

A Comparative Survey on Software Testing Tools

**Advaith Aditya Chevuturu, Divyendra Pratap Mathur, Byreddy Joseph Prasanth Kumar Reddy,
Charanya R.**



Abstract: Testing the software is an important part of the Software Development Life Cycle. In present day, the software industry's focus is on developing quality applications. In order to save time and costs when evaluating a program, most testers have moved from manual testing to automated testing. A large variety of software testing tools, open-source as well as commercial, are available in the market. It is quite a challenging task to select a testing tool that fits best to their software, as the choices keep getting wider. With more choices comes also the increased spectrum of software testing tool features with cost variations. Web application tools help the developers to test their software quite comfortably and are widely used today. With the integration of testing tools and web browsers, testing has become modular. When choosing an appropriate software testing tool, there are many factors which come into play to make this decision apt for a particular software to be tested. Testing tools are either automated or manual. The tool selection is done on parameters which best suit the tester. The purpose of this study is to identify and compare the popular testing tools and provide a review based on parameters which are suitable. This paper provides a comparative review of features of open source and commercial testing tools in a tabular format, based on these different parameters. It also lists down descriptions and features of various testing tools so that users and developers can opt for the appropriate tools based on their demands and requirements.

Keywords: Automation Testing, Manual Testing, Software Testing, Testing Framework, Testing Tools

I. INTRODUCTION

The advancement of software technologies has been radical over the last decade but, with the increase in the complexity of such softwares so comes the need of testing such softwares for bugs and logical errors to help facilitate better use of software for the user. This highlights the importance of software testing tools and the constant need to improve such tools. An estimate of around 30% of software development activities comprises software testing. Every software has its own functionalities to be tested at various stages throughout the testing process and hence, requires various software testing tools.

Manuscript received on 14 June 2022 | Revised Manuscript received on 20 June 2022 | Manuscript Accepted on 15 August 2022 | Manuscript published on 30 August 2022.

* Correspondence Author

Advaith Aditya Chevuturu*, School of Information Technology, Vellore Institute of Technology, Vellore (Tamil Nadu), India. E-mail: achevuturu@gmail.com

Divyendra Pratap Mathur, School of Information Technology, Vellore Institute of Technology, Vellore (Tamil Nadu), India. E-mail: divyendra.dpm@gmail.com

Byreddy Joseph Prasanth Kumar Reddy, School of Information Technology, Vellore Institute of Technology, Vellore (Tamil Nadu), India. E-mail: josephprasanth11@gmail.com

Dr. Charanya R, Assistant Professor, School of Information Technology, Vellore Institute of Technology, Vellore (Tamil Nadu), India. E-mail: charanya.r@vit.ac.in

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](#) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Software testing is an integral and vital part of the software development life cycle (SDLC) and includes various different types of testing. Testing can be broadly divided into two main categories of testing namely, Manual testing and Automation Testing. Manual testing is where the software is tested manually and is slower and a more comprehensive approach but can handle complex test cases better. Automation testing executes a test script covering many permutations for testing and is a faster approach but code has to be maintained. Both have their own merits and demerits and their usage depends upon the tester. With growing needs, the need for Automation testing has also increased due to its ability to test an entire suite of test cases in a more time efficient manner. Testing can be further classified into Functional testing and Non-Functional testing. Functional testing tests the important functions for a particular product and it includes Unit testing, Integration Testing, Regression Testing, Smoke testing, Critical path testing and Extended testing. Non-Functional testing does not involve testing core functions of the product but rather how the system operates, mainly the performance, reliability, usability and security. This type of testing includes Performance testing which further includes Load testing, stress testing, endurance testing and spike testing. Non-Functional testing also includes UI testing, configuration testing and security testing. With such varied tests, it is necessary to choose software testing tools which are efficient and cost effective, preferably containing a suite of different tools for different types of testing. Some of the popular automated testing tools such as Selenium, Cypress and WinRunner are used to execute a test case suite for various testing methodologies. Some other testing tools such as LoadRunner and Jmeter are used for performance testing of web applications [2]. This paper aims to provide a comparison between different automation testing tools for web applications and analyse data to improve software quality.

