

Problem Statement: Download and install selenium and setup the environment using the pycharm.

Aim: To Download and install selenium and setup the environment using the pycharm.

Description:

Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#, among others.

Selenium WebDriver is a web framework that permits you to execute cross-browser tests. This tool is used for automating web-based application testing to verify that it performs expectedly. Selenium WebDriver allows you to choose a programming language to create test scripts.

Procedure:

- 1.Install Python in your system.
- 2.Install selenium.
- 3.Install web drivers.
- 4.Install pycharm.

Install selenium:

- 1.Open terminal (cmd)
- 2.type the following command
- 3.pip install selenium

Install web drivers:

Selenium requires a driver to interface with the chosen browser.

supported browsers will have their own drivers available. Links to some of the more popular browser drivers follow.

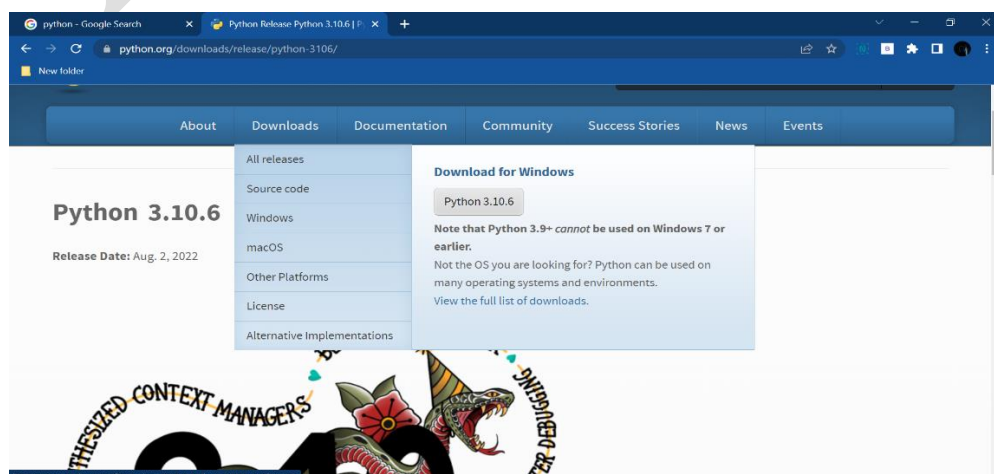
Chrome:	https://sites.google.com/chromium.org/driver/
Edge:	https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/
Firefox:	https://github.com/mozilla/geckodriver/releases
Safari:	https://webkit.org/blog/6900/webdriver-support-in-safari-10/

Install pycharm:

- 1.Download pycharm community application from jetbrains.com/pycharm/download/windows
- 2.Install the pycharm.exe and complete the installation process.
- 3.open the pycharm application setup the project.
- 4.In pycharm open terminal and run the command: pip install selenium
- 5.Now create project and perform the selenium tests.

Output:

Python install



Install Selenium

```

Microsoft Windows [Version 10.0.22610.1]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sandu>pip install selenium
Requirement already satisfied: selenium in c:\programdata\anaconda3\lib\site-packages (4.3.0)
Requirement already satisfied: trio~=0.17 in c:\programdata\anaconda3\lib\site-packages (from selenium) (0.21.0)
Requirement already satisfied: urllib3[secure,socks]~=1.26 in c:\programdata\anaconda3\lib\site-packages (from selenium) (1.26.13)
Requirement already satisfied: trio-websocket~=0.9 in c:\programdata\anaconda3\lib\site-packages (from selenium) (0.9.1)
Requirement already satisfied: sortedcontainers in c:\programdata\anaconda3\lib\site-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: attrs>=19.2.0 in c:\programdata\anaconda3\lib\site-packages (from trio~=0.17->selenium) (23.1.0)
Requirement already satisfied: async-generator>=1.9 in c:\programdata\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.10.3)
Requirement already satisfied: idna in c:\programdata\anaconda3\lib\site-packages (from trio~=0.17->selenium) (3.2)
Requirement already satisfied: outcome in c:\programdata\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.2.0)
Requirement already satisfied: sniffio in c:\programdata\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.2.0)
Requirement already satisfied: cffi>=1.14 in c:\programdata\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.15.1)
Requirement already satisfied: pycparser in c:\programdata\anaconda3\lib\site-packages (from cffi>=1.14->trio~=0.17->selenium) (2.21)
Requirement already satisfied: wsproto>=0.14 in c:\programdata\anaconda3\lib\site-packages (from trio-websocket~=0.9->selenium) (1.0.0)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\programdata\anaconda3\lib\site-packages (from urllib3[secure,socks]~=1.26->selenium) (1.7.1)
Requirement already satisfied: pyOpenSSL>=0.14 in c:\programdata\anaconda3\lib\site-packages (from urllib3[secure,socks]~=1.26->selenium) (23.2.0)
Requirement already satisfied: cryptography>=1.3.4 in c:\programdata\anaconda3\lib\site-packages (from urllib3[secure,socks]~=1.26->selenium) (41.0.7)
Requirement already satisfied: certifi in c:\programdata\anaconda3\lib\site-packages (from urllib3[secure,socks]~=1.26->selenium) (2022.12.7)
Requirement already satisfied: six>=1.5.2 in c:\users\sandu\appdata\roaming\python\python39\site-packages (from pyOpenSSL>=0.14->selenium) (1.16.0)
Requirement already satisfied: h11<1,>=0.9.0 in c:\programdata\anaconda3\lib\site-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)

C:\Users\sandu>

