



Load testing method for web applications

Dr. Rehab F. Hassan
110019@uotechnology.edu.iq
 009647700481784
 University of technology
 Department of computer science
 Iraq, Baghdad

Hussein Kutaiba,
Husseinkutaiba40@yahoo.com
 +9647713212992
 University of technology
 Department of computer science
 Iraq, Baghdad

Dr. Raheem Oglaa
110027@uotechnology.edu.iq
 009647902426768
 University of technology
 Department of computer science
 Iraq, Baghdad

Abstract

Web testing is very important method for users and developers because it's give the ability to detect errors in performance, user interface, security and other different types of errors that may occur in web application. This study focuses on a major branch of performance testing, which is called load testing. Load testing depends on an important element called response time. From this element (response time) its can be decide if the performance of a web application is poor or not. It's shown in results section if the test requests of the users takes more time than is determined in global standards then the load testing of this web application is weak . The connectivity speed plays also important role in perform the results. The proposed method depends on the request and response time to determine if the results are good or bad. Load testing applies on (Facebook) and after extract results it was found that the number of users and the speed of internet plays a major role in the response time.

Keywords--performance testing, load testing, response time, web testing.

طريقة اختبار التحميل لتطبيقات الويب

د. رحاب ف. حسن

009647700481784/ 110019@uotechnology.edu.iq

جامعة التكنولوجيا / قسم علوم الحاسوب / العراق بغداد
 د. حسين قتيبة

Husseinkutaiba40@yahoo.com +9647713212992

جامعة التكنولوجيا / قسم علوم الحاسوب / العراق بغداد
 د. رحيم عقلة

009647902426768 / 110027@uotechnology.edu.iq

جامعة التكنولوجيا / قسم علوم الحاسوب / العراق بغداد

الملخص - يعد اختبار الويب طريقة مهمة جداً للمستخدمين والمطورين لأنه يمنح القدرة على اكتشاف الأخطاء في الأداء وواجهة المستخدم والأمان وأنواع أخرى مختلفة من الأخطاء التي قد تحدث في تطبيقات الويب. تركز هذه الدراسة على فرع رئيسي من اختبار الأداء ، والذي يسمى اختبار الحمل. يعتمد اختبار الحمل على عنصر مهم يسمى وقت الاستجابة. من هذا العنصر (وقت الاستجابة) يمكن تحديد ما إذا كان أداء تطبيق الويب ضعيفاً أم لا. يظهر في قسم النتائج إذا كانت طلبات الاختبار للمستخدمين تستغرق وقتاً أطول مما هو محدد في المعايير العالمية ، فإن اختبار التحميل لتطبيق الويب هذا يكون ضعيفاً. تلعب سرعة الاتصال أيضاً دوراً مهماً في أداء النتائج. تعتمد الطريقة المقترحة على الطلب ووقت الاستجابة لتحديد ما إذا كانت النتائج جيدة أم سيئة. يتم تطبيق اختبار التحميل على (Facebook) وبعد استخراج النتائج تبين أن عدد المستخدمين وسرعة الإنترنت يلعبان دوراً رئيسياً في وقت الاستجابة.

الكلمات المفتاحية - اختبار الأداء ، اختبار الحمل ، وقت الاستجابة ، اختبار الويب.



Related work

Rijwan Khan, ,Mohd Amjad [1] explain that the Performance testing determines how fast some aspects of the system perform under pre-defined workload and it is calculated by analyzing the production, which comes from the application hosted on the server. The main performance is calculated when business is at its peak by its hits. So [1] used load testing with the help of HP ALM tool. Throughput and hits per second observed during the load test have been found. An application has been developed and tested on this tool, and it observed all the load tests that have been found more suitable with all other tools. Throughput and hits per second were shown to be stabilized during the load testing with HP ALM tool.

