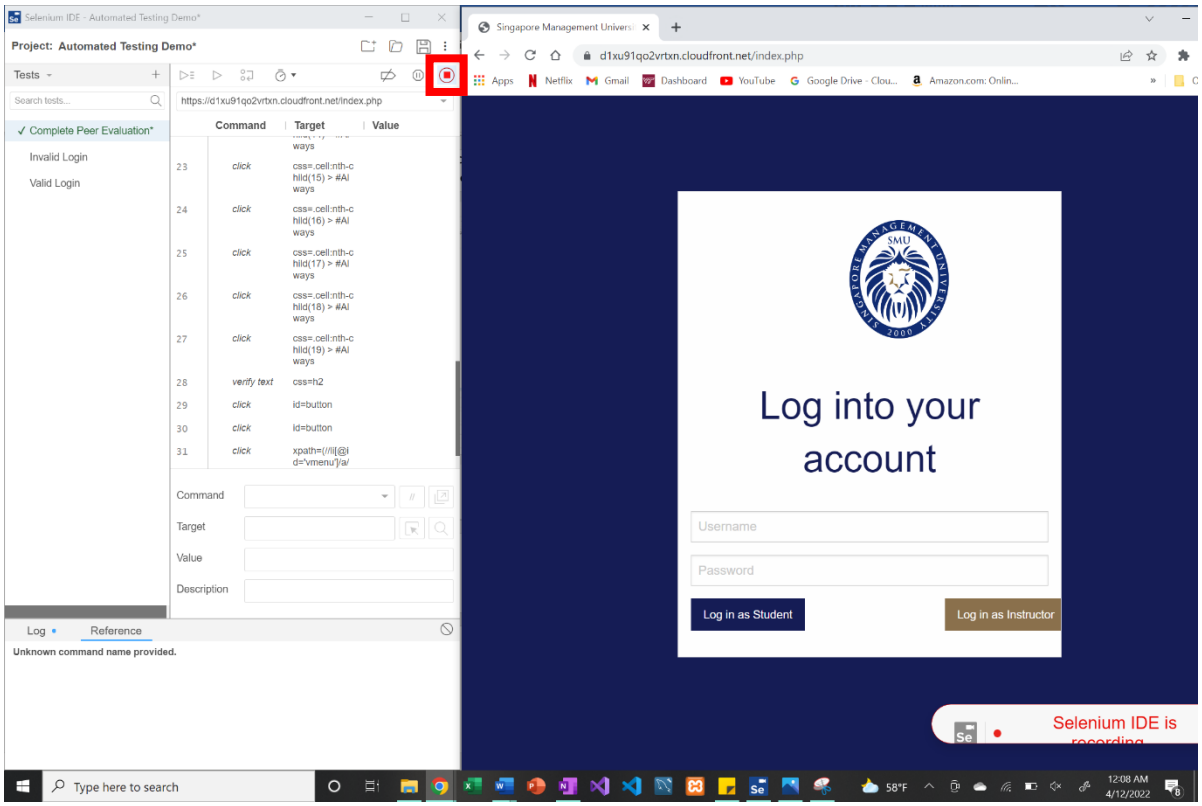
18. Click on the “Go Back” button.



19. Click on the “Log Out” button.



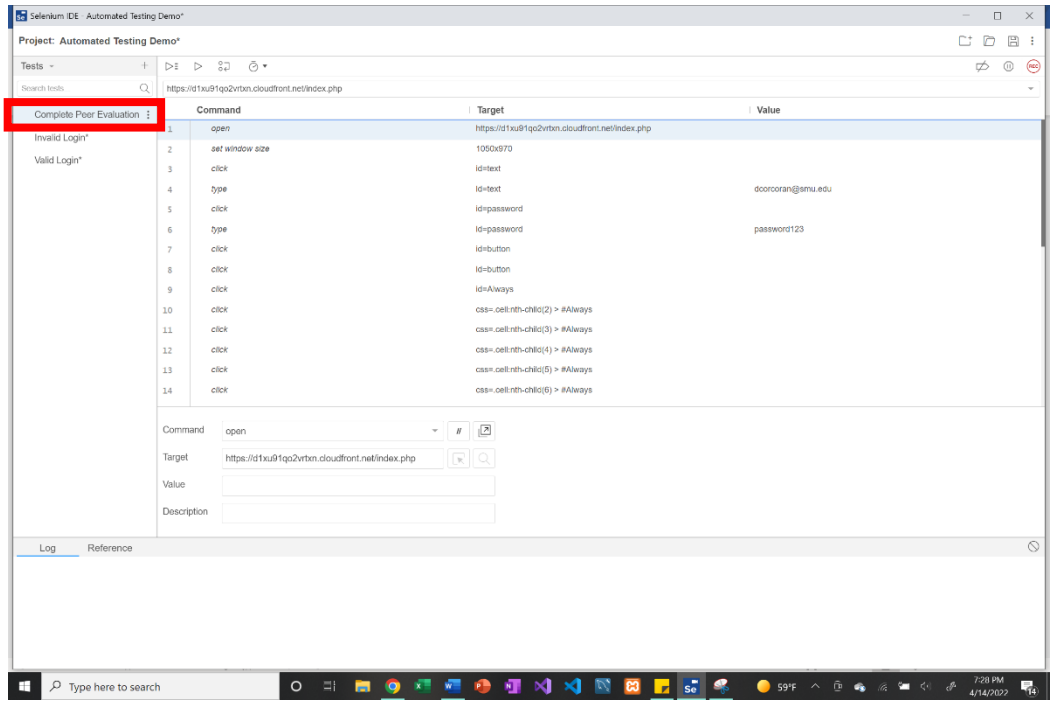
20. Press the “REC” button again to close the recording. The complete peer evaluation test case has been fully recorded.