```

Install Webdrivers



Index of /105.0.5195.19/

Name	Last modified	Size	ETag
Parent Directory		-	
chromedriver_linux64.zip	2022-08-08 09:16:09	6.15MB	b3944a67d3dd6f9369cec74c1cc69c41
chromedriver_mac64.zip	2022-08-08 09:16:11	8.07MB	43d4460f63ef5bc8a5e711aff5ca00e3
chromedriver_mac64_m1.zip	2022-08-08 09:16:14	7.46MB	70f5243170812fe88f53e4502e03fb10
chromedriver_win32.zip	2022-08-08 09:16:16	6.24MB	eff85de8df53272a1e18ab65fdbcbebf5
notes.txt	2022-08-08 09:16:22	0.00MB	5c39d0751819d261ae8b2072976494dd

Install Pycharm

pycharm - Google Search

jetbrains.com/pycharm/download/#section=windows

PyCharm

What's New Features Learn Pricing [Download](#)

Version: 2022.2
Build: 222.3345.131
28 July 2022

[System requirements](#)
[Installation instructions](#)
[Other versions](#)

Download PyCharm

[Windows](#) [macOS](#) [Linux](#)

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

[Download](#)

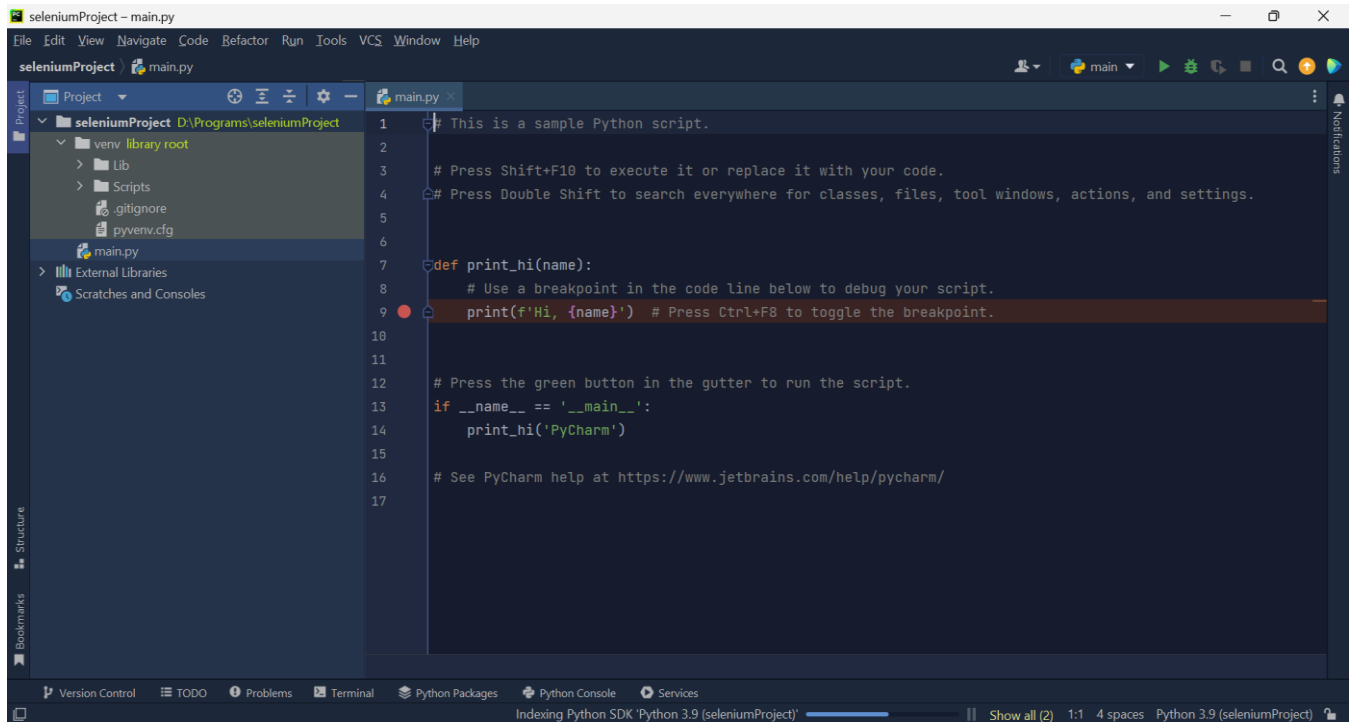
Free 30-day trial available

Community

For pure Python development

[Download](#)

Free, built on open-source



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The left sidebar shows the Project view with the following structure:

- seleniumProject
 - venv
 - library root
 - Lib
 - Scripts
 - .gitignore
 - pyvenv.cfg
 - main.py
- External Libraries
- Scratches and Consoles

The main editor window displays the file 'main.py' with the following Python code:

```

1  This is a sample Python script.
2
3  # Press Shift+F10 to execute it or replace it with your code.
4  # Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
5
6
7  def print_hi(name):
8      # Use a breakpoint in the code line below to debug your script.
9      print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.
10
11
12 # Press the green button in the gutter to run the script.
13 if __name__ == '__main__':
14     print_hi('PyCharm')
15
16 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
17

```

The bottom status bar indicates 'Indexing Python SDK 'Python 3.9 (seleniumProject)' and shows a progress bar. The bottom right corner displays 'Show all (2) 1:1 4 spaces Python 3.9 (seleniumProject)'.

Results:

Download and install selenium and setup the environment using the pycharm was completed successfully.

Problem Statement: Automate a web browser to capture title of the page, URL of the page using selenium.

Aim: To Automate a web browser

- 1.to capture title of the page
- 2.to get URL of the page using selenium.