II. LITERATURE REVIEW

As new softwares are being developed, so are the tools to test them and one of the leading tools for web application testing is Selenium. Selenium is a highly versatile automation test tool that supports almost all programming languages and features a suite of testing tools and IDE's [12]. Before identifying the tool, it has to be clear as to the requirement of the tool and the function it has to perform. Hence before approaching any tool, Selection has to be done depending upon factors such as identifying the test required, whether manual testing or automated testing and the importance of testing[1]. There are a number of software testing tools which can be used for various purposes ranging from unit testing to integration testing and end-to-end testing.



Published By:

Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Journal Website: www.ijeat.org

© Copyright: All rights reserved.

A Comparative Survey on Software Testing Tools

Every application has its own unique test case suite depending upon which an appropriate tool is to be selected [11]. Some of the other popular tools in this respective space are LoadRunner, JUnit, Cypress, UFT etc. each used for a different purpose, with UFT, LoadRunner and JUnit being contenders for performance testing of web applications [4]. Each of these tools have advantages and disadvantages depending upon their features and usability. According to [6], a performance test of web application tools Selenium and Katalon proved that each tool has its own properties which make it more of a user's choice and their preferences with respect to the tool. The preference of open source tools has increased because of their cost effectiveness and also that they have been performing better when compared to proprietary tools[5], which gives tools like Selenium the upper hand when compared to tools like UFT. Various Testing tools have been compared and contrasted to provide better insight as to what the user needs and that a suitable test strategy is required when approaching any testing activity [8]. Automation of testing has been very important for many industries and amongst automation tools, Selenium has been used quite popularly [14]. Although there are many tools available, we compared some of the most popular tools used and provided a survey on the tool and its properties. Here, a brief description of the different software testing tools is given. Their features, advantages and disadvantages are mentioned in the following Table1 below: LoadRunner [4][7-9][13][15][16][17][20] is a software testing tool developed by Mercury which was later acquired by Hewlett Packard in 2006 and later by Micro Focus in 2016. It is used to test applications, to measure system behaviour and performance under load. JUnit [1][4] is a test unit for the Java programming language unit. JUnit has been instrumental in the development of experimental development, and is one of the family of joint unit testing units known as xUnit, founded by JUnit. Cypress [6][21][22], developed by Cypress.io, is a JavaScript-based frontend testing tool. It is built in keeping with the changing needs of the modern web.

QTP (Quick test professional) renamed as UFT (Unified

Functional Testing) [1][4-8][10][13][14][19-21] is an automated testing tool that helps testers perform automatic debugging tests to detect any errors, defects or gaps as opposed to the expected results of the application under test. It was designed by Mercury Interactive and later acquired by Hewlett Packard and now Micro Focus. Selenium [1][3][5-10][12-14][18-20] is an open-source web browser automation tool. One of the most widely used automated testing frameworks is Selenium. Selenium is a framework for supporting and encouraging automated testing of web-based applications across a variety of browsers and platforms. Selenium IDE, Selenium RC, WebDriver, and Selenium Grid are the four main components of the Selenium suite. Rapise [21][22] is a powerful script less test automation platform for online, mobile, and desktop apps, as well as APIs. For automated acceptance and regression testing, Serenity BDD (Behaviour-driven development) [21][22] is a selenium alternative. It's one of the best Selenium alternatives for generating test reports that document and describe functional test coverage. Galen [21][22] was originally designed to test the layout of web applications in a real browser. It has since evolved into a fully functional testing framework. For visual and layout testing of online applications, this automation test tool can be integrated with Selenium. Avo assure [19][20] is a codeless automation testing tool, It enables the user to test end to end business applications easily. One can seamlessly perform testing across web applications, desktops, associated emulators, mobile, etc. In Kobitan [19], development and testing teams can automate performance, compatibility, functional and visual testing across IoT and real mobile devices. It is a mobile testing platform that accelerates delivery and testing of mobile apps. It offers manual and automated testing on real devices, in the cloud or on premise. For organisations that use software for operations and development, Zaptest [20][21] is a Software Automations solution. Organisations can also automate their back-office operations to develop a seamless and effective automation framework.

