National Workshop-Cum-Conference on Recent Trends in Mathematics and Computing (RTMC) 2011
Proceedings published in International Journal of Computer Applications® (IJCA)

Comprehensive Approach for Cross Compatibility Testing of Website

Sanjay Dahiya[1], Ved Parkash[1], T.R. Mudgal[2]
[1]Lecturer, Deptt. of CSE, CDLM Govt. Engineering College, Panniwala - Mota (Sirsa) (Haryana)
2Lecturer, Deptt. of Physics, CDLM Govt. Engineering College, Panniwala - Mota (Sirsa) (Haryana)

ABSTRACT
Developing a website is one of the most challenging processes in computer science and the instant worldwide audiences of any Web Browser Enabled Application make its quality and reliability crucial factors in its success. But the technical complexities of a website and variances in the browsers, operating systems, and devices make testing and quality control much more difficult and in some ways, more subtle, than "conventional" client/server or non-web application. So which configuration should we test for our website, become a major ongoing challenging effort. Automated testing of websites is an opportunity and a challenge. The aim of this paper is to discuss the parameters that a website should meet before its launching worldwide and suggest some cross browsers automated testing tools which help in website testing on various browsers, operating systems, and devices and meet technical needs for assuring website quality.

Keywords:
Web-testing, Web-browser, Web quality and reliability, Client/Server, Compatibility Testing.

1. INTRODUCTION:
Web testing is the name given to software testing that focuses on web applications. Comprehensive testing of a web-based system before going live can help address issues before the system is revealed to the public. Generally the websites are essentially client/server applications with web servers and 'browser' clients. Consideration should be given to the interaction between html pages, TCP/IP communications, Internet connections, firewalls, application that run in web pages (such as applet, JavaScript, plug-in application), and applications that run on the server side (such as cgi script, database interfaces, logging applications, dynamic page generators, asp etc.)[5]. Web testing is done for 3 tier applications (developed for Internet/Intranet/Extranet). Here we will be having browser, web server and Data Base server. The applications accessible in browser would be developed in HTML, DHTML, XML, JavaScript etc. Applications for the web server would be developed in Java, ASP, JSP, VBScript, JavaScript, Perl, Cold Fusion, PHP etc.[5][7]. All the manipulations are done on the web server with the help of these programs developed. The Data Base-server would be having oracle, SQL server, Sybase, my-sql etc. All data is stored in the database available on the Data Base server.

Also the issues such as the security of the web application, the basic functionality of the site, its accessibility to users as well as readiness for expected traffic and number of users and the ability to survive a massive spike in user-traffic,

both of which are related to load testing. One of the most challenging processes is testing the site in wide array of different browser, operating systems, and devices. Gone are the days where all we needed to test in was IE and Netscape, for displaying resolution of 600*800 via window 98. Now we are taking dozens of browsers, handful operating systems, and a mix of monitor resolutions [8]. With a wide variety of servers, browsers, operating systems and different devices, website

Testing can become a major ongoing effort. So which dimensions should we test for our website is very challenging task. It is not possible for us to test all the quality dimensions of the website manually. Therefore, automated testing plays a very important role in website testing and automated testing tools are extremely helpful in the testing of website with all dimensions.

2. GENERAL DIMENSIONS OF WEBSITE QUALITY & RELIABILITY:
There are many dimensions of quality; each measure will pertain to a particular website in varying degrees. Here are some common measures [2]:

