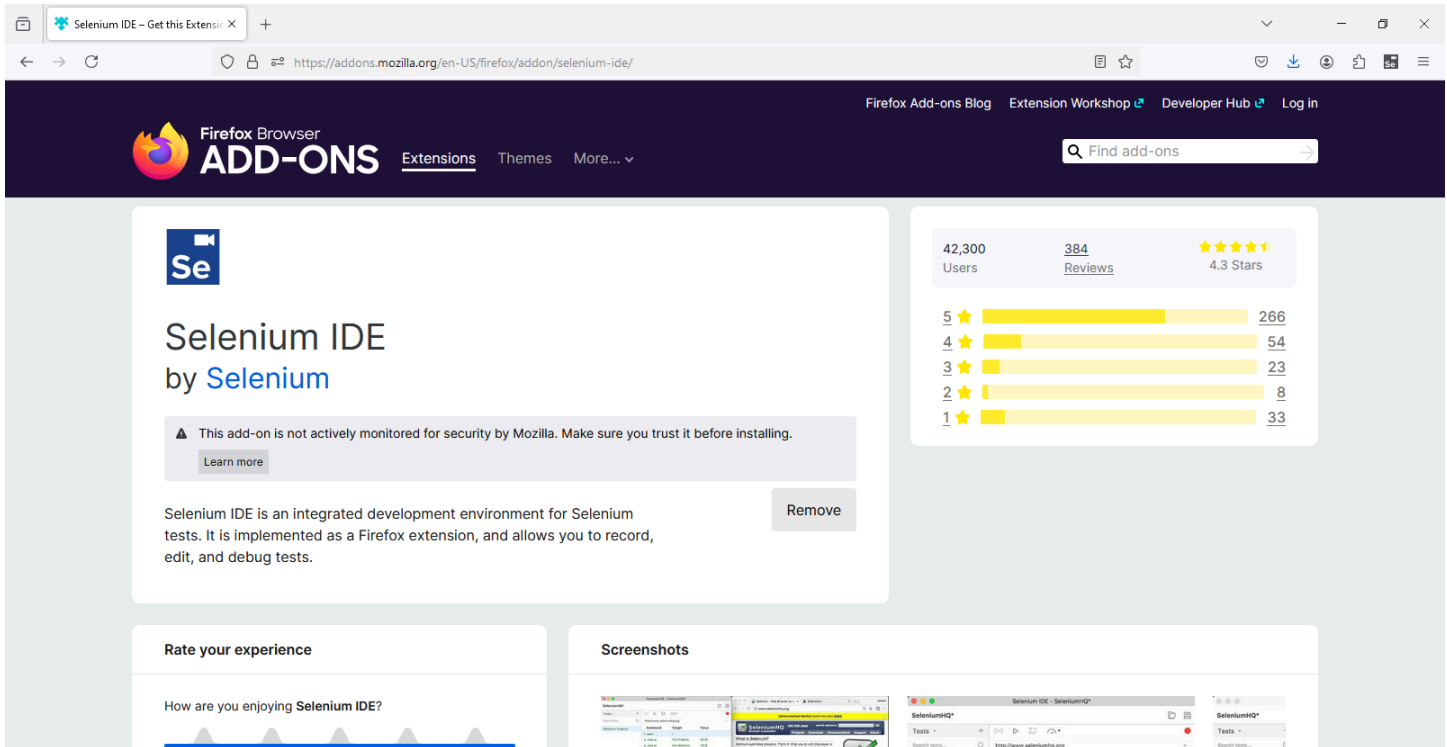


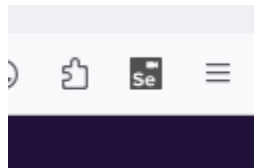
PRACTICAL NO 01

Aim : Install Selenium IDE and create a test suite containing a minimum of 4 test cases for different web page formats (e.g., HTML, XML, JSON, etc.).

1. Installing Selenium Add-ons



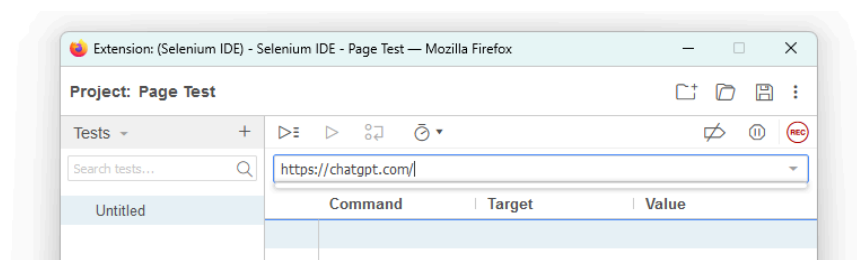
2. Click on the 'Se' icon which is on the Firefox toolbar



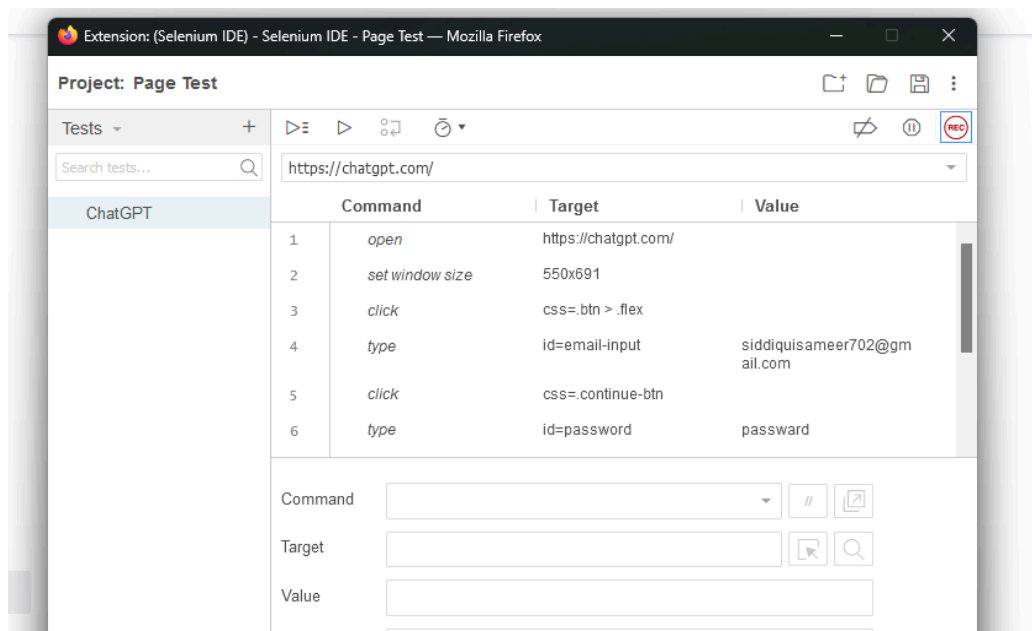
Running 4 Test on Different we

1. ChatGPT ->

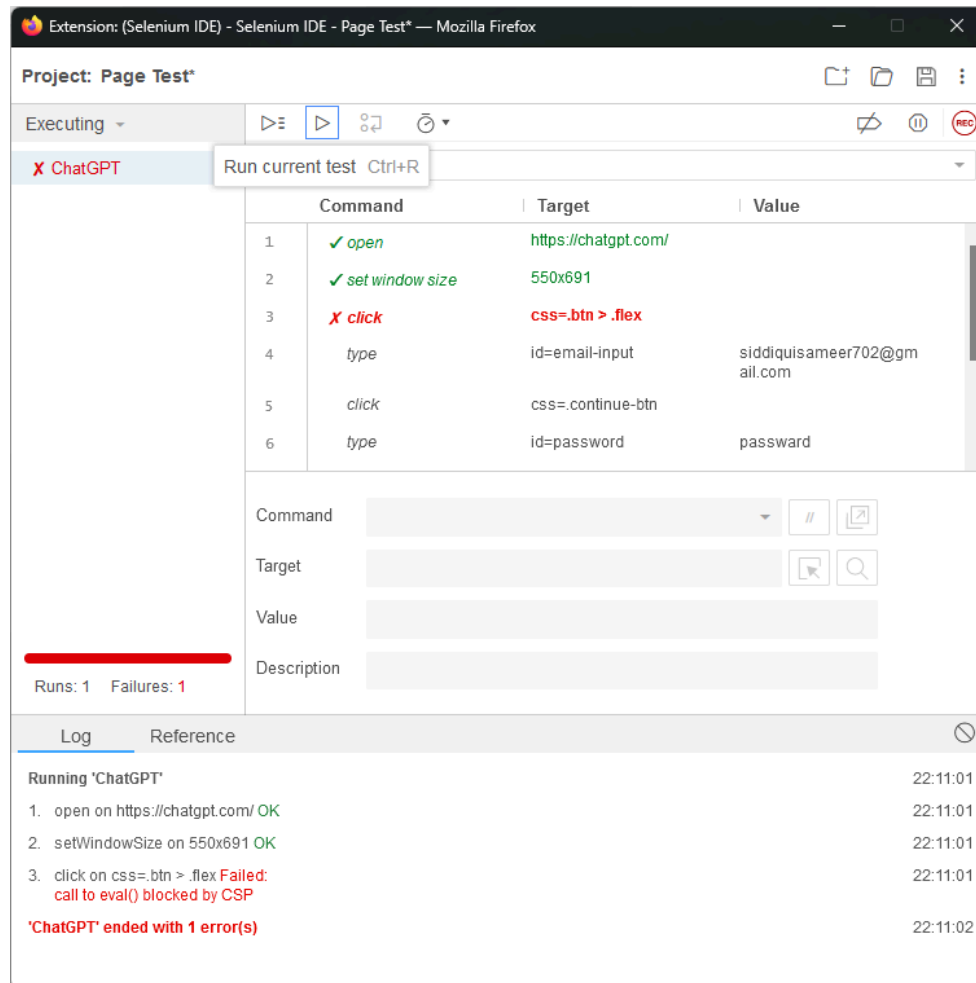
Step 1 : Open Mozilla Firefox and search for <https://chatgpt.com/> Copy and paste the same URL into Selenium IDE, start recording subsequently.