Table-I: Software testing tools

Tool	Features	Advantages	Disadvantages
Load Runner [4][7-9][13][15][20]	<ul style="list-style-type: none"> 1. Can simulate thousands of users simultaneously using program software, record and later analyse the performance of key application components. 2. Widely used Load test tool. 3. LoadRunner Product Performance Test Results are used as a benchmark against other tools. 4. Simplify testing with a project-based testing solution that supports a wide range of technologies and protocols in the industry. 	<ul style="list-style-type: none"> 1. There is no need to install it on the server under test. Uses native monitors 2. Excellent lessons, complete documentation and practical tool support from HP/Micro Focus 3. Excellent visual monitoring and analysis interface where you can see reports on colour charts and easily recognizable images. 	<ul style="list-style-type: none"> 1. The tool works with a paid licence which could be costly 2. Some level of experience is required to operate the tool
JUnit [1][4]	JUnit is a simple framework for writing repetitive tests.	<ul style="list-style-type: none"> 1. The tool is open source and free to use 2. The tool is easy to learn and work with 	<ul style="list-style-type: none"> 1. Dependency testing is not supported 2. Only Java programming language is supported

Cypress [6][21][22]	<ol style="list-style-type: none"> 1. Cypress is mainly used for frontend integration and unit testing. 2. The flexible Cypress method allows frontend developers to build their own unique test cases. 3. Cypress is a tool designed to perform the development and evaluation of compatible processes. This goal is supported by the ability of the Test-Driven Development (TDD) tool for complete end-to-end testing. 	<ol style="list-style-type: none"> 1. Browser support is present for most of the popular browsers like Chrome, Edge and Firefox 2. Easy to set-up and operate 3. Support for network traffic control which allows for edge case scenario tests without server involvement 	<ol style="list-style-type: none"> 1. This tool is limited to single instances and restricts you to use multiple sessions of browsers. 2. Does not support Safari browser for Mac and iOS users 3. Javascript and other dependencies need to be known and installed
QTP/UFT [1][4-8][10][13][14][19-21]	<ol style="list-style-type: none"> 1. It is an icon-based tool that automatically activates the operation and active testing of the application. 2. Both devices, as well as the non-technical tester, can use Micro Focus QTP. 3. Provides features such as Recording and Playing. 4. We can explore Desktop and web-based applications through this tool. 	<ol style="list-style-type: none"> 1. Supports more than 200 apps and environments and provides test automation for web and mobile applications 2. Easy to learn and understand the tool and can be used for any client server application 	<ol style="list-style-type: none"> 1. The tool is licenced which can incur costs 2. Only support tests in VB script
Selenium [1][3][5-10][12-14][18-20]	<ol style="list-style-type: none"> 1. Selenium IDE is a Firefox extension that allows you to record and replay web application tests. 2. WebDriver automates by communicating directly with the web browser and taking advantage of its inherent compatibility. 3. It provides a single interface for writing test scripts in a variety of programming languages, including Ruby, Java, NodeJS, PHP, Perl, Python, and C#. 	<ol style="list-style-type: none"> 1. Selenium is a simple web application functional testing framework that is easy to get started with. 2. It allows you to record and playback web-based application tests. Selenium enables multithreading, which means that the same script can be run many times on separate browsers. 	<ol style="list-style-type: none"> 1. Selenium necessitates a large amount of skill. In addition, the resource should be well-versed in framework architecture. 2. It only supports web-based applications and does not support applications that run on Windows. Also Built-in add-ins are not supported by Selenium.
Rapise [21][22]	<ol style="list-style-type: none"> 1. Record and replay in any browser, with real-time confirmation during the recording process. 2. With its object-based approach, you may create and refine tests with drag and drop. Rapise's robust maintenance features and self-healing AI-driven locators ensure that objects update in sync. 3. RVL is an easy-to-use visual, keyword-driven framework for editing recorded tests. 4. Rapise supports data-driven testing out of the box and interfaces with third-party CI/CD/ALM systems as well as sophisticated applications (MS Dynamics, Salesforce.com, SAP). 5. Rapise is based on a JavaScript engine and uses open-source standards (Selenium, Appium) 	<ol style="list-style-type: none"> 1. Automation tasks are much easier. Lot of features support the desktop application's automation. 2. The customer support is excellent. 3. No tool offers this much functionality and flexibility for less cost. 4. Automation of test tasks increases productivity and significantly reduces test times 	<ol style="list-style-type: none"> 1. The tool is licenced. Integration with some tools (Dynamics) is only with the basic functions.
Serenity [21][22]	<ol style="list-style-type: none"> 1. It's one of the greatest best Selenium replacements because it includes comments/narrative and screenshots for each test step. 2. Test results aggregated by Requirements or Release. 3. This selenium replacement tool aids in the creation of more readable and maintainable automation code. 4. Relate your automated tests to your specifications. 5. Provides Test Coverage. 	<ol style="list-style-type: none"> 1. Serenity allows business users, developers, and testers to work together effectively. It employs Domain-Specific Language (DSL) and also allows you to establish requirements and accept tests in reasonable chunks. 2. It facilitates the rapid construction of functional features. It enables you to focus test runs on certain functional areas. 	<ol style="list-style-type: none"> 1. Creating and maintaining feature files takes time. Writing feature files necessitates a high level of communication. 2. The BDD documents should be kept up to date. More time is required to write automation code.

