IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A SURVEY ON SELENIUM AUTOMATION TESTING TOOL

[1]Nilesh Tiwari, [2]Prachi Agrawal, [3]Megha Chouhan, [4]Hitesh Kag

[1]Computer Science Student, [2]Computer Science Student, [3]Computer Science Student, [4]Assistance Professor [1]Department of Computer Science and Engineering

[1]Medicaps University, Indore, India

Abstract

Before any web application or software is made available to general public; it goes through a process called Testing. It checks for security, compatibility, functionality and performance of the Software or Web Application. Testing can be done manually as well as automatically. Automated testing is a technique that performs testing using special automated testing software tools whereas manual testing is done by a human where one has to carefully execute each and every test case. In software development cycles, often same test case has to be run multiple times. An Automation tool records the test case and then it can be used as often as it is required. In this study, Selenium as an automation testing tool has been discussed in depth as well as why the Selenium Web Driver is ideally equipped for automation testing. Through our project, we identified the benefits of automation testing over manual testing.

Keywords: Selenium, Selenium Suite, Automated Testing, Manual Testing, Web Driver.

I. Introduction

Software testing [1] is a technique for determining whether the software product meets the intended specifications and ensuring that it is defect-free. It entails the use of manual or automated methods to test one or more properties of interest by executing software or system components. In comparison to real specifications, the aim of software testing is to discover bugs, errors, and missing requirements.

There are numerous reasons to implement Program Testing. A few of them are:

- Identifying bugs and errors that occurred during the development phase
- Maintain the Quality of Software.
- Timely testing of software allows you to save money in the long run. It is less expensive to patch bugs discovered earlier in the software testing phase.
- Helps to maintain security and deploy trusted products.

Manual Testing: It is a process of identifying errors in the source code manually without any involvement of automated tool. It is an integral part of Software Development Lifecycle to ensure that code is error free.

Manual Testing is done through a sequence of processes. In the first place, tester notices all records identified with programming, to choose testing territories and investigates prerequisite reports to cover all necessities expressed by the client. All experiments are executed physically by utilizing Black Box testing and White box testing [13]. On the off chance that bugs happened, the testing group illuminates the development team.. After fixing the bugs, application is handed over to the Testing Team for a recheck.

Automated Testing: It is a software testing technique that involves using automated software tools with built-in feature to control the execution of tests and then comparing the actual test results to the anticipated or planned results.

Automated Testing is also done through a sequence of processes. It starts with selecting the best tool suited for the software testing. The scopes of testing are figured out based on data set and complexity. Further, the strategies and plan to execute testing is created. Finally, test automation scripts are executed to check for anticipated outputs.

Difference between Manual and Automation Testing [2]:

Table 1. Difference between Manual Testing and Automation Testing

PARAMETER	MANUAL TESTING	AUTOMATED TESTING
Definition	Test cases are executed by human	Automation testing tools are used to execute test
Processing time	It is time consuming	It is significantly faster
Exploratory testing	Random testing is allowed.	Random testing is not allowed.
Initial investment	Initial investment is low and ROI is also low in long run	Initial investment is significantly higher and ROI is also higher.
Reliability	It is prone to human error thus not accurate	It is a reliable method as it is performed by tools

I. PURPOSE OF RESEARCH

The aim of this study is to examine software and web testing tools such as Selenium, WATIR, and UFT for automated testing using previously collected research. Based on previous reports, we also intend to analyze and comprehend the Selenium Suite which comprises Selenium Integrated Development (IDE), Selenium Remote Control (RC), Web Driver and Selenium Grid. The research work should be handy for organizations because of multiple benefits of automation testing; a number of organizations are working to improve it.

II. LITERATURE SURVEY

There have been a decent number of previous works on Automation Testing. We have gathered several papers in order to analyze the testing tools and then reviewed the findings.

Neha Chaurasia et al. [3] presented a study with an aim of comparing and contrasting the principles, builds, and features of automated and manual testing. Furthermore, it emphasizes the significance of automated software testing. This paper focuses on the selenium testing method, and based on it they have identified them at the appropriate stage. They spoke about the different types of software testing and the differences between manual and automated testing.