Step 2 : Abreast, you will see the commands in Selenium IDE which you have performed on the website.

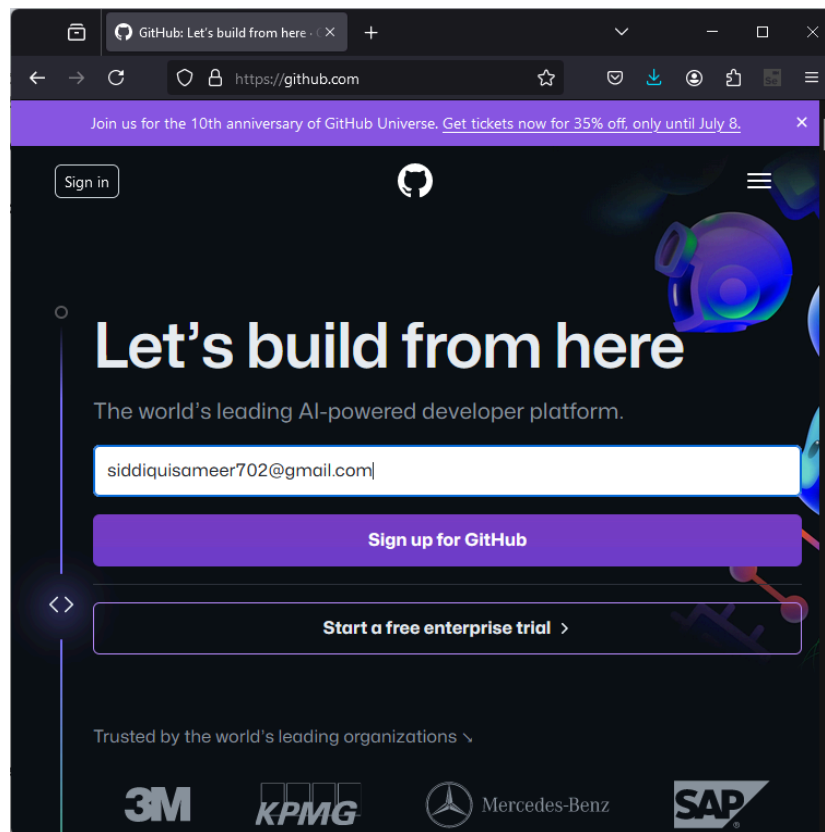


Step 3 : Go to the Selenium IDE and click onto Run current test then you will see the 'test case has completed successfully' in the Log section.

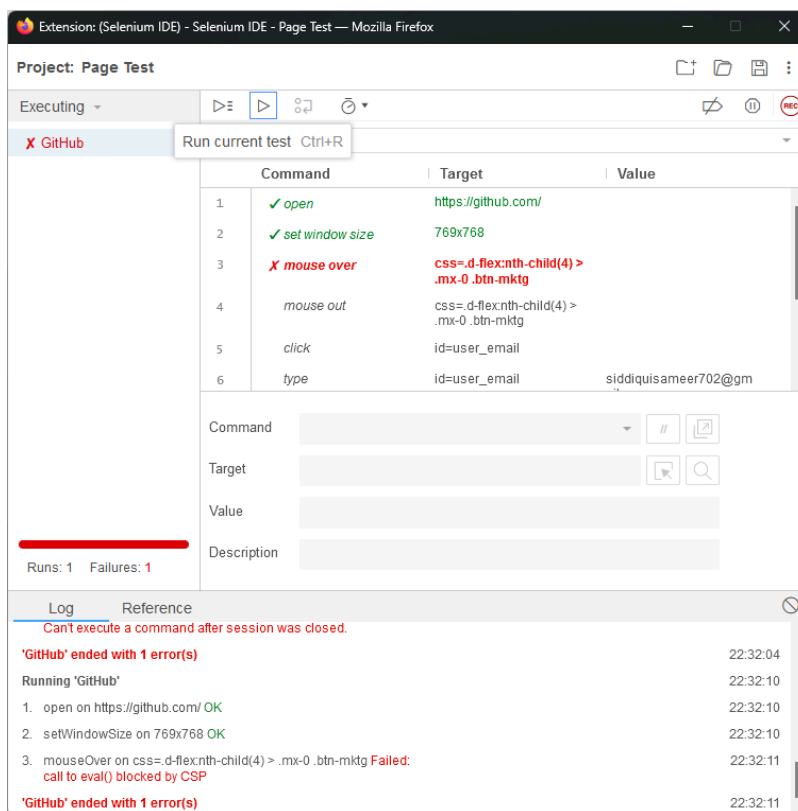


2. GitHub ->

Step 1 : Go to www.google.com and search for GitHub.

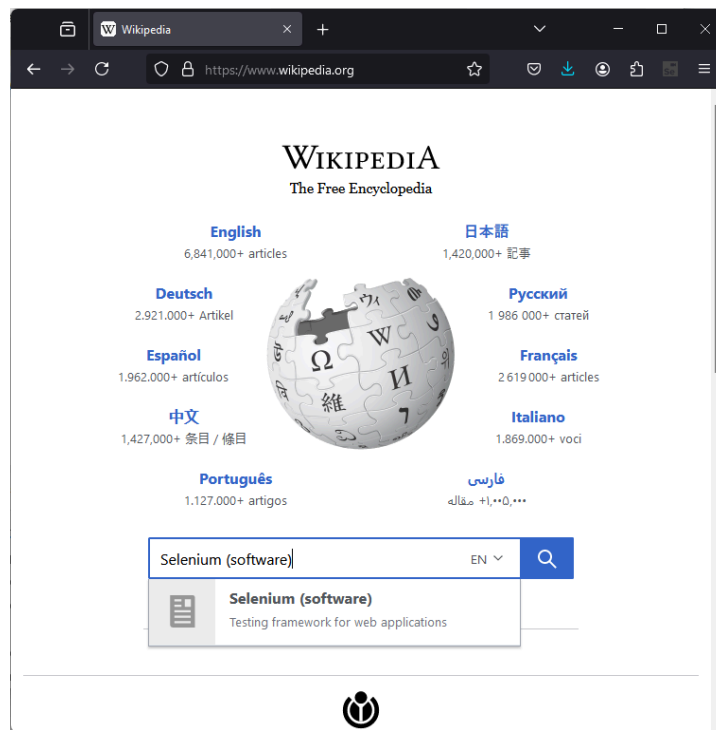


Step 2 : Logging in GitHub

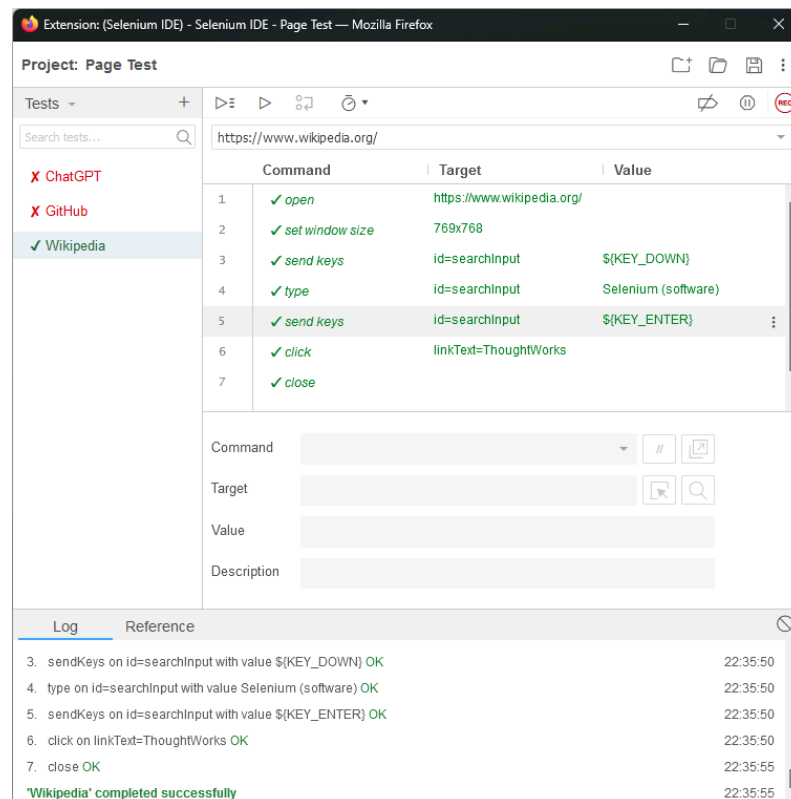


3. Wikipedia ->

Step 1 : Go to www.wikipedia.org and search for selenium software.