A Comparative Survey on Software Testing Tools

Galen Framework [21][22]	<ul style="list-style-type: none"> 1. Error reporting using HTML with screenshots 2. Syntax is simple to write and read. 3. It's one of the best Selenium alternatives because it supports multiple browsers. 4. You may specify any complex layout using Galen specifications language, including diverse screen widths and browsers. 5. Selenium Grid works well with Galen Framework. This enables the creation of cloud-based tests, such as Sauce Labs or the BrowserStack. 	<ul style="list-style-type: none"> 1. Galen is an excellent framework for automating layout testing. The Galen Spec Language is a human-readable language with two syntactic versions (basic and advanced). 2. Tests can be written in a variety of languages (Galen Test Suite Syntax, JavaScript, and Java API), reports in HTML format are very detailed, and the documentation is excellent. 	<ul style="list-style-type: none"> 1. Galen doesn't put everything to the test. Doesn't put what the user sees to the test. 2. The Galen Framework is difficult to utilise when validating huge websites. 3. Galen Framework isn't a good tool for visual testing.
Avo Assure [19][20]	<ul style="list-style-type: none"> 1. We can perform testing, create and execute test cases without the need of writing any code. 2. It helps us to achieve End to End test automation. 3. Using the Mindmap feature, we can define and design the test cases and test plans. 4. Can be integrated with tools like Salesforce, Sauce Labs, Jenkins, etc. 5. Multiple scenarios can be executed using smart scheduling. 	<ul style="list-style-type: none"> 1. Achieve upwards of 2x productivity since you can do more in less time and with less effort. 2. Test applications 85% faster than manual testing and double your application release time. 3. Know exactly where you are in your test automation journey through the Mindmaps feature. 	<ul style="list-style-type: none"> 1. Does not support requirement based testing and security testing. 2. Does not support desktop based Linux and Chromebook.
Kobiton [19]	<ul style="list-style-type: none"> 1. Automated Functional, Compatibility, Visual and Performance testing can be scripted or script less. 2. Solutions provided on the premises and there is unlimited user policy. 3. We can access real devices in Private as well as Public Clouds. 4. With every script less test, a 100% open standard Appium code is generated. 5. For rapid debugging, users have access to real devices within their IDE. 6. For Functional and Visual issues, an AI assisted remediation is provided. 	<ul style="list-style-type: none"> 1. This app assists us in testing, diagnosing, and resolving device/OS issues. It also allows us to collect screenshots on a variety of different screen sizes and devices. 2. Easy to use and useful scripting to integrate on several programming languages. 	<ul style="list-style-type: none"> 1. Without going via support, couldn't get access to Beta versions. When it comes to managing devices and groups, the UI is a little disjointed. 2. Some public cloud devices will occasionally be unfit (no memory, no network, etc.) and will fail the tests at random.
ZapTest [20][21]	<ul style="list-style-type: none"> 1. We can perform API Testing and DevOps Automation. 2. Automation for Functional and Performance Tests. 3. We have cross platform executions and Auto Documentation. 4. No API dependency 	<ul style="list-style-type: none"> 1. Agile: Flexible reaction to any application change, as well as ongoing test development with the R&D team. 2. Cross Platform Compatibility: the ZAP Objective Engine is non-intrusive, ZAPTEST can run alongside any program. 	<ul style="list-style-type: none"> 1. Free Trial is not available for the users. 2. No deployment for Desktop based Linux and Chromebook, and On-Premise Windows and Linux.