Procedure:

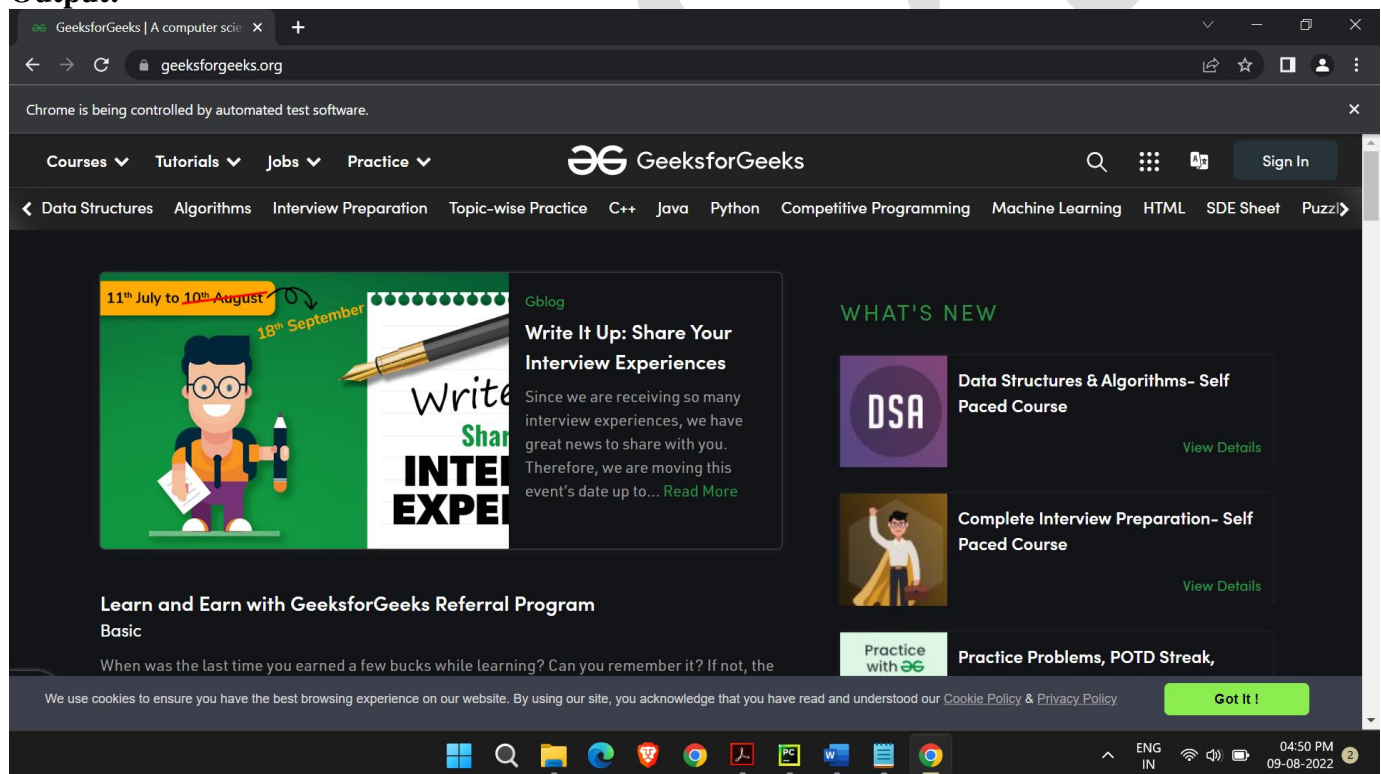
- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.common import keys
import time
```