Guangzhou Jiang and Shujuan Jiang [2] have presented a quick test model for performance testing, based on testing flow of web applications. They introduced a new performance index called successful request rate in order to test the performance of web application. They also contributed a testing method, combined with Load Runner tool to provide effective solution to quick performance testing. The quick testing process begins with planning the test, followed by Load Runner script creation, scenario definition, execution and result analysis.

Menascé, Daniel [3] discover way to assess IT infrastructure performance is through load testing, which lets you assess how your Web site supports its expected workload by running a specified set of scripts that emulate customer behavior at different load levels. [3] Describe the QoS factors load testing addresses, how to conduct load testing, and how it addresses business needs at several requirement levels. High-volume Web sites are becoming more complex due to several factors, including the use of third-party services, such as CDNs and ad networks, geographical distribution and duplication, streaming media features, and wireless access. These factors significantly affect Web sites performance and scalability, and pose challenges to designers and users of load-testing tools and services.

I. INTRODECTION

In web world, every user wants everything to be fast but at the same time there are fears about reliability of usage. Many users decide to give up if the page takes long time to be loaded. The interest of people will be lost in business if web application is not running fast. In these days every organization use the internet to conduct there on line business, so the performance of web application is very important for any organization [1] [3]. Performance means information providing in an accurately and quickly way despite the interaction of high multi-user or resources of constrained hardware and the ability of system transactions finishing [4]. Performance Testing include recording and monitoring the levels of performance during high, orderly and low stress load.

The process of executing application with the intent of discovering software errors called software testing [5]. This process is very important level of software development life cycle



(SDLC). The role of Software testing is to detect bugs and tries to remove it; in turn this leads to increase the software reliability and quality of software [6]. The main goal of software testing is to evaluate the capacity or attribute of program and determine if it satisfies the quality of serves (QOS) [7]. Software testing that focuses on web application called web testing and its consist of several types as shown in figure(1):

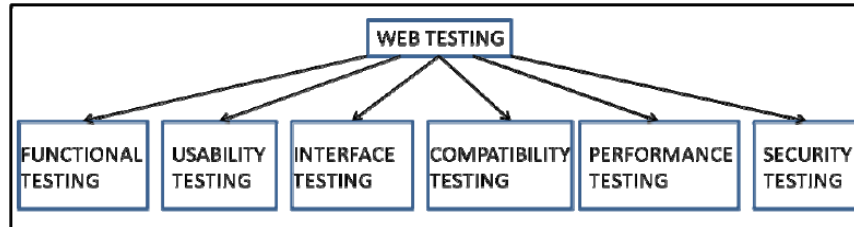


Fig.1. Web testing methods

- **Functional testing:** The main role of this testing is to check all links in webpage as (test all internal links, test the outgoing links from all pages, and check for all broken links), cookie testing (cookie is very small files that store in the user machine, it's used to maintain login sessions and its can test the application in browsers options by disabling or enabling the cookie), test forms (forms used to get information from users and to keep the interaction with them and test forms include option to create form if any user try to view or delete or modify the forms, check all validation on each field, check for field default values). Functional testing also includes database testing that is used to check the data integrity in web application.

- **Usability testing:** It's include navigation and content testing, navigation means different controls like buttons and the main role of navigation is how the user surfs the webpage and how user use the link to surf different pages. Content testing means that the content must be easy and logical to understand and check for spelling errors.

- **Interface testing:** It's consisting of different steps :

1. Web server: its include testing all application request without any rejection.
2. Database server: it's makes sure that the queries had been sent to database and give Expected results [8].
3. App: it must send the test request to database correctly and must display the output at client side. Error should be detected by the application and must perform to the administrator not end user.

- **Compatibility testing:** is important testing and it's include:

1. Operating system compatibility: some web application functionality may not compatible with all operating system such as graphics design and API may not be available in all operating system.