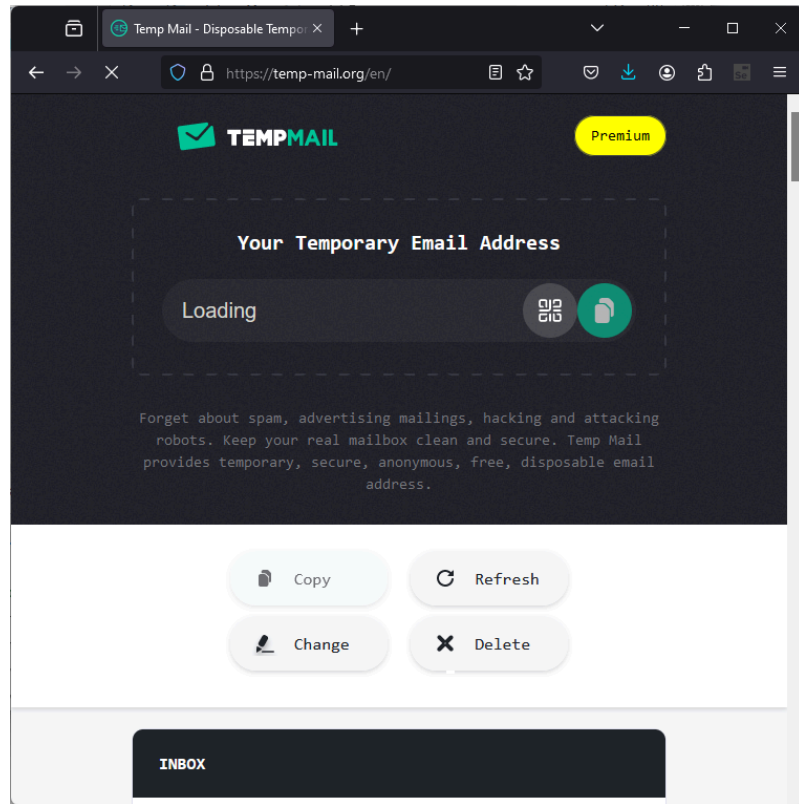


Step 2 : Save and Run the test.

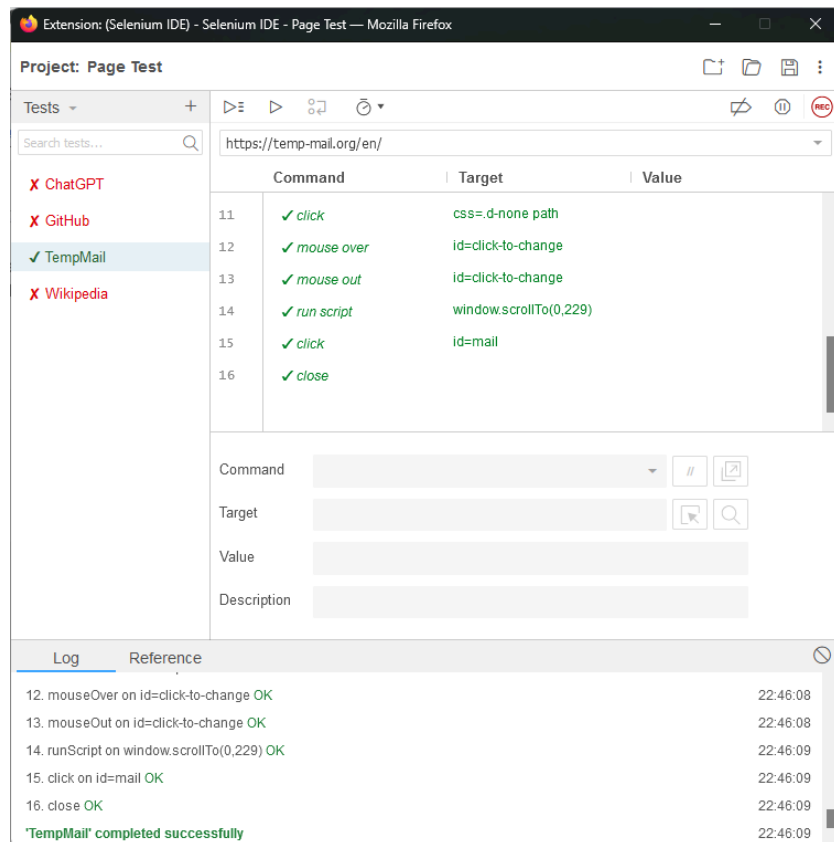


4. TempMail ->

Step 1 : Go to <https://temp-mail.org/en/> and copy the email address



Step 2 : Save and Run the test.

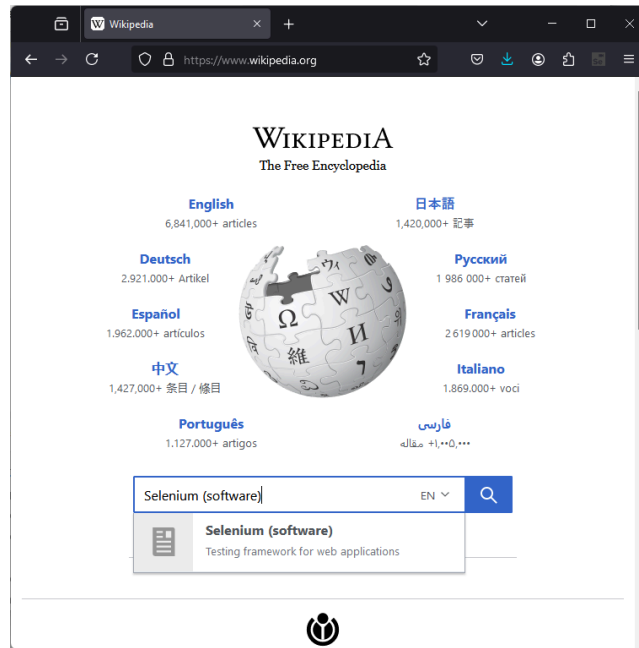


PRACTICAL NO 02

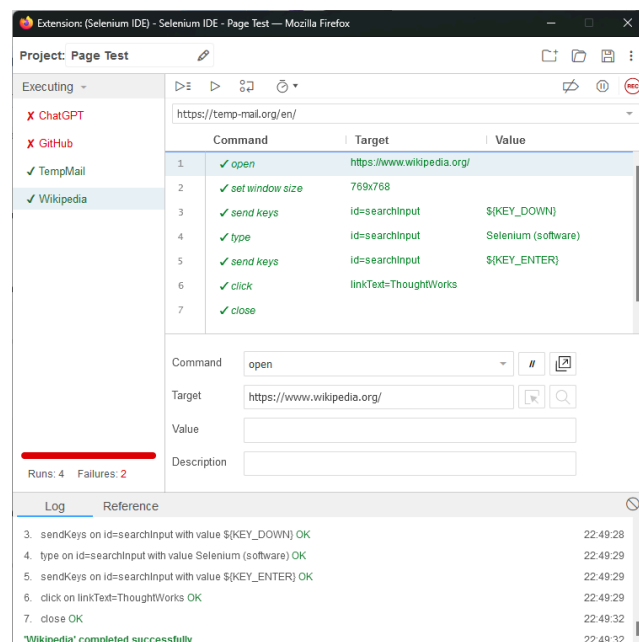
Aim : Conduct a test suite for two different websites using Selenium IDE. Perform various actions like clicking links, filling forms, and verifying content.

A. For www.wikipedia.org

1. Open the website and open selenium IDE. Copy the URL and paste it in the given space for further test. Then save & run the test to see log files.



2. After every successful run it gives a message “completed successfully” with green text highlighted.



3. Save it in the HTML format and open it on your browser

Extension: (Selenium IDE) - Selenium IDE - Page Test — Mozilla Firefox

Project: Page Test

Tests +

Search tests...

ChatGPT

GitHub

✓ TempMail

Wikipedia

https://temp-mail.org/en/

| | Command | Target | Value |
|---|-------------------|---------------------------|-------|
| 1 | ✓ open | https://temp-mail.org/en/ | |
| 2 | ✓ set window size | 769x768 | |
| 3 | ✓ mouse over | id=mail | |
| 4 | ✓ mouse out | id=mail | |
| 5 | ✓ mouse over | css=btn-rds:nth-child(4) | |
| 6 | ✓ mouse out | css=btn-rds:nth-child(4) | |
| 7 | ✓ mouse over | css=btn-rds:nth-child(4) | |
| 8 | ✓ mouse out | css=btn-rds:nth-child(4) | |