Prasad Mahajan [4] et al. covered Test Automation and its pre-requisites, working measures, when to use automation testing, advantages over manual testing, and how to choose which test cases to automate in this document. At last they concluded that the test automation mechanism is used to reduce costs and overheads. The time spent on regression tests is decreased as a result of automation. However, the success of automation testing is dependent on choice of a suitable and compatible method.

Nisha Gogna [5] in her study discussed basic features of browser-based automated test tools WATIR and Selenium RC, Selenium IDE and Selenium GRID. The study concluded that to use WATIR one must know Ruby .She mentioned Selenium has its own integrated development environment (IDE) for running test cases. In this study it can be easily seen that scope of Selenium is more than that of WATIR when it comes to automated testing.

Harshali Patil [6] drew a comparison between Selenium and UFT testing tools to choose a testing method based on various factors like the type of application to be evaluated, the budget, reusability, language support, application support and the efficiency required. For test scripts based on web browser, it is better to use Selenium to reduce cost and for high quality software, we should use UFT for testing.

Inderjeet Singh et al. [7] analyzed and compared the three testing tools based on factors like execution time, recording capabilities, scripts generation, data driven testing, ease of learning and supplement features. In every department, Selenium stood out as the best tool for testing.

Paruchuri Ramya et al. [8] discussed about Selenium Web Drivers, which is advanced version of Selenium RC and its features. The authors have implemented and executed test cases in a Lawyer's Login Page with multiple test data. It can also include addition tools like Maven together with Selenium Web Driver. They have concluded that Selenium Web Driver is a very fast and easy tool for automating web application testing, and the results are more reliable. It encourages the use of additional resources for better results accuracy and usability.

Rigzin Angmo et al. [9] evaluated the performance analysis of the Selenium Suite. Execution speed is better in Selenium IDE as compared to Selenium Chrome Driver. Since Selenium Web Driver does not require any server, it is better than other selenium suite. They have also analyzed Watir-Web Driver and Selenium Web Driver using criterion such as execution speed, recording and playback, report generation, browser compatibility, platform compatibility, and language support and concluded that Selenium is more suited for general purpose in comparison to Watir-Web Driver, which is suitable under certain specific situation.

III. SELENIUM SUITE

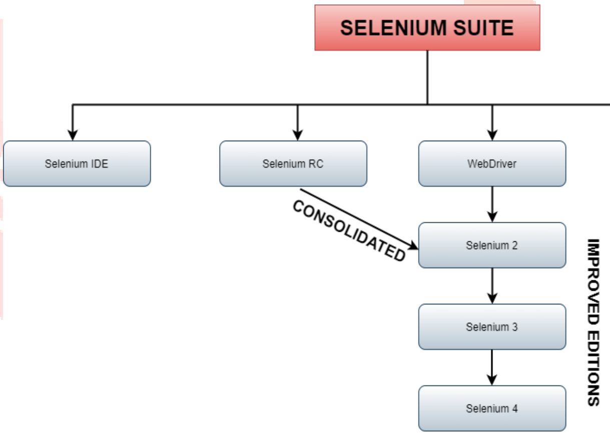


Figure1: Selenium Suite

Selenium was first released as an open source tool by Jason Huggins and Paul Hammant in 2004[8]. Selenium is an open source automation testing tool used to test web applications.

Selenium[10] Software is more than just a single tool; it's a set of tools, each of which caters to a specific requirements.