- Timeliness: websites change often and rapidly. How much has a website changed since the last upgrade? How do we highlight the parts that have changed?
- Structural Quality: How well do all of the parts of the website hold together? Are all links inside and outside the website working? Do all of the images work? Are there parts of the website that are not connected?[3]
- Content: Does the content of critical pages match what is supposed to be there? Do key phrases exist continually in highly-changeable pages? Do critical pages maintain quality content from version to version? What about dynamically generated HTML /DHTML pages?
- Accuracy and Consistency: Are today's copies of the pages downloaded the same as yesterday's? Close enough? Is the data presented to the user accurate enough? How do we know?
- Response Time and Latency: Does the Website server respond to a browser request within certain performance parameters? In an e-commerce context, how is the end-to-end response time after a SUBMIT? Are there parts of a site that are so slow the user discontinues working? [3]
- Performance: Is the Browser --> Web --> ebSite --> Web --> Browser connection quick enough? How does the performance vary by time of day, by load and usage? Is performance adequate for e-commerce applications? Taking 10 minutes -- or maybe even only 1 minute -- to respond to an e-commerce purchase may be unacceptable!

3. PARAMETERS OR WEB TESTING CHECKLIST THAT A WEBSITE MEET

- Here we will discuss some parameters that a website should meet in its life span in concerned of its quality and reliability, so that websites fulfill its objectives.

3.1 Functionality Testing
test for all the links in web pages, database connection, forms used in the web pages for submitting or getting information from user, Cookie testing [2]:

➢ **Check all the links:**

- Test the outgoing links from all the pages from specific domain under test.
- Test all internal links.
- Test links jumping on the same pages.
- Test links used to send the email to admin or other users from web pages.
- Test to check if there are any orphan pages.

- Lastly in link checking, check for broken links in all above-mentioned links.

➢ **Test forms in all pages:** Forms are the integral part of any web site. Forms are used to get information from users and to keep interaction with them. So what should be checked on these forms? [3]

- First check all the validations on each field.
- Check for the default values of fields.
- Wrong inputs to the fields in the forms.
- Options to create forms if any, form delete, view or modify the forms.

➢ **Cookies testing:** Cookies are small files stored on user machine. These are basically used to maintain the session mainly login sessions. Test the application by enabling or disabling the cookies in our browser options. Test if the cookies are encrypted before writing to user machine. If we are testing the session cookies (i.e. cookies expire after the session's ends) check for login sessions and user stats after session end. Check effect on application security by deleting the cookies.

➢ **Validate our HTML/CSS:** If we are optimizing our site for Search engines then HTML/CSS validation is very important. Mainly validate the site for HTML syntax errors. Check if site is crawlable to different search engines.

➢ **Database testing:** Data consistency is very important in web application. Check for data integrity and errors while we edit, delete, modify the forms or do any DB related functionality. Check if all the database queries are executing correctly, data is retrieved correctly and also updated correctly. More on database testing could be load on Database; we will address this in web load or performance testing below [2].

3.2 Usability Testing:

➢ **Test for navigation:** Navigation means how the user surfs the web pages, different controls like buttons, boxes or how user using the links on the pages to surf different pages. Usability testing includes: Web site should be easy to use. Instructions should be provided clearly. Check if the provided instructions are correct means whether they satisfy purpose. Main menu should be provided on each page. It should be consistent [3].

➢ **Content checking:** Content should be logical and easy to understand. Check for spelling errors. Use of dark colors annoy users and should not be used in site theme. We can follow some standards that are used for web page and content building. Content should be meaningful. All the anchor text links should be working properly. Images should be placed properly with proper sizes. These are some basic standards that should be followed in web development.

➢ **Other user information for user help:** Like search option, sitemap, help files etc. Sitemap should be present with all the links in web sites with proper tree view of navigation. Check for all links on the sitemap[2].

3.3 Interface Testing:

The main interfaces are Web server and application server interface & Application server and Database server interface. Check if all the interactions between these servers are executed properly. Errors are handled properly. If database or web server returns any error message for any query by application server then application server should catch and display these error messages appropriately to users. Check what happens if user interrupts any transaction in-between? Check what happens if connection to web server is reset in between?

