

American International University-Bangladesh (AIUB) Department of Computer Science

Faculty of Science &Technology (FST) Fall 2022-2023

Section: A

Software Quality and Testing

**Assignment on Automated Testing Tool**

|  |  |  |
| --- | --- | --- |
| SN | Student Name | Student ID |
| 1 | Nibal, Tasnim Alam | 20-42891-1 |

Under the supervision of

ABHIJIT BHOWMIK

Associate Professor & Special Assistant [OSA], Computer Science

Faculty of Science and Technology (FST),

American International University-Bangladesh

Table of Contents

[1. First approach (Selenium Web Driver using dot net) : 3](#_Toc130850406)

[2. Test code and test run : 5](#_Toc130850407)

[3. Test results : 6](#_Toc130850408)

[4. Second Approach (Selenium IDE by chrome extension) : 7](#_Toc130850409)

[5. Test Process : 8](#_Toc130850412)

[6. Test Result : 9](#_Toc130850413)

# First approach (Selenium Web Driver using dot net) :

* **Installation Process -**

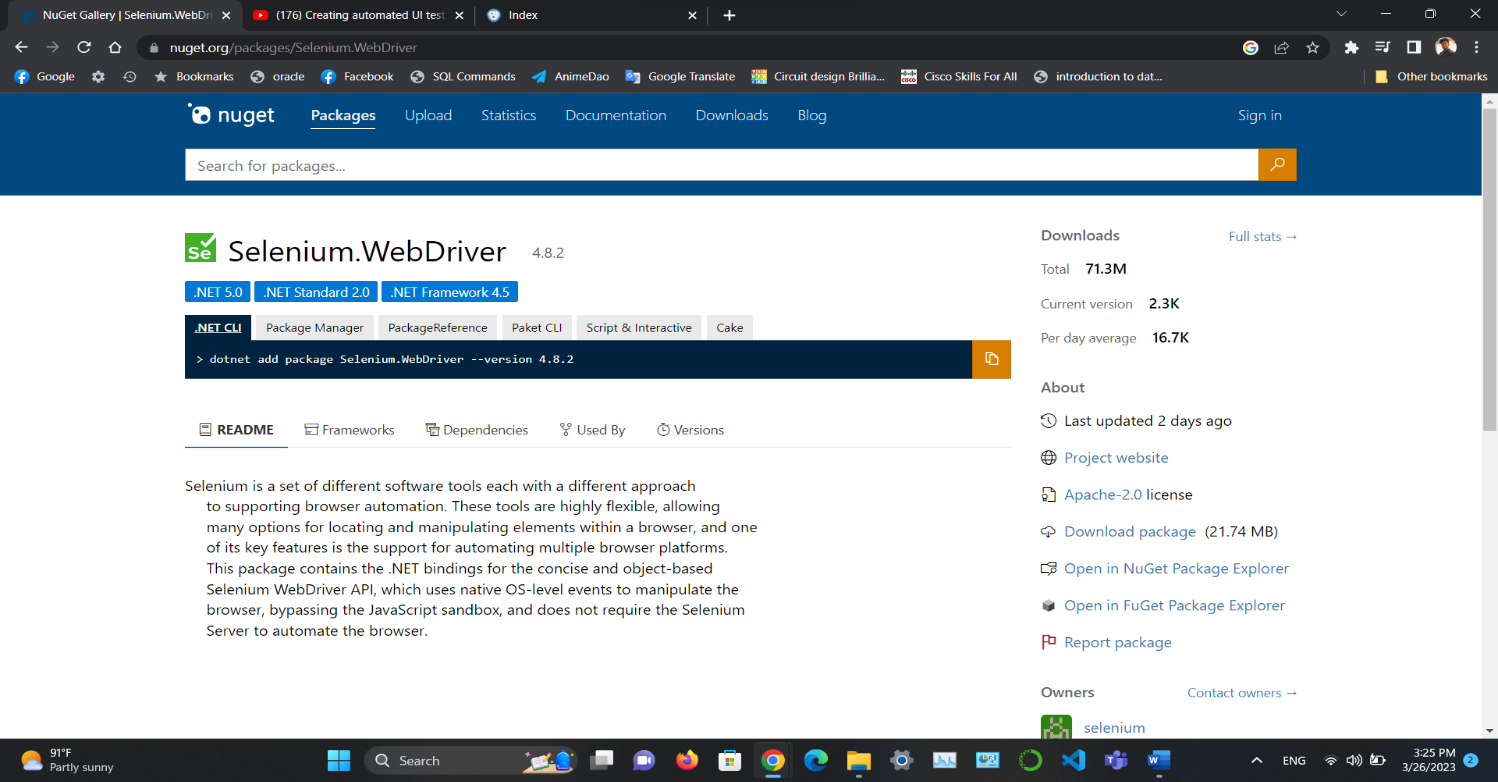


Figure 1: Website addressing nuget management to use selenium.

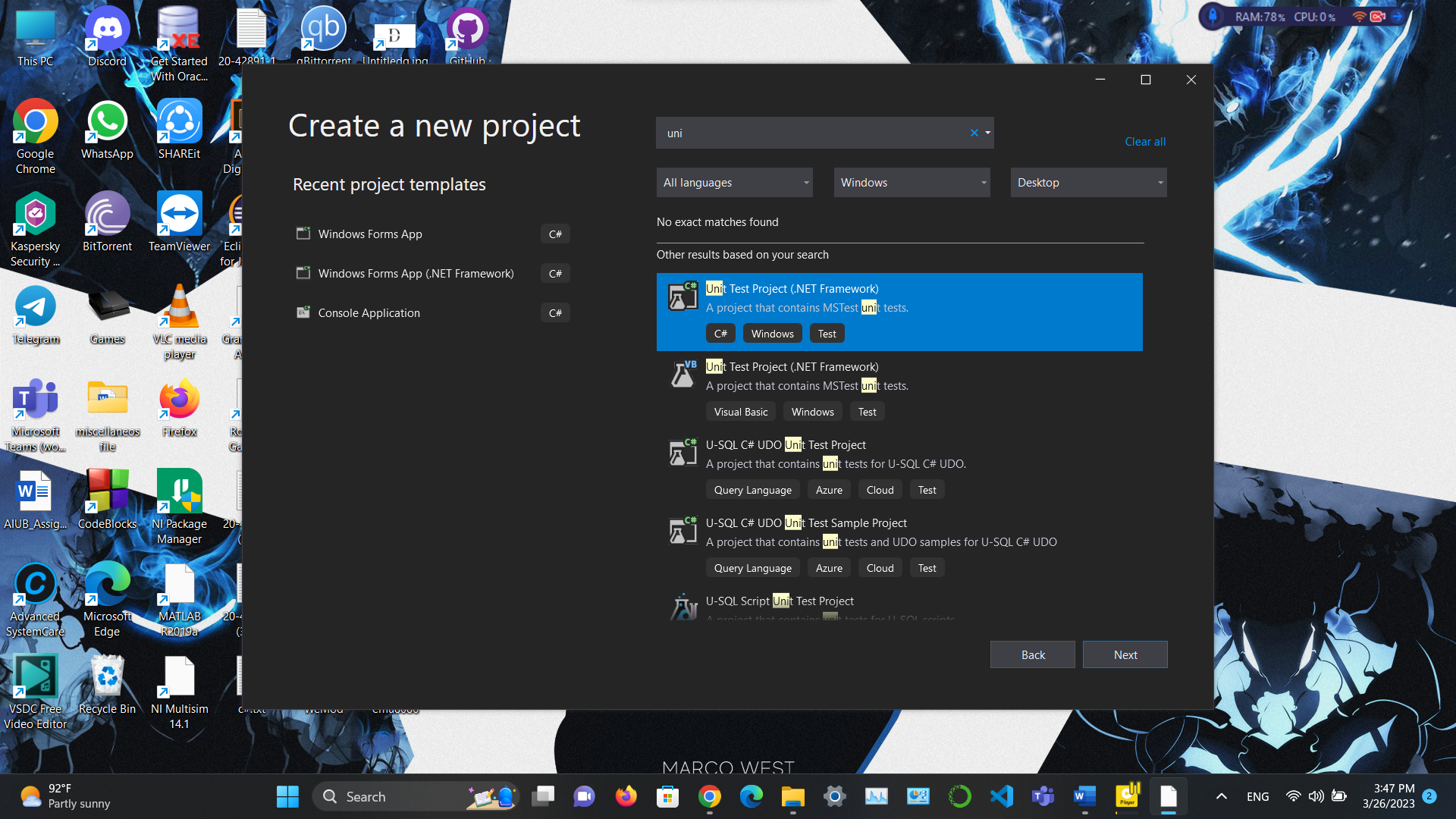
In **figure 1** As I have chosen to introduce my testing tool as selenium web driver it focuses on and advice on to install it by my preferable ide platform, so I choose visual studio as for my unit testing or automated testing to run on.