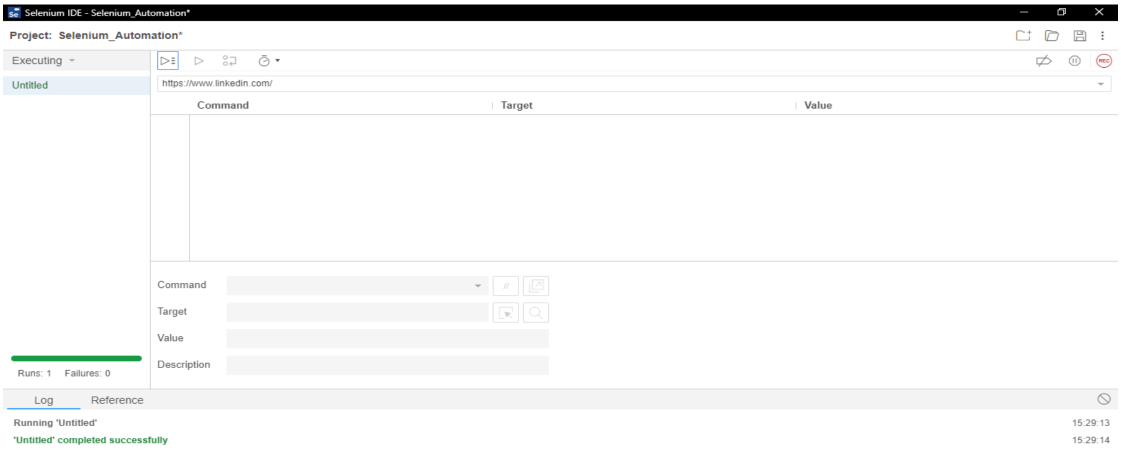
It comprises of 4 major components (fig 1):

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control
- Web Driver
- Selenium Grid

A. Selenium IDE

Selenium IDE (fig 2) is an integrated development environment for Selenium tests. It is implemented as a Firefox extension and Chrome Extension which allows us to edit, record, and playback test. It provides a way to save tests as HTML, Java, Ruby Scripts, etc [1][11].

Selenium IDE has impressive features. It has a record and play feature, as well as a user-friendly testing environment and some other useful features. The installation of an IDE is simple and convenient. To run the scripts, you only need a general understanding of programming. It is one of the primary resources in the Selenium Suite because of its simple user interface.



Running 'Untitled' 15:29:13

'Untitled' completed successfully 15:29:14

Figure2: Selenium IDE in Chrome extension.

B. Selenium Remote Control (RC)

Selenium Remote Control is a solution to cross browser testing [1]. Before Selenium Web Driver (Selenium 2.0), Selenium Remote Control (RC) was the main Selenium project that lasted for a long time. Selenium RC is no longer widely used because Web Driver has more powerful features.

Selenium RC (fig 3) is split into two parts, Selenium Server and Client libraries.

Selenium Server helps to start and stop browsers. Furthermore, it interprets and executes the Selenese instructions. It also serves as an HTTP proxy, accessing and validating HTTP messages sent between the browser and the test programme.

Client libraries function as a bridge between various programming languages (Java, C#, Perl, Python, and others)

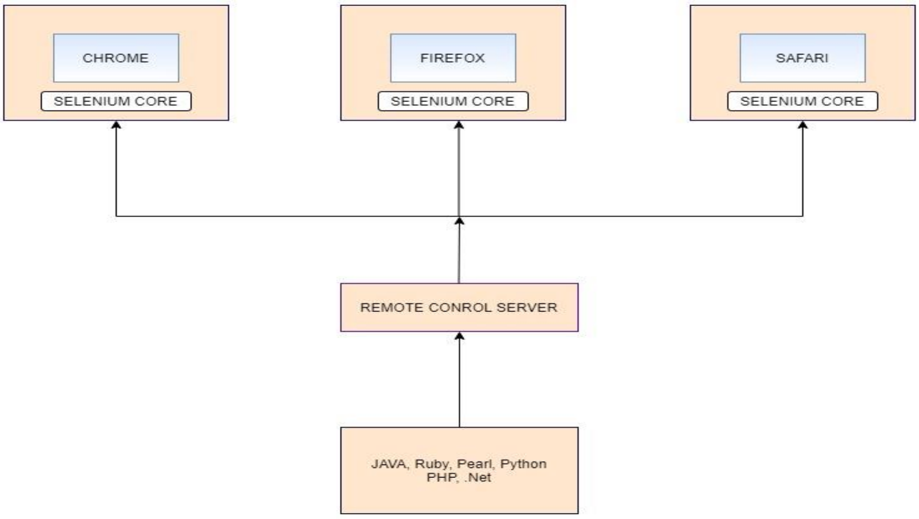


Figure3: Architecture of Selenium RC

C. Selenium Web Driver

Web Driver (fig 4) is a web application testing automation tool to verify the expected output. Selenium Web Driver is more advanced version of selenium RC which in addition to addressing the drawbacks of Selenium RC also adds new features. It is considered to be fastest among all the component of Selenium toolkit. For testing dynamic web pages, it is a better support than others. It can be implemented on browsers like Google Chrome, Firefox, Macintosh[12].