3.4 Compatibility Testing:

Cross compatibility of our website is very significant testing aspect. Which compatibility test to be executed is a major issue? At least following compatibility test should be executed:

➢ **Browser compatibility:** Some applications are very dependent on browsers. Different browsers have different configurations and settings that our web page should be compatible with. Our web site coding should be cross browser platform compatible. If we are using java scripts or AJAX calls for UI functionality, performing security checks or validations then give more stress on browser compatibility testing of our web application. Test web application on different browsers like Internet explorer, Firefox, Netscape navigator, AOL, Safari, Opera browsers with different versions [3].

➢ **OS compatibility:** Some functionality in our web application may not be compatible with all operating systems. All new technologies used in web development like graphics designs, interface call like different API's may not be available in all Operating Systems. Test our web application on different operating systems like Windows, UNIX, MAC, Linux and Solaris with different OS flavors.

➢ **Mobile browsing:** This is a new technology era. So in future Mobile browsing will rock. Test web pages on mobile browsers. Compatibility issues may be there on mobile.

➢ **Printing options:** If we are giving page-printing options then make sure fonts, page alignment and page graphics getting printed properly. Pages should be fit to paper size or as per the size mentioned in printing option.

3.5 Performance testing:

Web application should sustain to heavy load. Web performance testing should include Web Load Testing and Web Stress Testing. Test application performance on different internet connection speed.

➢ **Web load testing:** Test if many users are accessing or requesting the same page. Can system sustain in peak load times? Site should handle many simultaneous user requests, large input data from users, Simultaneous connection to DB, heavy load on specific pages etc.

➢ **Stress testing:** Generally stress means stretching the system beyond its specification limits. Web stress testing is performed to break the site by giving stress and checked how system reacts to stress and how system recovers from crashes. Stress is generally given on input fields, login and sign up areas.

3.6 Security Testing:

Following are some test cases for web security testing [2] [4]:

- Test by pasting internal URL directly into browser address bar without login. Internal pages should not open.
- If we are logged in using username and password and browsing internal pages then try changing URL options directly.
- Try some invalid inputs in input fields like login username, password, input text boxes. Check the system reaction on all invalid inputs.
- Web directories or files should not be accessible directly unless given download option.
- Test the CAPTCHA for automates scripts logins.
- Test if SSL is used for security measures. If used proper message should get displayed when user switch from non-secure http:// pages to secure https:// pages and vice versa.
- All transactions, error messages, security breach attempts should get logged in log files somewhere on web server.

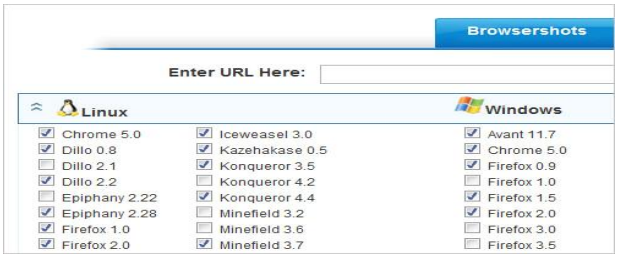
4. WEBSITE CROSS BROWSER COMPATIBILITY TESTING TOOLS.

Cross-browser testing of web applications is an important part of any developer's routine. As the number of browsers increases, and they certainly have in recent years, the need for automatic tools that can assist us in the process becomes ever greater. Here, we present an overview of different cross-browser testing applications and their services. Also, at the end we summarize key metrics and unique features for each service.

4.1 Browser Shots

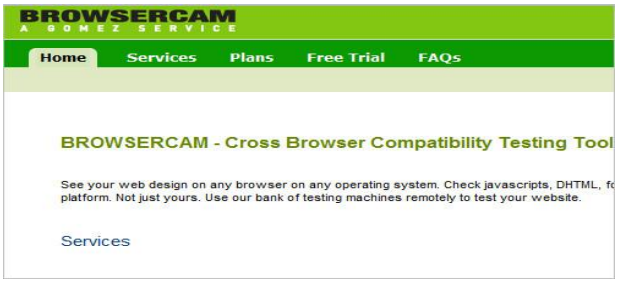
It is the oldest and best known free online multi-browser screenshot service. It supports the largest number of browsers: a total of 61 different browser versions and operating systems, which is great. It

allows us to enable and disable JavaScript, Java and Flash and change the screen size [6].