```
driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
driver.get("https://www.geeksforgeeks.org/")
driver.maximize_window()
print(driver.title)
print(driver.current_url)
time.sleep(3)
driver.close()
```

Output:



GeeksforGeeks | A computer science portal for geeks

<https://www.geeksforgeeks.org/>

Results:

Automation of a web browser to capture title of the page, URL of the page using selenium was completed successfully.

Problem Statement: Automate a web browser to test for Title matching with selenium tool.

Aim: To Automate a web browser to test for Title matching with selenium tool.

Procedure:

- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

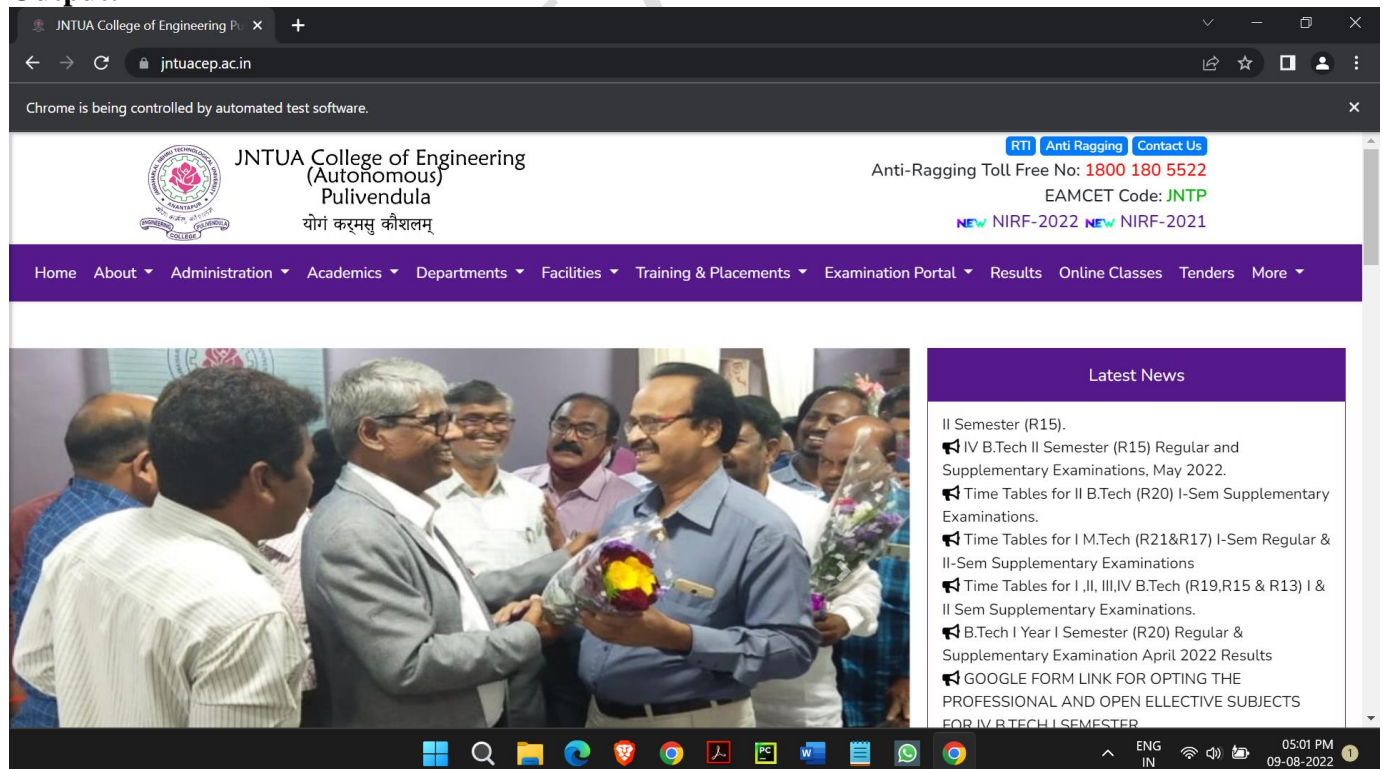
```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
import time
```

```
serv_obj = Service("C:\\Drivers\\chromedriver_win32\\chromedriver.exe")
driver = webdriver.Chrome(service=serv_obj)
driver.get("https://jntuacep.ac.in/")
driver.maximize_window()
actualTitle = driver.title
expectedTitle="JNTUA College of Engineering Pulivendula"
time.sleep(3)
```

```
if (actualTitle == expectedTitle):
    print("title matched")
else:
    print("title not matched")
```

```
driver.quit()
```

Output:



title matched

Results:

Automation of a web browser to test for Title matching with selenium tool was completed successfully.

Problem Statement: Automate a web browser for filling the web forms with selenium tool.

Aim: To Automate a web browser for filling the web forms with selenium tool.

Procedure:

- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")

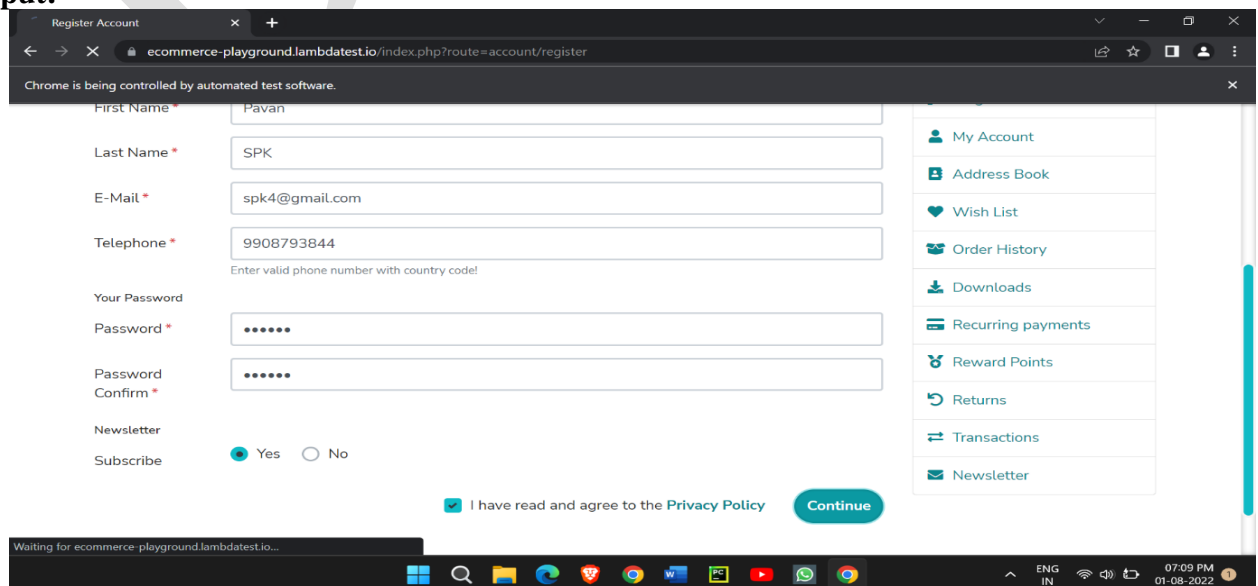
driver.get("https://ecommerce-playground.lambdatest.io/index.php?route=account/register")

inputboxes=driver.find_element(By.CLASS_NAME,'form-control')
driver.find_element(By.ID,'input-firstname').send_keys("Pavan")
driver.find_element(By.ID,'input-lastname').send_keys("SPK")
driver.find_element(By.ID,'input-email').send_keys("spk4@gmail.com")
driver.find_element(By.ID,'input-telephone').send_keys("9908793844")
driver.find_element(By.ID,'input-password').send_keys("123321")
driver.find_element(By.ID,'input-confirm').send_keys("123321")
newsletter = driver.find_element(By.XPATH, value="//label[@for='input-newsletter-yes']")
newsletter.click()
terms = driver.find_element(By.XPATH, value="//label[@for='input-agree']")
terms.click()

continue_button = driver.find_element(By.XPATH, value="//input[@value='Continue']")
continue_button.click()
continue_button = driver.find_element(By.XPATH, value="//input[@value='Continue']")
continue_button.click()

assert browser.title == "Your Account Has Been Created!"
time.sleep(10)
driver.close()
```

Output:



Your Account Has Been Created!