Command

Target

Value

Description

Log Reference

12. mouseOver on id=click-to-change OK 23:00:57

13. mouseOut on id=click-to-change OK 23:00:57

14. runScript on window.scrollTo(0,229) OK 23:00:57

15. click on id=mail OK 23:00:57

16. close OK 23:00:58

TempMail completed successfully 23:00:58

3. Save it in the HTML format and open it on your browser

SOFTWARE TESTING & QUALI x SOFTWARE TESTING & QUALI x Temp.html

C:/Users/Sameer/Downloads/Temp.html

```
{ "id": "60cb7e40-ec3d-43be-96da-11b76ab5c6ab", "version": "2.0", "name": "Page Test", "url": "https://temp-mail.org/en/", "tests": [ { "id": "158361cd-645d-4c9c-81d0-2856f818f3d7", "name": "TempMail", "commands": [ { "id": "847726ae-0219-449d-b254-e07d7075ecd8", "comment": "", "command": "open", "target": "https://temp-mail.org/en/", "targets": [ ], "value": "" }, { "id": "04a53b4a-9eda-4418-aa19-c075becc316", "comment": "", "command": "setWindowSize", "target": "769x768", "targets": [ ], "value": "" }, { "id": "ca174874-717e-44c3-ae04-7a599d8cf18a", "comment": "", "command": "mouseOver", "target": "id=mail", "targets": [ "id=mail", "id", "css=#mail", "css=finder" ], "xpath": "/input[@id='mail']", "xpath.attributes": [ "xpath=/body[@id='tm-body']/div/div/div/div[2]/div/form/div/div/input", "xpath=idRelative" ], "xpath=/input", "xpath.position": [ ], "value": "" }, { "id": "738378cf-d4d7-496b-9782-4161940244e5", "comment": "", "command": "mouseOut", "target": "id=mail", "targets": [ "id=mail", "id", "css=#mail", "css=finder" ], "xpath": "/input[@id='mail']", "xpath.attributes": [ "xpath=/body[@id='tm-body']/div/div/div/div[2]/div/form/div/div/input", "xpath=idRelative" ], "xpath=/input", "xpath.position": [ ], "value": "" }, { "id": "476da347-3032-4400-ad68-444a8e814996", "comment": "", "command": "mouseOver", "target": "css=btn-rds:nth-child(4)", "targets": [ "css=btn-rds:nth-child(4)", "css=finder" ], "xpath": "/button[@type='button'] [2]", "xpath.attributes": [ "xpath=/body[@id='tm-body']/div/div/div/div[2]/div/form/div/div/button[2]", "xpath=idRelative" ], "xpath=/button[2]", "xpath.position": [ ], "value": "" }, { "id": "6a02604-1dae-4104-ab62-0d74fd3f4e02", "comment": "", "command": "mouseOut", "target": "css=btn-rds:nth-child(4)", "targets": [ "css=btn-rds:nth-child(4)", "css=finder" ], "xpath": "/button[@type='button'] [2]", "xpath.attributes": [ "xpath=/body[@id='tm-body']/div/div/div/div[2]/div/form/div/div/button[2]", "xpath=idRelative" ], "xpath=/button[2]", "xpath.position": [ ], "value": "" }, { "id": "157e83d1-c8f4-454a-a186-f979b2bdf07", "comment": "", "command": "mouseOver", "target": "css=btn-rds:nth-child(4)", "targets": [ "css=btn-rds:nth-child(4)", "css=finder" ], "xpath": "/button[@type='button'] [2]", "xpath.attributes": [ "xpath=/body[@id='tm-body']/div/div/div/div[2]/div/form/div/div/button[2]", "xpath=idRelative" ], "xpath=/button[2]", "xpath.position": [ ], "value": "" }, { "id": "108182f2-33ec-4b74-9884-8bf5a052a27", "comment": "", "command": "mouseOut", "target": "css=btn-rds:nth-child(4)", "targets": [ "css=btn-rds:nth-child(4)", "css=finder" ], "xpath": "/button[@type='button'] [2]", "xpath.attributes": [ "xpath=/body[@id='tm-body']/div/div/div/div[2]/div/form/div/div/button[2]", "xpath=idRelative" ], "xpath=/button[2]", "xpath.position": [ ], "value": "" }, { "id": "2042959d-fce9-4c3c-b4ca-41a94c8e2136", "comment": "", "command": "mouseOver", "target": "css=btn-rds:nth-child(4)", "targets": [ "css=btn-rds:nth-child(4)", "css=finder" ], "xpath": "/button[@type='button'] [2]", "xpath.attributes": [ "xpath=/body[@id='tm-body']/div/div/div/div[2]/div/form/div/div/button[2]", "xpath=idRelative" ], "xpath=/button[2]", "xpath.position": [ ], "value": "" }, { "id": "6a51cbd3-9c7c-46bf-8d0a-c32fc77f744b", "comment": "", "command": "mouseOut", "target": "css=btn-rds:nth-child(4)", "targets": [ "css=btn-rds:nth-child(4)", "css=finder" ], "xpath": "/button[@type='button'] [2]", "xpath.attributes": [ "xpath=/body[@id='tm-body']/div/div/div/div[2]/div/form/div/div/button[2]", "xpath=idRelative" ], "xpath=/button[2]", "xpath.position": [ ], "value": "" }, { "id": "f55385a8-b1fd-4f4e-b560-af9e9f465d91", "comment": "", "command": "click", "target": "css=d-none path", "targets": [ "css=d-none path", "css=finder" ], "value": "" }, { "id": "23cfd9dc-5690-4c09-a0f8-6ff5201b0afe", "comment": "", "command": "mouseOver", "target": "id=click-to-change", "targets": [ "id=click-to-change", "id", "linkText=Change", "linkText" ], "css=#click-to-change", "css=finder", "xpath=/a[@id=click-to-change]", "xpath.attributes": [ "xpath=/body[@id='tm-body']/div/div/div/a[2]", "xpath=idRelative" ], "xpath=/a[contains(,Change)]", "xpath=innerText" ], "value": "" }, { "id": "9816a456-905e-4f52-831f-47670dde0c7e", "comment": "", "command": "mouseOut", "target": "id=click-to-change", "targets": [ "id=click-to-change", "id", "linkText=Change", "linkText" ], "css=#click-to-change", "css=finder", "xpath=/a[@id=click-to-change]", "xpath.attributes": [ "xpath=/body[@id='tm-body']/div/div/div/a[2]", "xpath=idRelative" ], "xpath=/a[contains(,Change)]", "xpath=innerText" ], "value": "" }, { "id": "bedf39d6-89ed-4b8a-95f7-bcb1647b3ab0", "comment": "", "command": "runScript", "target": "window.scrollTo(0,229)", "targets": [ ], "value": "" }, { "id": "1b1ee16-8751-43c7-b3ed-eeb1f6cc17d8", "comment": "", "command": "click", "target": "id=mail", "targets": [ "id=mail", "id", "css=#mail", "css=finder" ], "xpath": "/input[@id='mail']", "xpath.attributes": [ "xpath=/body[@id='tm-body']/div/div/div/div[2]/div/form/div/div/input", "xpath=idRelative" ], "xpath=/input", "xpath.position": [ ], "value": "" }, { "id": "b4e9f908-c6ed-4ac3-86ba-8046b6035317", "comment": "", "command": "close", "target": "", "targets": [ ], "value": "" }, { "id": "2f0b49d9-7b27-4b93-a0ae-5f3c0d6bba2", "name": "Default Suite", "persistSession": false, "parallel": false, "timeout": 300, "tests": [ ], "urls": [ "https://chatgpt.com/", "https://github.com/", "https://www.wikipedia.org/", "https://temp-mail.org/en/" ], "plugins": [ ] }
```


PRACTICAL NO 03

Aim : Install Selenium Server (Selenium RC) and demonstrate its usage by executing a script in Java or PHP to automate browser actions.

A. Installing PHP

1. Download PHP:

- Go to [PHP for Windows download page](<https://windows.php.net/download/>).
- Download the latest `Thread Safe` version of PHP (zip package).

VS16 x64 Thread Safe (2024-Jun-04 19:11:24)

▪ [Zip](#) [30.81MB]

sha256: 4cd0ccdcf613be57fb5c437a10b6e96c5d69016ec1c7f4c514398a72e36e4006

2. Install PHP:

- Extract the downloaded zip file to a directory, e.g., `C:\php`.

3. Add PHP to System PATH:

- Open the Start Menu, search for "Environment Variables", and select "Edit the system environment variables".
- In the System Properties window, click on the "Environment Variables" button.
- In the Environment Variables window, find the "Path" variable in the "System variables" section and click "Edit".
- Click "New" and add the path to your PHP directory, e.g., `C:\php`.
- Click "OK" to close all windows.

4. Verify PHP Installation:

- Open Command Prompt.
- Type `php -v` and press Enter.
- You should see the PHP version information.

B. Installing Composer

1. Download Composer:

- Go to the [Composer download page](<https://getcomposer.org/Composer-Setup.exe>).
- Download the Composer-Setup.exe file.

2. Install Composer:

- Run the Composer-Setup.exe file and follow the installation instructions.
- Ensure that you select the PHP executable (php.exe) when prompted.

3. Verify Composer Installation:

- Open Command Prompt.
- Type `composer --version` and press Enter.
- You should see the Composer version information.

C. Downloading Selenium Server

1. Download Selenium Server:

- Go to the [Selenium downloads page]

(<https://github.com/SeleniumHQ/selenium/releases/download/selenium-4.22.0/selenium-server-4.22.0.jar>).

- Download the `selenium-server-4.22.0.jar` file.

2. Save the Jar File:

- Save the jar file in a directory of your choice, e.g., `C:\Selenium`.

D. Starting Selenium Server

1. Open Command Prompt:

- Navigate to the directory where you saved the Selenium Server jar file:
`cd C:\Selenium`

```
C:\Users\Sameer>cd C:\Selenium  
C:\Selenium>|
```

2. Start Selenium Server:

- Run the following command:
`java -jar selenium-server-4.22.0.jar standalone`

```
C:\Selenium>java -jar selenium-server-4.22.0.jar standalone
```

- The server will start, and you should see a message indicating that it is up and running on `http://localhost:4444`.

