

Selenium Automation

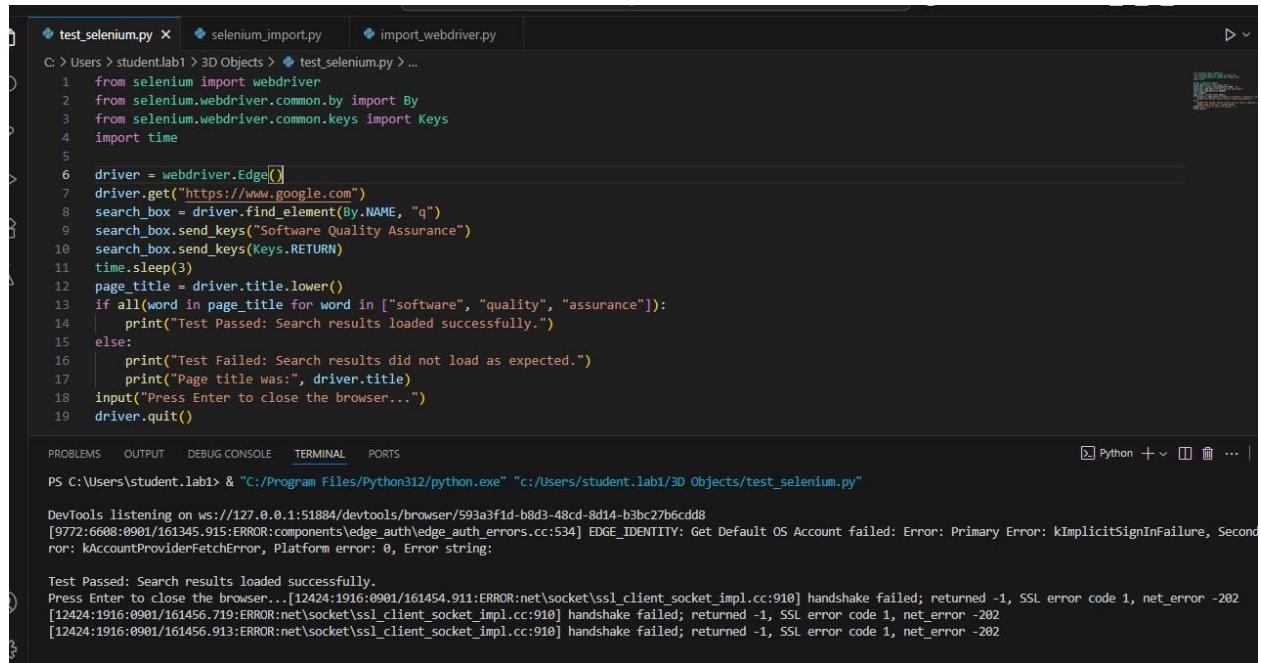
Introduction

This practical activity focuses on the implementation of **Selenium Automation**. The main purpose of this activity is to gain practical experience in automating web application testing using the Selenium framework, thereby enhancing their understanding of software quality assurance techniques.

Selenium is an open-source automation tool primarily used for testing web-based applications across different browsers and platforms. It supports multiple programming languages such as Python, Java, and C#, enabling developers and testers to write test scripts that interact directly with web elements. Through this automation, repetitive and time-consuming manual testing tasks can be minimized, improving both accuracy and efficiency in the testing process.

In this practical, students are guided through several essential steps, including the installation and verification of **Python** and **Selenium**, and the setup of a suitable development environment such as **Visual Studio Code**. The exercises involve automating functional tests on popular web applications, starting with a **Google Search Test** to validate search functionality, followed by a **Login Function Test** using both a practice testing website and the **OrangeHRM** demo site. These tasks help students understand how Selenium can simulate real user interactions—such as entering text, clicking buttons, and verifying responses—to ensure that web applications perform as expected.

By completing this activity, students gain practical insights into the importance of automation in software quality assurance. They also develop the technical skills required to design, execute, and maintain automated test scripts that contribute to improving the reliability, efficiency, and maintainability of software systems.