Congratulations! Your new account has been successfully created!

You can now take advantage of member privileges to enhance your online shopping experience with us.

If you have ANY questions about the operation of this online shop, please e-mail the store owner.

A confirmation has been sent to the provided e-mail address. If you have not received it within the hour, please contact us.

[Continue](#)

- My Account
- Edit Account
- Password
- Address Book
- Wish List
- Notification
- Order History

Waiting for eCommerce-playground.lambdatest.io...

07:08 PM
01-08-2022

Your Account Has Been Created!

Results:

Automation of a web browser for filling the web forms with selenium tool was completed successfully.

Problem Statement: Automate a web browser to upload a file in web forms using selenium tool.

Aim: To Automate a web browser to upload a file in web forms using selenium tool.

Procedure:

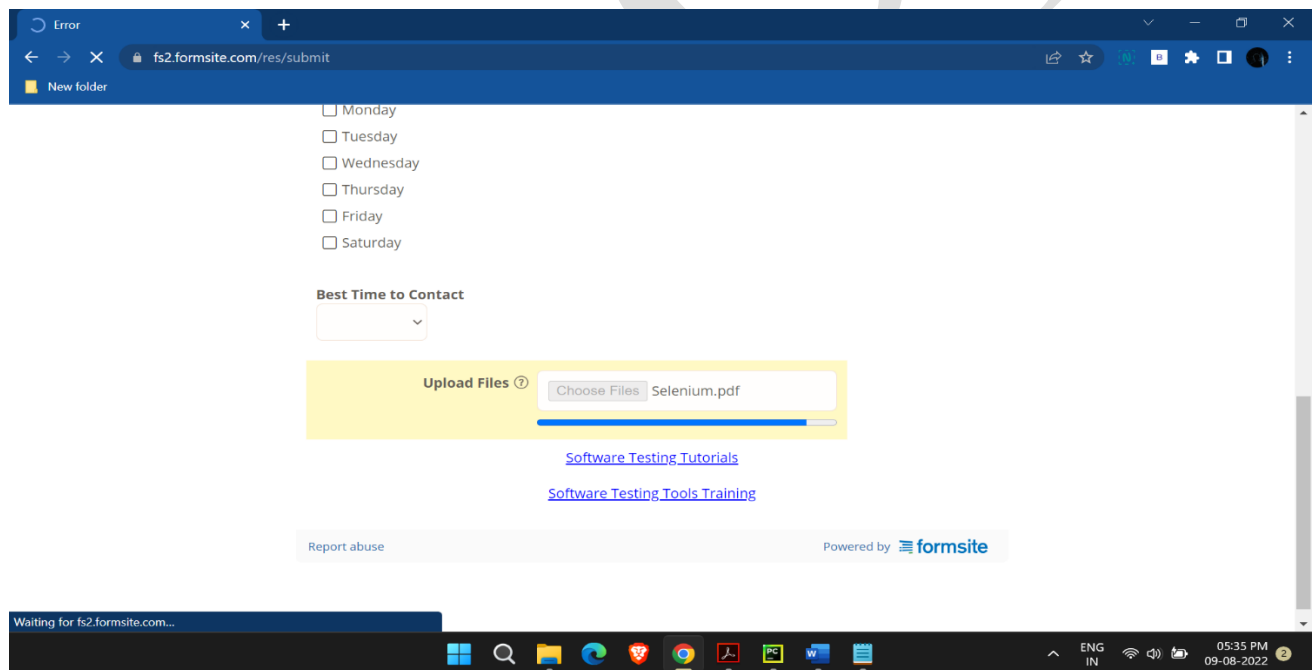
- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