III. PARAMETERS FOR COMPARISON

To study the different tools discussed, we need some parameters to identify the features of each tool and how that tool is different from another. According to [13], the key points to consider when selecting a tool are cross platform support for flexibility, ease of use, user experience, function and cost. Considering such parameters can make it more viable to choose one tool over the other depending upon the user requirement and expertise.

Table-II: Parameters for tools

Parameters	Definition
Platforms	Operating systems supported
Browsers	Web browsers supported
Ease of Learning	How easy is the tool to learn
Programming Skills	Required prerequisite programming skills

Published By:

Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

© Copyright: All rights reserved.

Retrieval Number: 100.I/ijeat.F36640811622

DOI: [10.35940/ijeat.F3664.0811622](https://doi.org/10.35940/ijeat.F3664.0811622)

Journal Website: www.ijeat.org



Report Generation	How the result is generated	
Cost	Whether the tool is free, licensed or limited	
Function	Types of testing supported by the tool	

IV. COMPARATIVE REVIEW OF TOOLS

Table-III: Comparative review of software testing tools

	Selenium[1][3][5-10][12-14][18-20]	LoadRunner [4][7-9][13][15][20]	JUnit [1][4]	Cypress [6][21][22]	QTP/UFT [1][4-8][10][13][14][19-21]	Rapise [21][22]	Serenity [21][22]	Galen [21][22]	Avo Assure [19][20]	Kobiton [19]	ZapTest [20][21]
Developer	Jason Huggins	Micro Focus	Kent Beck, Erich Gamma, David Saff, Kris Vasudevan	Cypress.io	Micro Focus	Inflectra	John Ferguson Smart	Ivan Shubin	SLK Software Services	Vu Lam	Alex Chernyak
Platforms	Windows, Mac OS X, Linux.	Windows, Linux	Windows, Linux, Mac	Windows, Linux, Mac	Windows, Linux, Mac	Windows, Java, Oracle, Microsoft Dynamics, QT Framework	Linux, Windows, Mac.	Linux, Windows, Mac.	Linux, Windows, Mac	Linux, Windows, Mac	Linux, Windows, Mac, Android
Browsers	IE, Chrome, Firefox, Opera, Edge	IE, Chrome, Firefox, Opera, Edge	Chrome, Opera, Firefox, Edge, IE, Safari	Chrome, Edge, Firefox	Chrome, Firefox, Edge, IE, Safari	IE, Mozilla Firefox and Google Chrome.	Chrome, Firefox, Safari.	IE 11, Firefox, Chrome, Opera and Safari	Chrome, Firefox, Edge, IE	Chrome, Firefox, Edge, IE	Chrome, Firefox, Edge, IE, Safari
Ease of Learning	Challenging	Experience needed	Relatively easy to learn	Easy to learn	Easy to learn	Easy to learn	Easy to learn	Easy to learn	Easy to learn	Easy to learn	Moderately easy
Programming Skills	Definite programming languages (Java, C/C++, Ruby, Python, Perl) required.	Partial (script can be complex and difficult to understand)	Partial, Basic programming skills in Java required	Partial, Easy to navigate but Javascript skills required	Partial Quite easy to edit, navigate, parametrize	Easy extensibility using JS	Programming skills in Java required.	Programming skills related to JS and Java are required.	No programming skills required	Programming skills required	Programming experience required
Report generation	Html report of current execution which contains information like an error, groups, time, reporter logs, testing XML files.	Provides graphical representation of results through LoadRunner Analysis	XML-Gives summary of test, errors, success rate etc. in the form of values	Html-Report generation using Mocha	Html Xml - gives executive summary of test, gives statistics in the form of pie charts	HTML reports can be exported which contains full execution results and images are saved as separate files.	Test outcomes by default are generated in XML and JSON format, both containing the same information ; test case, suites, test results, rest queries.	It generates Html reports where you can see all your test objects on the page.	Detailed and automatic-generated reports	Interactive report generation	PDF and HTML reports generated.
Cost	Open Source	Licensed	Open Source	Freeware/Licensed	Licensed	Licensed	Open Source	Licensed	Free for use case of user's choice	Paid, free trial available	Free version available