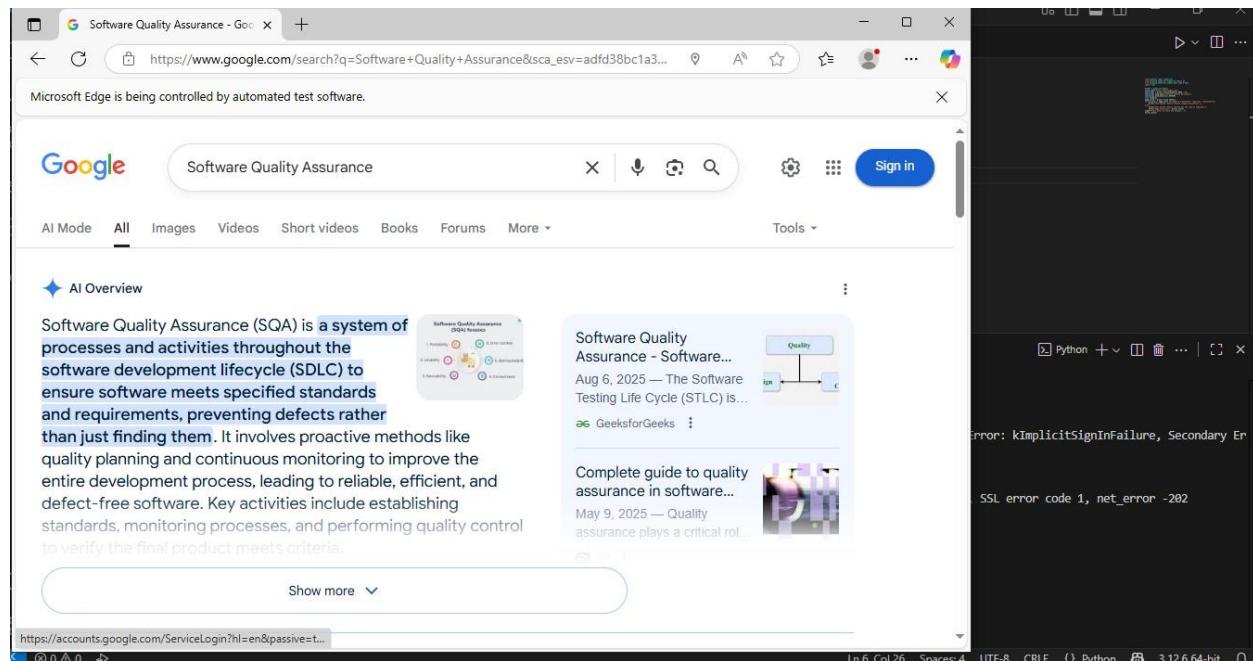


The screenshot shows a code editor with three tabs: `test.selenium.py`, `selenium_import.py`, and `import_webdriver.py`. The `test.selenium.py` tab contains the following Python code:

```
C:\ > Users > student.lab1 > 3D Objects > test.selenium.py > ...
1  from selenium import webdriver
2  from selenium.webdriver.common.by import By
3  from selenium.webdriver.common.keys import Keys
4  import time
5
6  driver = webdriver.Edge()
7  driver.get("https://www.google.com")
8  search_box = driver.find_element(By.NAME, "q")
9  search_box.send_keys("Software Quality Assurance")
10 search_box.send_keys(Keys.RETURN)
11 time.sleep(3)
12 page_title = driver.title.lower()
13 if all(word in page_title for word in ["software", "quality", "assurance"]):
14     print("Test Passed: Search results loaded successfully.")
15 else:
16     print("Test Failed: Search results did not load as expected.")
17     print("Page title was:", driver.title)
18 input("Press Enter to close the browser...")
19 driver.quit()
```

The terminal below shows the command run and the output of the test:

```
PS C:\Users\student.lab1> & "C:/Program Files/Python312/python.exe" "c:/Users/student.lab1/3D Objects/test.selenium.py"
DevTools listening on ws://127.0.0.1:51884/devtools/browser/593a3f1d-b8d3-48cd-8d14-b3bc27b6cdd8
[9772:6688:0901/161345.915:ERROR:components\edge_auth\edge_auth_errors.cc:534] EDGE_IDENTITY: Get Default OS Account failed: Error: Primary Error: kImplicitSignInFailure, Secondary Error: kAccountProviderFetchError, Platform error: 0, Error string:
Test Passed: Search results loaded successfully.
Press Enter to close the browser...[12424:1916:0901/161456.719:ERROR:net\socket\ssl_client_socket_impl.cc:910] handshake failed; returned -1, SSL error code 1, net_error -202
[12424:1916:0901/161456.913:ERROR:net\socket\ssl_client_socket_impl.cc:910] handshake failed; returned -1, SSL error code 1, net_error -202
[12424:1916:0901/161456.913:ERROR:net\socket\ssl_client_socket_impl.cc:910] handshake failed; returned -1, SSL error code 1, net_error -202
```