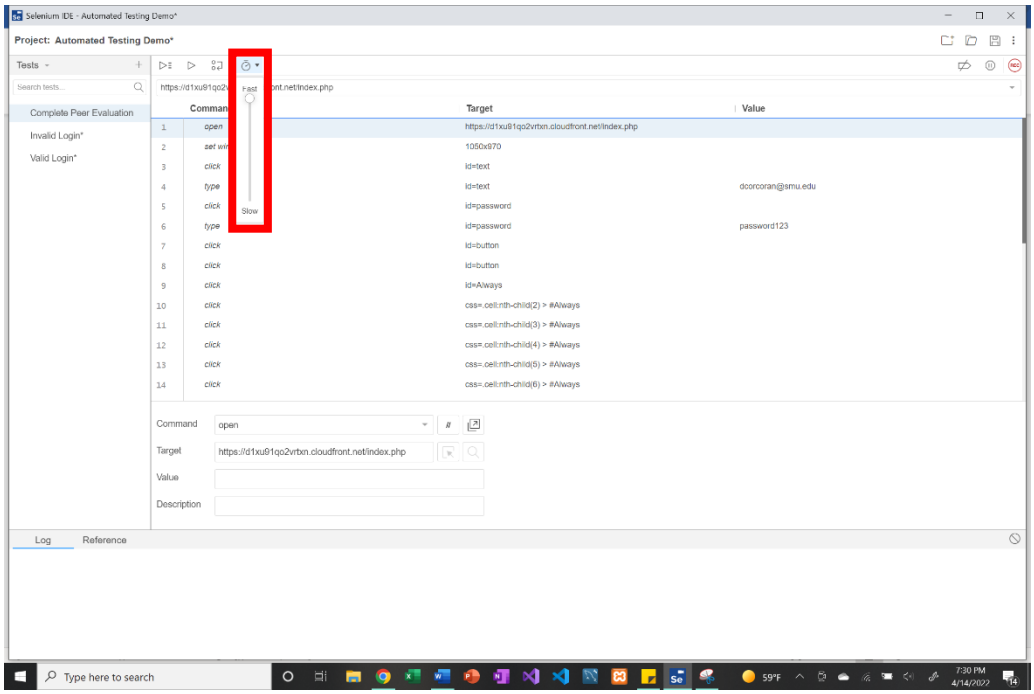
Test Playback

Replaying a test is very important is requirements testing. Replaying allows you to see your recorded actions in real time, or at a variable speed, as well as the running test's outcome.

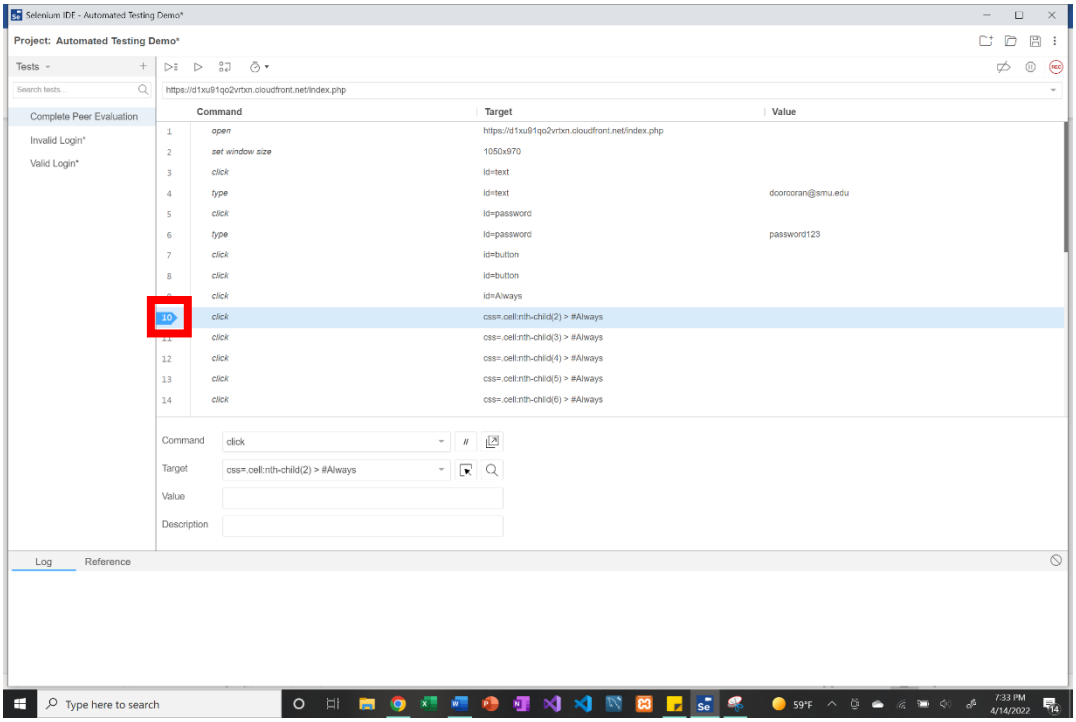
1. Click the "Complete Student Evaluation" test in the test case pane. The entire test case should already be completed from the previous steps.