Figure 2:Visual Studio Project Insertion

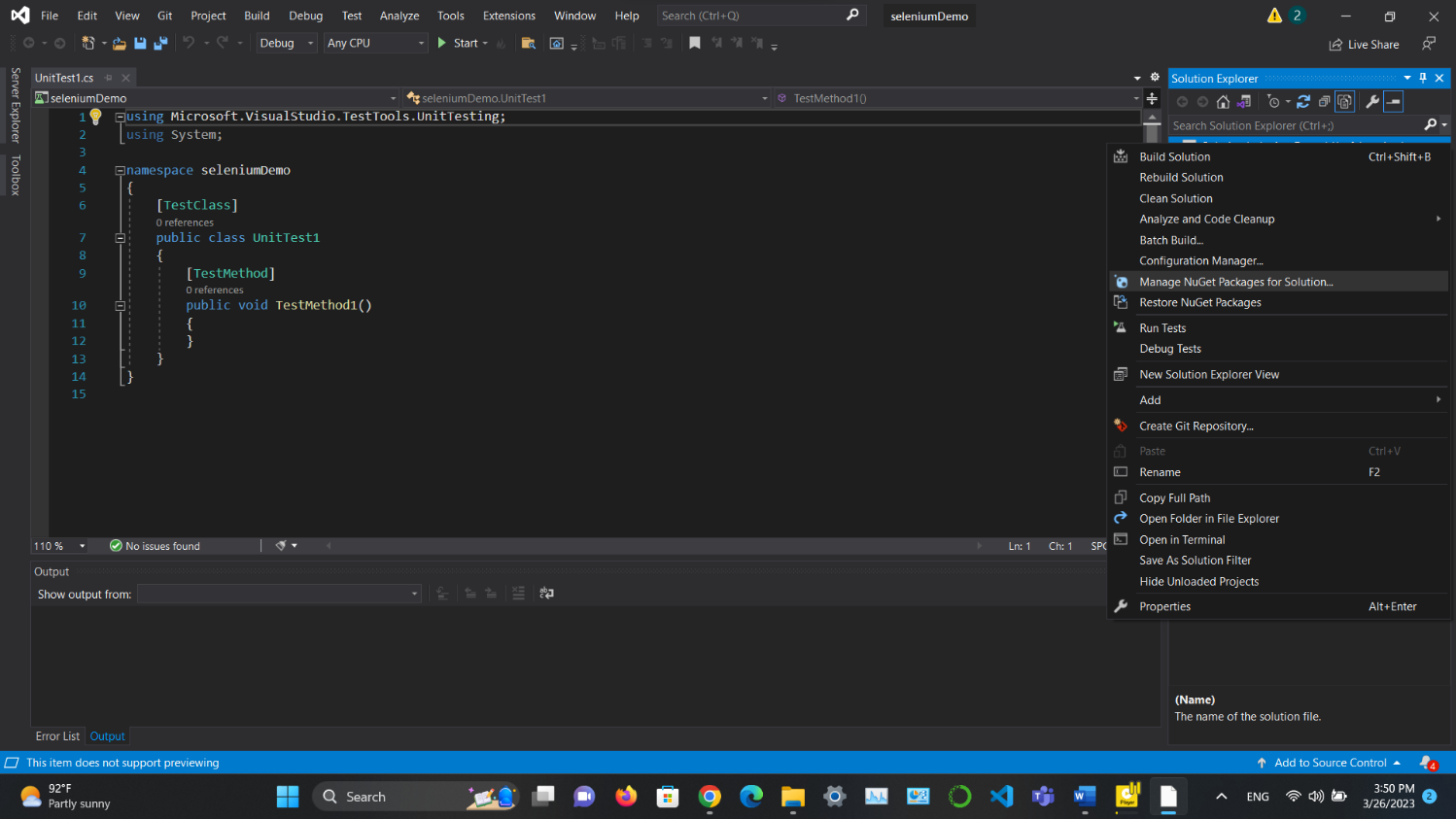
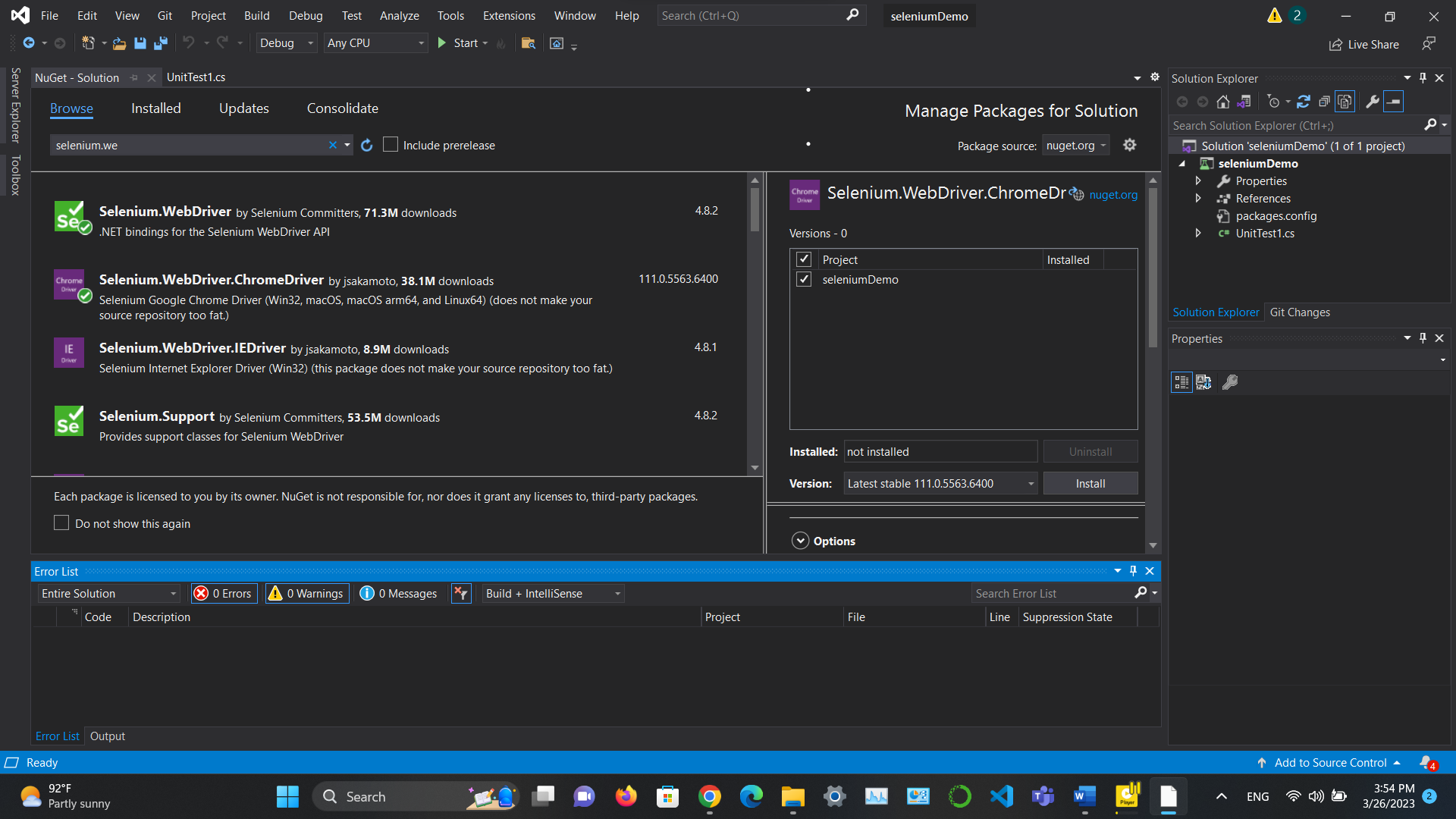
In **figure 2** I did not show the installation process of visual studio because I already have it. And after opening visual studio we are going to create a project to do a unit test project by dot net.

