Software Testing Tool: Selenium

Harpreet Kaur¹, Dr. Gagan Gupta²
¹M. Phil in Computer Applications (Research Scholar), Guru Kashi University, Talwandi Sabo, Punjab, INDIA
²Department of Applied Sciences, Guru Kashi University, Talwandi Sabo, Punjab, INDIA

ABSTRACT
Software testing is an area of software development where persistence is essential. Software testing is a process used to identify the correctness, completeness, and quality of developed computer software. We describe that how automated testing tools are very much convenient and easy to use which also makes testing faster and more effective in less time. The Objective of this paper is to conduct Automation Testing using Software Testing Tool “Selenium”. Selenium IDE is a free and open source add-on for the Firefox web browser. It can be easily downloaded from the internet using selenium web site. It is primarily used by the Web development community to perform automated testing of web applications. In this paper firstly we analyzed the Integrated Development Environment of Selenium a Software testing tool. Secondly we have performed the black box testing of web application.

Keywords— Selenium, Manual Testing, Automation testing White box testing, black box testing, SDLC, STLC.

I. INTRODUCTION

Software testing refers to process of evaluating the software with the intention to find out error in it. Software testing is a technique aimed at evaluating an attribute or capability of a program or product and determining that it meets its quality [1].

Software testing is the process of executing a program to verify its functionality and correctness [2].

Software Testing can be divided into two types. Manual testing is in which the testing activity is performed by testing persons. Manual testing requires a tester to perform manual test operations on the test software without the help of Test automation. Manual Testing is a process where in a tester often follows a written test plan that leads them through a set of important test cases. A test case in software testing is a set of conditions written for particular applications & tester run all these conditions to ensure the correct functionality of the software applications [3].

Automation Testing is running test cases where manual intervention is not required to run each one. It uses special software to write & execute test cases to compare the actual outcome with the predicted outcome. Once tests have been automated; they can be run quickly and repeatedly. Automated software testing is the best way to increase the effectiveness, efficiency and coverage of software testing. Automation testing requires considerable amount of investment for buying the software & compatible hardware resources. Automation testing does what manual testing does not. Automation testing improves the accuracy & it saves the time of the tester & organization’s money [4].

It is the important phase of any system development life cycle of a product. For software testing, there is a complete System testing life cycle followed by software companies. There are number of techniques for software testing[5].

Automated software testing is an essential element component of successful development projects. Automation testing draws maximum benefits with minimum efforts by increasing the quality and reliability of the products. There are various tools which are used in automation testing [6].

The main two techniques are Back box testing and White box testing. In back box testing technique tester will test the Graphical User Interface of software application [7].

Tester uses this approaches both for window based applications and Web applications. The other testing technique is White Box Testing, in this tester will check for the code of the software application [8].

The Web GUI applications are becoming more and more complex due to development of internet technology. Software development cycle is becoming shorter, which makes GUI application testing [9].

The web application testing is more complicated as it involves different testing strategies and tools, because the tool has to support different characteristic properties and also different frameworks [10].

In this paper, the software testing tool selenium is a free (open source) automated testing suite for web applications across different browsers & platform. It is primarily developed in Java Script and browser technologies. Hence it supports all the major browsers on all the platforms. Selenium is not just a single tool but a suite of software, each catering to different testing
needs. It is primarily developed in JavaScript and browser technologies. Hence it supports all the major browsers on all the platforms. Selenium is not just a single tool but a suite of tools, each catering to different testing needs. Selenium is a plug-in that you can install as easily as you can install other plug-ins.

II. INTRODUCTION TO SELENIUM

Selenium is an open source, robust set of tools that supports rapid development of test automation for web based applications. Selenium provides a record/playback tool for authoring tests without learning a test scripting language. Selenium is written in JavaScript. Its support different browsers and multiple platforms. Selenium provides a rich set of testing functions specifically geared to the needs of testing of a web application. These operations are highly flexible, allowing many options for locating UI elements and comparing expected test results against actual application behavior.

Selenium Components
It has four components:-
- Selenium integrated development environment (IDE)
- Selenium remote control (RC)
- Web driver
- Selenium grid

The selection of particular automated testing tool is based on the type of application we are testing and the cost associated with the tool. In the present work, we have evaluated the open source software testing tool Selenium. Our main motive is to perform black box testing on the web application www.fontconvertergosht.in

III. SELENIUM IDE

In Selenium IDE scripts may be automatically recorded and edited manually providing auto-completion support and ability to move commands around quickly. Selenium IDE is an integrated development environment for Selenium scripts. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests. It was previously known as Selenium Recorder. Selenium IDE is not only recording tool: it is a complete IDE.