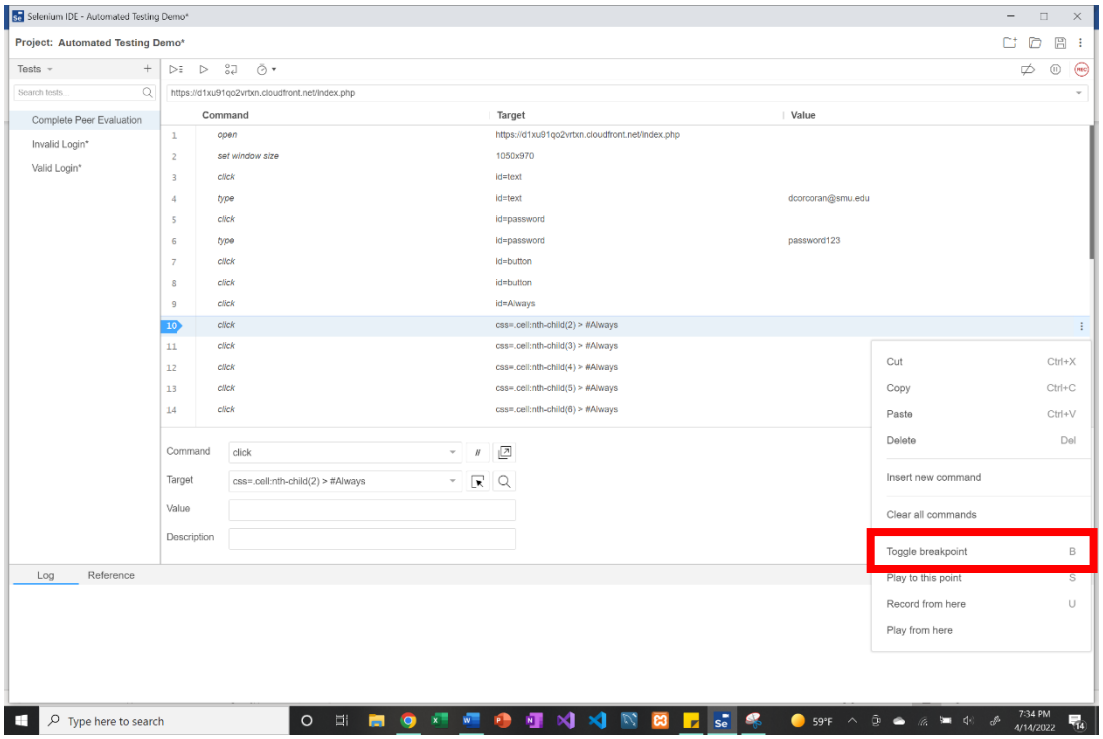


2. Set the replay speed with the slider at the top of the screen. You are able to slow down or speed up the replay of the test. I will put my test on the fastest setting.