2. Browser Compatibility: many applications depend on browser. Different browser has different setting and configurations, so that web application must compatible with specific application like explorer, Firefox, safari.
- **Performance testing:** It's the validation that the system meets performance requirements and it consists of load and stress testing. This paper will focuses on load testing which is The main goal of it is to test the application for measuring and make the report under an anticipated live load. The result during this type of testing can be end-user response time, CPU response time and memory statistics. These results give tester data to perform on it for better response on the website application and will explain in detail in next sections.
- **Security testing:** Web application security testing is the process of testing, analyzing and reporting on the security level and/or posture of a Web application. It is used by Web developers and security administrators to test and gauge the security strength of a Web application using manual and automated security testing techniques. The key objective behind Web application security testing is to identify any vulnerabilities or threats that can jeopardize the security or integrity of the Web application [8].

II. PERFORMANCE TESTING

When web application takes minutes to load, it can say that it is frustrating comparing with other sites that download similar content in seconds, when someone trying to log in web application and receiving a message like "server busy" its can say that it is aggravating, when web application responds in some situations and its goes into infinite wait state in other different situation its can say that it is disconcerting. All these problems happen in web and these problems are related to performance. Performance testing is used to discover performance defects that can results from weakness of network bandwidth, lack of server side, weakness of operating system capabilities and other software or hardware problems. Performance testing used to simulate the real word loading situations like increasing in number of online transactions, amount of data, the increasing number of simultaneous web application users [9].

The advantage of Performance testing is to determine the throughput and response time of any required web application [10]. Throughput observes how many transactions per second an application can handle during the test, throughput indicting the amount of transactions produced over time. Throughput depends on different elements like,

1. Types of transactions being processed
2. Specifications of the host computer
3. Processing overload in software [10]

Response time is the time that begin at sending the request by user until the application observe that the request has completed as shown in details in section (A. Response time) [11].



Tools of Performance testing are used to determine how much time will be taken for executing a task by the system [12]. Performance testing is used to check if the non-functional requirements in document of Software Requirement Specification (SRS) are meets the system or not. In the current time most of the websites builders or developers knows that it's necessary to test the performance before starting [13]. Performance testing consists of load and stress testing. Load testing measures the response time and the main goal of it is to see if the application can maintain the increasing load on the server or not and this paper focus on the load testing and will explain in detail in section (3). Stress testing is just like load testing but is go further than load testing in increasing the load on the server, the main goal of stress testing is to test the insane limits of the application [14].

III. LOAD TESTING

It is the most used techniques for measuring performance, the major goal for this method is to specify anticipated peak load condition and behavior of web applications, it means while handling the specific load given by the user to the system this testing provide many information about system behavior [3]. This testing process is based on gradual rising in resources. It means that the test usually start to load the tested web application with limited number of virtual users and gradually increasing the number from normal to peak As shown in Figure(2). Load testing provides the measurement for the application QOS (quality of services) performance by depending on actual customers' behavior. To build interaction scripts, a script recorder uses customers' requests. Then scripts will be replayed by the load generator module, possibly modified by test parameters against the Web site.

A. Response time

The major key in (QOS) is response time. The response time must be measured to define how customers perceive things, for instance keyword search times and page download. When defining response time, the different between the bases HTML pages download time and the other page components like (images, ad banners) should be notice. Response time for web application varies according to many different elements such as:

1. The customer's *internet service provider*.
2. The testing site IPS.
3. Bandwidth.
4. Which network route packets from the customer to the testing website?

Obviously, measuring response time from specific time window and geographical location will not give a full picture. Response time depends on time and space. It must be known how the customers from different locations with different connectivity perceive your site performance at different times [3].

B. How load testing works?

To accomplish the load testing process, the load generator module of the tester, see figure (2) simulates browser behavior; it continuously sends requests to the Web site and then after the site sends a reply to the request. It waits for a period of time and this is called (the think time) and then submits a new request. The load generator can simulate many of concurrent users to test the scalability of web. Each simulated browser is called a virtual user, which is a key load-testing concept. If virtual users' behavior has characteristics similar to those of actual users then the load test is valid. So it must be sure that those virtual users:

- Use realistic think times.
- react like abandoning a Web session if response time is excessive, frustrated users
- Follow patterns similar to real users.