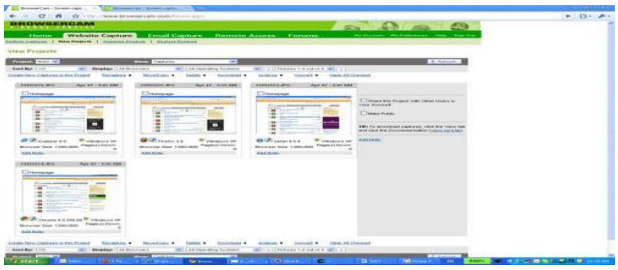


The interface is not very user-friendly. Selecting the browsers and options we want to take time, and because it is a Web service we have to do it over every time we want to take a screenshot. When we finally get our screenshots, there is no easy way to compare different captures in order to find rendering inconsistencies. HTTP redirect is not fully automated: BrowserShots displays the URL we are being redirected to, but we have to start the screenshot again manually. The biggest disadvantage of BrowserShots which makes it practically unusable for a professional developer is the response time.

4.2 BrowserCam: It is another well-known screenshot service. Unlike BrowserShots, this is a commercial service. The cheapest plan cost $159.80 a year and provides access for five users. The interface is nice. It allows us to create a project and specify the URL and browsers we want to capture, so that we do not have to do it all over again to re-test the page. But because it is a non-AJAX Web-based interface, its response time is not comparable to that of a native application, which is a bit annoying [1].

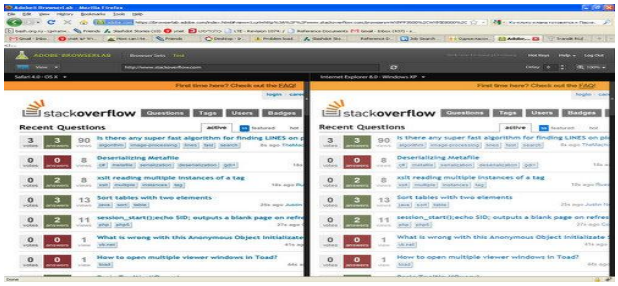


Browser support is slightly more limited than that of BrowserShots, but it is good enough for practical purposes; it supports multiple versions of IE, Firefox, Safari, Opera and Chrome, as well as some older browsers on OS X, Linux and multiple versions of Windows. Capture speed is decent: it took about two minutes to take a screenshot of our test scenario. BrowserCam supports multiple resolutions and has window and full-page capture, which means scroll bar support. Another nice feature is mobile device capture: it supports Blackberry, iPhone, Android and Windows Mobile devices. Note that mobile capture support is not part of the browser capture plan and costs $999.95 extra annually. It also has an email capture service, which in our opinion is of limited use, and remote access, which can be useful for troubleshooting rendering inconsistencies that are detected from a screen capture. Both services cost extra. The screenshot below is of a BrowserCam results window. Remote access packages allow us to connect using VNC to our choice of Linux, Windows and Mac machines with different browser versions. This can be a good option for debugging on hardware that we do not have, such as Mac. But the price of $499.95 a year is not far from the price of Mac mini, and because the VNC protocol is not terribly efficient, extensive remote debugging via VNC can be daunting.