E. Create Project Directory

1. Create a Directory:

- Create a new directory for your project, e.g., `C:\Selenium\PHPProject`.
- Open Command Prompt and navigate to this directory:

```
mkdir C:\Selenium\PHPProject
```

```
C:\Users\Sameer>mkdir C:\Selenium\PHPProject
```

```
cd C:\Selenium\PHPProject
```

```
C:\Users\Sameer>cd C:\Selenium\PHPProject  
C:\Selenium\PHPProject>|
```

F. Create Composer Configuration

1. Create composer.json:

- Inside your PHP project directory, create a file named `composer.json` with the following code:

```
{  
  "require": {  
    "php-webdriver/webdriver": "^1.15"  
  }  
}
```

2. Install Dependencies:

- Run `composer install` in the Command Prompt to install the necessary PHP WebDriver library:

```
composer install
```

```
c:\Selenium\PHPProject>composer install
```

G. Create the PHP Script

1. Create test.php:

- Inside your project directory, create a file named `test.php` with the following code:

```
<?php
require 'vendor/autoload.php';

use Facebook\WebDriver\Remote\RemoteWebDriver;
use Facebook\WebDriver\Remote\DesiredCapabilities;
use Facebook\WebDriver\WebDriverBy;

$host = 'http://localhost:4444'; // Selenium Server URL
$capabilities = DesiredCapabilities::chrome(); // or firefox, etc.
$driver = RemoteWebDriver::create($host, $capabilities);

// Navigate to a webpage
$driver->get('http://www.google.com');

// Find the search box element
$searchBox = $driver->findElement(WebDriverBy::name('q'));
// Enter text into the search box
$searchBox->sendKeys('Selenium');

// Submit the search form
$searchBox->submit();

// Close the browser
$driver->quit();
?>
```

H. Run the PHP Script

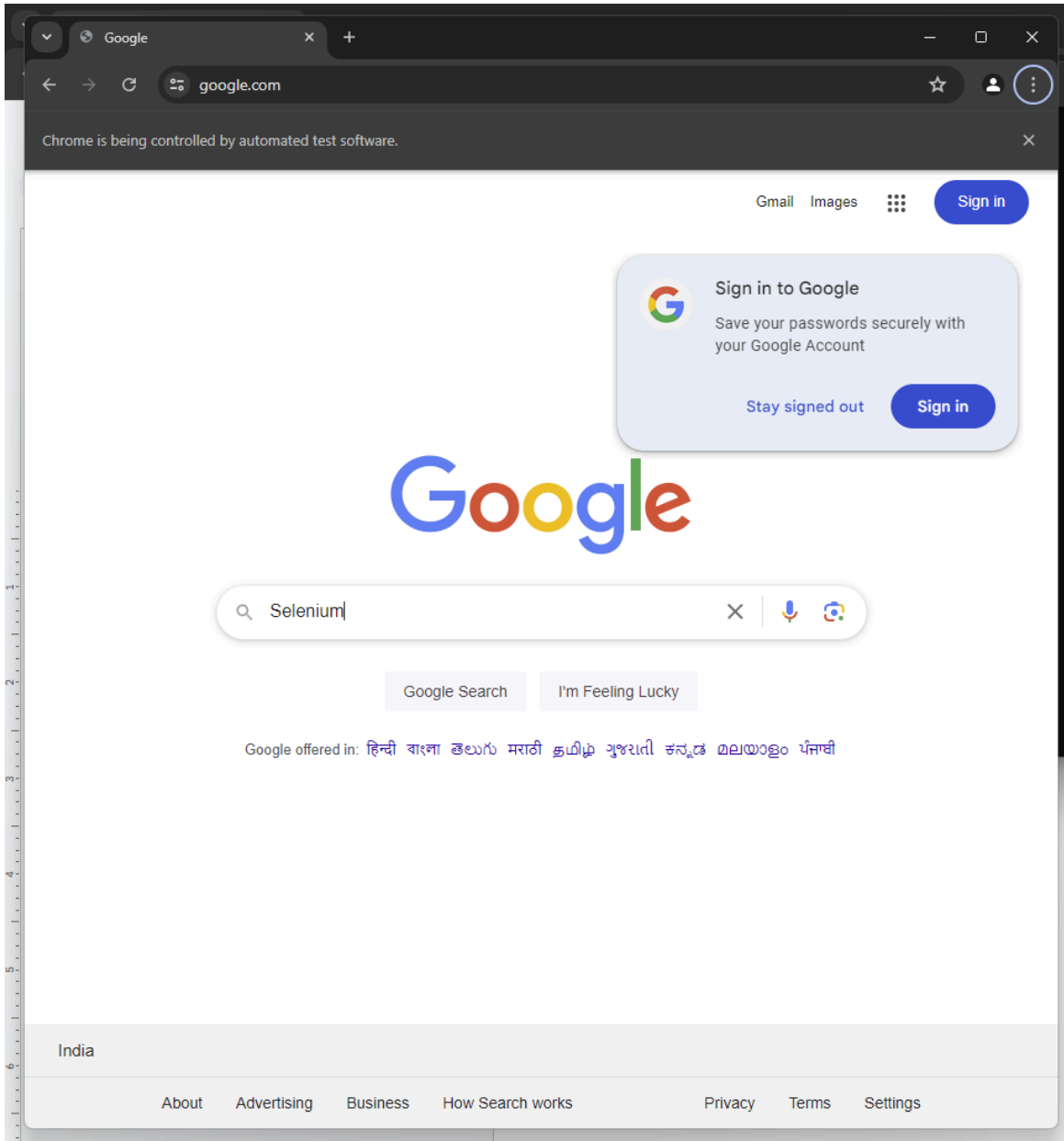
1. Run Your Script:

- Open Command Prompt and navigate to your project directory:
cd C:\Selenium\PHPProject

- Run your PHP script:
php test.php

```
C:\Users\Sameer>cd C:\Selenium\PHPProject  
C:\Selenium\PHPProject>php test.php
```

OUTPUT :



PRACTICAL NO 04

Aim : Write a program using Selenium WebDriver to automate the login process on a specific web page. Verify successful login with appropriate assertions.

Installation :

1. Run this command “pip install selenium” in cmd
2. To check installation run “pip show selenium” in cmd

Code :

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from time import sleep

options = Options()
options.add_experimental_option("excludeSwitches", ["enable-logging"])

print("Testing Started")

driver = webdriver.Chrome(options=options)
driver.get("https://www.saucedemo.com/")
sleep(3)

title = driver.title
print(f"Page title is {title}")

driver.find_element(By.ID, "user-name").send_keys("standard_user")
driver.find_element(By.ID, "password").send_keys("secret_sauce")
driver.find_element(By.ID, "login-button").click()
sleep(3)

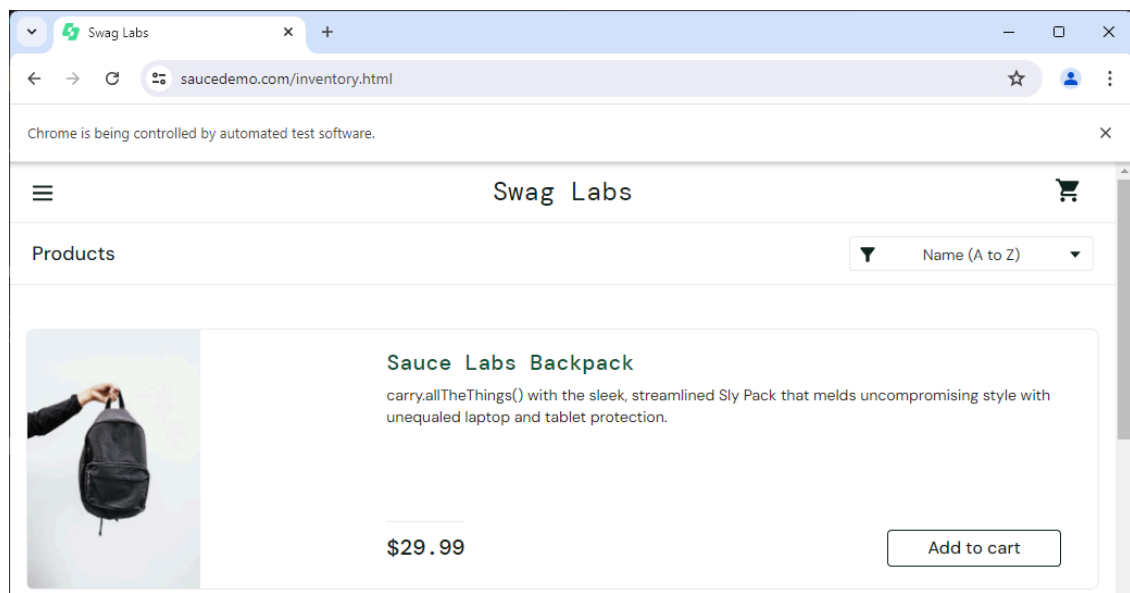
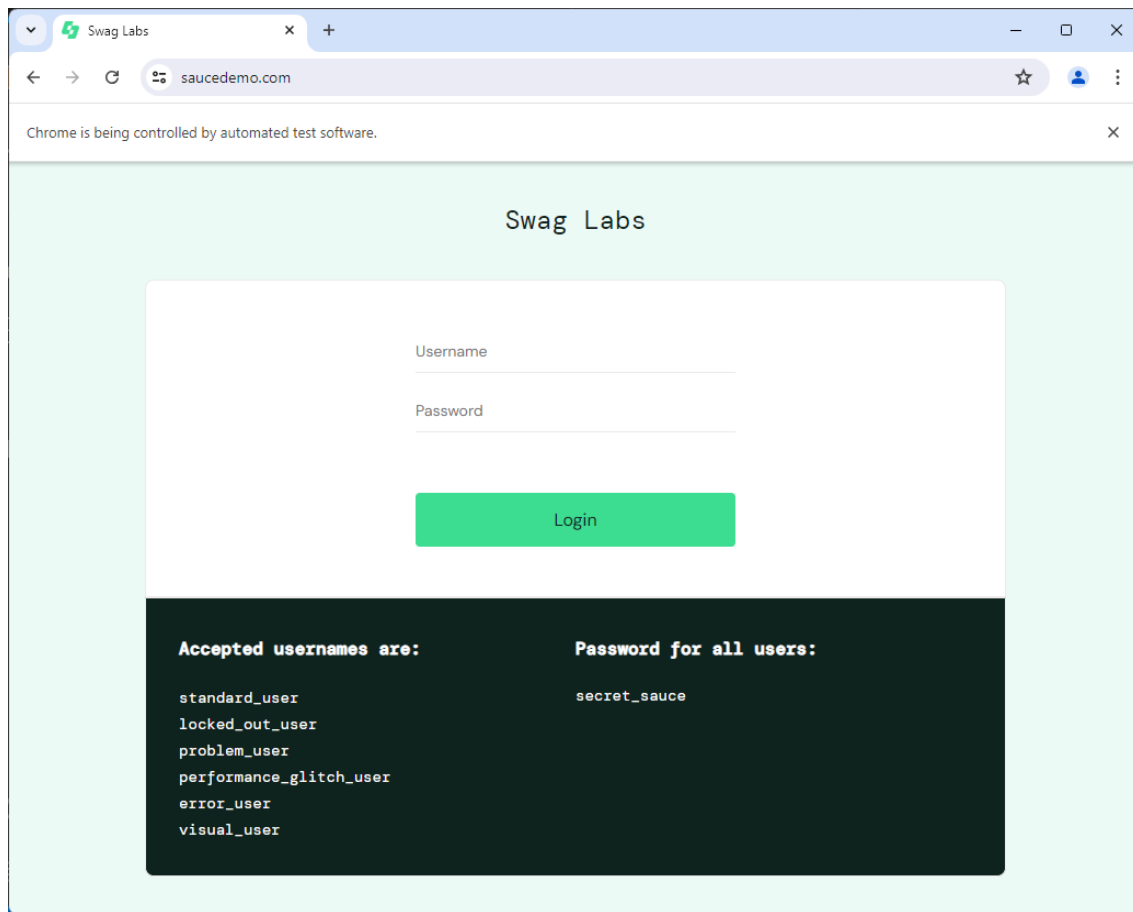
first_item = driver.find_element(By.CLASS_NAME, "inventory_item")
assert first_item.is_displayed(), "First inventory item is not displayed"