Figure 4: Installation and Completion of Selenium

Figure 3:Managing NuGet packages.

In **figure 3 and 4** complete focuses on how we are going to install the Selenium web driver and how we are going to use it. As we can see we are focusing on solution explorers NuGet packages and the we browse for Selenium web driver and chrome driver to test a project.

# Test code and test run :

Figure 5: Test code and Test run

In **figure 5** we have written test code using C# script and Selenium WebDriver APIs. To do our automated test run for our project.

# A picture containing graphical user interface Description automatically generatedGraphical user interface, text Description automatically generatedA screenshot of a computer Description automatically generatedTest results :

Figure 6: Viewing source web project.

Figure 7: Stored keys of login criteria’s.

Figure 8: Successful login and end of automated test run.

the C# script and Selenium WebDriver APIs used on Microsoft Visual Studio and the NUnit testing framework and contained code to run and show results like such as maximize the browser window, navigate to the web application's home page, find, and click the login link, enter the username and password values, submit the login form, and close the browser. Which is shown in the figures stated above. **(Figure 6 – 8)**

# Second Approach (Selenium IDE by chrome extension) :

# Installation Process –

# 

Figure 9: Adding selenium IDE as extension of chrome.

In **figure 9** it is shown that we have added selenium IDE as our testing tool in our chrome extension.

# Test Process :

Figure 10: Create a test record.

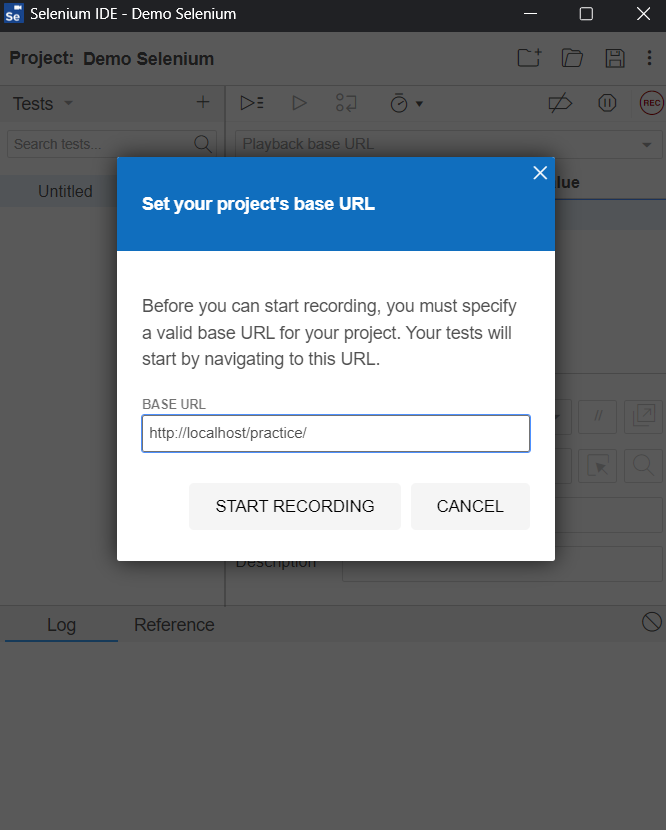


Figure 12: Record ongoing.