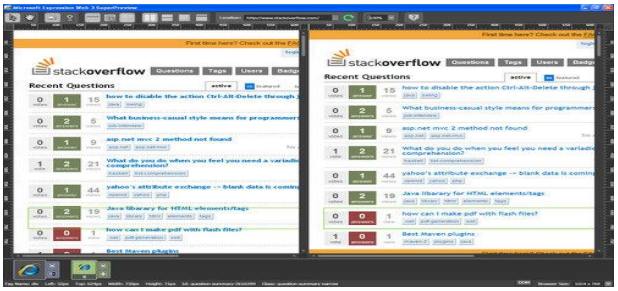
4.3 Browser Lab: It is a new offering from Adobe and was previously known as *Meer-Meer*. It is written in Flash and as such has the advantage of being cross-platform compatible and of having the look, feel and response time of an application. It is currently offered free of charge in preview mode while Adobe "is monitoring the performance." Because it will monitor it for more than one year, one wonders whether it has other reasons for this. According to Adobe, it will charge $10 to $20 per month for this service starting in 2011.

The interface is good-looking, polished and easy to use, as we can see from the screenshot below. We can inspect captures one by one or view two captures side by side, which is more useful. The much lauded "onion skin" option is not very practical: most of the time, browsers will not render a page identically pixel by pixel, but the page might still look the same. Browser support is modest compared to the competition. At the time of writing, BrowserLab supports only Chrome, Firefox, IE and Safari: a total of 12 browsers and OS version combinations. But it looks like the quality of the product is still at beta level; in two captures, it actually cut the image horizontally. Scroll bar support is buggy, too. Screenshot speed is very good [1].



4.4 Super Preview [1]: It is a new addition to Microsoft's Expression Web WYSIWYG development environment. This is the standalone version, limited to Internet Explorer and available for download free of charge. Browser support is limited. The standalone version supports only IE 6, 7 and 8, while the full version has support for Firefox and Safari.

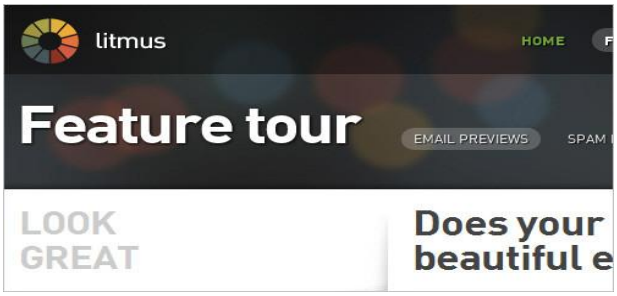
Because it is an application that runs on our PC, the response time and screenshot delay are among the best in class. It loads the website in a matter of seconds. It is noted that because SuperPreview works with only two browsers at a time and does not support Chrome, this test was not identical to that of other services. SuperPreview cannot be purchased without the Expression Web, whose retail price is $149.



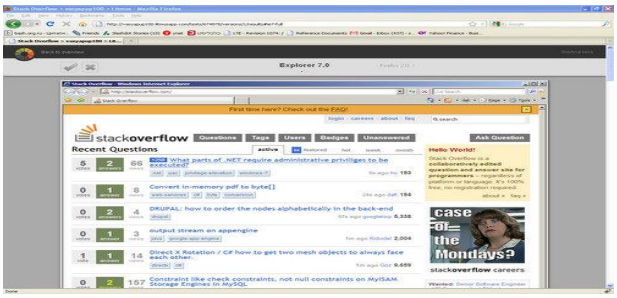
4.5 Browser Seal: It is a new tool. Similar to SuperPreview, it is an application rather than a Web service, and as such its capture speed is very good. BrowserSeal finished the test case in less than one minute. Browser support is quite broad. With multiple versions of IE, Firefox, Safari, Opera and Chrome, it pretty much covers every browser anyone would want to test. It has two good features that set it apart from the competition: standalone browser support and a command-line interface for automation scripting.

All browsers supported by BrowserSeal can be launched manually, which means that once we have found a rendering inconsistency in a browser, we can actually run the problem browser and troubleshoot the issue[1]. This is something most other services do not offer. The price is very competitive, too: the standard version sells for $49. There is also an automation version with a command-line interface that lets us capture multiple URLs from a script or batch file. The interface is attractive and easy to use. The optional tabbed interface makes it easy to spot even the slightest rendering difference when switching from one capture to another.