first_item_name = first_item.find_element(By.CLASS_NAME, "inventory_item_name").text
print(f"First item name: {first_item_name}")

print("TEST PASSED: Mr. Sameer, your LOGIN is SUCCESSFUL and first inventory item is verified")

driver.quit()
```

Output :



Testing Started

Page title is Swag Labs

First item name: Sauce Labs Backpack

TEST PASSED: Mr. Robot, your LOGIN is SUCCESSFUL and first inventory item is verified

PRACTICAL NO 05

Aim : Write a program using Selenium WebDriver to update 10 student records in an Excel file. Perform data manipulation and verification.

Code :

```
import java.io.File;
import java.io.IOException;
import jxl.Sheet;
import jxl.Workbook;
import jxl.read.biff.BiffException;
import jxl.write.Number;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import jxl.write.WriteException;
import org.testng.annotations.Test;

public class StudentUpdate {
    private static final String INPUT_FILE = "C:\\Users\\Sameer\\student_records.xls";
    private static final String OUTPUT_FILE =
"C:\\Users\\Sameer\\student_records_updated.xls";

    @Test
    public void testImportExport() throws IOException, BiffException, WriteException {
        File inputWorkbook = new File(INPUT_FILE);
        Workbook w;
        try {
            w = Workbook.getWorkbook(inputWorkbook);
            Sheet s = w.getSheet(0);
            WritableWorkbook copy = Workbook.createWorkbook(new File(OUTPUT_FILE), w);
            WritableSheet sheet = copy.getSheet(0);
            int studentsAbove60 = 0;

            for (int i = 1; i <= 10; i++) {
                String studentStr = s.getCell(0, i).getContents();
                String marksStr = s.getCell(1, i).getContents();
                try {
                    int studentNumber = Integer.parseInt(studentStr);
                    int marks = Integer.parseInt(marksStr);
                    int updatedMarks = marks + 10;

                    Number updatedMarksCell = new Number(1, i, updatedMarks);
                    sheet.addCell(updatedMarksCell);

                    if (marks > 60) {
                        studentsAbove60++;
                    }
                }
            }
        }
    }
}
```



```

    }
    System.out.println("Record successfully updated for student " +
studentNumber + ": " + marks + " -> " + updatedMarks);
    } catch (NumberFormatException e) {
        System.out.println("Invalid data for student in row " + (i + 1) +
": " + studentStr + ", " + marksStr);
    }
}
copy.write();
copy.close();
w.close();
System.out.println("All records successfully updated");
System.out.println("Number of students who scored above 60 (before update):
" + studentsAbove60);
    } catch (IOException | BiffException | WriteException e) {
        e.printStackTrace();
    }
}
}
}

```

Output :

```

[RemoteTestNG] detected TestNG version 7.4.0
Record successfully updated for student 1: 20 -> 30
Record successfully updated for student 2: 30 -> 40
Record successfully updated for student 3: 40 -> 50
Record successfully updated for student 4: 50 -> 60
Record successfully updated for student 5: 60 -> 70
Record successfully updated for student 6: 70 -> 80
Record successfully updated for student 7: 80 -> 90
Record successfully updated for student 8: 90 -> 100
Record successfully updated for student 9: 100 -> 110
Record successfully updated for student 10: 110 -> 120
All records successfully updated
Number of students who scored above 60 (before update): 5
PASSED: testImportExport1

```

```

=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====

=====
Default suite
Tests run: 1, Passes: 1, Failures: 0, Skips: 0
=====

```

| | A | B | | | A | B |
|----|----------|-------|--|----|----------|-------|
| 1 | Students | Marks | | 1 | Students | Marks |
| 2 | 1 | 20 | | 2 | 1 | 30 |
| 3 | 2 | 30 | | 3 | 2 | 40 |
| 4 | 3 | 40 | | 4 | 3 | 50 |
| 5 | 4 | 50 | | 5 | 4 | 60 |
| 6 | 5 | 60 | | 6 | 5 | 70 |
| 7 | 6 | 70 | | 7 | 6 | 80 |
| 8 | 7 | 80 | | 8 | 7 | 90 |
| 9 | 8 | 90 | | 9 | 8 | 100 |
| 10 | 9 | 100 | | 10 | 9 | 110 |
| 11 | 10 | 110 | | 11 | 10 | 120 |
| 12 | | | | 12 | | |

Before

After

PRACTICAL NO 06

Aim : Write a program using Selenium WebDriver to select the number of students who have scored more than 60 in any one subject (or all subjects). Perform data extraction & analysis.

Code :

```
import java.io.File;
import java.io.IOException;

import jxl.Sheet;
import jxl.Workbook;
import jxl.read.biff.BiffException;

import org.testng.annotations.Test;

public class StudentUpdate {
    private static final String INPUT_FILE =
"C:\\Users\\rdnc2\\Pictures\\student_records.xls";

    @Test
    public void testImportExport1() throws IOException, BiffException {
        File inputWorkbook = new File(INPUT_FILE);
        Workbook workbook;

        try {
            // Open the workbook
            workbook = Workbook.getWorkbook(inputWorkbook);
            Sheet sheet = workbook.getSheet(0); // Access the first sheet

            int studentsAbove60 = 0;
            // Loop through student data (assuming the first 10 rows after the header
contain data)
            for (int i = 1; i <= 10; i++) {
                String studentStr = sheet.getCell(0, i).getContents(); // Student
Number

                String marksStr = sheet.getCell(1, i).getContents(); // Marks

                try {
                    int studentNumber = Integer.parseInt(studentStr);
                    int marks = Integer.parseInt(marksStr);

                    // Check if the marks are above 60
                    if (marks > 60) {
                        studentsAbove60++;
                        System.out.println("Student " + studentNumber + " scored above
60: " + marks);
                    }
                }
            }
        }
    }
}
```

```

    }
} catch (NumberFormatException e) {
    // Handle case when student number or marks are not valid numbers
    System.out.println("Invalid data for student in row " + (i + 1) +
": " + studentStr + ", " + marksStr);
}
}
// Close the workbook after processing
workbook.close();
System.out.println("Number of students who scored above 60: " +
studentsAbove60);
} catch (IOException | BiffException e) {
    // Print the stack trace if there's an issue reading the file
    e.printStackTrace();
}
}
}

```

Output :

```

[RemoteTestNG] detected TestNG version 7.4.0
Student 6 scored above 60: 70
Student 7 scored above 60: 80
Student 8 scored above 60: 90
Student 9 scored above 60: 100
Student 10 scored above 60: 110
Number of students who scored above 60: 5
PASSED: testImportexport1

```

```

=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====

```

```

=====
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
=====

```

PRACTICAL NO 07

Aim : Write a program using Selenium WebDriver to provide the total number of objects present or available on a web page. Perform object identification and counting.

Code :

```
package javapract7;
import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class WebElementCounter {
    @SuppressWarnings("deprecation")
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
"C:\\Users\\Sameer\\Desktop\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        String url = "https://wikipedia.com";
        driver.get(url);
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

        List<WebElement> buttons = driver.findElements(By.tagName("button"));
        System.out.println("The number of buttons is " + buttons.size());

        List<WebElement> inputs = driver.findElements(By.tagName("input"));
        System.out.println("The number of input fields is " + inputs.size());
        for (WebElement button : buttons) {
            System.out.println("Button text: " + button.getText());
        }
        for (WebElement input : inputs) {
            System.out.println("Input type: " + input.getAttribute("type"));
        }
        driver.close();    }
}
```

Output :

