

## A Comprehensive Overview of Web-Based Automated Testing Tools

Ruya SAMLI\*<sup>1</sup> , Zeynep ORMAN<sup>2</sup> 

<sup>1,2</sup>Istanbul University-Cerrahpasa, Engineering Faculty, Computer Engineering Department, Istanbul, 34320, Turkey

Research Article, Received Date: 18.05.2023, Accepted Date: 14.06.2023

### Abstract

In the software testing process, automation testing is generally required for obtaining better test results and saving time and money. Automation testing is beneficial, especially when web applications are being tested according to performance, vulnerability, security, load, and stress. Today, there are many web-based automated tools with various properties, which are developed by different producers. Some of them can only be used for specific purposes and produce pretty good results, while others can be used in a broader area but produce average results. Web-based automated tools differ from each other according to various criteria such as cost, license, technical support, language support, user experience, documentation, browser support, environment support, testing type, and hardware requirements. It is often hard to determine which web-based automated tool is appropriate for a specific testing process. Since many of these tools are licensed and costly, it is sometimes impossible to make trials and decide on the proper automated tool. Some studies in the literature address this problem by examining only a few tools and only a few comparison criteria. However, the comparison of automated tools is an important issue that needs to be addressed in more detail. In this paper, for the first time in the literature, 14 web-based automated tools are compared according to 20 different criteria, and the results of this comprehensive review are presented.

**Keywords:** Software testing, Software tool, Web application testing, Automated testing tool.

### 1. INTRODUCTION

Software testing has become a popular concept in the last decade because of the tremendous growth in software applications. The primary motivation of software testing is to prove that the software is working as required by customers, programmers and users, and always producing the correct results. It is also known that risk management is a critical concept in software engineering (Masso et al., 2020), and the testing process can be used to determine the risks in the software. Testing process includes test plan, test environment, documentation, reporting and test cases/scenarios. Software tests can be grouped according to the tests' content such as database testing, mobile application

testing, interface testing, and web application testing. Apart from these, software tests can be categorized into several groups according to different criteria. Some of the categorization is as follows. According to their primary purposes: functional testing, non-functional testing; according to the testing times: active testing, passive testing, according to the levels: unit testing, integration testing, system testing, acceptance testing (Elbaum et al., 2003; Bertolino, 2007; Kundu, 2012; Hedayati et al., 2015; Falah et al., 2015) according to testers' knowledge of the system: the black-box testing, the white-box testing (Monier & El-mahdy, 2015). All software testing processes about all these testing styles can be performed by two different methods: (1) by human beings (manual testing), and

\*Corresponding author ruyasamli@iuc.edu.tr, <sup>2</sup>ormanz@iuc.edu.tr

(2) by machines/other software (automated testing). Automated testing includes many different abilities about testing and it is faster than manual testing (Catelani et al., 2011; Gupta et al, 2015). Because of the unbelievable growth of the softwares, automated testing is often preferred over manual testing in recent years.

This paper discusses web application testing and web-based automated system testing explicitly. In web application testing, web security, which is the site's ability to handle the traffic and the site's regular users, is also checked (Petukhov & Kozlov, 2008).

This paper examines web-based automated testing tools. The main aim is to assist the software testers in finding out the best web-based automated testing tools, thus efficiently and effectively performing web-based testing. There are many factors to be considered to refer to the very best tools, such as browser compatibility, technical support, and integration. Some studies in the literature have already analyzed some of the software testing tools. Therefore, a review is given below.

**Table 1.** A literature review of web-based automation tools

Study	Tools	Comparison Features	Major findings
(Križanić et al., 2010)	Grinder, JMeter, NeoLoad, WebLoad	Security support, Test results report	Although commercial tools offer a richer set of features and are, in general, easier to use, available open-source tools proved to be quite sufficient to perform given tasks successfully.
(Gogna & Kumari, 2011)	Watir, WET	Developer(s), Stable release, Language, The platform, Testing type, License	They have different limitations, so both can be better than the other in different situations.
(Kaur & Kumari, 2011)	QTP, TestComplete	Data-driven testing, Test result reports, Reusability, Execution speed, Record and playback, Recording efficiency, User experience, Cost	TestComplete is better used for applications with low-security requirements, and QTP is better where data security is required even during testing.
(Chandel et al., 2013)	JMeter, Loadrunner	Performance, Speed, Throughput, Efficiency	They have different limitations, so both can be better than the other in different situations.
(Kaur & Gupta, 2013)	QTP, Selenium, Test complete	Usability, Efficiency	These tools can be selected based on the type of application, the budget, and the efficiency required.
(Al-Zain et al., 2013)	Sahi, Selenium, TestComplete, VS Web Test	Record and playback, Handling of page waits, Browser support, Technical support	It has been shown that free and straightforward automation tools can be much more potent than commercially sophisticated and expensive tools in many situations.
(Dukes et al., 2013)	Paros, WebScarab, JbroFuzz, Acunetix, Fortify	Functionality, Cost, Platform support, Language support	Open-source tools can conduct security testing with limited results.
(Gogna, 2014)	Selenium, Watir	Functionality	They have different limitations, so both can be better than the other in different situations.
(Singh & Tarika, 2014)	Selenium, Skilu, Watir	Data-driven testing, Efficiency, Language support, Reusability of code	They have different limitations, so all can be better than others in different situations.
(Dubey & Shiwani, 2014)	Ranorex, TestComplete	Cost, Efficiency, Capability, Data-driven testing, Test result reports, Reusability, Record and playback, User experience	They have different limitations, so both can be better than the other in different situations.
(Daud et al., 2014)	Acunetix, Nessus	Vulnerability, Cost, Software support, Efficiency	Vulnerability scanning is a useful feature for organizations to prefer any of these tools.
(Sharmila & Ramadevi, 2014)	JMeter, WAPT	Average response/sec, Successful sessions, Bandwidth, CPU utilization, Total throughput/sec, Total KBytes sent, Standard deviation, Median	JMeter has beneficial and efficient bandwidth for testing web applications.
(Bharti & Dutt, 2014)	QTP, Selenium, Test Complete	Ease of operation, Usability, Area of application, Efficiency	These tools can be selected based on the type of application, the budget, and the efficiency required.
(Bindal & Gupta, 2014)	QTP, Selenium, Test Complete, SOAP UI	Usability, Efficiency	They have different limitations so each one can be better than the others in different situations.
(Angmo & Sharma, 2014)	Selenium, Watir	Execution speed, Record and playback, Platform support, Browser support, Language support, Test result reports, Accessibility, Usability	Selenium is a better choice in various conditions but Watir is suitable under certain specific situations.