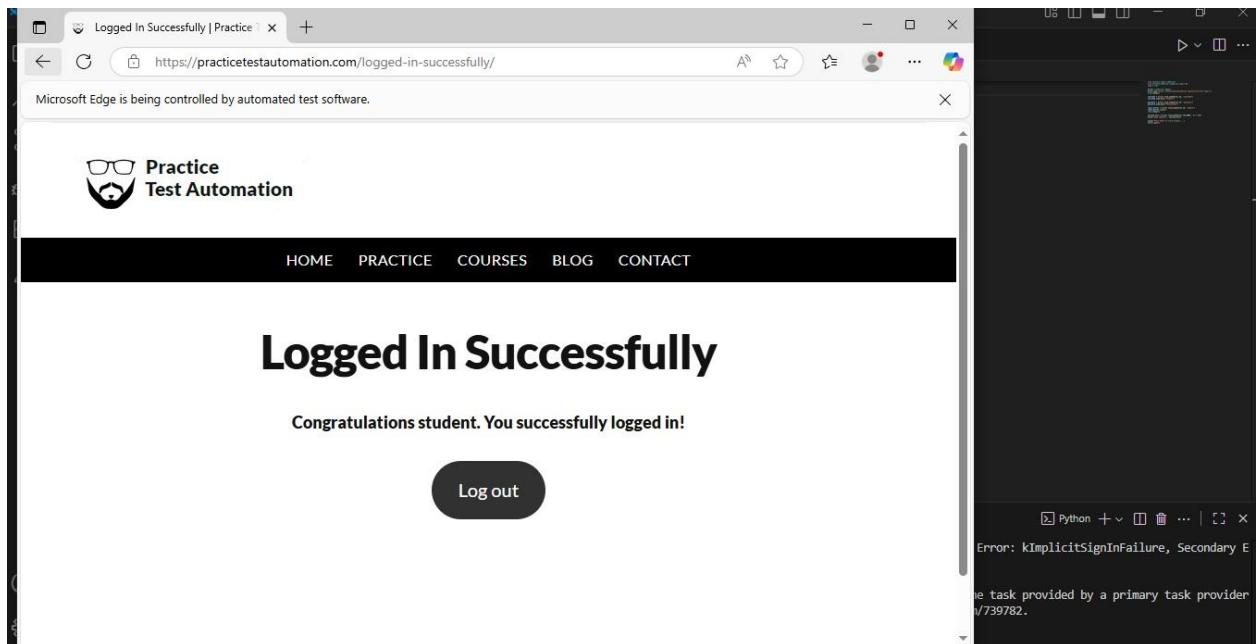


A screenshot of a code editor window. The top bar shows tabs for 'test_selenium.py', 'selenium_import.py', and 'import_webdriver.py'. The main area displays Python code for a Selenium test. The code imports webdriver, common, time, and selenium.webdriver.common.by. It creates an Edge driver, navigates to a practice login page, enters 'student' for the username and 'Password123' for the password, clicks the submit button, prints the welcome text, and quits the browser. Below the code, the terminal tab is active, showing the command 'Test Passed: Search results loaded successfully.' followed by several SSL error messages.

```
C: > Users > studentlab1 > 3D Objects > selenium_import.py > ...
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 import time
4
5 driver = webdriver.Edge()
6 driver.get("https://practicetestautomation.com/practice-test-login/")
7 time.sleep(2)
8
9 username = driver.find_element(By.ID, "username")
10 username.send_keys("student")
11
12 password = driver.find_element(By.ID, "password")
13 password.send_keys("Password123")
14
15 login_button = driver.find_element(By.ID, "submit")
16 login_button.click()
17 time.sleep(2)
18
19 welcome_text = driver.find_element(By.TAG_NAME, "h1").text
20 print("page heading:", welcome_text)
21
22 input("Press Enter to close browser...")
23 driver.quit()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

Test Passed: Search results loaded successfully.
Press Enter to close the browser...[12424:1916:0901/161454.911:ERROR:net\socket\ssl_client_socket_impl.cc:910] handshake failed; returned -1, SSL error code 1, net_error -202
[12424:1916:0901/161456.719:ERROR:net\socket\ssl_client_socket_impl.cc:910] handshake failed; returned -1, SSL error code 1, net_error -202
[12424:1916:0901/161456.913:ERROR:net\socket\ssl_client_socket_impl.cc:910] handshake failed; returned -1, SSL error code 1, net_error -202
[12424:1916:0901/161657.124:ERROR:net\socket\ssl_client_socket_impl.cc:910] handshake failed; returned -1, SSL error code 1, net_error -202
[12424:1916:0901/161858.476:ERROR:net\socket\ssl_client_socket_impl.cc:910] handshake failed; returned -1, SSL error code 1, net_error -202