A Comparative Survey on Software Testing Tools

Function	Performs functional, regression, load testing on web applications	Load testing and performance testing	Unit testing, regression testing, web applications	Web testing, end-to-end testing, not a unit testing framework	Web testing -regression, unit, distributed, manual	Test web applications , Create one test script and execute the same script without modification across the browsers	Helps you write cleaner and more maintainable automated acceptance and regression tests faster.	Test framework for testing layout of web applications but is also a great tool for functional testing	heterogeneous test automation solution that helps you test applications	cloud-based platform enables the execution of manual and automated mobile and web testing	automating testing at the very beginning of the software development lifecycle
-----------------	---	--------------------------------------	--	---	--	---	---	---	---	---	--

V. RESULTS AND CONCLUSION

In the software industry, automation testing is preferred to improve efficiency and productivity. As automated software testing has become a necessity for companies, we can choose which automated testing tool can be used for a certain type of testing purpose. The testing tool should be installed easily and quickly, and it should support both users with no programming skills, and those with good programming skills. The tool should support integration with other frameworks and reporting tools, to make it easy to understand the cause of the failure. Testing tools have their advantages and disadvantages. Choosing an unsuitable tool can incur heavy costs and loss of time which can be problematic to the testers and the company. Out of all the tools discussed here, Selenium proves to be better with adaptability and cross compatibility and has a host of features but to choose a perfect testing tool as per their requirements, the testers or developers have to deeply analyse various tools. In the comparative study presented, the developer can decide upon which tools they need to consider for their applications.