(Sharma & Angmo, 2014)	FitNesse, LoadRunner, QTP, Selenium, SilkTest, TestComplete, TestNg, TOSCA, Watir, WinRunner	Language support, Platform support, Testing type, Development status, Stable/initial release/development year, License, Developer, Browser support	Selenium is determined as the best tool according to the criteria in this study.
(Singh & Sharma, 2015)	Sahi, Selenium	Cost, Record and playback, Efficiency, Browser support, Platform support, Language support, Test result report, User experience	Sahi performs better than Selenium. However, Selenium is a better choice when platform compatibility and programming language support are required.
(Gupta et al., 2015)	QTP, Selenium	Cost, Language support, Application support, File upload system, Effort, and skill, Execution speed	They have different limitations, so both can be better than the other in different situations.
(Meenu & Kumar, 2015)	QTP, Selenium, SoapUI, Test Complete	Cost, Platform support, Application support, Browser support, Language support, Technical support, Record and playback, User experience, Test result report, Data-driven framework	They have different limitations, so both can be better than the other in different situations.
(Kunte & Mane, 2017)	QTP, Selenium	Cost, Efficiency, User experience, Platform support, Language support, Execution speed	Selenium is a better choice for web applications than QTP, which provides a tester with more flexibility to test the applications under various complex scenarios.
(Radhakrishna & Nachamai, 2017)	JMeter, Soap	Response time	They have different limitations, so both can be better than the other in different situations.
(Abbas et al., 2017)	JMeter, LoadRunner, Microsoft Visual Studio, Siege	Test scripts generation, Test result report, Application support, Cost	JMeter provides the best results, but the proper tool should be chosen based on the budget, time, and nature of the software system under consideration that has to be tested.
(Kakaraparthi, 2017)	Ranorex, Selenium, Test Complete	Cost, Platform support, Language support, Data-driven testing, Record and playback, Application support, User experience, the Test result report	They have different limitations, so both can be better than the other in different situations.
(Mahmood & Sirshar, 2017)	Selenium, SAHI, WATIR, QTP	Language Support, User experience, Hardware Requirement, Cost	It can be practical to utilize open-source tools.
(Qasaimeh et al., 2018)	Acunetix, Burp Suite, NetSparker, Nessus, OWASP ZAP	Vulnerability	ZAP vulnerability is the highest. However, Acunetix and NetSparker have more accurate results.
(Jain & Rajnish, 2018)	OpenScript, Selenium	Cost, Efficiency, Hybrid-driven framework, Test result report, Reusability of code, Execution speed, Record and playback, IDE support, User experience, Application support	Selenium has an easy-to-use and efficient playback.
(Kaur, 2021)	JMeter, Load Runner, Silk, Selenium	Performance, Load, Browser Supported, Language Supported, License Types	Automation testing is more useful and time-saving than manual testing.
(Patil & Pawar, 2021)	Many different tools	Reports, level of skills, documentation, DevOps support, integration with ALM	The selection of test automation tools in the BFSI business unit would depend on the required budget and type of the project
(Prasad et al., 2021)	Many different tools	Modifiability, understandability, extensibility, modularity, compatibility, cost-effectiveness, flexibility, reliability	The study summarizes the existing literature.
(Srivastava et al., 2021)	Many different tools	Test development platform, scripting languages, programming skills, script creation time	Different types of applications need different types of protocols.

From the literature studies, it can be realized that the comparative studies of web-based automated testing tools have examined only a few tools and only a few comparison criteria. The most used criteria can be summarized as cost, user experience, test result report, language support, platform support, and environment support. In almost every study, it has been concluded that it is hard to select the best testing tool as they have

different advantages and disadvantages. Whether one tool is better than the other could be decided based on the existing criteria, but not in general. With the motivation of these results from the previous studies, a large number of web-based testing tools (14) and a large number of comparison criteria (20) have been examined in this paper. We have compared both the open-sourced and licensed tools, whereas most studies have only made

comparisons with the open-sourced tools. Therefore, our research is more comprehensive than the other literature on web-based automated testing tools.

## 2. WEB-BASED AUTOMATED TESTING TOOLS

The web-based automated testing tools discussed in this study are Acunetix, FitNesse, JMeter, Katalon Studio, LoadRunner, QTP (QuickTest Professional), which is also known as HP Unified Functional Testing (HP UFT), Ranorex Studio, Sahi Pro, Selenium, Telerik Test Studio, TestComplete, TestIO, TestingWhiz, and Webload. Some brief explanations of these tools were given below.

Acunetix, which was developed in 2005, can detect and report over 4500 web application vulnerabilities. Some of them are: Cross-Site Scripting, SQL Injection, XML Injection, Malicious File Inclusion, Unrestricted URL Access, HTTP Response Splitting, Information Leakage, Improper Error Handling, Weak Authentication, Session Fixation and so on. The vulnerability scanner tool was initially developed for Windows until 2014. A Linux version and Acunetix 360 for enterprises were formed in 2018, respectively. An example scene of the Acunetix test report is given in Fig. 1 (Acunetix, 2020).

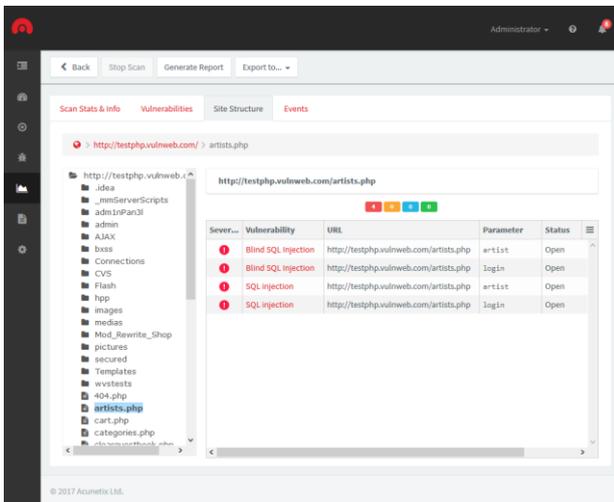


Fig. 1. An example Acunetix test report

FitNesse is all of a web server, a wiki, and an automated testing tool for software specially used for acceptance testing. The FitNesse project started in 2004 in Java with a fork of Fit classic Java, and new versions were created with new plug-ins until 2010. A FitNesse test report example is given in Fig. 2 (FitNesse, 2020).

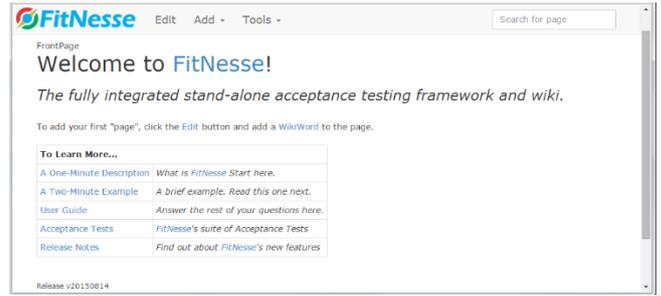


Fig. 2. An example FitNesse test report

JMeter, which was developed by Apache in 1998, is a tool for both dynamic and static resources. JMeter is used to simulate a heavy load on a server, object, or network group or to analyze the overall performance of a web application under many load types. JMeter can analyze the ability to load and perform tests of many different applications such as databases, message-oriented middleware, and internet protocols. An example JMeter test report is given in Fig. 3 (JMeter, 2020).

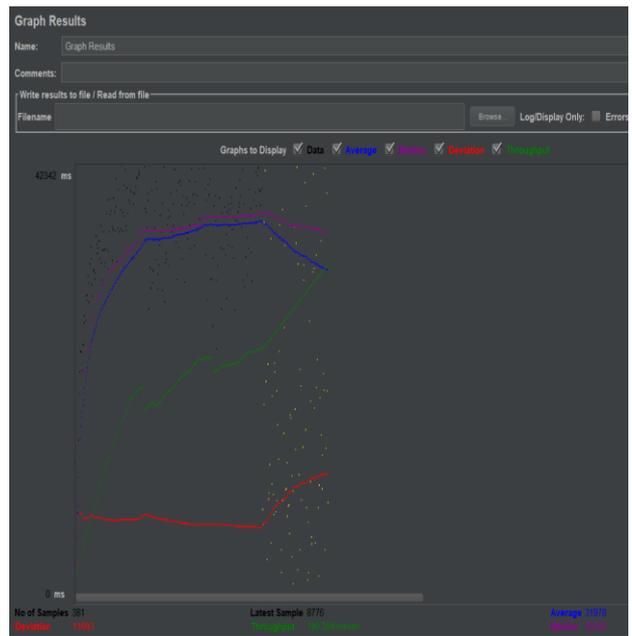


Fig. 3. An example JMeter test report

Katalon Studio (or informally Katalon) is used to help the users to generate automated tests cross-platform. The tool developed in 2015 is compatible with all operating systems, browsers, and devices. Differently from many web-based automated tools, Katalon has both open-sourced (Katalon Studio) and licensed (Katalon Studio Enterprise and Katalon Runtime Engine) versions. An example scene of the Katalon

Studio test report is given in Fig. 4 (Katalon Studio, 2020).



Fig. 4. An example Katalon Studio test report

LoadRunner was presented by MicroFocus in 1993. It offers a simplified and faster testing experience to the users, and it supports performance testing for the most comprehensive range of protocols and over 50 technologies and application environments. LoadRunner, which has more than 40 versions up to today, emulates many concurrent virtual users to apply real workloads to any application and capture end-user response times. An example LoadRunner test report is given in Fig. 5 (LoadRunner, 2020).