4.6 Litmus: It is another Web-based screenshot service. Its browser support is impressive, with 23 browser versions and operating system combinations, including IE, Firefox, Chrome, Safari, Opera, Flock, Camino, SeaMonkey and Netscape. Capture speed is okay but not comparable to that of native applications: our test took five minutes [1].

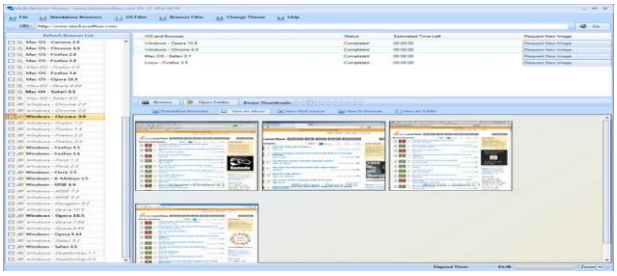


The interface is clear and simple but lacks some features. For instance, there is no easy way to compare capture results. All we can do is view them one by one or download them to our PC. The application, though, does support projects, so we don't have to enter URLs and change browser settings every time we want to take a screenshot, but this is pretty much all it does. Litmus does not support scrolling; that is, it captures only the top of long pages, which is a major drawback. The price is a bit high for a service that has such basic features: a single-user license costs $588 annually [6].



4.7 Multi-Browser Viewer: It is an application but relies on a server farm for browser rendering; in other words, the application is just a graphical interface, so it is as easy to use as an application but suffers the delays of a typical Web-based service.

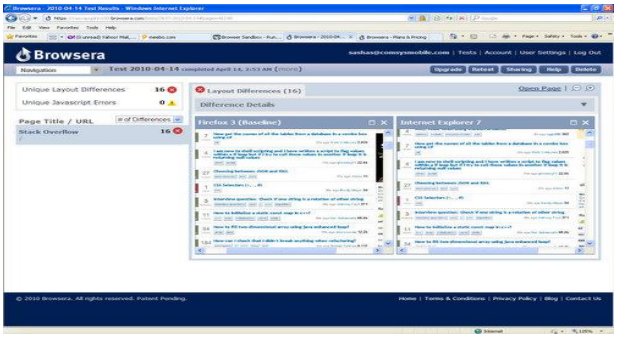
One interesting feature is that it comes with standalone browsers that can be used for debugging. But note that these are not the same browsers used for screen capture. Multi-Browser Viewer has standalone browsers that can be used for debugging, and it has a rendering farm with many more browsers that can be used for screen capture. Browser support is impressive, with 54 browser and OS version combinations (out of which 17 are available in standalone versions), including IE, Firefox, Chrome, Opera, Safari, Camino, Konqueror. The price is reasonable: a single-user license costs $129.95 annually. Feature-wise, it does lag significantly behind the competition: there is no support for authentication or capture delay. Scroll bar support is buggy; in some test cases, it worked for IE, Firefox and Safari, but not for Opera [1].



4.8 Browsera:

It is a Web-based screenshot service. Browser support is limited compared to that of most competitors: only IE, Firefox and Safari are supported. The standard plan costs $588 annually. The interface is attractive, fast and clean. We can conveniently organize our screenshot sessions into projects [1].

Browsera supports authentication, scroll bars and page crawling (i.e. Browsera to crawl your website recursively and take a screenshot of every page). The screenshot response time is very fast for a Web-based service.



4.9 Xenocode Browser Sandboxes:

The Xenocode Browser Sandbox is a game-changer for browser testing on Windows-based machines. With a single click of our mouse we can have an open and working browser without any installation. We can test in various IE versions, Firefox, Google Chrome, and even Safari. And really test, too, not just screenshots. To top everything off, the entire service is provide free of charge [6].