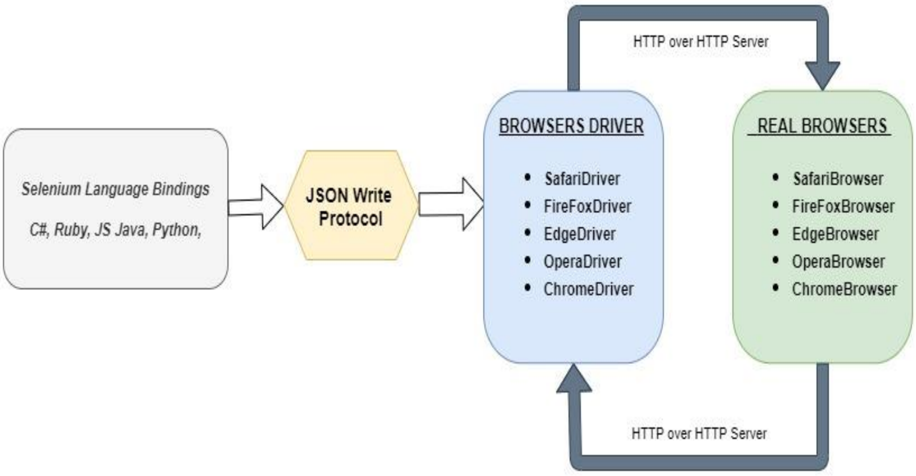


Figure4: Architecture of Selenium Web Driver

Difference between Selenium RC and Selenium Web Driver [12]:

Table 2. Difference between Selenium RC and Selenium Web Driver

SELENIUM RC	SELENIUM WEB DRIVER
Architecture is quite complicated	Architecture is comparatively simpler
Script execution in Selenium RC is slower as it uses JavaScript to interact with RC	Execution in Web Driver is faster as it directly interacts with browser.
Selenium RC's API (application programming interface) is small and simple	Selenium Web Driver's API is comparatively large and complex.
It is unable to test mobile application	It is able to test mobile application.
Selenium server act as a mediator in between browser and commands.	Web Driver directly interacts with the browser and controls it.

D. Selenium Grid

Selenium Grid (fig 5) allows running multiple tests simultaneously across different operating system, browsers and machines. The test is run in a central machine called 'hub' and parallel execution of test is conducted on a different machine called node.