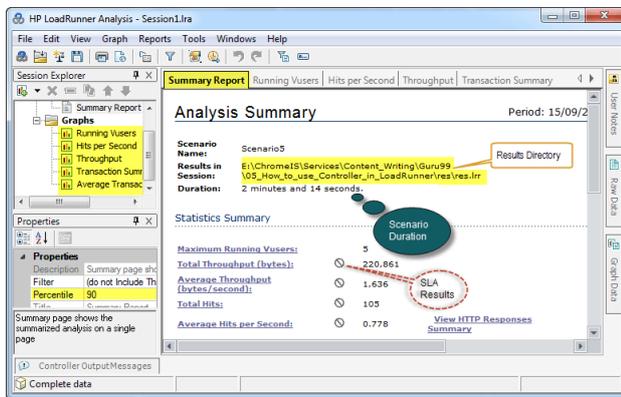


Fig. 5. An example LoadRunner test report

QTP (QuickTest Professional), which is also known as HP Unified Functional Testing (HP UFT), provides users to execute and create automated apps on web, mobile, and desktop platforms. QTP was developed in 2001, and there are many versions today. The tool's some features are; automated documentation, exception handling, data-driven testing, extensibility, error handling mechanism, and unique handling mechanism. An example QTP test report is given in Fig. 6 (QTP, 2020).

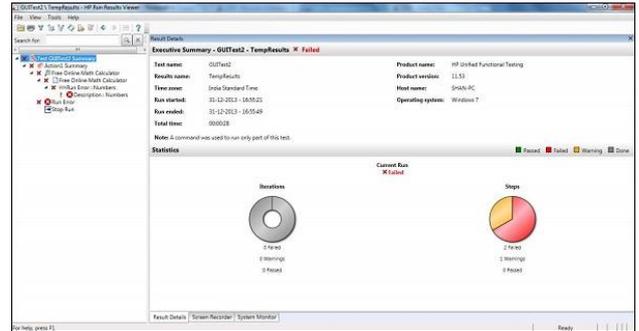


Fig. 6. An example QTP test report

Ranorex Studio is one of the web application testing automation tools covering all mobile, desktop, and web applications. Some of the features are record and playback, GUI recognition, reusable test code, and integration with various tools. The tool delivers robust object recognition, supports web frameworks and web technologies, performs what testers refer to as web element identification, integrates with the current solutions, and has various useful properties. An example Ranorex Studio test report is given in Fig. 7 (Ranorex Studio, 2020).

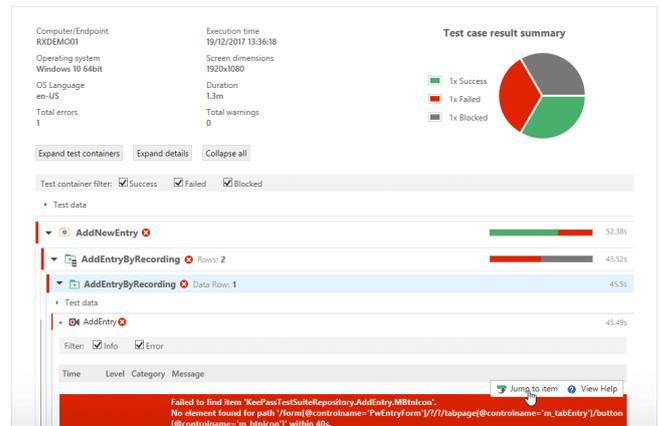


Fig. 7. An example Ranorex Studio test report

Sahi Pro (or informally Sahi) is a tool designed to solve the everyday problems faced by testers. Sahi is uniquely tester-centric in its design and functionality. It can be used on all browsers on all operating systems for desktop, mobile, and web testing. While it can be integrated with many other tools, it is an efficient tool for different testing types. An example Sahi Pro test report is given in Fig. 8 (SahiPro, 2020).

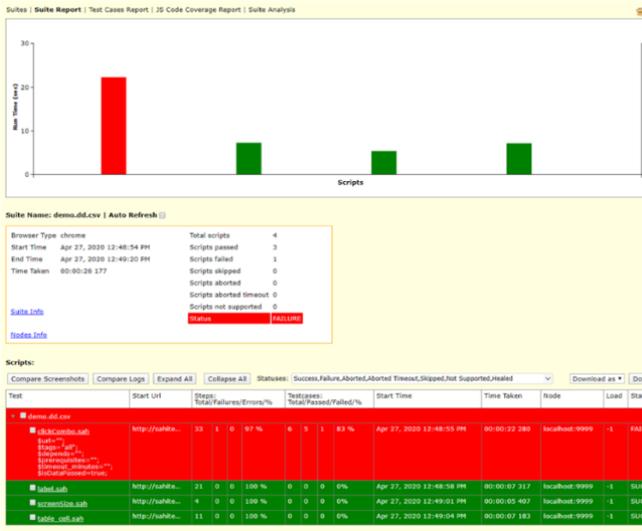


Fig. 8. An example Sahi Pro test report

Selenium, which is the first tool that comes to people's minds when anyone is talking about web-based automated tools, was first developed in 2004. Selenium is a popular web application tool that supports different types of software tests. It is a sophisticated tool composed of several components that have specific roles in helping in the development of web applications. Not all of them are used together; most of the Selenium Quality Assurance engineers focus on one or two tools for their projects. Selenium is a widely used web-based automated tool because it supports various platforms, operating systems, programming languages, and browsers. However, Selenium is used to test web applications only; it does not have any ability to test desktop-based applications, mobile applications, or any other software. It does not offer a native reporting facility, one of the most important disadvantages of the tool. An example Selenium test report is given in Fig. 9 (Selenium, 2020).

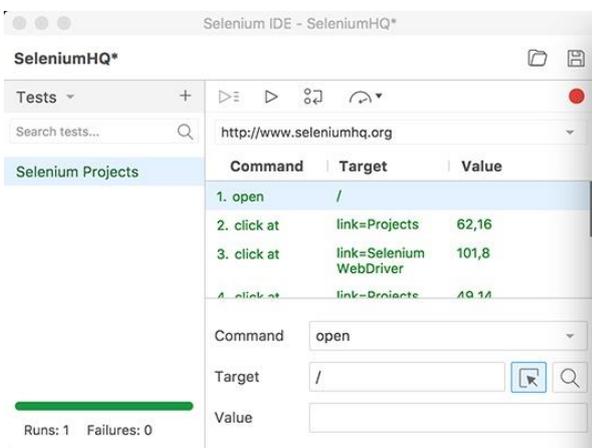


Fig. 9. An example Selenium test report

Telerik Test Studio is a Windows-based automated tool for different types of applications. It empowers the users to quickly and easily craft automated tests, integrate them in CI/CD (Continuous Integration/Continuous Delivery) environment following the workflow, find defects earlier, and ship a better quality software product. It can perform various testing types, such as functional testing, performance testing, load testing, and mobile testing. An example Telerik Test Studio test report is given in Fig. 10 (Telerik Studio, 2021).

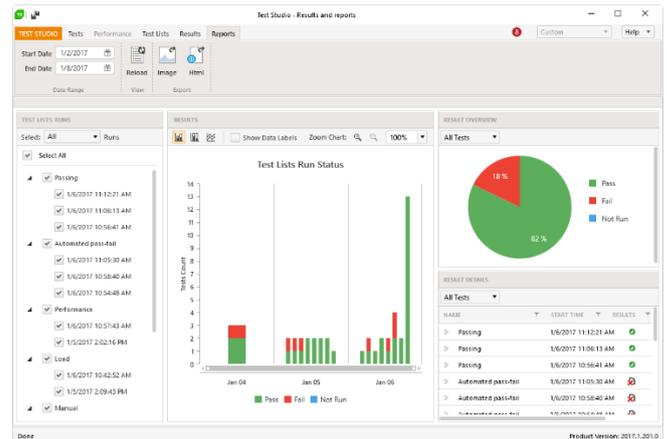


Fig. 10. An example Telerik Test Studio test report

TestComplete is a functional automated testing platform developed in 1999. TestComplete contains three significant modules, which are mobile, web, and desktop testing, and each of the modules contains its functionality for creating automated tests. TestComplete is an easy, reliable, fast tool that has a capable supporting team. An example TestComplete test report is given in Fig. 11 (TestComplete, 2021).