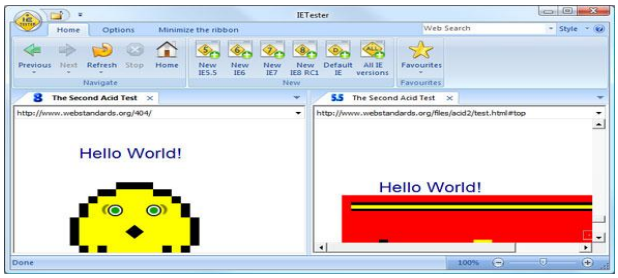
Alas, this isn't yet the perfect solution. There is currently no Macintosh support, which is definitely a significant problem. It is a rumor that this may be coming in the future, though, and at that time this service will be in a class of its own.

4.10 CrossBrowserTesting.com

Free 5 minute test sessions for registered users, and a lot more than that for paid users. CrossBrowserTesting.com makes things as easy as logging in and selecting an available machine with the browser/os we want. Once we pick our machine and browser we can begin our testing.

We can use a web-based java applet to connect to their remote test machines, or we can use a local VNC client if we have one installed. Our system allows full testing of a site's interactivity and, like Xenocode's solution, is not just screenshots [6].

4.11 IE Tester:

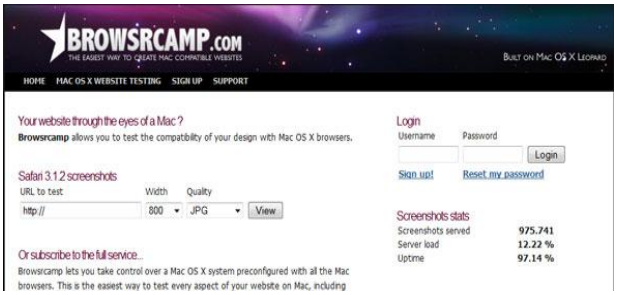


This is a free downloadable windows program that is still in the early stages of development. That being said, it is a single free resource that will allow us to fully test all of the relevant versions of Internet Explorer. Just download and install their free browser, and we can easily select which IE rendering version we want to browse in. The program even allows comparing two different versions side-by-side [6].

Sr. No.	Name of Browsers Automated Testing Tools	Supported Browsers	Capture Speed	Paid / Free	Inter-Face	Authentication	Capture Delay	Scroll Bars	Extra Special Features	Prons	Cons
1	**BrowserShots**	IE,Firefox,Chrome,Safari,Dilo,SeaMonkey,Minefield,Epiphany,Flock, Galeon,Konqueror,KMeleon,Avant,Netscape,Shireteko,Kazehakase,Iceweasel	45 Mins	Free	Avg.	No	No	No	None	Less time to install	Slow working Painfully slow.
2	**BrowserCam**	IE,Firefox,Chrome,Safari,Konqueror,Camino,Netscape,AOL	2Min	Paid	Good	Yes	Yes	Yes	Mobilebrowserssupport, remote access service	Thoughtful Interface	Expensive
3	**BrowserLab**	IE, Firefox,Chrome,Safari	1 Min	Free	Good	No	Yes	Buggy	None	Good Response Time,nice interface	Modest Browser support,Minor Bugs
4	**SuperPreview**	IE, Firefox,Safari	1 Min	Paid	Good	No	No	Yes	None .	Incredible Speed, interface is extremely easy to use	Limited support Browser, Expensive
5	**BrowserSeal**	IE, Firefox,Chrome,Safari, Opera	1 Min	Paid	Good	Yes	Yes	Yes	Standalone browser versions, support for automation scripts	Attractive Interface, Good Response Time,comprehensivebrowser support	Only Windows Support
6	**Litmus**	IE, Firefox,Chrome,Safari, Flock,Camino,SeaMonkey,Netscape	5 Min	Paid	Basic	Yes	No	No	None	Goodbrowser support,and Enough for moderate users,Average capture speed	Does not support scrolling, and lacks other standard features.
7	**Multi Browser Viewer**	IE,Firefox,Chrome,Safari,Opera,Flock,SeaMonkey,Netscape, K-Meleon,Camino,Konqueror,Epiphany,Kazehakase,	2 Min	Paid	Good	No	No	Buggy	Standalone Browser Versions	Good interface, impressive browser support	Only Windows Support, Buggy scroll bar support
8	**Browsera**	IE,Firfox,Safari	3 Min	Paid	Good	Yes	No	Yes	Recursive Crawling	Good interface	Limited Brower support, Expensive
9	**Xenocode Browser Sandboxes:**	IE, Firefox,Chrome,Safari, Opera	Quick	Free	Good	Yes	No	Yes	No Macintosh support.	Good interface	Limited Brower support
10	**CrossBrowser Testing.com**	IE, Firefox,Chrome,Safari, Opera	Avg.	Free / Paid	Good	Yes	No	Yes	Support full website testing.	Good interface	Limited Brower support
11	**IE Tester:**	All IE Version	Quick	Free / Paid	Avg.	Yes	No	Yes	Early stage of development	Compare two IE version side by side	Limited Brower support
12	**BrowsCamp**	All IE Version, safari, Mac	Avg.	Free / Paid	Good	Yes	Yes	Yes	Paid version perform depth testing	Good interface	Limited Brower support,
13	**NetRenderer**	All IE Version	Quick	Free	Good	Yes	No	Yes	Provide browser toolbar	Good interface	Limited Brower support