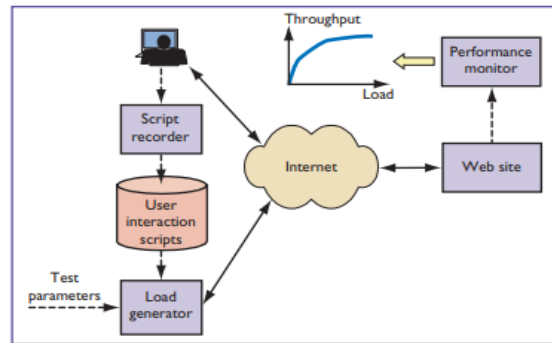


Fig.2. Load testing process

Failure to simulate real user behavior can create inconsistent results. Because customers who left a session use lower site resources than those who finish it. Failure to simulate behavior of real user can create totally unbalance results.

IV. METHOD

In order to implement load testing a specific web site should be selected and the following steps should be performed to get the desired results:

1. Give the complete URL for the tested site.
2. Specify the number of virtual users that will access the site. It is preferable for the user number to be in a range between 100 – 1000 to get the desirable results.
3. Specify the time interval accepted for adding virtual users, for example 10 users/ 5 seconds.
4. Start the simulation process, where users should access different accounts, if the tested web provides accounts for each user.
5. The simulation results such as statistics results and other information will be displayed, as will explain in the next section.
6. Terminate the simulation process at a specific time, and the gathered results should be analyzed and studied to decide the loading capability of the tested web.



Four elements should be considered when designing load testing module as shown in figure(3) and should be shown within the results, these elements are:-

A- Statistics: it's consist of several elements as shown in tables (1, 2), these elements are:

- **Type:** includes the type of http protocol request or method, if it is (get) then it deals with asking the users to input homepage and a specific account or profile page and they must be entered before the program started. If it is (post) then its deals with the login (), logout () functions.
- **Name:** consist of (homepage, profile page, login (), logout ()) as shown in tables (1, 2).
- **Requests:** In this option, it must be collect the tests requests that will be executed from homepage, profile page, login () and logout ().
- **Fails:** This element explains the users test requests that will be not executed because of different reasons like connection broken or low bandwidth etc... there are different faults that will explain in detail in failures field.
- **Median (MS):** Is a filter that measured by millisecond and it used to arrange the waiting time to execute test requests from maximum time to minimum time or on the contrary and then (median) choose the middle wait time.
- **Average:** also measured by millisecond, this option collects all waiting times to execute the request and divided it on its numbers to get the average time.
- **Min (MS) & Max (MS):** (Min) is the Minimum time to execute user request and the (Max) is maximum time to execute user request.
- **Content size (bytes):** This field is calculates the weight of the required page in bytes.
- **Request/sec:** it calculates how much time will take to execute total requests.

B- Chart: It consists of number of users, response time and total request per second, it will explain in detail in result section.

C- Failures: It explains why the requests have been failed as shown in figure (7).

D- Download data: The purpose of this field is to download the statistics, charts; data and failures in excel files.

Statistics Charts Failures Exceptions Download Data									
Type	Name	# requests	# fails	Median (ms)	Average (ms)	Min (ms)	Max (ms)	Content Size (bytes)	# reqs/sec
GET	//	90	331	87000	100762	39877.8185844214	223667.81783103943	653283	0.3
POST	//login	6	363	176000	225590	98720.72744369507	450281.0177707672	616365	0
POST	//logout	0	789	0	0	0	0	0	0
GET	//profile	38	176	100000	120953	42328.59396934509	510384.07921791077	653273	0.1
Total		134	1659	95000	112077	39877.8185844214	510384.07921791077	651627	0.4

Fig.3. Load testing form

V. RESULTS AND DISCUSSION



Table (1) represents the statistics for entering 100 users. It's shown in (C column) that (206) requests has been implemented, in (failures) the requests that doesn't implemented and the reason why it's been failed shown in figure (7), (median) its take the middle response time and (requests/sec) represent the time needed to implement the request.