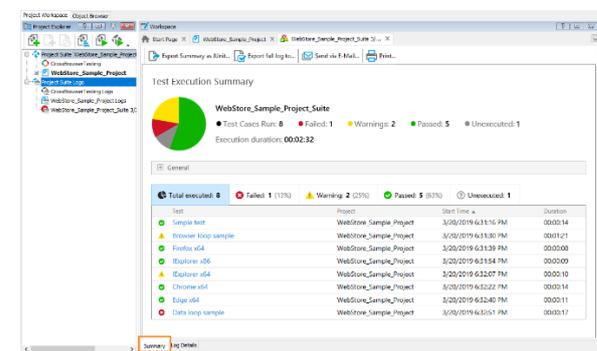


Fig. 11. An example TestComplete test report

TestIO is another web testing tool that is used generally for crowd-testing. It can remove quality assurance bottlenecks with flexible testing, enable one to expand one coverage to hundreds of platforms and devices and ensure one of the professional testers has unbiased eyes

on the product. It has high compatibility with devices, browsers, and platforms. An example TestIO test report is given in Fig. 12 (TestIO, 2022).

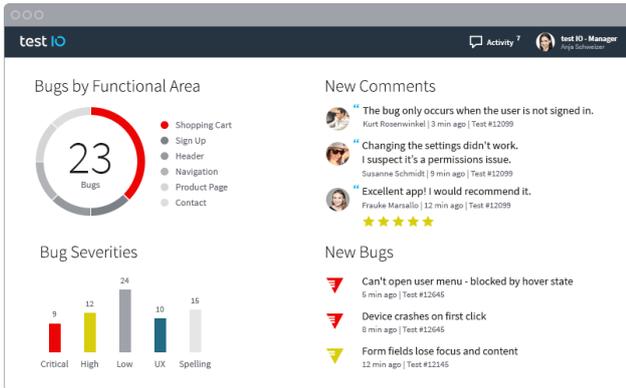


Fig. 12. An example TestIO test report

TestingWhiz supports various testing types such as mobile testing, regression testing, data-driven testing, database testing, big bang testing, and so on. TestingWhiz is very easy to use for both large and small applications. It has intelligent and reusable recording techniques while testing. Also, the test commands applied by the tool are usable even to users who have no coding skills to optimize testing workloads and boost the efficiency of the automation projects. However, the tool is not provided for free; it is subscription-based, which is readily available on request, and it cannot thoroughly scan a web application. An example TestingWhiz test report is given in Fig. 13 (TestingWhiz, 2022).

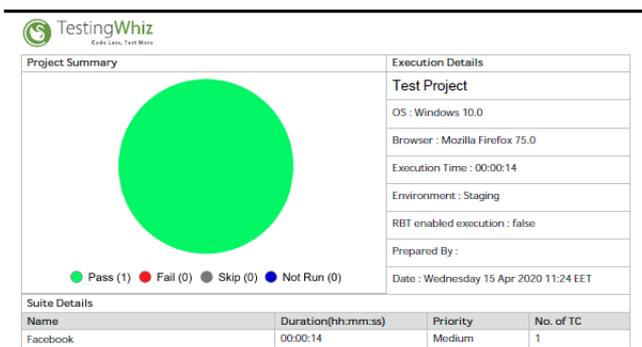


Fig. 13. An example TestingWhiz test report

Webload, which was developed in 1997, is one of the testing tools which are specially used for load/performance testing. There is a long period between the first and second versions of Webload. The second and third versions were presented in 2010, and after that, new versions were developed continuously. Webload supports many integration tools, browsers, platforms, and also cloud applications. With Webload, a tester can pinpoint issues and the bottlenecks which may

stand in the way of achieving one load response requirement. An example Webload test report is given in Fig. 14 (Webload, 2021).

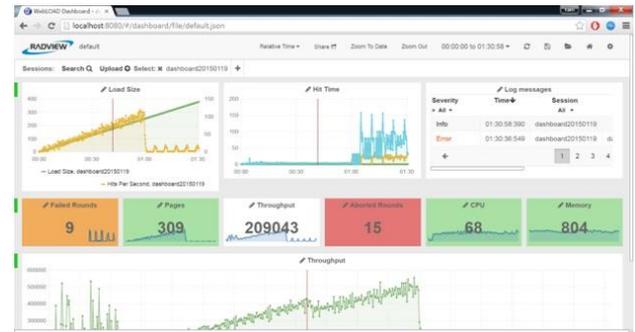


Fig. 14. An example Webload test report

### 3. RESULTS AND DISCUSSIONS

In this section, a comparison of the web-based automated tools according to various criteria has been performed. The tools were compared according to 20 different features. These features are gathered in 5 separate tables according to their relevance to facilitate the comparisons. The comparisons were made by examining each tool's official website, the related studies and tutorials in the literature, and our test simulations. For this study, many sample tests have been carried out such as implementing tests operations on Facebook by using Selenium, TestingWhiz, and Webload, using sample university websites with JMeter, TestingWhiz and Webload, shopping on sample shopping sites with JMeter, TestingWhiz, and Katalon Studio, comparing Facebook and Twitter, Google, Yahoo and Yandex with JMeter and some other test simulations. The tools are classified according to their general characteristics in Table 2, requirements in Table 3, technical compatibility in Table 4, testing properties in Table 5, and technical supports in Table 6. For each comparison table, the features and the values of attributes are briefly discussed.

#### 3.1. Analysis of General Properties

General Properties refer to the properties of a tool that come to mind in the first place. These features are the Testing Style, Open Sourced/Licensed, Cost, and Stable Release. The features discussed in this section are described below.

“Testing style” explains how automation software does the test. It can take values in the form of API Testing, Desktop Testing, Mobile Testing, Web Testing, and combinations thereof. Open-sourced/Licensed indicates whether the tool has a license or not. Tools with free versions of open source, licensed for later versions, are

also specified. This feature is important because it handles both open-source and licensed tools, unlike other literature studies. "Cost" is a numerical value that expresses the price of the tool. The intermediaries' fees may be specified in different ways, such as daily, monthly, or 1-time purchases. "Stable release" means the most recently updated version of a tool and is

expected to be recent for the tool to be used. Since software engineering is a continually evolving field, software varies and changes over time, and also different necessities may occur for other software. In this respect, a tool needs to update tools frequently and meet the changing software needs.

**Table 2.** Comparison of web-based automated tools according to general properties

	Testing Style	Open-sourced/Licensed	Cost	Stable release
<b>Acunetix</b>	Web Testing	Licensed	Customizing according to customer necessities	-
<b>FitNesse</b>	Web Testing	Open Sourced	Free	April 2019
<b>JMeter</b>	Mobile Testing, Web Testing	Open Sourced	Free	November 2019
<b>Katalon Studio</b>	API Testing, Desktop Testing, Mobile Testing, Web Testing	Katalon Studio (Open Sourced) Katalon Studio Enterprise (Licensed) Katalon Runtime Engine (Licensed)	Katalon Studio (Free) Business technical support (\$2,500 per year) Enterprise technical support (\$5,500 per year) Enterprise Premium technical support (customizable price)	October 2019
<b>LoadRunner</b>	Desktop Testing, Web Testing	Licensed	50 virtual user licenses (free) More virtual users (\$1,40 per day)	2020
<b>QTP (HP UFT)</b>	Desktop Testing, Web Testing	Licensed	Trial (Free) Runtime Engine (\$2,300 per year) Uft One (\$3,200 per year) Volume Pricing (Customizing according to customer necessities)	July 2019
<b>Ranorex Studio</b>	Desktop Testing, Mobile Testing, Web Testing,	Licensed	Trial (Free) Runtime License Floating (\$890) Premium Node-Locked (\$2,990) Premium Floating (\$4,990)	October 2019
<b>Sahi Pro</b>	Desktop Testing, Mobile Testing, Web Testing	Licensed	Trial (Free) Sahi Pro (\$695 per year)	-
<b>Selenium</b>	Web Testing	Open Sourced	Free	2018
<b>Telerik Test Studio</b>	Desktop Testing, Mobile Testing, Web Testing	Licensed	Test Studio Web & Desktop (\$2,499 per year) Test Studio Ultimate (\$3,499 per year)	June 2019
<b>TestComplete</b>	Desktop Testing, Web Testing	Licensed	Trial (Free) TestComplete Base (Euro 5,170) TestComplete Pro (Euro 8,023) TeamSuite Bundle (Customizing according to customer necessities)	June 2019
<b>TestIO</b>	Web Testing	Licensed	Available on request	-
<b>TestingWhiz</b>	Desktop Testing Web Testing	Licensed	Available on request	-
<b>Webload</b>	Web Testing	Licensed	Available on request	February 2016