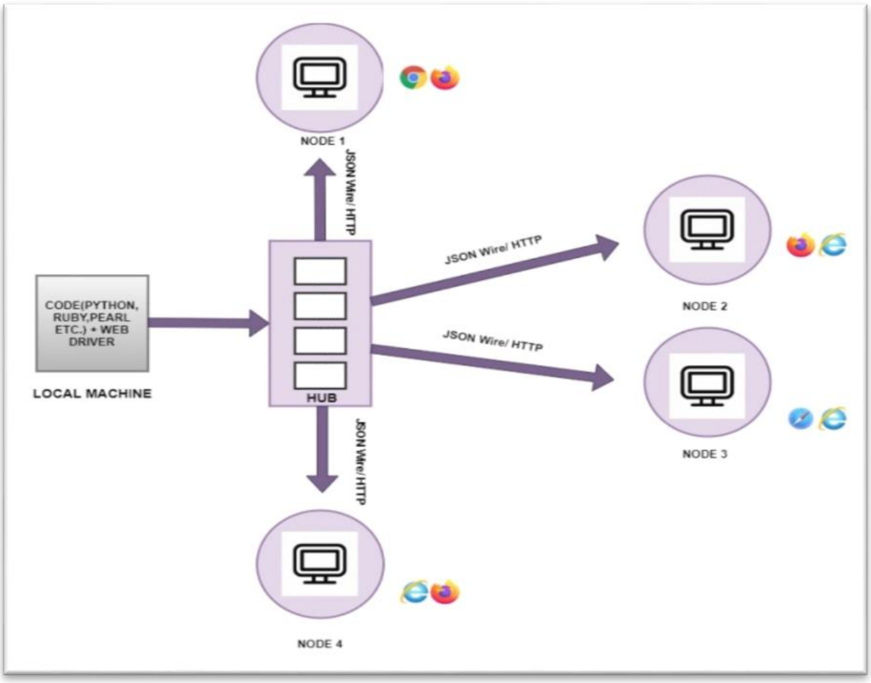


Figure5: Architecture of Selenium Grid

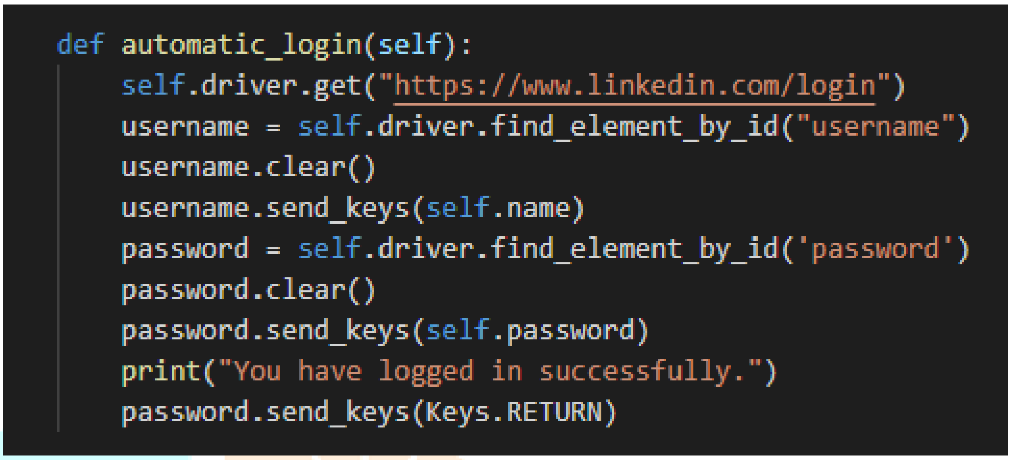
There are numerous benefits of using Selenium Grid. Since Selenium Grid supports concurrent test execution, it saves us a significant amount of time. It gives us the opportunity to run test cases in a variety of browsers. We can use multi-machine nodes to distribute and execute tests after we have created them. There are also some drawbacks in Selenium Grid. It raises the project's expense because it necessitates the use of additional machines as nodes. Furthermore, it is pretty stagnant. Each server is set up ahead of time to provide a subset of the appropriate browser instances. You must reconfigure if you want to change this which takes up a lot of additional time. Our study consist of understanding the need of automated testing tools over manual testing tools as former is more efficient and reduces time and cost factors. Any testing tool should be selected considering the cost factor and type of software to be tested. With related works in automation testing tools, we have assessed that Selenium testing is more efficient and has impressive features and expected to more in demand in near future as compared to its competitors. We also examined Selenium Suite to explore about its components and concluded that Selenium Web Driver is best suited for selenium testing due to its cross-browser synchronization and advanced interaction techniques.

With past studies, it has been concluded that Selenium Web Driver is considered to be the fastest among all its components in Selenium Suite [8].The main reason is that there is no involvement of Selenium Server. Commands given in the code gets translated into Web service methods (JSON Wired Protocol), and the Remote Driver accepts HTTP requests (commands) and executes them in the browser itself. This helps to reduce the execution time of Web Driver and improves the overall efficiency.

To explore more about Selenium Web Driver, we attempted to automate job applications on a Job Searching Portal, LinkedIn.