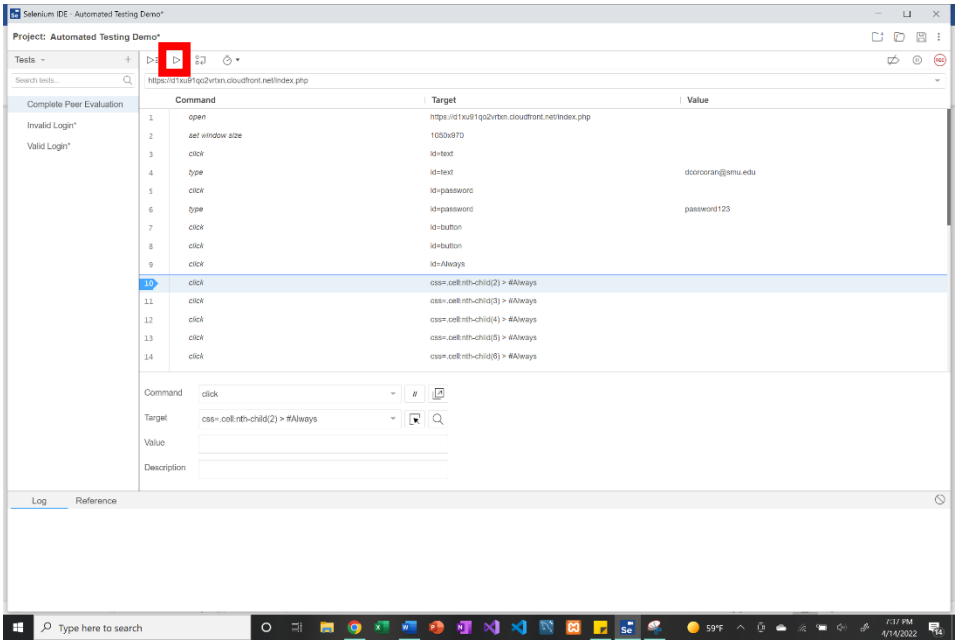


3. Set a breakpoint on step 10 by clicking the step number on the left side of testing dashboard. Alternatively, you can click the menu button at the right of the step and select “toggle breakpoint” from there. Breakpoints stop pause the breakpoint in the middle of testing.