driver=webdriver.Chrome(executable_path="C:\\Drivers\\chromedriver_win32\\chromedriver.exe")
driver.get("https://fs2.formsite.com/meherpavan/form2/index.html?1537702596407")
driver.maximize_window()
driver.find_element(By.ID,"RESULT_FileUpload-10").send_keys("C://Users/sandu/Pictures/Selenium.pdf")
time.sleep(5)
driver.close()
```

Output:



Results:

Automation of web browser to upload a file in web forms using selenium tool was completed successfully.

Problem Statement: Automate a web browser for working on a drop-down list in web forms using selenium tool.

Aim: To Automate a web browser for working on a drop-down list in web forms using selenium tool.

Procedure:

- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import Select
import time

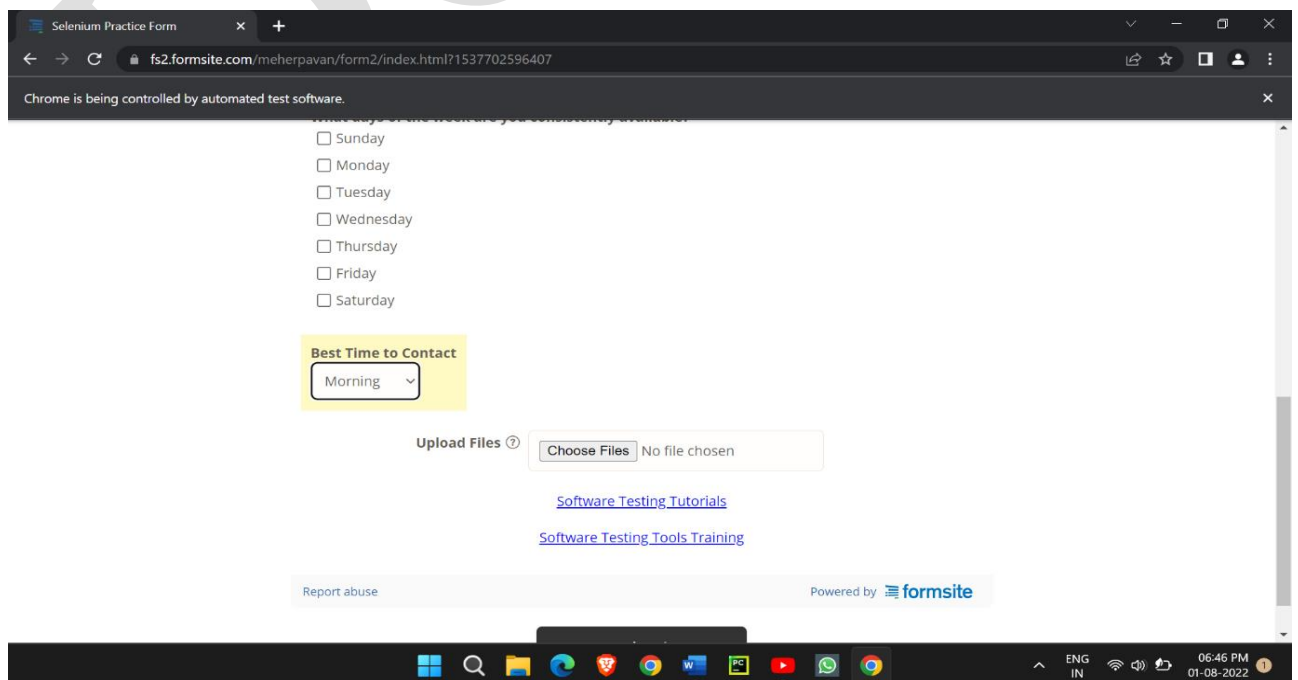
driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
driver.get("https://fs2.formsite.com/meherpavan/form2/index.html?1537702596407")
driver.maximize_window()

element=driver.find_element(By.NAME,"RESULT_RadioButton-9")
drp=Select(element)
print(len(drp.options))

all_options=drp.options
for option in all_options:
    print(option.text)

drp.select_by_value("Radio-0")
time.sleep(5)
print("Morning selected")
driver.close()
```

Output:



```

11 drp=Select(element)
12 print(len(drp.options))
13
14 all_options=drp.options
15 for option in all_options:
16     print(option.text)
17
18 drp.select_by_value("Radio-0")
    
```

Run: dropdown (1)

```

D:\Programs\Selenium\venv\Scripts\python.exe D:/Programs/selenium/venv/dropdown.py
D:\Programs\selenium\venv\dropdown.py:6: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
4
Morning
Afternoon
Evening
Morning selected

Process finished with exit code 0
    
```

Results:

Automation of a web browser for working on a drop-down list in web forms using selenium tool was completed successfully.

Problem Statement: Automate a web browser for working on a Radio and check boxes in formfilling using selenium tool.

Aim: To Automate a web browser for working on a Radio and check boxes in formfilling using selenium tool.

Procedure:

- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

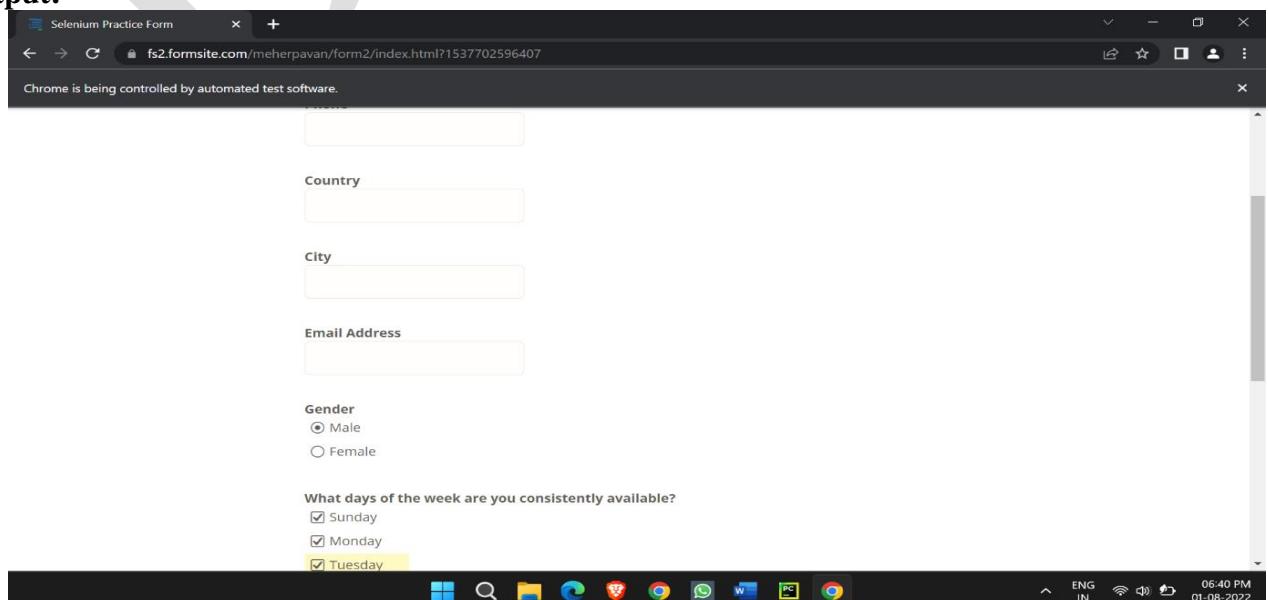
driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
driver.get("https://fs2.formsite.com/meherpavan/form2/index.html?1537702596407")
driver.maximize_window()