Table 1 statistics result for 100 users.

	A	B	C	D	E	F	G	H	I	J
1	Method	Name	# requests	# failures	Median re	Average re	Min respo	Max respc	Average C	Requests/s
2	GET	//	206	122	58000	85315	22276	642899	669873	0.23
3	POST	//login	29	58	69000	154626	38070	520548	633365	0.03
4	POST	//logout	0	80	0	0	0	0	0	0
5	GET	//profile	87	51	59000	72537	24391	282200	669873	0.1
6	None	Total	322	311	59000	88105	22276	642899	666585	0.36

Figure (4) show the maximum response time in (600000 MS), when users reach the peak. Figure (5) shows the number of 100 users that have been entered. Figure (6) shown the total requests per- second, how many user requests in one second. Figure (7) shows the failure requests and the reasons why it's had been failed.

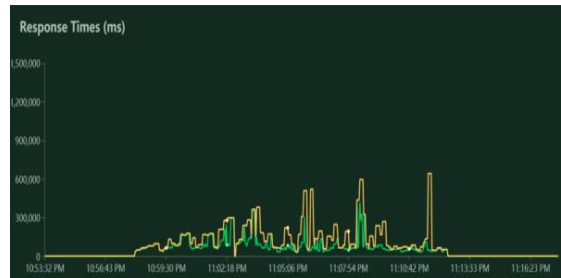


Fig.4. Response time for 100 users, X-axis is response time, Y-axis estimation time

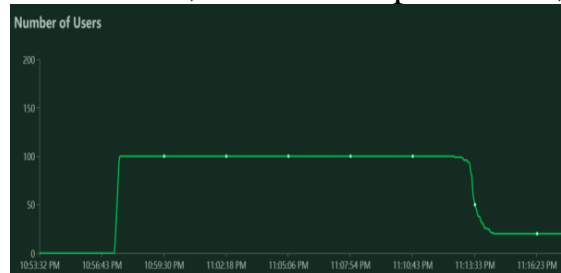


Fig.5. Number of 100 users, X-axis is the taken Time to reach 100 users, Y--axis is number of users