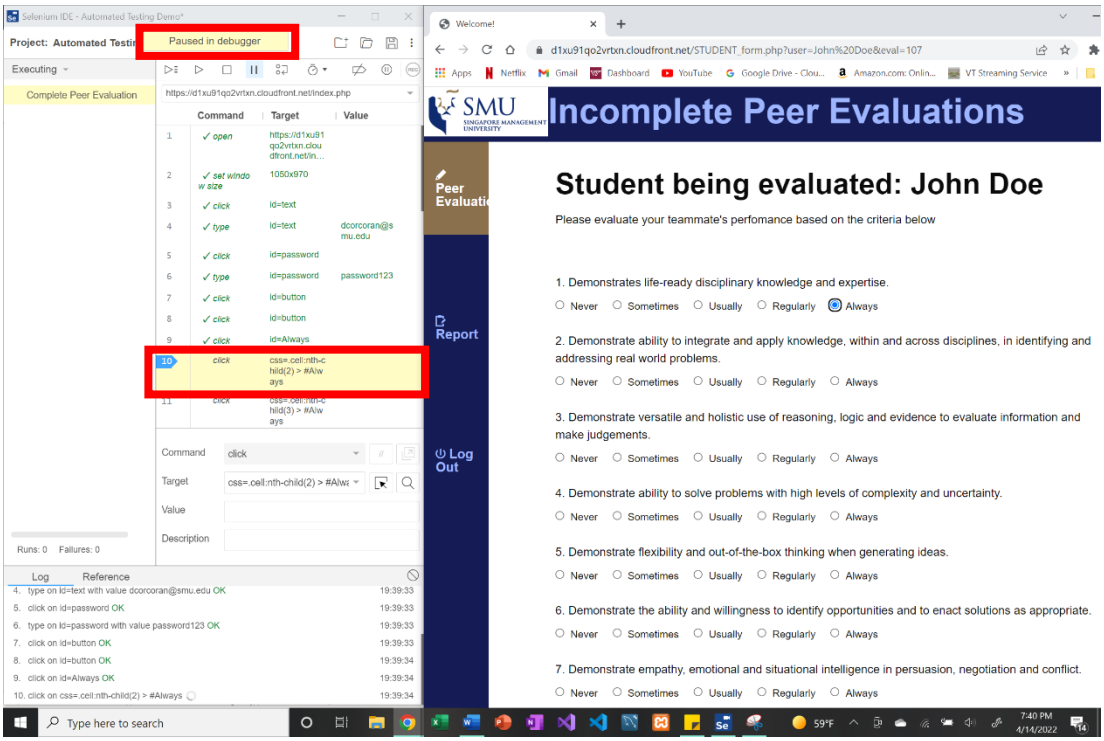




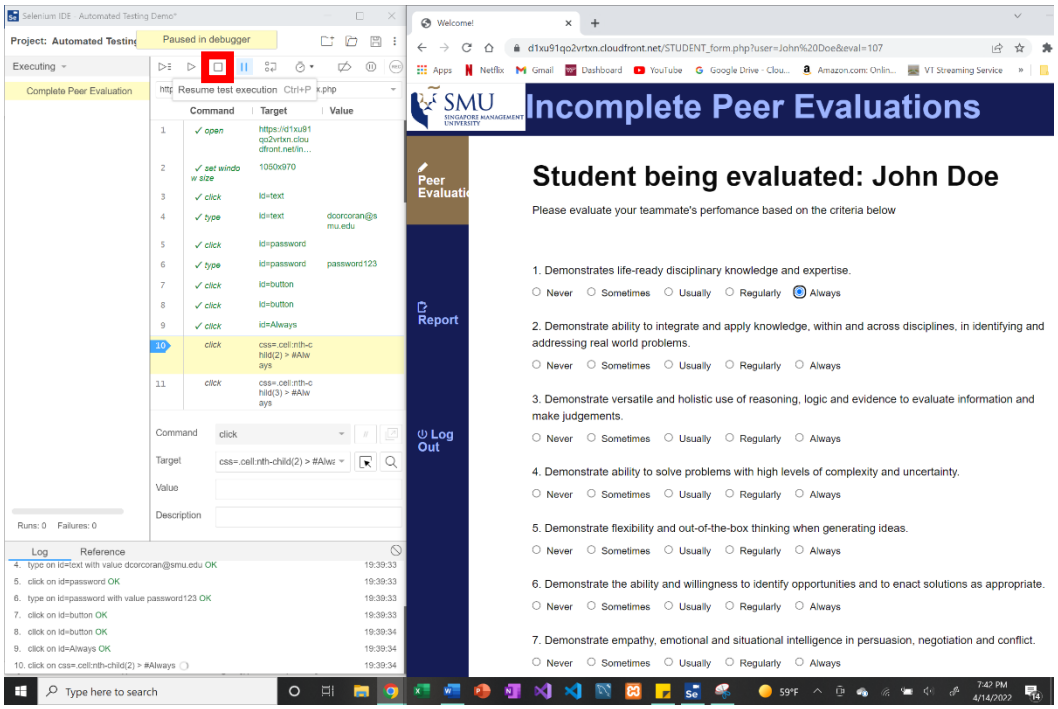
4. The first play button with three horizontal lines is a “play all” button. This button will replay all tests one after another. Because we have only completed one test, we will skip over this function.
5. Press the play button, the small triangle button at the top of the selenium IDE window. The replay will begin automatically once pressed.



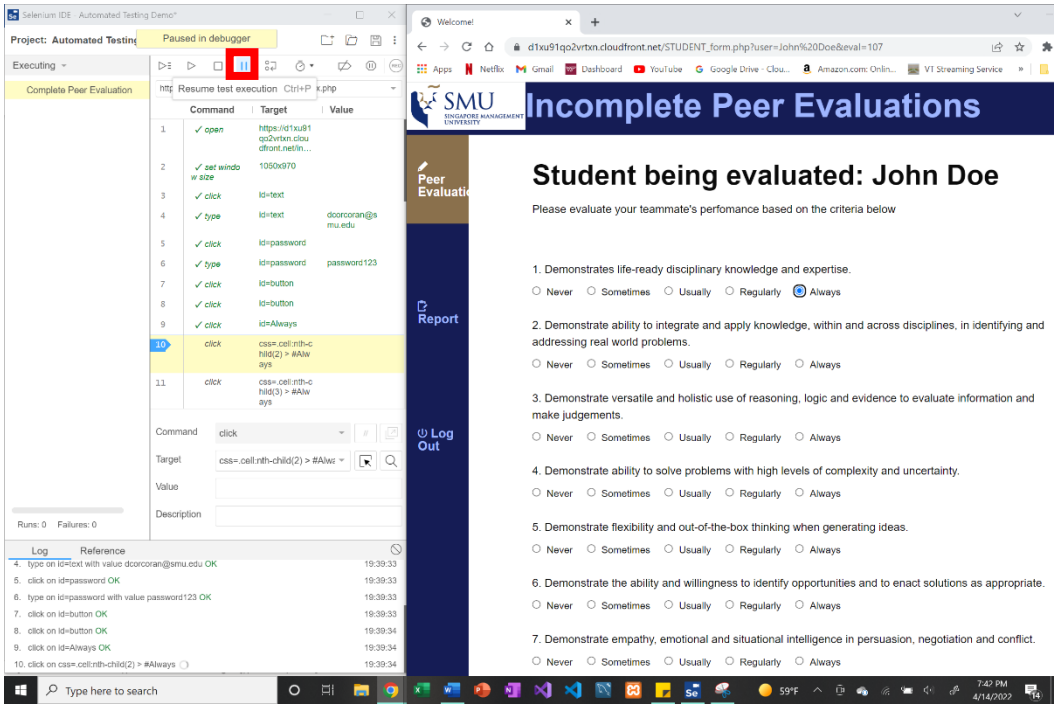
6. The replay will automatically stop at the previously selected breakpoint. A message saying “Paused in Debugger” will appear at the top of Selenium IDE.