```
test_selenium.py import_webdriver.py
C:\> Users > studentlab1 > 3D Objects > import_webdriver.py > ...
1  from selenium import webdriver
2  from selenium.webdriver.common.by import By
3  import time
4
5  driver = webdriver.Edge()
6  driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login")
7  time.sleep(20)
8
9  username = driver.find_element(By.NAME, "username")
10 username.send_keys("Admin")
11
12 password = driver.find_element(By.NAME, "password")
13 password.send_keys("admin123")
14
15 # ✅ Fixed 'button' spelling
16 login_button = driver.find_element(By.XPATH, "//button[@type='submit']")
17 login_button.click()
18 time.sleep(20)
19
20 welcome_text = driver.find_element(By.CSS_SELECTOR, ".oxd-userdropdown-name").text
21 print("Logged in user:", welcome_text)
22
23 input("Press Enter to close browser...")
24 driver.quit()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

[6040:14084:0901/162019.646:ERROR:components\edge_auth\edge_auth_errors.cc:534] EDGE_IDENTITY: Get Default OS Account failed: Error: Primary Error: kImplicitSignInFailure, Secondary Error: kAccountProviderFetchError, Platform error: 0, Error string:

[6040:14084:0901/162020.668:ERROR:chrome\browser\task_manager\providers\fallback_task_provider.cc:126] Every renderer should have at least one task provided by a primary task provider. If a "Rendererr" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of cbug.com/739782.

page heading: Logged In Successfully
Press Enter to close browser...