We executed the test script using following steps:

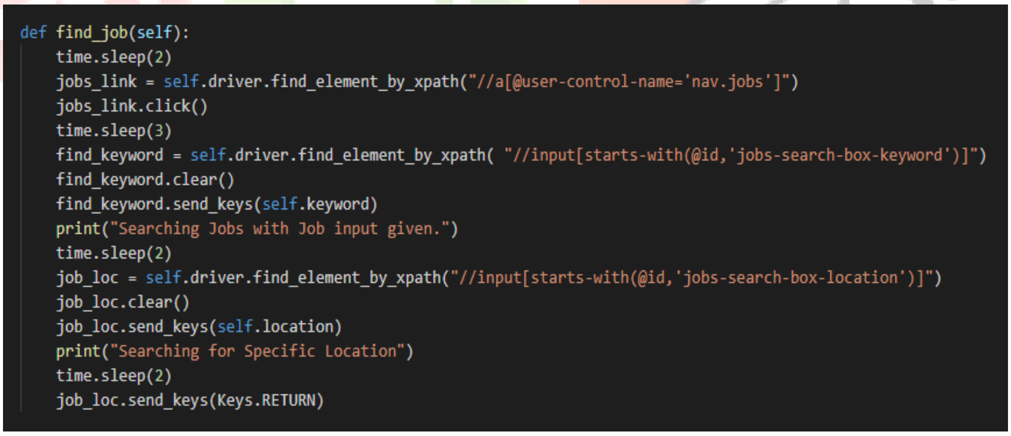
1. Automated Login by loading JSON File



```python
def automatic_login(self):
    self.driver.get("https://www.linkedin.com/login")
    username = self.driver.find_element_by_id("username")
    username.clear()
    username.send_keys(self.name)
    password = self.driver.find_element_by_id('password')
    password.clear()
    password.send_keys(self.password)
    print("You have logged in successfully.")
    password.send_keys(Keys.RETURN)
```

Figure6: Automated Login

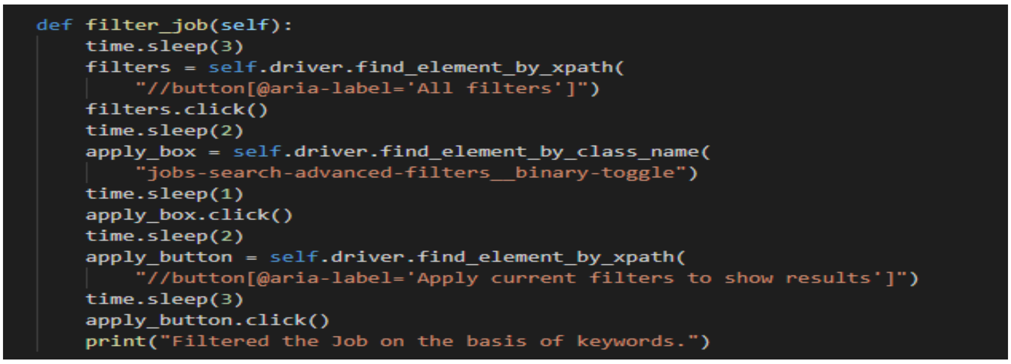
2. Finding Jobs



```python
def find_job(self):
    time.sleep(2)
    jobs_link = self.driver.find_element_by_xpath("//a[@user-control-name='nav.jobs']")
    jobs_link.click()
    time.sleep(3)
    find_keyword = self.driver.find_element_by_xpath( "//input[starts-with(@id,'jobs-search-box-keyword')]")
    find_keyword.clear()
    find_keyword.send_keys(self.keyword)
    print("Searching Jobs with Job input given.")
    time.sleep(2)
    job_loc = self.driver.find_element_by_xpath("//input[starts-with(@id,'jobs-search-box-location')]")
    job_loc.clear()
    job_loc.send_keys(self.location)
    print("Searching for Specific Location")
    time.sleep(2)
    job_loc.send_keys(Keys.RETURN)
```