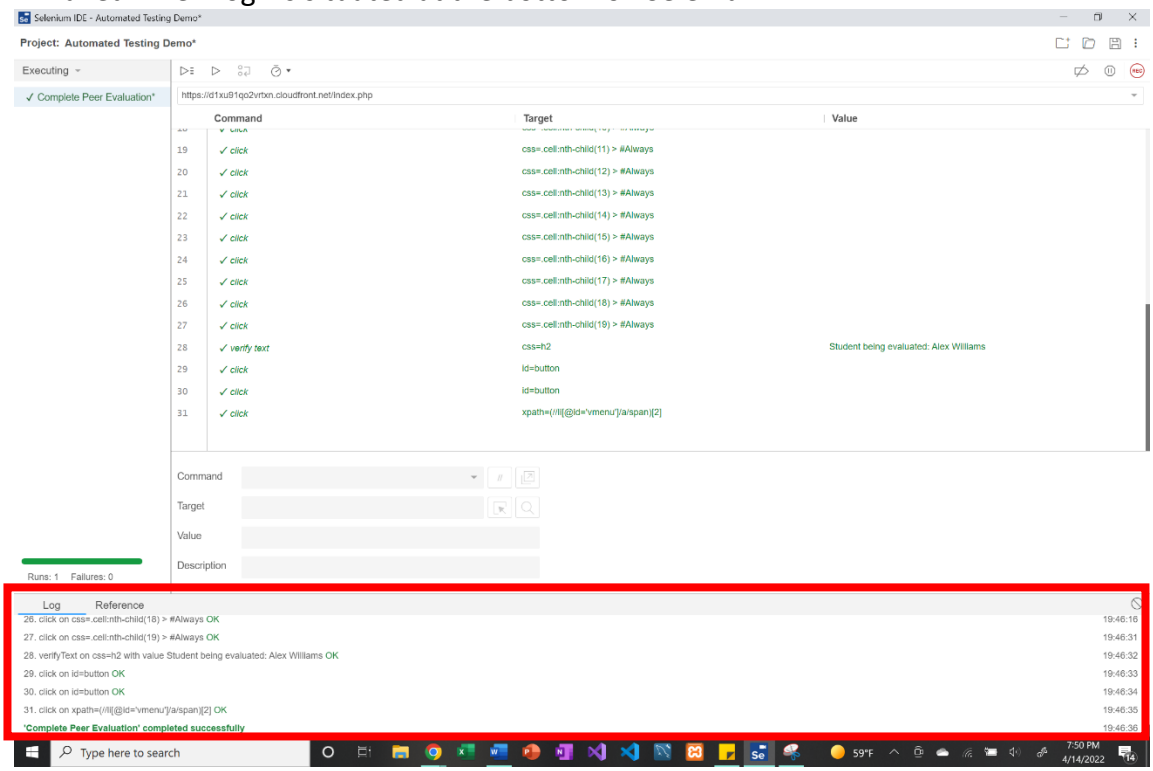
7. Every step previously played that pass should be highlighted in green. Steps that don't pass will be highlighted in red.
8. If at any time you want to stop the test, you can press the square button at the top of the testing page. However, do not press the button now.



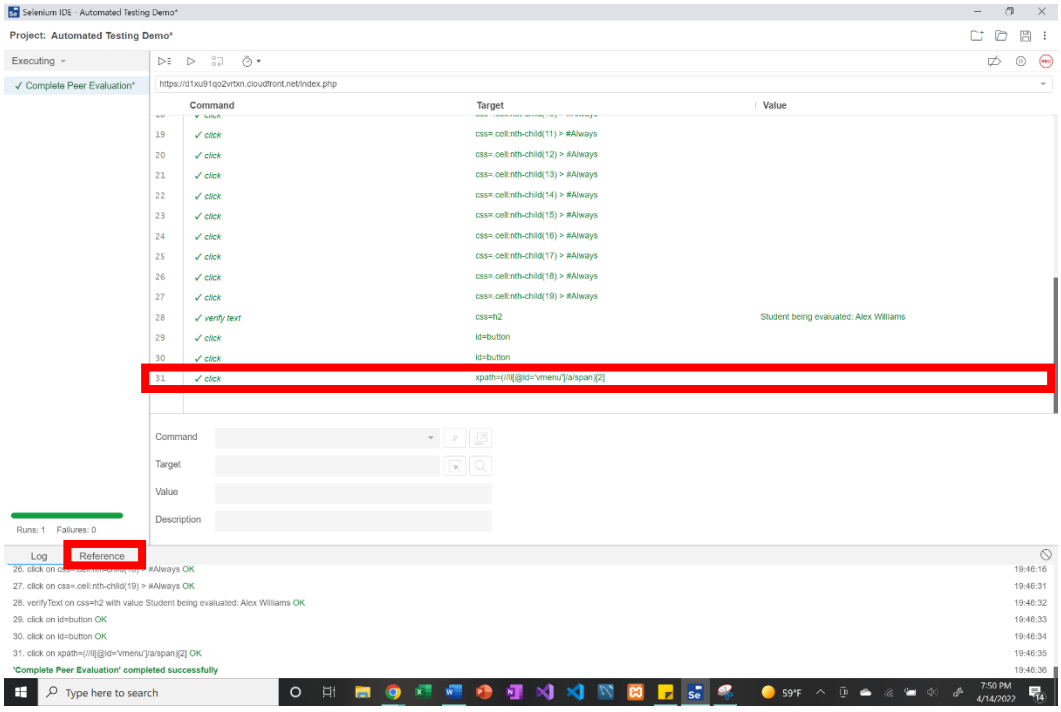
9. Resume the test by pressing the pause button, or the two vertical lines at the top of Selenium.