As shown in Table 2, while some tools have only one unique test style (web testing or desktop testing), some other tools consist of a combination of tests and have many test styles such as API Testing, Desktop Testing, Mobile Testing, and Web Testing. This feature has of great importance in the selection of the tool. For a tool to be used on an application, it must have a structure suitable for that application. As shown from the table, only three tools within 14 tools are entirely open-source and free. The remaining licensed vehicles have considerably higher fees. While specific versions or trial versions of some tools are free, a professional version must be paid for. While the prices of some tools are

specified as daily or monthly, the fees of some other tools or some versions can be customized according to customer demands. This is an essential criterion regarding the selection of the tool. This table shows that different market tools for companies seeking professional support, individual nonprofessional users, and actors with different roles state that each of these actors can find a suitable tool according to their needs. When the stable release features are examined for the tools, it is seen that all of them are updated frequently, and in this respect, it can be stated that all tools can meet the changing software needs.

### 3.2. Analysis of Requirements

In this section, the required features for a tool to be used efficiently are discussed. Since the most crucial feature for the efficient use of a tool is the hardware requirements, both CPU and RAM requirements are presented. Besides, it is another matter to consider whether any programming knowledge is required to use a tool. The features discussed in this section are described below. “Hardware requirement (CPU)”

expresses the minimum CPU features required to use the tool. CPU speed and other features are given in this feature. “Hardware requirement (RAM)” referred to the minimum and recommended RAM amounts (if detected) required to use the tool. “User experience requirement” indicates whether user experience is necessary for the use of the tool or not. It can take two values “Easy to use” (can be used without any knowledge) or Requires some programming skills.

**Table 3.** Comparison of web-based automated tools according to requirements

	Hardware requirement (CPU)	Hardware requirement (RAM)	User experience requirement
<b>Acunetix</b>	64-bit any processor	Minimum: 2 GB	Easy to use
<b>FitNesse</b>	-	-	Easy to use
<b>JMeter</b>	Multicore CPU with four or more cores	Minimum: 16 GB	Requires some programming skills
<b>Katalon Studio</b>	1 GHz, 32-bit (x86) or 64-bit (x64) processor or higher	Minimum: 1 GB (for 32-bit) 2 GB (for 64-bit)	Easy to use
<b>LoadRunner</b>	Intel Core, Pentium, Xeon, AMD or compatible Speed: 1.6 GHz or higher	Minimum: 2 GB Recommended: 4 GB	Requires some programming skills
<b>QTP (HP UFT)</b>	1.6 GHz or higher	Minimum: 2 GB when no more than three add-ins are loaded simultaneously.	Easy to use
<b>Ranorex Studio</b>	2 GHz dual-core or higher	Minimum: 1 GB	Easy to use
<b>Sahi Pro</b>	-	Minimum: 1.5 GB	Requires some programming skills
<b>Selenium</b>	4x Dual-core AMD Opteron or higher	Minimum: 4 GB	Requires some programming skills
<b>Telerik Test Studio</b>	x86 or x64 1 GHz Pentium processor or higher	Minimum: 1 GB Recommended: 2 GB	Easy to use
<b>TestComplete</b>	Intel Core i5 or Intel Core i7 (the 3rd generation) or higher	Minimum: 8 GB	Easy to use.
<b>TestIO</b>	800 MHz or higher	Minimum: 4 GB	Easy to use
<b>TestingWhiz</b>	Intel Pentium 4 or higher	Minimum: 4 GB Recommended: 8 GB	Requires some programming skills
<b>Webload</b>	IBM-compatible PC (x86-32) with Pentium III 800 MHz or higher	Minimum: 1 GB Recommended: 4 GB	Easy to use

Table 3 presents a comparison of the web-based automated tools according to hardware and user experience requirements. It has all the information for all tools except FitNesse and Sahi Pro tools. Although the criteria in this table do not directly affect the tool's selection, it provides information on whether the currently selected tool can be used or not and what requirements must be met for the tool before it is used. For example, it would be beneficial for a new user to start the test process with the FitNesse tool in terms of being both open source and free and easy to use. However, FitNesse only performs web testing and cannot be used for mobile testing. Also, since there is no full information about the CPU configuration it requires, there may be a compatibility problem with the hardware to be tested. Table 3 also indicates that the tools usually need high CPUs and RAMs. This situation can be stated as the fact that at least one other software is running while the tools are running, various data operations are performed, and reports are generated on the software. JMeter needs a higher memory than other tools because

it is cross-platform; it supports all browsers and IDEs that support Java; in other words, it has high compatibility. One of the first actions to be taken after deciding to use a tool is to provide the tool's requirements. As it is seen, while some tools are easy to use, the user should have specific programming knowledge in some popular and frequently used tools such as Selenium. In other words, it is not enough for testers to know and apply the test cases and test scenarios; they need to write these cases and procedures and make changes if necessary.

### 3.3. Analysis of Technical Compatibility

In this section, programming languages, platforms, browsers, and IDEs supported by tools are discussed. The feature that maximizes an automation tool's usability can be considered compatibility because it provides a tool for many projects. The features discussed in this section are described below.

“Language compatibility” expresses which programming languages the tool can test on applications.

“Platform compatibility” determines the operating systems and other platforms on which the tool can

operate. “Browser compatibility” expresses the lowest level of web browsers that the tool can work with. “IDE compatibility” gives the IDEs supported by the tool.

**Table 4.** Comparison of web-based automated tools according to technical compatibility

	Language Compatibility	Platform Compatibility	Browser Compatibility	IDE Compatibility
<b>Acunetix</b>	CSS, HTML, Java, JavaScript, .NET, Node.js., PHP	Web-based environments	Any browser	-
<b>FitNesse</b>	C++, C#, Delphi, Python, Ruby	Cross-platform	Any browser	Maven or Ivy in Eclipse
<b>JMeter</b>	-	Cross-platform	Any browser	All IDEs which support Java
<b>Katalon Studio</b>	Java/Groovy	Windows 7, 8, 10, MacOS 10.11+, Linux (Ubuntu-based)	Any browser	Java, Android SDK, Web drivers
<b>LoadRunner</b>	C#, Java, JavaScript, VB, VBScript	Windows 7 (SP1) 32/64 bit, 8 64 bit, Windows Server 2012 64 bit, R2 (SP1) 64 bit	Any browser	-
<b>QTP (HP UFT)</b>	VBScript	Windows 7 32/64 bits, Windows 7 (SP1) 32/64 bits	IE 6,7,8,10, Firefox 3.0 and later, Google Chrome	Its IDE
<b>Ranorex Studio</b>	No specific scripting language (it is written in .NET using C hash, Iron python, and VB.net)	Windows 2000, XP, Vista, 7, Windows Server 2003, 2008	Any browser	Its IDE
<b>Sahi Pro</b>	JavaScript	Cross-platform	Any browser	Selenium IDE
<b>Selenium</b>	C#, Java, JavaScript, Perl, PHP, Python, R, Ruby	Windows, macOS X, Linux	All major browsers	All IDEs which support Java
<b>Telerik Test Studio</b>	AJAX, Angular, HTML, MVC, Silverlight, WPF	Windows 7, 8, 10, Windows Server 2008, 2012	Internet Explorer, Microsoft Edge, Firefox, Google Chrome, Opera, Safari	Visual Studio IDE
<b>TestComplete</b>	C++, C#, DelphiScript, JavaScript, Jscript, Python, VBScript	Windows 7 (SP1), 8, 8.1, 10, Windows Server 2008 R2, 2012, 2012 R2, 2016, 2019	IE, Firefox, Google Chrome	Its IDE
<b>TestIO</b>	-	Web-based environments	Any browser	-
<b>TestingWhiz</b>	-	Windows	Any browser	Its IDE
<b>Webload</b>	JavaScript	Windows, Linux	Any browser	Its IDE