gender=driver.find_element(By.XPATH, value="//label[@for='RESULT_RadioButton-7_0']")
gender.click()

status=driver.find_element(By.XPATH, value="//label[@for='RESULT_RadioButton-7_0']").is_selected()
print(status)

driver.find_element(By.XPATH, value="//label[@for='RESULT_CheckBox-8_0']").click()
#sunday
status=driver.find_element(By.XPATH, value="//label[@for='RESULT_CheckBox-8_1']").click()
#monday
driver.find_element(By.XPATH, value="//label[@for='RESULT_CheckBox-8_2']").click()
#tuesday
print(status)
time.sleep(5)
driver.quit()
```

Output:



The screenshot shows a Selenium IDE script for automating a radio button. The script is titled 'radio_checkboxes.py' and is located in the 'D:\Programs\selenium\venv\radio_checkboxes.py' file. The script contains the following code:

```

8  """status=driver.find_element(By.ID,'RESULT_RadioButton-7_0').is_selected()
9  print(status)"""
10
11  gender=driver.find_element(By.XPATH,value="//label[@for='RESULT_RadioButton-7_0']")
12  gender.click()
13
14  status=driver.find_element(By.XPATH,value="//label[@for='RESULT_RadioButton-7_0']").is_selected()
15  print(status)
16
17  driver.find_element(By.XPATH,value="//label[@for='RESULT_RadioButton-7_0']").click()
18

```

The script is executed in the Selenium IDE interface, which shows the 'Run' button and the 'Run' status. The output of the script is displayed in the 'Run' console, showing a deprecation warning and the final status of the radio button.

Run: radio_checkboxes (1) -