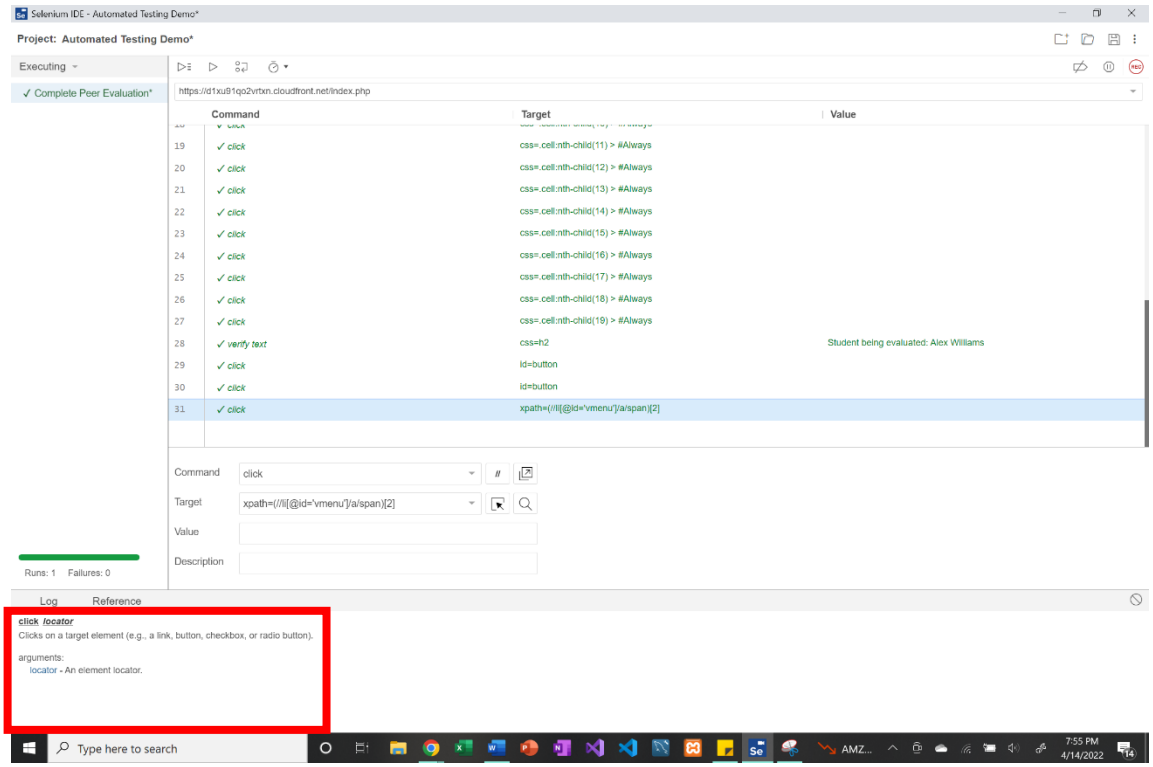


10. Once the replay is complete, the Selenium window will take control. The test will automatically stop once the last step is reached.
11. The “Log” keeps track of the steps taken and decides whether they have passed or failed. The “Log” is situated at the bottom on Selenium.