```
The number of buttons is 2
The number of input fields is 4
Button text: Search
Button text: Read Wikipedia in your language
Input type: hidden
Input type: hidden
Input type: search
Input type: hidden
```

PRACTICAL NO 08

Aim : Write a program using Selenium WebDriver to update 10 student records in an Excel file. Perform data manipulation and verification.

Code :

```
Package javapract8;
import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class l1l1 {
@SuppressWarnings("deprecation")
public static void main(String[] args) {
Replace with the actual path
WebDriver driver = new ChromeDriver();
String url = "https://wikipedia.com"; // Replace with the actual URL of the page
containing the combo box
driver.get(url);
// Set an implicit wait time
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
// Locate the combo box (select element) using CSS Selector
WebElement comboBox = driver.findElement(By.cssSelector("select")); // This Locates
the first <select> element on the page
List<WebElement> options = comboBox.findElements(By.tagName("option"));
System.out.println("Number of items in the combo box: " + options.size());
for (WebElement option : options) {
System.out.println(option.getText());
}
driver.close();
}
}
```

Output :

```
Number of items in the combo box: 71
Afrikaans
Polski
العربية
Asturianu
Azərbaycanca
```

PRACTICAL NO 09

Aim : Write a program using Selenium WebDriver to update 10 student records in an Excel file. Perform data manipulation and verification.

Code :

```
Package javapract9;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import java.util.List;

public class WebElementCheckboxCounter {
    public static void main(String[] args) {
        WebDriver driver = new FirefoxDriver();
        try {
            driver.get("https://www.w3schools.com/html/html_forms.asp");
            List<WebElement> checkboxes =
driver.findElements(By.cssSelector("input[type='checkbox']"));
            int totalCheckboxes = checkboxes.size();
            int checkedCheckboxes = 0;
            int uncheckedCheckboxes = 0;
            for (WebElement checkbox : checkboxes) {
                if (checkbox.isSelected()) {
                    checkedCheckboxes++;
                } else {
                    uncheckedCheckboxes++;
                }
            }
            System.out.println("Total checkboxes: " + totalCheckboxes);
            System.out.println("Checked checkboxes: " + checkedCheckboxes);
            System.out.println("Unchecked checkboxes: " + uncheckedCheckboxes);
        } finally {
            driver.quit();
        }
    }
}
```

Output :

