1. **Write a Python program using Selenium WebDriver to launch the Chrome browser, open Google's homepage, retrieve and display the page title, and then close the browser.**
   Step 1: create a folder
   mkdir   program1
   code:
   ```
   from selenium import webdriver
   from selenium.webdriver.chrome.service import Service
   from selenium.webdriver.common.by import By

   # Launch Chrome browser
   driver = webdriver.Chrome()

   # Open Google homepage
   driver.get("https://www.google.com")

   # Retrieve and display page title
   print("Page Title is:", driver.title)

   # Close the browser
   driver. quit()
   ```
   Run the code :
   python program1.py
   OR
   python.exe program1.py
   output: Page title is: Google

2. **Write a Python program using Selenium WebDriver to launch the Chrome browser, navigate to a webpage, locate a button using its element ID, click the button, wait for a few seconds to observe the result, and finally close the browser.**
   Create a folder
   mkdir program2
   code :
   ```
   from selenium import webdriver
   from selenium.webdriver.common.by import By
   from selenium.webdriver.chrome.service import Service
   ```

```
from webdriver_manager.chrome import ChromeDriverManager
import time
driver =
webdriver.Chrome(service=Service(ChromeDriverManager().install()))
driver.get("https://the-internet.herokuapp.com/add_remove_elements/")
driver.find_element(By.XPATH, "//button[text()='Add Element']").click()
time.sleep(3)
driver.quit()
```

Run the code

Python program2.py

3. **Write a Selenium script in Python to automate the login process for a web application. The script should open a browser, navigate to the login page, enter the username and password, click the login button, verify successful login, and then close the browser.**

   Code:

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

driver =
webdriver.Chrome(service=Service(ChromeDriverManager().install()))
driver.get("https://the-internet.herokuapp.com/login")

driver.find_element(By.ID, "username").send_keys("tomsmith")
driver.find_element(By.ID,
"password").send_keys("SuperSecretPassword!")
driver.find_element(By.CSS_SELECTOR, "button.radius").click()

print("Login Successful")

driver.quit()
```

output:

python program3.py

Login successful

4. **Write a Selenium script in Python to open any webpage and capture a screenshot of the page. The script should launch a browser, navigate to a given URL, take a screenshot, save it with a specified filename, and then close the browser.**
   Code:
   ```
   from selenium import webdriver

   driver = webdriver.Chrome()
   driver.get("https://example.com")
   driver.save_screenshot("screenshot.png")
   print("Screenshot saved successfully")
   driver.quit()
   ```
   output:python program4.py
   Screenshot saved successfully
5. **Write a Selenium script in Python to automate the submission of a web form. The script should open a browser, navigate to a given form page, enter values into multiple input fields (such as name, email, and phone number), select an option from a dropdown or radio button, submit the form, and then close the browser.**
   Code:
   ```
   from selenium import webdriver
   from selenium.webdriver.common.by import By
   from selenium.webdriver.support.ui import Select

   driver = webdriver.Chrome()
   driver.get("https://www.techlistic.com/p/selenium-practice-form.html")

   driver.find_element(By.NAME, "firstname").send_keys("John")
   driver.find_element(By.NAME, "lastname").send_keys("Doe")
   driver.find_element(By.ID, "sex-0").click()
   driver.find_element(By.ID, "exp-2").click()
   Select(driver.find_element(By.ID,
   "continents")).select_by_visible_text("Asia")

   print("Form submitted successfully")

   driver.quit()
   ```

run the code : python program5.py
output: Form submit successfully

6. **Write a Selenium script in Python to automate handling of a JavaScript alert. The script should open a browser, navigate to a webpage that contains an alert box, trigger the alert, accept or dismiss the alert as required, display the alert message in the console, and then close the browser.**

   Code :

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

d = webdriver.Chrome()
d.get("https://the-internet.herokuapp.com/javascript_alerts")

d.find_element(By.XPATH, "//button[text()='Click for JS Alert']").click()
print(d.switch_to.alert.text)
d.switch_to.alert.accept()

d.find_element(By.XPATH, "//button[text()='Click for JS
Confirm']").click()
print(d.switch_to.alert.text)
d.switch_to.alert.dismiss()

d.quit()
```

   Run the code: python program6.py
   Output: C:\selenium\program6>python program6.py
   Alert message: I am a JS Alert
   Confirmation message: I am a JS Confirm

7. **Write a Selenium script in Python to perform data-driven login automation for a web application. The script should read multiple sets of login credentials from an external file (such as Excel or CSV), attempt login for each set, verify whether the login is successful or**

not, capture a screenshot for each attempt, log the result (pass/fail), and then close the browser.

Code:

Csv file

```
username,password
tomsmith,SuperSecretPassword!
wronguser,wrongpass
```

program7.py

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

for r in csv.DictReader(open("login_data.csv")):
    d = webdriver.Chrome()
    d.get("https://the-internet.herokuapp.com/login")
    d.find_element(By.ID,"username").send_keys(r["username"])
    d.find_element(By.ID,"password").send_keys(r["password"])
    d.find_element(By.CSS_SELECTOR,"button.radius").click()
    msg = d.find_element(By.ID,"flash").text
    res = "PASS" if "secure area" in msg else "FAIL"
    print(r["username"],":",res)
    d.save_screenshot(r["username"]+".png")
    d.quit()
```

output:

```
C:\selenium\program7>python program7.py
tomsmith : PASS
wronguser : FAIL
```

8. **Problem Statement 2: End-to-End E-Commerce Workflow Automation Write a Selenium script in Python to automate an end-to-end e-commerce workflow. The script should open an e-commerce website, perform user login, search for a product, apply filters, add the product to the cart, verify cart details, proceed to checkout (without payment), capture screenshots at key steps, and then close the browser.**

   Code:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
d = webdriver.Chrome()
w = WebDriverWait(d,10)
d.get("https://www.saucedemo.com/")
d.save_screenshot("1_login.png")
w.until(EC.presence_of_element_located((By.ID,"user-
name"))).send_keys("standard_user")
d.find_element(By.ID,"password").send_keys("secret_sauce")
d.find_element(By.ID,"login-button").click()
w.until(EC.presence_of_element_located((By.CLASS_NAME,"inventory
_list")))
d.find_element(By.XPATH,"(//button[text()='Add to cart'])[1]").click()
d.find_element(By.CLASS_NAME,"shopping_cart_link").click()
w.until(EC.presence_of_element_located((By.ID,"checkout"))).click()
w.until(EC.presence_of_element_located((By.ID,"first-
name"))).send_keys("John")
d.find_element(By.ID,"last-name").send_keys("Doe")
d.find_element(By.ID,"postal-code").send_keys("12345")
d.find_element(By.ID,"continue").click()
d.save_screenshot("2_final.png")
print("E-Commerce Workflow Completed Successfully")
d.quit()
```

output:

```
python program8.py
C:\selenium\program8>python program8.py
E-Commerce Workflow Completed Successfully
```