For this study we use the current version of selenium that is 2.0.0 we will discuss the following parameters of Selenium 2.0.0

Recording efficiency: -It is a recording and playback type tool. When Selenium IDE is first opened the record button is ON by default. We can pause recording in between. Recording commands are inserted in the table to check whether application works as intended. When we play the recorded test case the commands execute. We can see the commands executed in the log mentioned in the selenium IDE.

Capability of generation of scripts: Selenium generates html scripts. We can modify the scripts. The main advantage of these scripts is that all test cases written can be run against most modern web browsers.

Data driven testing: Selenium doesn’t support data driven testing by default we can use sel blocks plug-in to perform data driven testing in selenium.

Test result report: We can check the result of executed test case by using plug in test result.

Execution Speed: - we can adjust the execution speed of a test case from the bar.

Playback Capability: - when script is played back it replaced the user action data performed by the user during recording.

Easy to Learn: - we can easily learn the feature of this tool we can also enhance the functionality of exiting features.

It is free software easily available on internet.

Execution of test case: - A test designed to confirm a case scenario is called a test case. In a software company, it is the responsibility of tester to write the test cases for the whole application. We have developed two test cases to test the web application. In these test cases, we have checked the validity of the login page. Our approach was to perform negative testing. We planned to check the application flow and input data. The various fields of test case are as follows:-

Test Case ID: - It can be any alpha numeric character to uniquely identify a test case. We generally define test case Id according to the nature of test case.

Objective: - It describes the function of the test case.

Inputs/Steps/Action/Description: - It describes the steps that needed to be performed before execution of a particular test case.

Expected result: -It describes the tester’s expectation from the execution of the test case.

Actual result: - it describes the result of the executed test case.

Status: - Tester mentions the result of the test case as fail or pass.

Remarks: - Tester writes the remarks, if required.

Test cases for the user login form are as follows:-

<table>
<thead>
<tr>
<th>Test ID</th>
<th>Objective</th>
<th>Input/Steps/Action/Description</th>
<th>Expected Result</th>
<th>Actual Result</th>
<th>Status</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>login1</td>
<td>To check the user login</td>
<td>User will provide the correct user name and password</td>
<td>User will be redirected to the login page</td>
<td>User login successful</td>
<td>Pass</td>
<td>Login successful</td>
</tr>
</tbody>
</table>
Implementation of test cases using SELENIUM tool

Selenium IDE is an add-on feature in Firefox. To activate selenium, the following steps are to be followed. Select tools option from Firefox menu bar. Select selenium IDE option from tools popup. The screen shown in figure will appear. It has a recording button on the right side of the screen. If mouse pointer is placed on the button then there is a popup message “Now recording, click to stop Recording”. This message shows that selenium IDE is ready for recording test cases.

Figure 1: Screen of Selenium for recording Test Case

Figure 2: Application login window with correct user name and password

Figure 3: Selenium window with recorded user actions

The other test case we have written is for incorrect user name and password. According to the expected result, application should show the error message as wrong user name and wrong password.

Figure 4: Application login window with incorrect user name and incorrect password

When the tester fills wrong user name and wrong password, the expected outcome should be an error message on the application window. As the actual result and the expected result is the same, this shows that test case executed is passed.
IV. ADVANTAGES OF SELENIUM IDE

- Can only test web applications.
- Automates at a slower rate because it does not have a native IDE and only third party IDE can be used for development.
- Clever option for adding different asserts options in scripts.
- Cannot access elements outside of the web application under test.
- No official user support is being offered.
- Has no native capability to export runtime data onto external formats.
- Parameterization can be done via programming but is difficult to implement.
- No native support to generate test/bug reports.

V. LIMITATIONS TO SELENIUM IDE

- Selenium IDE tool can only be used in Mozilla Firefox as it is an Add-on for only that browser and hence it cannot be used with other browsers.
- It is a record and playback tool and the script format can be written in Selene

VI. CONCLUSIONS

This is a good automated software testing tool to test the web applications. The tester can understand the tool as it is open source. By using selenium tool, we can increase the effectiveness, efficiency and coverage of software testing. With the help of the case study, we analyze the features of selenium. In this paper we have implemented the record, replay and run feature of the selenium IDE.

VII. ACKNOWLEDGMENTS

We wish our sincere gratitude to Guru Kashi University Talwandi sabo (Bathinda), India for providing the facilities to carry out the work.

REFERENCE