Figure7: Finding Jobs

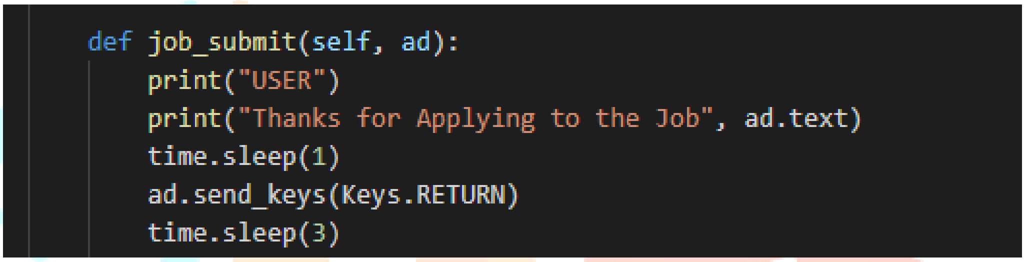
3. Filtering the jobs



```
def filter_job(self):
    time.sleep(3)
    filters = self.driver.find_element_by_xpath(
        "//button[@aria-label='All filters']")
    filters.click()
    time.sleep(2)
    apply_box = self.driver.find_element_by_class_name(
        "jobs-search-advanced-filters__binary-toggle")
    time.sleep(1)
    apply_box.click()
    time.sleep(2)
    apply_button = self.driver.find_element_by_xpath(
        "//button[@aria-label='Apply current filters to show results']")
    time.sleep(3)
    apply_button.click()
    print("Filtered the Job on the basis of keywords.")
```

Figure8: Filtering Jobs with Keywords

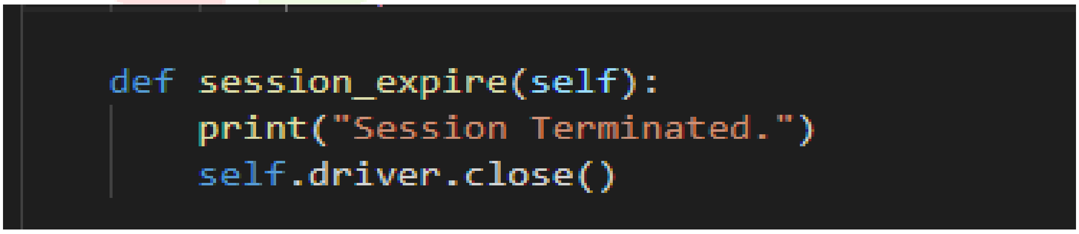
4. Applying to the job



```
def job_submit(self, ad):
    print("USER")
    print("Thanks for Applying to the Job", ad.text)
    time.sleep(1)
    ad.send_keys(Keys.RETURN)
    time.sleep(3)
```

Figure9: Applying Jobs

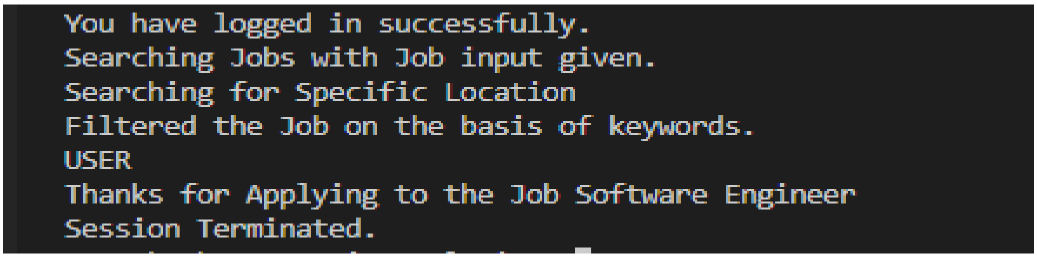
5. Session Termination



```
def session_expire(self):
    print("Session Terminated.")
    self.driver.close()
```

Figure10: Session Termination

6. Results



```
You have logged in successfully.
Searching Jobs with Job input given.
Searching for Specific Location
Filtered the Job on the basis of keywords.
USER
Thanks for Applying to the Job Software Engineer
Session Terminated.
```

Figure11: Results on Terminal

Methods Used:

- find_elements_by_xpath() : It is a method which uses XML path expression to navigate any feature on a web page
- find_elements_by_class_name(): It is used to locate the first element with the matching class name attribute.
- clear(): It is a specified method in Selenium which is used to empty any text box field.
- click(): It is used to click on any element in a web page like buttons, box, anchor tags etc.
- sleep(): It is used for waiting for a certain amount of time before executing other process. It is generally done to get the web page loaded completely.
- send_keys(): It is used to send text to any field in the form of input.
- close(): It is used to close the current browser and hence, terminate the process.