The screenshot shows the Microsoft Edge browser window with the OrangeHRM application open. The URL in the address bar is <https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index>. The page title is "OrangeHRM". The dashboard features a navigation sidebar on the left with links like Admin, PIM, Leave, Time, Recruitment, My Info, Performance, and Dashboard (which is currently selected). The main content area includes a "Time at Work" section showing a "Punched Out" status for Sep 22nd at 05:02 PM (GMT 5.5) with a duration of 397h 45m Today. It also displays a weekly calendar for Sep 01 - Sep 07 and a "My Actions" section with items like "Pending Self Review" and "Candidate to Interview". A "Quick Launch" button is visible at the bottom left. The status bar at the bottom right shows "In 25 Col 1 Spaces:4 UTE-8 CRLE {} Python 3.12.6 64-bit".

```
Microsoft Windows [Version 10.0.19045.6093]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student.lab1>py --version
Python 3.12.6

C:\Users\student.lab1> -m pip install selenium
'-m' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\student.lab1>py -m pip install selenium
Defaulting to user installation because normal site-packages is not writeable
Collecting selenium
  Downloading selenium-4.35.0-py3-none-any.whl.metadata (7.4 kB)
Collecting urllib3<3.0,>=2.5.0 (from urllib3[socks]<3.0,>=2.5.0->selenium)
  Downloading urllib3-2.5.0-py3-none-any.whl.metadata (6.5 kB)
Collecting trio<=0.30.0 (from selenium)
  Downloading trio-0.30.0-py3-none-any.whl.metadata (8.5 kB)
Collecting trio-websocket~=0.12.2 (from selenium)
  Downloading trio_websocket-0.12.2-py3-none-any.whl.metadata (5.1 kB)
Collecting certifi>=2025.6.15 (from selenium)
  Downloading certifi-2025.8.3-py3-none-any.whl.metadata (2.4 kB)
Collecting typing_extensions~=4.14.0 (from selenium)
  Downloading typing_extensions-4.14.1-py3-none-any.whl.metadata (3.0 kB)
Collecting websocket-client~=1.8.0 (from selenium)
  Downloading websocket_client-1.8.0-py3-none-any.whl.metadata (8.0 kB)
Collecting attrs>=23.2.0 (from trio~=0.30.0->selenium)
  Downloading attrs-25.3.0-py3-none-any.whl.metadata (10 kB)
Collecting sortedcontainers (from trio~=0.30.0->selenium)
  Downloading sortedcontainers-2.4.0-py2.py3-none-any.whl.metadata (10 kB)
Collecting idna (from trio~=0.30.0->selenium)
  Downloading idna-3.10-py3-none-any.whl.metadata (10 kB)
Collecting outcome (from trio~=0.30.0->selenium)
  Downloading outcome-1.3.0.post0-py2.py3-none-any.whl.metadata (2.6 kB)
Collecting sniffio>=1.3.0 (from trio~=0.30.0->selenium)
  Downloading sniffio-1.3.1-py3-none-any.whl.metadata (3.9 kB)
Collecting cffi>=1.14 (from trio~=0.30.0->selenium)
  Downloading cffi-1.17.1-cp312-cp312-win_amd64.whl.metadata (1.6 kB)
Collecting wsproto>=0.14 (from trio-websocket~=0.12.2->selenium)
  Downloading wsproto-1.2.0-py3-none-any.whl.metadata (5.6 kB)
Collecting pysocks!=1.5.7,<2.0,>=1.5.6 (from urllib3[socks]<3.0,>=2.5.0->selenium)
  Downloading PySocks-1.7.1-py3-none-any.whl.metadata (13 kB)
Collecting pycparser (from cffi>=1.14->trio~=0.30.0->selenium)
  Downloading pycparser-2.22-py3-none-any.whl.metadata (943 bytes)
```

```
----- 9.6/9.6 MB 32.4 kB/s eta 0:00:00
Downloading wsproto-1.2.0-py3-none-any.whl.metadata (5.6 kB)
Collecting pysocks!=1.5.7,<2.0,>=1.5.6 (from urllib3[socks]<3.0,>=2.5.0->selenium)
  Downloading PySocks-1.7.1-py3-none-any.whl.metadata (13 kB)
Collecting pycparser (from cffi>=1.14->trio~=0.30.0->selenium)
  Downloading pycparser-2.22-py3-none-any.whl.metadata (943 bytes)
Collecting h11<1,>0.9.0 (from wsproto>=0.14->trio-websocket~=0.12.2->selenium)
  Downloading h11-0.16.0-py3-none-any.whl.metadata (8.3 kB)
Downloading selenium-4.35.0-py3-none-any.whl (9.6 MB)
----- 9.6/9.6 MB 32.4 kB/s eta 0:00:00
Downloading certifi-2025.8.3-py3-none-any.whl (161 kB)
Downloading trio-0.30.0-py3-none-any.whl (499 kB)
Downloading trio_websocket-0.12.2-py3-none-any.whl (21 kB)
Downloading typing_extensions-4.14.1-py3-none-any.whl (43 kB)
Downloading urllib3-2.5.0-py3-none-any.whl (129 kB)
Downloading websocket_client-1.8.0-py3-none-any.whl (58 kB)
Downloading attrs-25.3.0-py3-none-any.whl (63 kB)
Downloading cffi-1.17.1-cp312-cp312-win_amd64.whl (181 kB)
Downloading outcome-1.3.0.post0-py2.py3-none-any.whl (10 kB)
Downloading PySocks-1.7.1-py3-none-any.whl (16 kB)
Downloading sniffio-1.3.1-py3-none-any.whl (10 kB)
Downloading wsproto-1.2.0-py3-none-any.whl (24 kB)
Downloading idna-3.10-py3-none-any.whl (70 kB)
Downloading sortedcontainers-2.4.0-py2.py3-none-any.whl (29 kB)
Downloading h11-0.16.0-py3-none-any.whl (37 kB)
Downloading pycparser-2.22-py3-none-any.whl (117 kB)
Installing collected packages: sortedcontainers, websocket-client, urllib3, typing_extensions, sniffio, pysocks, pycparser, idna, h11, certifi, attrs, wsproto, outcome, cffi, trio, trio-websocket, selenium
  WARNING: The script wsdump.exe is installed in 'C:\Users\student.lab1\AppData\Roaming\Python\Python312\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed attrs-25.3.0 certifi-2025.8.3 cffi-1.17.1 h11-0.16.0 idna-3.10 outcome-1.3.0.post0 pycparser-2.22 pysocks-1.7.1 selenium-4.35.0 sniffio-1.3.1 sortedcontainers-2.4.0 trio-0.30.0 trio-websocket-0.12.2 typing_extensions-4.14.1 urllib3-2.5.0 websocket-client-1.8.0 wsproto-1.2.0

[notice] A new release of pip is available: 24.2 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\student.lab1>py
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import selenium
>>> print(selenium.__version__)
4.35.0
>>>
```