Figure 11: Navigate the project website to run record.

In **figure 10-12** it shows how the IDE is going to work. First of all we will create a test record by navigating it’s source project website. Then we will run the record to create a test record result.

# Test Result :

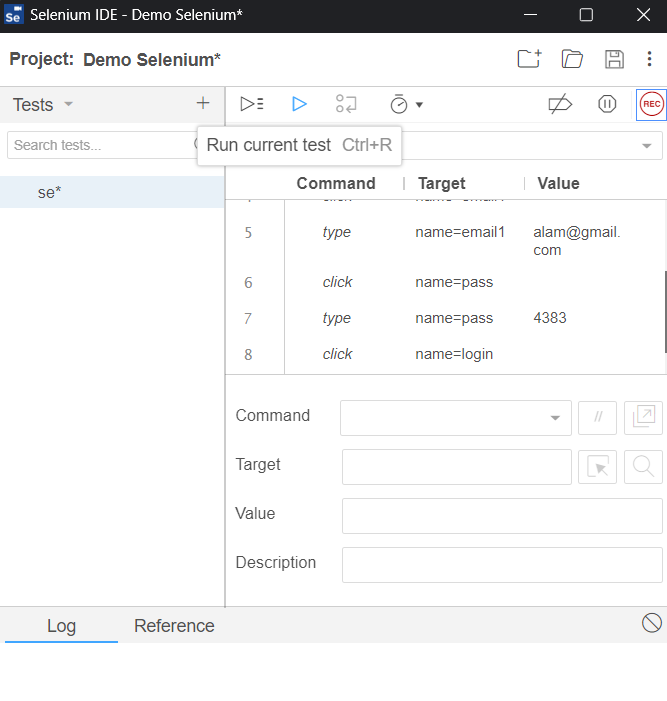
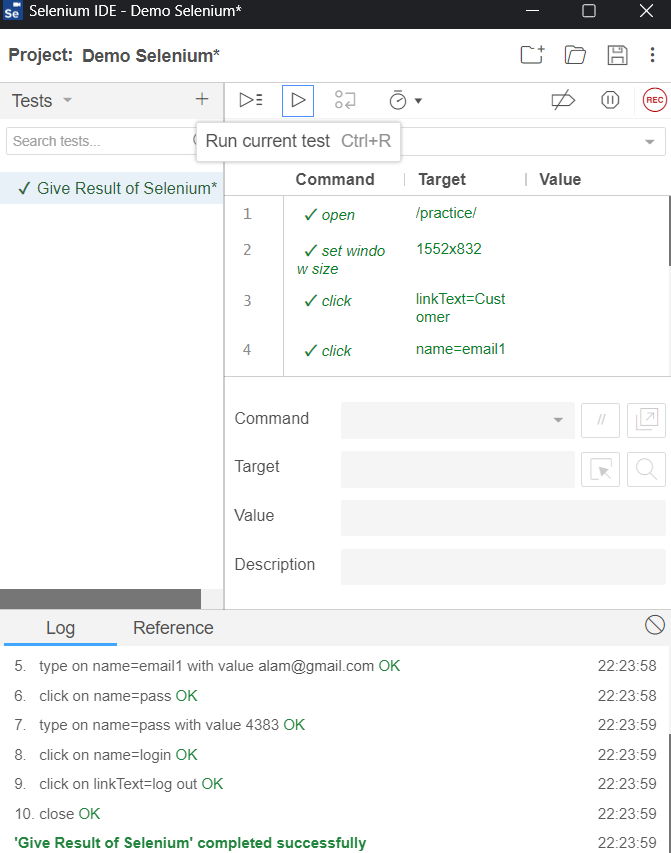


Figure 14: Successful test result.

Figure 13: Test run.

In **figure 13 and 14** it shows how the IDE is going to give us test result. After the record ends, we will have run the test and it will automatedly see through all our action and will give result of our testing.

**Noted:** This test was runed to test a course outlined project done in a recent course.