```
Total checkboxes: 3
Checked checkboxes: 0
Unchecked checkboxes: 3
```

PRACTICAL NO 10

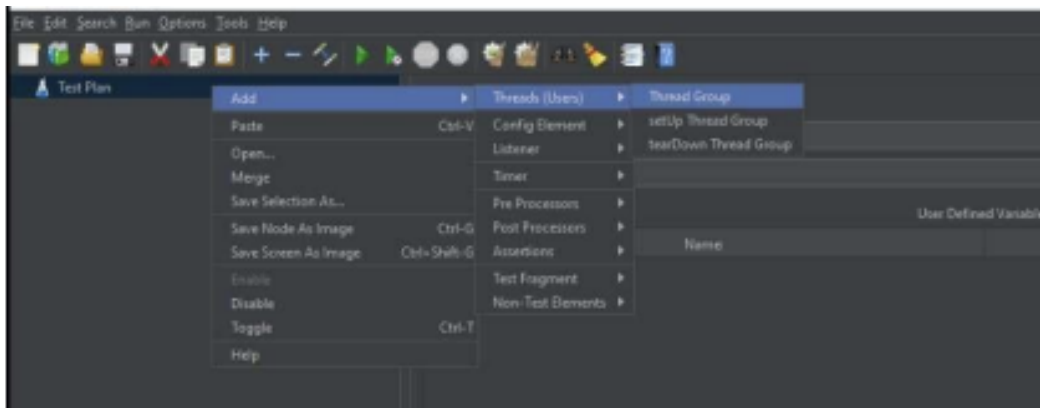
Aim : Perform load testing on a web application using JMeter. Generate and analyze load scenarios. Additionally, explore bug tracking using Bugzilla as a tool for logging and tracking software defects

Step-1 : Download and Unzip Jmeter package from Apache Website.

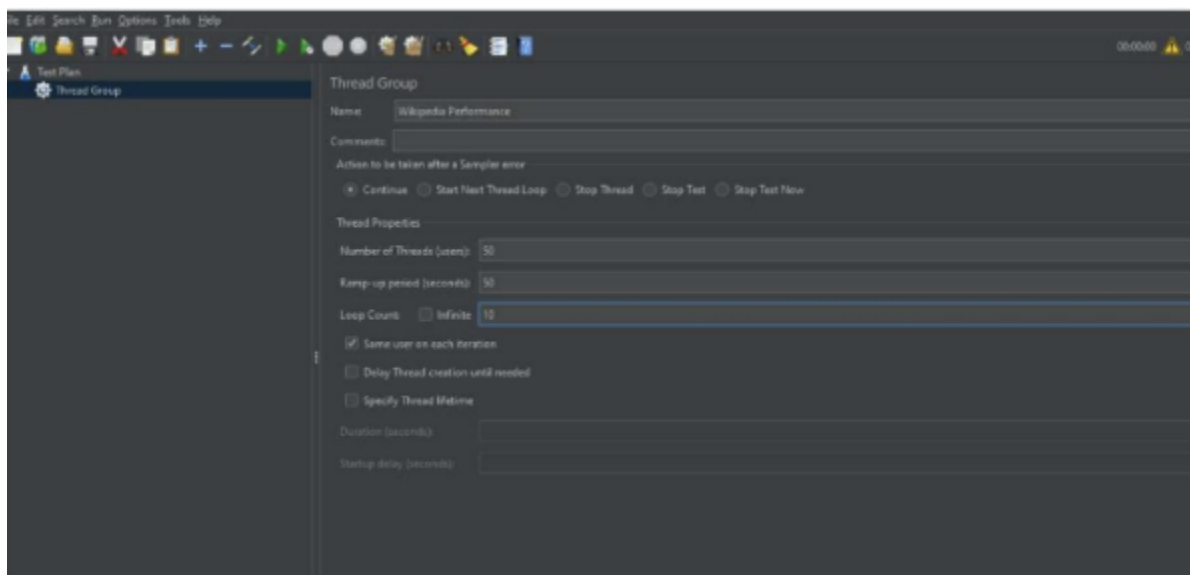
Step-2 : Navigate to the bin folder and run the "ApacheJMeter.jar" or "jmeter.bat"

file and it will launch a window. In case of error/failure refer to the PreInstallation Requirements.

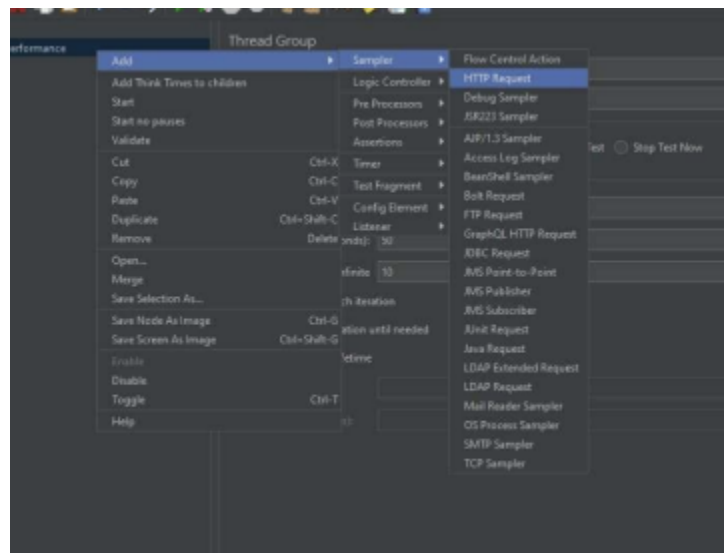
Step-3 : Now First step is to add an Thread Group, RightclickonTestPlan → Add → Threads(Users) → ThreadGroup.



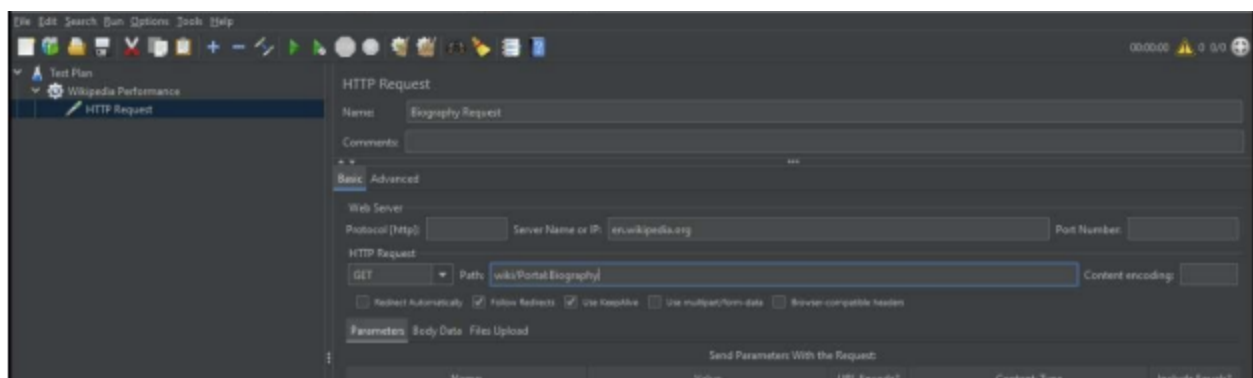
Step-4 : Next rename the Thread Group and set "Number of Threads (Users):" as 50 and "Ramp-up period(seconds):" as 50. Keep rest options as default.



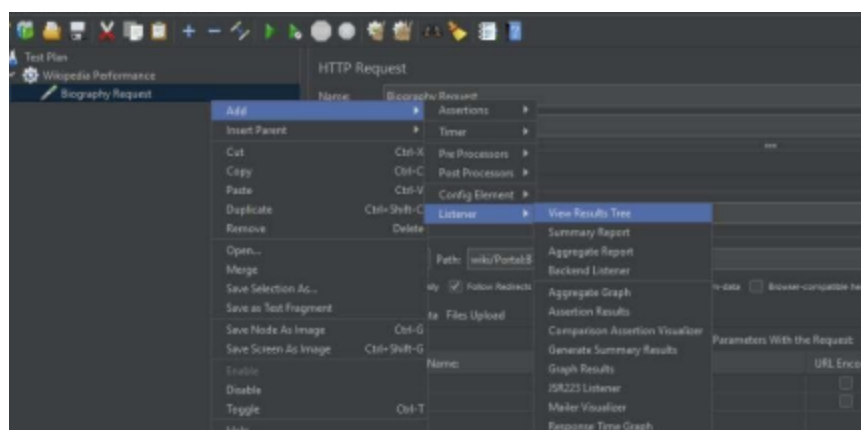
Step-5 : We add a Http Request Sample to the Thread Group Right Click ThreadGroup-> Add -> Sampler -> HTTPRequest



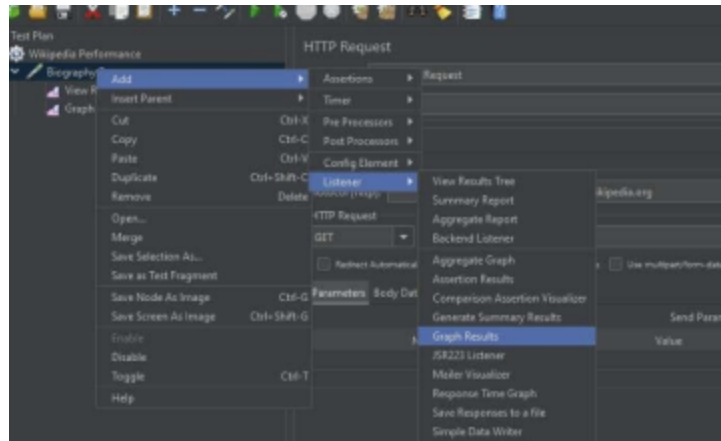
Step-6 : Rename Http Request to Biography Request. Next, Edit the “Server Name or IP:” to the specific website or Server IP address, optionally one can configure port number and Request parameters and path as well



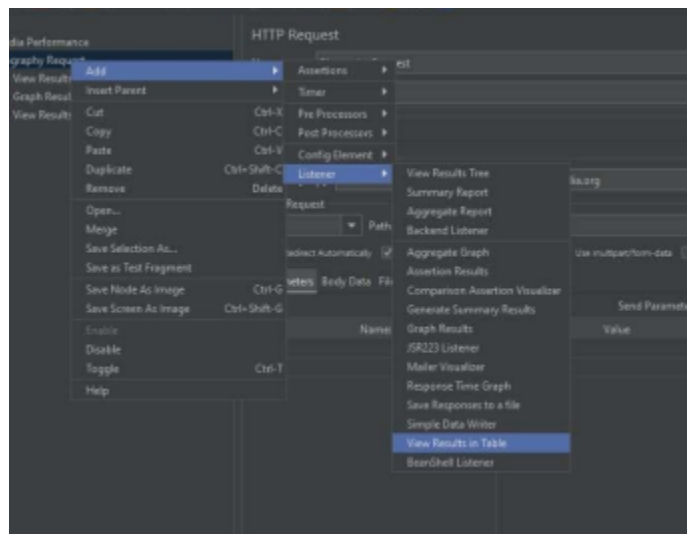
Step7: To view the log data add a “View Results Tree Listener” Right click on TestPlan →Add → Listener → View Results Tree



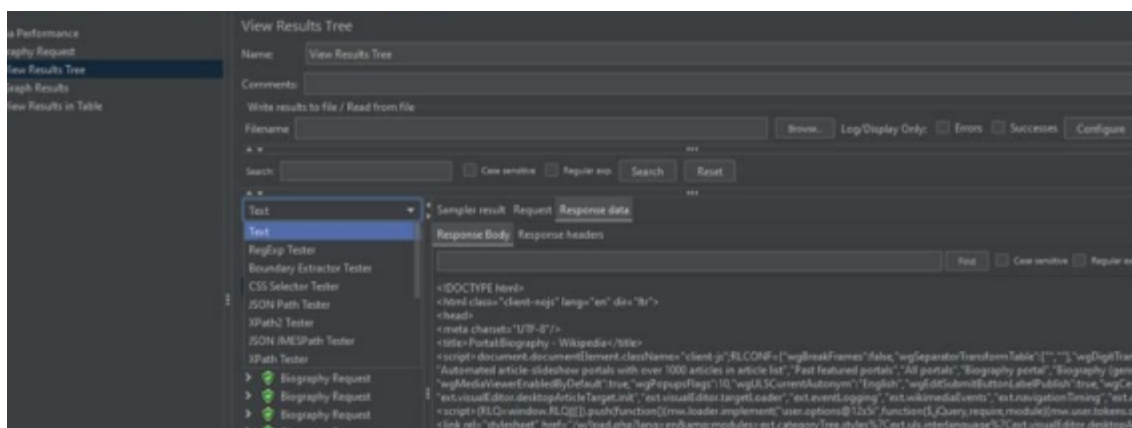
Step-8: To visualize the data we add a “Graph Results Listener” Right Click on TestPlan →Add → Listener →GraphResults



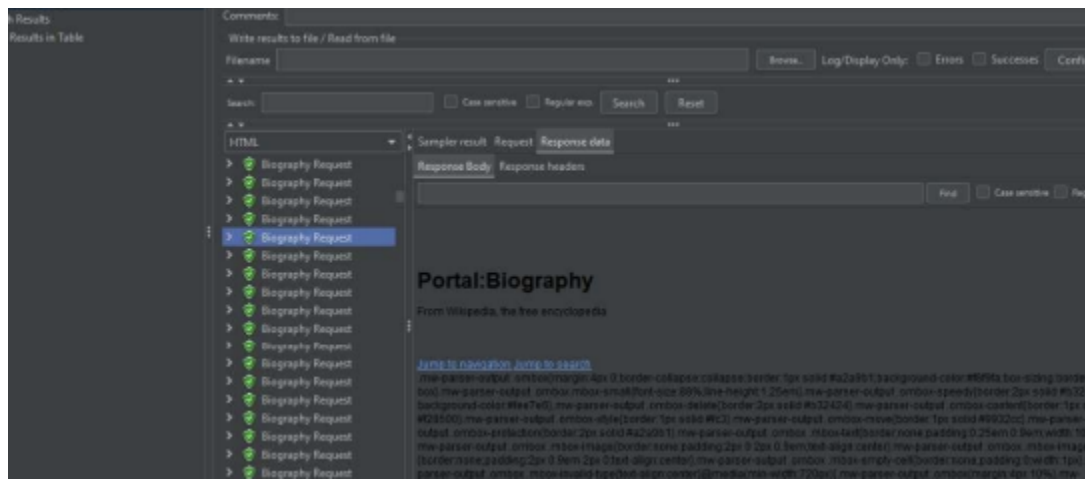
Step-9: To see the data we add a “View Results in Table Listener” Right Click on TestPlan→ Add → Listener →View Results in Table



Step 10: You can view the logged results of the test in “View Results Tree” in text.



Step 11: After the test completes successfully you can view the logged results of the test in “View Results Tree” in Html.



Step12 : One can view the parameters by looking at the “View Results in Table”

| Sample # | Start Time | Thread Name | Label | Sample Time | Status | Bytes | Sent Bytes | Latency | Connect Time |
|----------|--------------|-------------------|------------------|-------------|---------|--------|------------|---------|--------------|
| 1 | 12:02:43.675 | Wikipedia Perf... | Biography Req... | 1132 | Success | 448527 | 329 | 285 | |
| 2 | 12:02:43.840 | Wikipedia Perf... | Biography Req... | 2170 | Success | 448527 | 329 | 801 | |
| 3 | 12:02:45.009 | Wikipedia Perf... | Biography Req... | 433 | Success | 448527 | 329 | 281 | |
| 4 | 12:02:45.010 | Wikipedia Perf... | Biography Req... | 447 | Success | 448527 | 329 | 282 | |
| 5 | 12:02:44.674 | Wikipedia Perf... | Biography Req... | 551 | Success | 448527 | 329 | 204 | |
| 6 | 12:02:45.442 | Wikipedia Perf... | Biography Req... | 336 | Success | 448527 | 329 | 142 | |
| 7 | 12:02:45.457 | Wikipedia Perf... | Biography Req... | 325 | Success | 448527 | 329 | 129 | |
| 8 | 12:02:45.625 | Wikipedia Perf... | Biography Req... | 326 | Success | 448527 | 329 | 135 | |
| 9 | 12:02:45.762 | Wikipedia Perf... | Biography Req... | 152 | Success | 448527 | 329 | 147 | |

Step-13: The Load Produced on the Server is different for different parameters and also the throughput you get may be less because of hardware