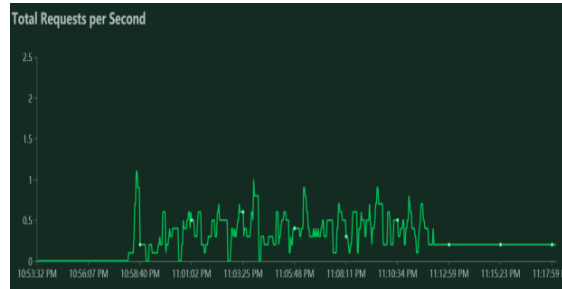


Fig.6. Total requests per second X-axis is response time, Y-axis estimation time

#	Method	Name	Type
35	POST	/logout	ChunkedEncodingError(ProtocolError('Connection broken: OSError("10054, WSARECONNRESET")', OSError("10054, WSARECONNRESET"))))
39	POST	/logout	HTTPError(404 Client Error: Not Found for url: https://www.facebook.com/logout/)
8	POST	/logout	SSLError(MaxRetryError(HTTPConnectionPool(host='www.facebook.com', port=443): Max retries exceeded with url: /logout (Caused by SSLError(SSLError('bad handshake: SysCallError(-1, 'Unexpected EOF'))',)))
1	POST	/logout	SSLError(MaxRetryError(HTTPConnectionPool(host='www.facebook.com', port=443): Max retries exceeded with url: /logout (Caused by SSLError(SSLError('bad handshake: SysCallError(-1, 'Unexpected EOF'))',)))
1	POST	/logout	ConnectionError(ProtocolError('Connection aborted.', OSError("10054, WSARECONNRESET"))))
49	GET	/profile	ChunkedEncodingError(ProtocolError('Connection broken: OSError("10054, WSARECONNRESET")', OSError("10054, WSARECONNRESET"))))
114	GET	/	ChunkedEncodingError(ProtocolError('Connection broken: OSError("10054, WSARECONNRESET")', OSError("10054, WSARECONNRESET"))))
4	GET	/	ConnectionError(ProtocolError('Connection aborted.', OSError("10054, WSARECONNRESET"))))
4	GET	/	SSLError(MaxRetryError(HTTPConnectionPool(host='www.facebook.com', port=443): Max retries exceeded with url: / (Caused by SSLError(SSLError('bad handshake: SysCallError(-1, 'Unexpected EOF'))',)))
51	POST	/login	ChunkedEncodingError(ProtocolError('Connection broken: OSError("10054, WSARECONNRESET")', OSError("10054, WSARECONNRESET"))))
3	POST	/login	SSLError(MaxRetryError(HTTPConnectionPool(host='www.facebook.com', port=443): Max retries exceeded with url: /login (Caused by SSLError(SSLError('bad handshake: SysCallError(-1, 'Unexpected EOF'))',)))
4	POST	/login	ConnectionError(ProtocolError('Connection aborted.', OSError("10054, WSARECONNRESET"))))

Fig.7. Failure requests with reasons

Table (2) represents the statistics for entering 1000 users. It's shown in (requests) that (126) requests have been implemented, in (failures) the requests that doesn't implemented and the reason why it's been failed shown in figure (7), (median) its take the middle response time and (requests/sec) represent the time needed to implement the request. Figure (8) shows that the maximum response time in (900000 MS) when users reach the peak. Figure (9) shows the number of 1000 users that been entered. Figure (10) shows total requests per second and how many user requests in one second.

Table 2 statistics for 1000 users

	A	B	C	D	E	F	G	H	I	J
1	Method	Name	# requests	# failures	Median re	Average re	Min respo	Max respc	Average C	Requests/s
2	GET	/	126	512	88000	99496	28325	407857	669881	0.14
3	POST	/login	10	395	217000	267587	95131	800591	632850	0.01
4	POST	/logout	0	546	0	0	0	0	0	0
5	GET	/profile	60	260	78000	101978	31829	615387	669884	0.07
6	None	Total	196	1713	88000	108832	28325	800591	667993	0.21

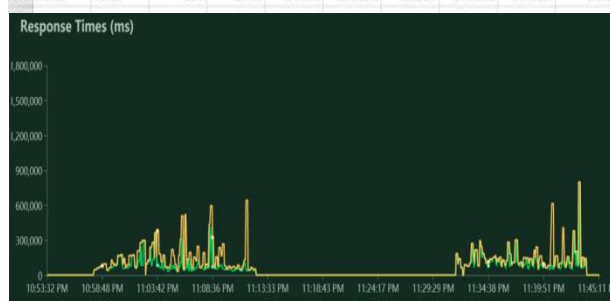


Fig.8. Response time for 1000 users, X-axis is response time, Y-axis estimation time

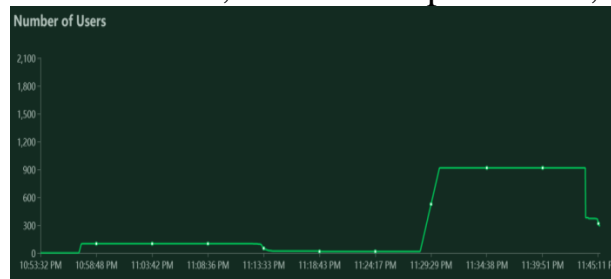


Fig.9. Number of 1000 users, X-axis is the taken Time to reach 1000 users, Y--axis is number of users