```

D:\Programs\Selenium\venv\Scripts\python.exe D:/Programs/selenium/venv/radio_checkboxes.py
D:\Programs\selenium\venv\radio_checkboxes.py:5: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
False
None
Process finished with exit code 0

```

Results:

Automation of a web browser for working on a Radio and check boxes in formfilling using selenium tool.

Problem Statement: Automate a web browser for unit-test framework on a simple project using selenium tool.

Aim: To Automate a web browser for unit-test framework on a simple project using selenium tool.

Procedure:

- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
chrome_executable = Service(executable_path='C:\\Drivers\\chromedriver_win32\\chromedriver.exe',
log_path='NULL')
from selenium.webdriver.common.by import By
import unittest
import time

class SearchEnginesTest(unittest.TestCase):
    @classmethod
    def testGoogle(self):

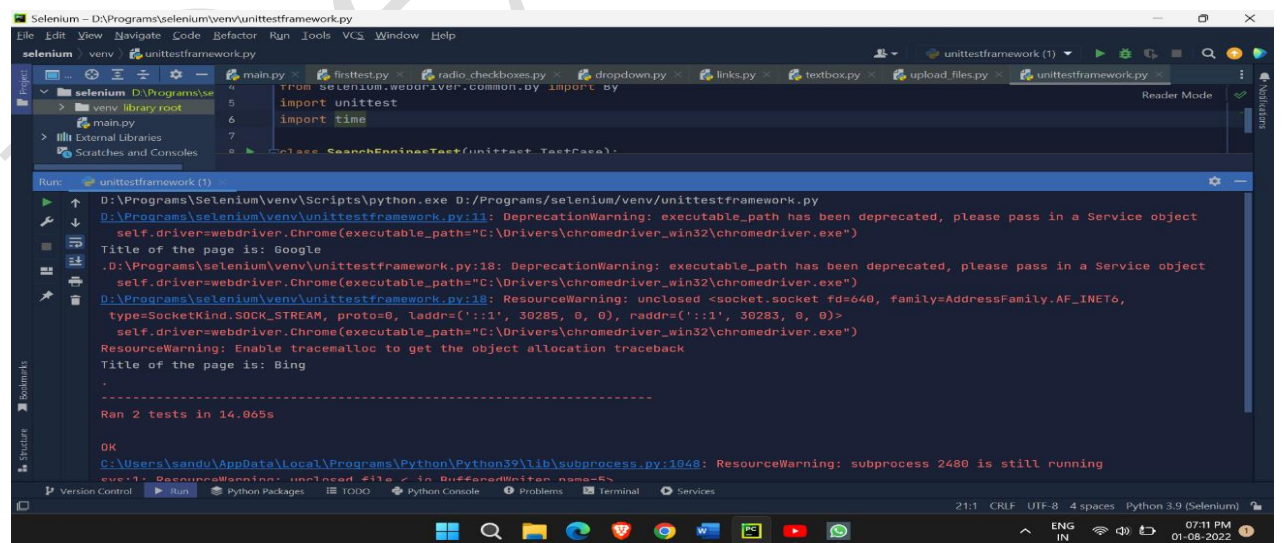
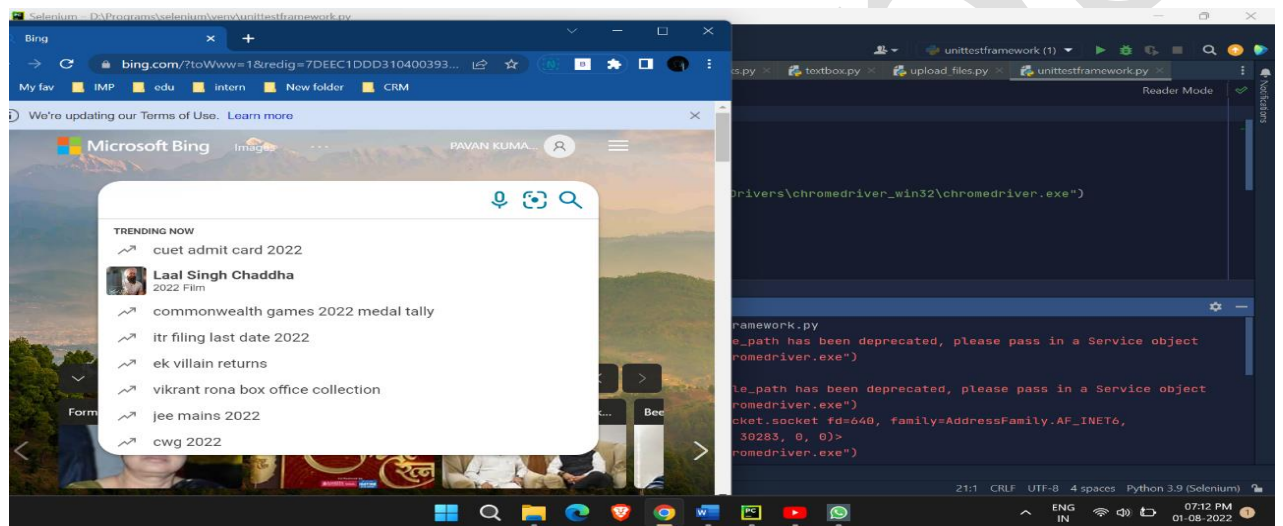
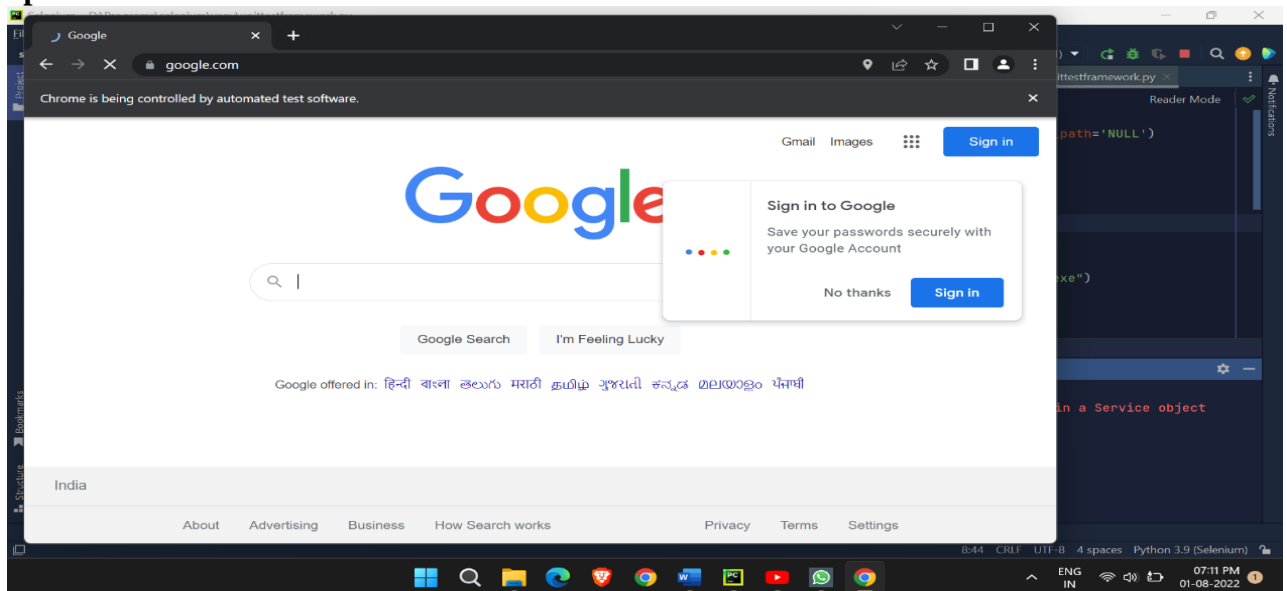
self.driver=webdriver.Chrome(executable_path="C:\\Drivers\\chromedriver_win32\\chromedriver.exe")
        self.driver.get("https://www.google.com/")
        print("Title of the page is:",self.driver.title)
        self.driver.close()

    @classmethod
    def test_Bing(self):

self.driver=webdriver.Chrome(executable_path="C:\\Drivers\\chromedriver_win32\\chromedriver.exe")
        self.driver.get("https://bing.com")
        print("Title of the page is:", self.driver.title)
        self.driver.close()

if __name__ == "__main__":
    unittest.main()
driver.quit()
```

Output:



Title of page is: Google

Title of page is: Bing

Closes the browser

Ran 2 tests in 14.065s

Status: OK

All Test classes finished

Results:

Automate a web browser for unit-test framework on a simple project using selenium tool.

DevOps