REFERENCES

1. (Critical Analysis of Software Testing Techniques and Automation Testing Tools - IJSER Journal Publication)
2. "Performance Testing: A Comparative Study and Analysis of Web Service Testing Tools." International Journal of Recent Trends in Engineering and Research, vol. 4, no. 3, Mar. 2018, pp. 95–100. [CrossRef]
3. Satheesh, Arjun, et al. "Comparative Study of Open Source Automated Web Testing Tools: Selenium and Sahi." Indian Journal of Science and Technology, vol. 10, no. 13, Apr. 2017, pp. 1–9. [CrossRef]
4. "Performance Testing Tools: A Comparative Study of QTP, Load Runner, Win Runner and JUnit." Journal of University of Shanghai for Science and Technology, 15 Dec. 2020.
5. Abhishek V, Naveen A, Vidya G, Dr.Jasmine K.S, 2014, Comparative study on Software testing tools used for GUI testing, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCSE – 2014 (Volume 2 – Issue 02),
6. E. Pelivani and B. Cico, "A comparative study of automation testing tools for web applications," 2021 10th Mediterranean Conference on Embedded Computing (MECO), 2021, pp. 1-6, doi: 10.1109/MECO52532.2021.9460242. [CrossRef]
7. R. K. Lenka, U. Satapathy and M. Dey, "Comparative Analysis on Automated Testing of Web-based Application," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2018, pp. 408-413, doi: 10.1109/ICACCCN.2018.8748374. [CrossRef]
8. Sundareswaran, Veena & Aarthi, S & Aishwarya, K & Arunadevi, P & Student. (2019). A Survey on Software Testing Tools for Web Applications.
9. Shuaibu, Isiaka & Mustapha, Musa & Ibrahim, Muazzamu. (2019). Investigation onto the Software Testing Techniques and Tools: An Evaluation and Comparative Analysis. International Journal of Computer Applications. 177. 24-30. 10.5120/ijca2019919685. [CrossRef]
10. Kaur, H. and Gupta, G., 2013. Comparative study of automated testing tools: selenium, quick test professional and testcomplete. Int. Journal of Engineering Research and Applications, 3(5), pp.1739-1743.
11. Hussain, Sharqi & Wang, Zhaoshun & Toure, Ibrahima & Diop, Abdoulaye. (2013). Web Service Testing Tools: A Comparative Study. International Journal of Computer Science Issues. 10. 641-647.
12. B. Majeed, S. K. Toor, K. Majeed and M. N. A. Chaudhary, "Comparative Study of Open Source Automation Testing Tools: Selenium, Katalon Studio & Test Project," 2021 International Conference on Innovative Computing (ICIC), 2021, pp. 1-6, doi: 10.1109/ICIC53490.2021.9693066. [CrossRef]
13. Gamido, Heidilyn & Gamido, Marlon. (2019). Comparative Review of the Features of Automated Software Testing Tools. International Journal of Electrical and Computer Engineering. 9. 4473-4478. 10.11591/ijece.v9i5.pp4473-4478. [CrossRef]
14. A. Sivaji et al., "Software Testing Automation: A Comparative Study on Productivity Rate of Open Source Automated Software Testing Tools For Smart Manufacturing," 2020 IEEE Conference on Open Systems (ICOS), 2020, pp. 7-12, doi: 10.1109/ICOS50156.2020.9293650. [CrossRef]
15. R. Abbas, Z. Sultan and S. N. Bhatti, "Comparative analysis of automated load testing tools: Apache JMeter, Microsoft Visual Studio (TFS), LoadRunner, Siege," 2017 International Conference on Communication Technologies (ComTech), 2017, pp. 39-44, doi: 10.1109/COMTECH.2017.8065747. [CrossRef]
16. C. -H. Hsieh et al., "Evaluation System for Software Testing Tools in Complex Data Environment," 2021 4th International Conference on Information Communication and Signal Processing (ICICSP), 2021, pp. 604-609, doi: 10.1109/ICICSP54369.2021.9611846. [CrossRef]
17. D. Saharan, Y. Kumar and R. Rishi, "Analytical Study and Implementation of Web Performance Testing Tools," 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIECE), 2018, pp. 2370-2377, doi: 10.1109/ICRIECE44171.2018.9008408. [CrossRef]
18. S. M. Shariff, H. Li, C. Bezemer, A. E. Hassan, T. H. D. Nguyen and P. Flora, "Improving the Testing Efficiency of Selenium-Based Load Tests," 2019 IEEE/ACM 14th International Workshop on Automation of Software Test (AST), 2019, pp. 14-20, doi: 10.1109/AST.2019.00008. [CrossRef]
19. Automation Testing Tools - <https://www.guru99.com/automation-testing.html>
20. Web Application Testing Tools In 2022 (Comprehensive List) <https://www.softwaretestinghelp.com/most-popular-web-application-testing-tools/>
21. 20 BEST Selenium Alternatives in 2022 - <https://www.guru99.com/selenium-alternatives.html>
22. Best Selenium Alternatives (Free and Paid) in 2022 - <https://www.softwaretestingmaterial.com/best-selenium-alternatives/>

AUTHORS PROFILE



Advaith Aditya Chevuturu, is a third-year undergraduate at Vellore Institute of Technology specializing in Information Technology. He has a keen interest in software and cloud related technologies and is an active part of the community. He has worked on projects and is looking forward to explore more on cloud-based technologies. achevuturu@gmail.com





Divyendra Pratap Mathur, is a third-year undergraduate at Vellore Institute of Technology specializing in Information Technology. He is interested in web development and related technologies and was part of the DBQC VIT quiz club. He has worked on many projects and is an active part of the software community.

divyendra.dpm@gmail.com



Byreddy Joseph Prasanth Kumar Reddy, is a third-year undergraduate at Vellore Institute of Technology specializing in Information Technology. He is passionate to learn new things related to software and technology and has worked on many projects in his course. He is a machine learning enthusiast who aspires for improvement. josephprasanth11@gmail.com



Dr. Charanya R, is an assistant professor at the school of information technology in Vellore Institute of Technology. She is experienced in the field of software testing and is an active member of the research community. Her expertise in the area of software testing has helped and inspired this paper.

charanya.r@vit.ac.in