The values in this table are related to the technical features of the tool. Programming language compatibility means which programming language can test on applications written, a fundamental selection criterion. While some tools support only one language (JavaScript, VBScript) in this regard, some tools increase the probability of being used because they support many languages (C ++, C #, DelphiScript, JavaScript, Jscript, Python, VBScript, and so on). Another one of the most critical factors affecting the selection of the tool is platform compatibility. While researching this feature, it is observed that some tools work on only one system (usually Windows), and some tools work on cross-platforms. In some tools, only the name of the operating system is included, while in some tools, it is possible to get information about the platforms' versions. A remarkable point here is that the tools usually do not work in the MAC OS operating system. Accordingly, the testers who should work with this system should make the selection only among many tools.

Since the tools examined in this study are web-based tools, there is generally no restriction in the browser support section. In other words, the tools mostly work in all browsers. However, in some tools such as QTP, browsers are expected to be higher than a specific version. Looking at Table 4, it is seen that the tools either use existing IDEs or create their unique IDEs. The limitation here is to consider compatibility issues when using tools that use current IDEs. For example, IDE used in tools such as JMeter and Selenium is expected to have Java support.

### 3.4. Analysis of Testing Properties

In this section, the features of the test process, which is the main function of the tools, are discussed. The features discussed in this section are described below. “Other tools support for integration” refers to other tools that the tool can work with. “Testing type” describes the tests that the tool performs. There are areas where each of them specializes. Some tools focus on security like

penetration and vulnerability, while others are concerned with tests such as load, stress, and performance. This feature generally indicates which tests a test tool can perform best. “Test result

generation” expresses how the results of the test will be reported. In other words, it can be described as the format of the reports obtained.

**Table 5.** Comparison of web-based automated tools according to testing properties

	<b>Other tools support integration</b>	<b>Testing type</b>	<b>Test result generation</b>
<b>Acunetix</b>	GitHub, FortiWeb, Imperva, Jenkins, Jira, SecureSphere	Penetration, Vulnerability, Web, Security	HTML, PDF
<b>FitNesse</b>	-	Acceptance	-
<b>JMeter</b>	BlazeMeter, Dynatrace, Jenkins, Meliora, TestLab, Visual Studio	Load, Performance, Stress	CSV, HTML
<b>Katalon Studio</b>	Azure DevOps, Bamboo, CircleCI, Jenkins, Jira, qTest, TestLink, TestRail, TeamCity, Travis CI, Microsoft Teams	API, Mobile, Web	CSV, HTML, PDF
<b>LoadRunner</b>	-	Load, Performance	-
<b>QTP (HP UFT)</b>	CollabNet, iRise, Jira, Kovair, TeamForge, TestComplete, ServiceNow	API, Business Process, GUI	HTML
<b>Ranorex Studio</b>	Azure DevOps, Bamboo, Hudson, Jenkins, Team City	Data-Driven, GUI, Keyword-Driven, Quality Assurance, Regression	HTML, XML
<b>Sahi Pro</b>	Bamboo, Jenkins, Selenium, or any CI tool	Agile, Functionality, Load, Performance	XSL
<b>Selenium</b>	Can be integrated with many prices or free tools	Data-Driven, GUI, Keyword, Regression, Unit, Web	No reports
<b>Telerik Test Studio</b>	Bamboo, Jenkins, Jira, TeamCity, TFS	Load, Functional, Performance, Restful	Junit, XML
<b>TestComplete</b>	Bamboo, Jenkins, Jira, QAComplete, Selenium, Team Foundation Server	Coverage, Data-Driven, GUI, Keyword, Load, Mobile, Regression, Unit, Web	HTML, Junit, PDF, XML
<b>TestIO</b>	GitHub, Jira, Redmine, Trello	Beta, Exploratory, Regression, Usability	CSV, XLS
<b>TestingWhiz</b>	Bamboo, Jenkins	Cross-Browser, Database, Data-Driven, Mobile, Regression, Web	CSV, XLS
<b>Webload</b>	Amazon Web Services, Jenkins, Selenium	Capacity, Load, Stress, Soak	CSV, DOC, HTML, JUnit, ODT, PDF, RTF, XLS, XLSX

When looking at Table 5, it is seen that other tools that support integration information were not available for FitNesse and Load Runner. When looking at other tools, it is generally found that Bamboo, Jenkins, and Jira are common in all; apart from these, it can be seen that they can be integrated with many tools. One of the issues to be considered here is that SAHI, TestComplete, and Webload tools can be combined with Selenium. Apart from these, Selenium stands out in this feature as it can work integrated with many different tools, free or licensed except these tools.

Perhaps the most critical factor in choosing a testing tool is the types of tests it can perform. As it is known, not every tool can perform all kinds of tests. While some tools focus on security, some tools focus on performance balls and derivatives, and some tools deal with different types of tests. From Table 5, it is seen that Acunetix performs security tests; JMeter, Load Runner, SAHI, Telerik, and WebLoad perform tests, FitNesse performs acceptance tests, and other tools perform different types of tests. Accordingly, it is essential to choose a tool according to the test type to be performed on the application.

The reports obtained from the tests are not a direct criterion for selecting the tool, but it is a feature that can help to use the tool efficiently. It is seen that all of the tools support standard formats such as CSV, HTML, XML, and PDF, except for some tools that do not have information about the test report. Apart from that, it is seen that Telerik, TestComplete, and Webload also support the Junit format, and Webload has quite a lot of options regarding the test report. One of the surprising results of this feature is that there is no report support in Selenium, which is the first tool that many people think of when it comes to web-based automation tools. Selenium, which has been ahead of other tools in many respects, has become the most left-behind tool in this feature.

### 3.5. Analysis of Technical Support

In this section, the part's features of the technical support available for free or paid tools are discussed. These are essential points that ensure the continuity of the tool. The features discussed in this section are described below.

“Customer support” expresses whether the tool gives supporting the customers using it. It refers to a general

situation and takes one of the values—no professional support, Limited, Free Communities, and Dedicated. “Documentation titles” refers to the documentation titles on the official website of the tools. This feature can also be thought of as information someone can easily find on the tool’s website. “Articles” gives the various articles on the official website of the tools. Although it does not have a format, it is evaluated in two categories: A limited number of articles and A large number of well-

organized articles. Forums indicate whether there is a forum on the tool's website or not. It does not have a specific format; it gives general information about the forum's presence/absence and its features. The technical staff provides available information about whether there is a technical staff supporting the tools or not. “Bug tracking” expresses whether there is a bug tracking feature in the tool or not, and if so, what operations this feature does.