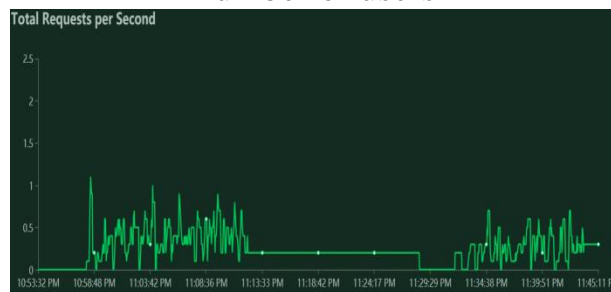


Fig.10. Total requests per second X-axis is response time, Y-axis estimation time

The load testing have been run for (100-1000) users the results between them are different because the connectivity speed and the time it had been tested (11.00pm). When the load testing had been runs for 100 users the connectivity speed was (11.6 Mbps to 8.7 Mbps) and the test was run for 15 minutes. Then the load testing run for 1000 users the connectivity speed was (5.8 Mbps to 8.7 Mbps) and the test was run for 15 minutes.

Finally, Users are simulated in this system dependent on the characteristics and human nature to keep pace with the free world.

VI. CONCLUSION

Load testing is very useful method for commercial center but From the results that shown above, its can concluded that the bandwidth or internet speed and the number of users plays the major role in the load testing and sometimes may adversely affect and could obstruct evaluated the application in the right way, so there are many requests that had been not tested for different reasons as shown in figure (7). so when load testing applied for 100 users the results was good but when load testing applied for 1000 the requests failure became increases because the number of users become high and the internet speed is less than pervious test and response time for 100 users and 1000 users was different. So the better results came with better internet speed.

REFERENCE

- [1] R. Khan and M. Amjad, "Performance testing (load) of web applications based on test case management," *Perspect. Sci.*, vol. 8, pp. 355–357, 2016.



- [2] G. Jiang and S. Jiang, "A quick testing model of Web performance based on testing flow and its application," in *Web Information Systems and Applications Conference, 2009. WISA 2009. Sixth*, 2009, pp. 57–61.
- [3] D. A. Menascé, "Load testing of web sites," *IEEE Internet Comput.*, vol. 6, no. 4, pp. 70–74, 2002.
- [4] H. Sarojadevi, "Performance testing: methodologies and tools," *J. Inf. Eng. Appl.*, vol. 1, no. 5, pp. 5–13, 2011.
- [5] E. T. Barr, M. Harman, P. McMinn, M. Shahbaz, and S. Yoo, "The oracle problem in software testing: A survey," *IEEE Trans. Softw. Eng.*, vol. 41, no. 5, pp. 507–525, 2015.
- [6] R. Bathla and S. Bathla, "Innovative approaches of automated tools in software testing & current technology as compared to manual testing," *Glob. J. Enterp. Inf. Syst.*, vol. 1, no. 1, pp. 127–131, 2009.
- [7] S. Maheshwari, D. C. Jain, M. S. Maheshwari, and D. C. Jain, "A Comparative Analysis of Different types of Models in Software Development Life Cycle," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 2, no. 5, pp. 285–290, 2012.
- [8] S. Kundu, "Web testing: Tool, challenges and methods," *IJCSI Int. J. Comput. Sci. Issues*, vol. 9, no. 2, pp. 814–1694, 2012.
- [9] R. Pressman and D. Lowe, *Web Engineering: A Practitioner's Approach*. McGraw-Hill, Inc., 2008.
- [10] M. S. Sharmila and E. Ramadevi, "Analysis of performance testing on web application," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 3, no. 3, pp. 2319–5940, 2014.
- [11] <https://loadfocus.com/blog/2013/07/04/what-is-throughput-in-performance-testing>. Have been visited at 29Nov18.
- [12] <https://loadfocus.com/blog/2013/07/03/what-is-response-time-in-performance-testing>. Have been visited at 02Dec18
- [13] M. Kaushik and P. Fageria, "A Comparative Study of Performance Testing Tools," vol. 3, no. 5, pp. 1300–1307, 2014.
- [14] D. M. Kaushik and P. Fageria, "Performance testing tools: A comparative study," *Int. J. Innov. Sci. Eng. Technol.*, 2014.