IV. RESULTS

Table 3. Analysis and Results

Average Time taken to (In sec) ->	Login	Search Jobs	Filter Location	Apply 1st Job	Apply 2nd Job	Apply More Jobs
1st Attempt	8.9	11.2	1.5	6.4	6.2	5.9
2nd Attempt	9.2	10.9	1.2	6.8	6.6	6.1
3rd Attempt	8.8	11.4	1.4	6.3	6.1	5.8
4th Attempt	9.1	11.7	1.6	6.8	6.5	6.0
5th Attempt	8.9	10.6	1.3	6.5	6.3	5.8
6th Attempt	8.7	10.8	1.5	6.4	6.1	5.6
Average	8.93	11.1	1.41	6.53	6.31	5.86

On executing test script multiple times, it was found that it took around 5.86 seconds to apply to a job(excluding time to login and search for a job) whereas same process if done manually took on an average 30.50 seconds(excluding time to login and search for a job) to be completed.

V. CONCLUSION

Our study consist of understanding the need of automated testing tools over manual testing tools as former is more efficient and reduces time and cost factors. Any testing tool should be selected considering the cost factor and type of software to be tested. With related works in automation testing tools, we have assessed that Selenium testing is more efficient and has impressive features and expected to more in demand in near future as compared to its competitors. We also examined Selenium Suite to explore about its components and concluded that Selenium Web Driver is best suited for selenium testing due to its cross-browser synchronization and advanced interaction techniques.

VI. REFERENCES

[1] Mahajan , P., Shedge, H., & Patkar, U. (2016). Automation Testing In Software Organization. *International Journal of Computer Application Technology and Research*, 5.

[2] https://www.guru99.com/difference-automated-vs-manual-testing.html

[3] Chaurasia, N., & Bahl, K. (2018). Analytical Study on Manual vs. Automation Testing Using Selenium. *IJISET-International Journal of Innovative Science, Engineering & Technology*, *5*(10)

[4] Mahajan, P., Shedge, H., & Patkar, U. (2016). Automation Testing In Software Organization. *International Journal of Computer Applications Technology and Research*, *4*, 198-201.

[5] Gogna, N. (2014). Study of browser based automated test tools watir and selenium. *International Journal of Information and Education Technology*, *4*(4), 336.

[6] Kumar, Y. (2015). Comparative study of automated testing tools: selenium, SoapUI, HP unified functional testing and test complete. *Journal of Emerging Tech-nologies and Innovative Research*, *2*(9), 42-48.

[7] Singh, I., & Tarika, B. (2014). Comparative analysis of open source automated software testing tools: Selenium, sikuli and watir. *International Journal of Information & Computation Technology*, *4*(15), 1507-1518.

[8] Ramya, P., Sindhura, V., & Sagar, P. V. (2017, February). Testing using selenium web driver. In *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (pp. 1-7). IEEE.

[9] Angmo, R., & Sharma, M. (2014, September). Performance evaluation of web based automation testing tools. In *2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence)* (pp. 731-735). IEEE.

[10] https://www.selenium.dev/documentation/en/selenium_installation/

[11] https://data-flair.training/blogs/category/selenium-tutorials/

[12] https://www.guru99.com/selenium-tutorial.html

[13] Malve, S., & Sharma, P. (2017). Investigation of Manual and Automation Testing using Assorted Approaches. *International Journal of Scientific Research in Computer Science and Engineering*, *5*(2), 81-87.

[14] https://www.guru99.com/automation-testing.html