**Table 6.** Comparison of web-based automated tools according to technical support

	Customer support	Documentation titles	Articles	Blog	Technical support	staff	Bug tracking
<b>Acunetix</b>	Dedicated	Case Studies, Support, Videos, Web Vulnerabilities, Webinars, Whitepapers and subtitles	A limited number of articles	An active blog with many posts	24/7 technical support	online staff	Assigning identified vulnerabilities
<b>FitNesse</b>	-	Features, Download Plug-ins, User Guide	A limited number of articles	No blogs	-	-	No bug tracking
<b>JMeter</b>	Free Communities	Get Started, User Manual, Best Practices, Component Reference, Functions Reference, Properties Reference, Change History, Javadocs, JMeter Wiki, FAQ Wiki and subtitles	-	An active blog with many posts	-	-	Creating a bug report and listing the bug logs quickly.
<b>Katalon Studio</b>	Dedicated	Products, Pricing, Support The company, Download Now, Sign In and subtitles	A limited number of articles	An active blog with many posts	4/7 technical support	online staff	No bug tracking
<b>LoadRunner</b>	Limited	Resources, Pricing, Contact Us, Education Central, Free Trial	-	No blog	-	-	No bug tracking
<b>QTP (HP UFT)</b>	Dedicated	Well-organized – considerable documentation with subtitles	A large number of well-organized articles	An active blog with many posts	24/7 technical support	online staff	No bug tracking
<b>Ranorex Studio</b>	Dedicated	Products, Solutions, Webinars, Support, Company Blog	A large number of well-organized articles	An active blog with many posts	24/7 technical support	online staff	No bug tracking
<b>Sahi Pro</b>	Limited	Home, Features, Services, Pricing, Try for Free	A limited number of articles	No blog	-	-	No bug tracking
<b>Selenium</b>	No professional support	About, Downloads, Projects, Documentation, Support, Blog	-	An active blog with many posts	-	-	Determining new bugs or checking the status of an existing one
<b>Telerik Test Studio</b>	Dedicated	Demos, Pricing, Blogs, Docs & Support, Search	A limited number of articles	An active blog with many posts	-	-	Determining new bugs
<b>TestComplete</b>	Dedicated	Features, Use Cases, Integrations, Resources Pricing, Start my Free Trial	A large number of well-organized articles	A general blog	24/7 technical support	online staff	No bug tracking
<b>TestIO</b>	Dedicated	Product, Testing Solutions, Customers, Pricing, Resources and subtitles	A limited number of articles	An active blog with many posts	24/7 technical support	online staff	No bug tracking
<b>TestingWhiz</b>	Dedicated	Solutions, Features, Integrations, Pricing, Resources, Blog, Company and subtitles	A large number of well-organized articles	An active blog with many posts	24/7 technical support	online staff	No bug tracking
<b>Webload</b>	Dedicated	Solutions, Features, Resources, Blog, Pricing and subtitles	A large number of well-organized articles	A general blog	24/7 technical support	online staff	No bug tracking

The features in this table have not been dealt with in any previous studies in the literature. This is one of the original features of this publication. The disadvantageous aspect of this table's features is that it

does not have a numerical or deterministic value, and it is tried to be tabulated with verbal expressions. Therefore, it is a little more subjective than other tables. Also, when compared to other tables, it is seen that the

most missing information is also here. For example, sufficient information about Load Runner for many features could not be obtained.

When the table is examined, it is seen that there is limited customer support for some tools and support in the form of free communities for JMeter, it is not possible to obtain customer support information for FitNesse, and there is no professional support for Selenium. Since one of the most detailed information about the test tools can be brought is the websites, the websites' documentation titles also provide information about the tool's support. In general, it is seen that many tools have detailed menus and subtitles, and some tools have only main titles, but as a result, it is understood that there is sufficient documentation information in each tool.

Another feature examined in this table is whether there is a blog on a tool's website or not. A blog, an area where both the manufacturers and users of the tools share their experiences, can sometimes be an effective problem-solving method. It is seen that there is a blog system in all tools except FitNesse, LoadRunner, and SAHI. In terms of a tool, technical staff support is also of great importance. It is a facilitating aspect for the user that there is active technical support, especially for the users who are not experienced in the test automation process. Online 24/7 technical support is provided in Acunetix, Katalon Studio, QTP (HP UFT), Ranorex, TestComplete, TestIO, TestingWhiz, and Webload tools. This technical support can be provided by phone, e-mail, website, or in different ways. No information on the technical support of other teams has been obtained. The last feature examined in this table, and this study is bug tracking. This feature, which means following and monitoring the bugs brought with a tool, is a feature that can provide statistical benefit, especially in the long-term use of the tool. However, there is a bug tracking feature only for Acunetix, JMeter, Selenium, and Telerik Test Studio. These bug tracking structures assign identified vulnerabilities as tasks to team members for remediation, create a bug report and list the bug logs quickly, determine new bugs, and check the status of an existing one.

#### 4. CONCLUSION

Automated software testing is one of the most useful and popular aspects of the software testing process. Automated software testing tools provide some advantages to the users, such as performing the test process quickly, trying as many test cases as possible,

and performing significant data operations. They are also used to measure some essential software metrics such as security, vulnerability, efficiency, performance, load, stress, and so on. Especially for web-based applications, automated software testing tools have a critical role. Since the number and type of automated tools are evolving and expanding today, compatible programming languages, operating systems, and browsers are also continually increasing and provide many options for those interested in software testing. Although this is advantageous, it can be a disadvantage if the right tool cannot be selected because every tool has different features and may not provide a useful test for every software. In this respect, a tester should decide the tool that best tests the application he/she will try. This selection process takes place not only by software features but also by many different criteria. These criteria can be related to the application's features to be used, financial conditions, technical support, or any other thing. The more we know about the tools, the more comfortable we choose the most appropriate one. For this reason, there are many studies in the literature comparing software testing tools.

This study has a wide-ranging analysis feature that addresses more tools and more criteria than previous tool comparison papers in the literature. The 14 free or paid tools discussed in this study, according to 20 different standards, are Acunetix, FitNesse, JMeter, Katalon Studio, LoadRunner, QTP (HP UFT), Ranorex, Sahi Pro, Selenium, Telerik Test Studio, TestComplete, TestIO, TestingWhiz, and Webload and the criteria are general properties (Testing style, Open-sourced/Licensed, Cost, Stable release); requirements (Hardware requirement (CPU), Hardware requirement (RAM), User experience requirement); technical compatibility (Language compatibility, Platform compatibility, Browser compatibility, IDE compatibility); testing properties (Other tools support for integration, Testing type, Test result generation) and technical support (Customer support, Documentation titles, Articles, Blog, Technical staff support, bug tracking). After all these reviews, the following results were obtained:

- If the vulnerability metric of the software will be explicitly measured; Acunetix,
- If there is a necessity for an open-sourced tool specializing in web testing only; FitNesse,
- If a tester with knowledge of programming will test different platforms, IDEs, and browsers on an environment with high hardware features; Jmeter,

- If test operations will be performed on APIs; Katalon Studio,
- If the software's metrics such as load, stress, and performance will be explicitly measured by a tester with knowledge of programming; LoadRunner,
- If testing will be done on an environment with high hardware features and an application written in VBScript and if the tool is required to use its IDE; QTP,
- If testing is done on the desktop, mobile, and web applications at the same time and the used programming languages are also various; Ranorex Studio,
- If there is a necessity for the testing report of type XSL, or if testing will be done using the current Selenium IDE; Sahi Pro,
- If testing is done on a hybrid project by a tester with knowledge of programming using different tools and the test reports are not necessary; Selenium,
- If different test types such as Load Testing, Functional Testing, Performance Testing, and Restful Testing will be performed and a bug tracking mechanism is needed to detect new errors; Telerik Test Studio,
- If different tests such as Coverage Testing, Data-Driven Testing, GUI Testing, Keyword Testing, Load Testing, Mobile Testing, Regression Testing, Unit Testing, and Web Testing should be applied to the same project; TestComplete,
- If a tool is required to be integrated with specialized tools such as Redmine or Trello and to perform usability testing; TestIO,
- If a database test is to be performed on a Windows-based system and good technical support is needed; TestingWhiz,
- If the software's metrics, such as load, stress, and performance, are to be measured, Webload would be the most appropriate tool.

In general, the result obtained from this study can be summarized as no tool is better than others. The suitability of a tool for a project depends on various criteria in the test process.