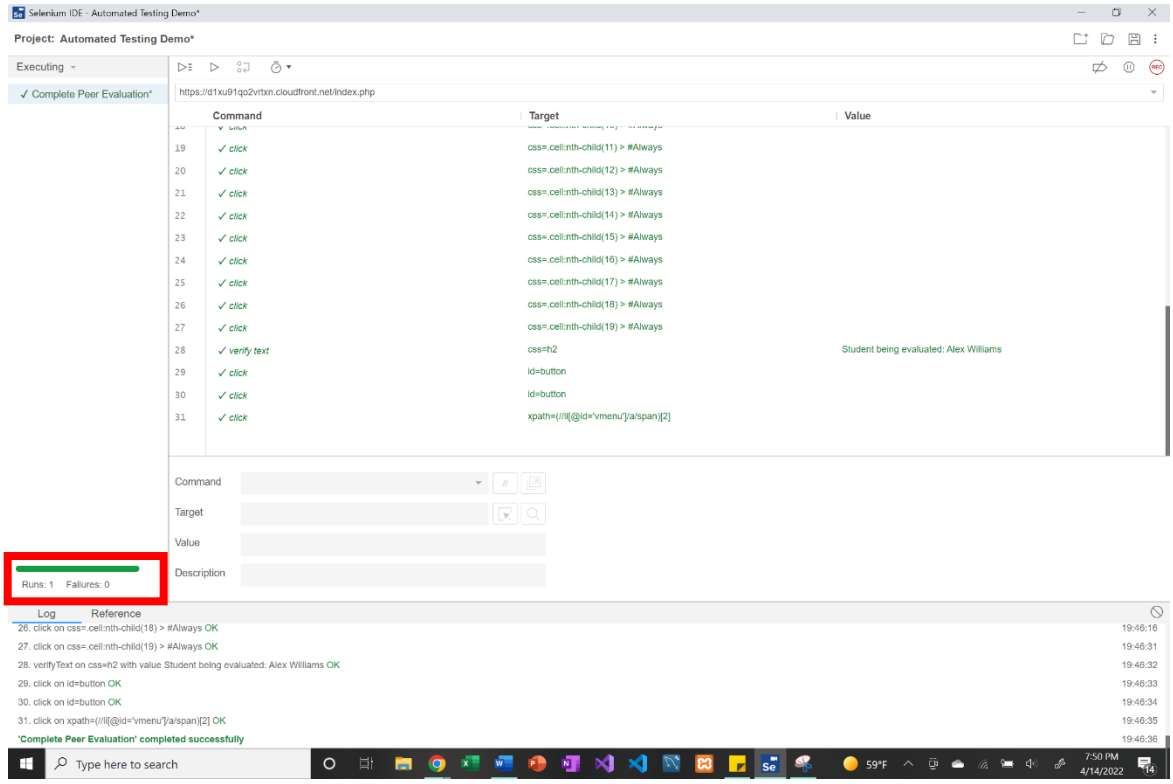


12. Click the last step and press the “reference” tab. Pressing the “Reference” tab next to the “Log” tab brings the user to a description of the highlighted step. In this cas it is the “click” command.

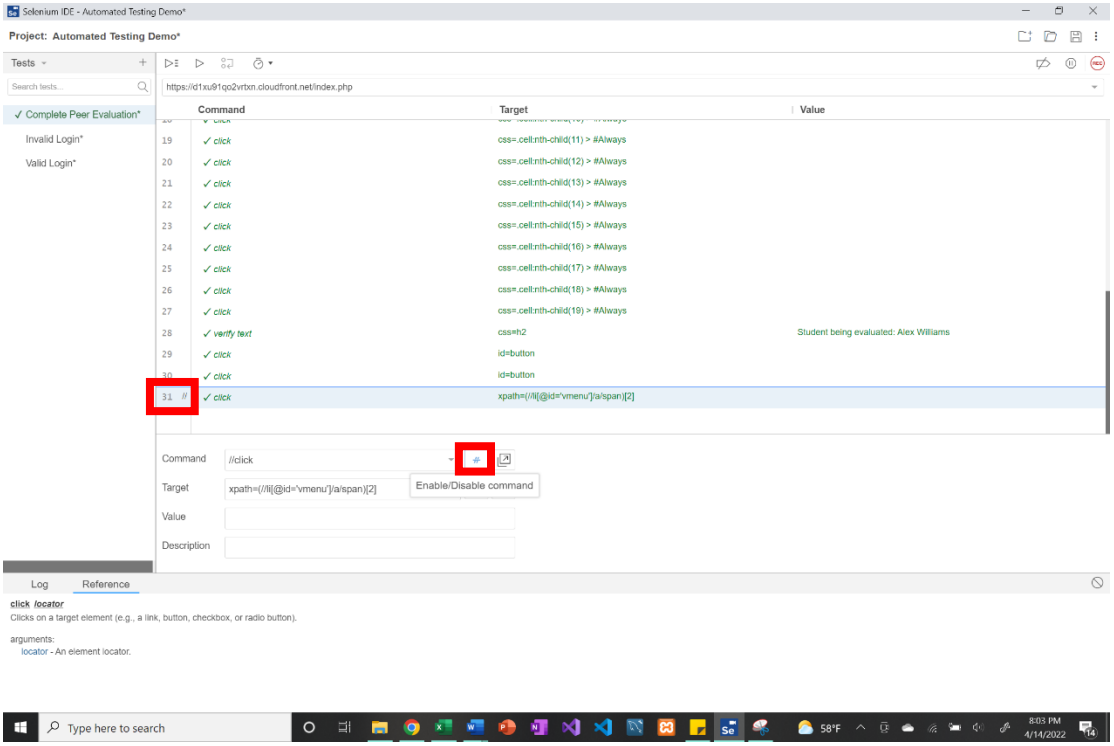




13. The number of runs and failures is also depicted at the bottom of Selenium's testing pane. This shows the result of the previously run tests and how many problems occurred during testing.



14. Click the comment button on the last test step. This button should look like two forward slashes in a row. This action nullifies the step altogether and will put the icon next to the step number.



Conclusion

Selenium IDE is a very helpful automated testing program. The software allows requirements analysts to run automated tests using a record and playback function, significantly reducing the time it can take to test website functionality. Selenium IDE is a completely free service with a wide variety of features useful in software testing. It has a simple graphical layout, does not require previous programming knowledge, and is easy to understand. Because the program is a browser extension, it very easy to install on any computer with internet connection and either Mozilla Firefox or Google Chrome. I highly recommend Selenium IDE to anyone in need of an automated testing technology especially if they have limited knowledge on the subject and technical requirements.