## REFERENCES

- Abbas, R., Sultan, Z., & Bhatti, S. N. (2017). Comparative Analysis of Automated Load Testing Tools: Apache JMeter, Microsoft Visual Studio (TFS), LoadRunner, Siege. *International Conference on Communication Technologies*.
- Acunetix, <https://www.acunetix.com>, Retrieved 13 April 2020.
- Al-Zain, S., Eleyan, D. & Hassounah, Y. (2013). Comparing GUI Automation Testing Tools for Dynamic Web Applications. *Asian Journal of Computer and Information Systems*, 1, 38–48.
- Angmo, R., & Sharma, M. (2014). Performance Evaluation of Web Based Automation Testing Tools. *International Conference-Confluence The Next Generation Information Technology Summit*.
- Bertolino, A. (2007). Software testing research: Achievements, challenges, dreams. *Future of Software Engineering Conference*.
- Bharti, T., & Dutt, E. V. (2014). Relative Review of Automated Testing Tools: (QTP) Quick Test Professional, Selenium and Test Complete. *International Journal of Computer Science Trends and Technology*, 2, 110–114.
- Bindal, P., & Gupta, S. (2014). Test Automation Selenium WebDriver using TestNG. *Journal of Engineering Computers and Applied Sciences*, 3, 18–40.
- Catelani, M., Ciani, L., Scarano, V.L., & Bacioccola, A. (2011). Software automated testing: A solution to maximize the test plan coverage and to increase software reliability and quality in use. *Computer Standards and Interfaces*, 33, 152–158.
- Chandel, V., Patial, S., & Guleria, S. (2013). Comparative Study of Testing Tools: Apache JMeter and Load Runner. *International Journal of Computing and Corporate Research*. 3, 1–17.
- Daud, N. I., Bakar, K. A. A., & Hasan, M. S. M. (2014). A Case Study on Web Application Vulnerability Scanning Tools. *Science and Information Conference*.

- Dubey, N., & Shiwani, S. (2014). Studying and Comparing Automated Testing Tools; Ranorex and TestComplete. *International Journal Of Engineering And Computer Science*, 3, 5916–5923.
- Dukes, L., Yuan, X., & Akowuah, F. (2013). A case study on web application security testing with tools and manual testing. *IEEE Southeastcon*.
- Elbaum, S., Karre, S. & Rothermel, G. (2003). Improving web application testing with user session data. *International Conference on Software Engineering*.
- Falah, B., Akour, M., & El Marchoum, N. (2015). Testing patterns in action: Designing a test pattern-based suite. *International Review on Computers and Software*, 10, 489–494.
- FitNesse, <http://docs.fitnessse.org/FrontPage>, Retrieved 15 April 2020.
- Gogna, N. (2014). Study of Browser-Based Automated Test Tools WATIR and Selenium. *International Journal of Information and Education Technology*, 4, 336–339.
- Gogna, N., & Kumari, R. (2011). Comparative Study of Browser Of Browser Based Open Source Testing Tools WATIR And WET. *International Journal on Computer Science and Engineering*, 3, 1910–1923.
- Gupta, S., Kumar, S., & Saxena, C. (2015). Review Paper on Comparison of Automation Testing Tools Selenium and QTP. *MIT International Journal of Computer Science and Information Technology*, 5, 55–57.
- Hedayati, A., Ebrahimzadeh, M., & Sori, A. A. (2015). Investigating Automated Test Patterns in Erratic Tests by Considering Complex Objects. *International Journal of Information Technology and Computer Science*, 7, 54–59.
- Jain, V., & Rajnish, K. (2018). Comparative Study of Software Automation Testing Tools: OpenScript and Selenium. *International Journal of Engineering Research and Application*, 8, 29–33.
- JMeter, <https://jmeter.apache.org>, Retrieved 5 April 2020.
- Kakaraparth, D. (2017). Overview And Analysis Of Automated Testing Tools: Ranorex, Test Complete, Selenium. *International Research Journal of Engineering and Technology*, 4, 1575–1579.
- Katalon Studio, <https://www.katalon.com>, Retrieved 1 April 2020.
- Kaur, H., & Gupta, G. (2013). Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete. *International Journal of Engineering Research and Applications*, 3, 1739–1743.
- Kaur, M., & Kumari, R. (2011). Comparative Study of Automated Testing Tools: TestComplete and QuickTest Pro. *International Journal of Computer Applications*, 24, 1–7.
- Kaur, P. (2021). Comparison of Automation Testing Tools for Regression Testing website, *International Journal of Innovative Science. Engineering & Technology*, 8, 259–265.
- Križanić, J., Grgurić, A., Mošmondor, M., & Lazarevski, P. (2010). Load testing and performance monitoring tools in use with AJAX based web applications. *International Convention on Information and Communication Technology, Electronics, and Microelectronics*.
- Kundu, S. (2012). Web Testing: Tool, Challenges, and Methods. *International Journal of Computer Science*, 9, 481–486.
- Kunte, P., & Mane, D. (2017). Automation Testing of Web-based application with Selenium and HP UFT (QTP). *International Research Journal of Engineering and Technology*, 4, 2579–2583.
- LoadRunner, <https://www.microfocus.com/en-us/products/loadrunner-professional/overview>, Retrieved 3 May 2020.
- Mahmood, H., & Sirshar, M. (2017). A Case Study of Web Based Application by Analyzing Performance of a Testing Tool. *International Journal of Education and Management Engineering*, 7, 51–58.
- Masso, J., Pino, F. J., Pardo, C., García, F., & Piattini, M. (2020). Risk management in the software life cycle: A systematic literature review. *Computer Standards and Interfaces*, 71, 103431.

- Meenu, & Kumar, Y. (2015). Comparative Study of Automated Testing Tools: Selenium, SoapUI, HP Unified Functional Testing and Test Complete. *Journal of Emerging Technologies and Innovative Research*, 2, 42–48.
- Monier, M., & El-Mahdy, M. M. (2015). Evaluation of automated web testing tools. *International Journal of Computer Applications Technology and Research*, 4, 405–408.
- Patil, D. S., & Pawar, P. B. (2021). Selection Of Test Automation Tools forBFSI Sector Web Based Applications. *Natural Volatiles & Essential Oils*, 8, 10645–1065.
- Petukhov, A., & Kozlov, D. (2008). Detecting security vulnerabilities in web applications using dynamic analysis with penetration testing. *Application Security Conference*.
- Prasad, L., Yadav, R., & Vore, N. (2021). A Systematic Literature Review of Automated Software Testing Tool. *International Conference on Computing Informatics and Networks*, 101–123.
- Qasaimeh, M., Shamlawi, A., & Khairallah, T. (2018). Black Box Evaluation Of Web Application Scanners: Standards Mapping Approach. *Journal of Theoretical and Applied Information Technology*, 96,4584–4596.
- QTP, <https://www.tutorialspoint.com/qtp>, Retrieved 10 May 2020.
- Radhakrishna, S., & Nachamai, M. (2017). Performance Inquisition of Web Services Using Soap UI and Jmeter. *IEEE International Conference on Current Trends in Advanced Computing*.
- Ranorex Studio, <https://www.ranorex.com>, Retrieved 12 May 2020.
- SahiPro, <https://sahipro.com>, Retrieved 2 May 2020.
- Selenium, <https://www.selenium.dev>, Retrieved 30 May 2020.
- Sharma, M., & Angmo, R. (2014). Web-based Automation Testing and Tools. *International Journal of Computer Science and Information Technologies*, 5, 908–912.
- Sharmila, S., & Ramadevi, E. (2014). Performance Evaluation and Comparison of Web Application Testing Tools: WAPT Pro and Apache Jmeter. *International Journal for Scientific Research and Development*, 2, 519–522.
- Singh, I., & Tarika, B. (2014). Comparative Analysis of Open Source Automated Software Testing Tools: Selenium, Sikuli and Watir. *International Journal of Information and Computation Technology*, 4, 1507–1518.
- Singh, J., & Sharma, M. (2015). Performance Evaluation and Comparison of Sahi Pro and Selenium Webdriver. *International Journal of Computer Applications*, 129, 23–26.
- Srivastava, N., Kumar, U., & Singh, P. (2021). Software and Performance Testing Tools. *Journal of Informatics Electrical and Electronics Engineering*, 2, 1–12.
- Telerik Studio, <https://www.telerik.com/teststudio>, Retrieved 7 September 2021.
- TestComplete, <https://smartbear.com/product/testcomplete/overview>, Retrieved May 3 2020.
- TestingWhiz, <https://www.testing-whiz.com>, Retrieved 12 May 2022.
- TestIO, <https://test.io>, Retrieved 1 May 2022.
- Webload, <https://www.radview.com>, Retrieved 3 October 2021.