4.12 Brows Camp:

With all of the IE-Only test sites out there, it was about time someone joined in and created a site that allows testing on Safari/Mac. The free version of their service offers near-instant screenshots on the newest stable release of safari, and though it's lacking a bit in browser versions it definitely makes up for it in rendering speed. For a few dollars extra, they offer the ability to take over an entire machine and perform much more in-depth testing.



4.13 Net Renderer:

NetRenderer is a slightly more humble-looking option for testing IE compatibility. Like many of the other services, NetRenderer creates screenshots of our website in various browsers. It currently supports everything from IE5.5 all the way to IE8, and creates our screenshots very quickly without need to wait. They also provide a browser toolbar that allows us to quickly test any of the pages we are visiting with their service. This is also a free service, and don't even offer a paid version of the tests [6].

4.14 Browser Packs: If we need to test our website in specific browsers with and we are willing to perform the tests manually, there are a few free services and applications that could help:

- Spoon
- BrowserSeal.BrowserPack
- Internet Explorer Collection

In the first look, Spoon looks convenient because it is a Web service, which relieves us from having to install many browsers locally. But it had some stability problems with this service. Meanwhile, both the IE Collection and BrowserSeal.BrowserPack work very reliably. It is not any issues with browsers installed by these packs. The IE Collection has every IE version we could think of. BrowserSeal.BrowserPack, which relies on the IE Collection for IE support, also supports two Firefox, three Opera and two Safari versions [6]

Precise concluding remarks: The table summarizes services that were tested and analyzed in the article. Based on some metrics for each service to make it easier for us to choose the best on price, features and performance trade-offs.

Obviously, we have no clear winner. Each service has its advantages and disadvantages, and you are left to decide that the best trade-off for your case is. Professional developers would likely not use BrowserShots because of the unreasonably long response time. SuperPreview and Browsera are probably also impractical because of their very limited browser support.

BrowserLab will probably remain popular as long as it is free. Once Adobe starts charging about $20 per month for it, one would hardly have reason to use it, unless we worked in Dreamweaver, which has a BrowserLab extension, because there are much better alternatives. When choosing a tool, one of the most important factors in our decision will be whether to use a Web service or application. Some people prefer Web-based tools because they do not require installation. It is a prefer applications, at least for the development tools that we use frequently. They generally have a better interface and faster response time. BrowserCam, BrowserSeal, Litmus and Multi-Browser Viewer are all very good choices. But they do vary significantly in price. If we need to test mobile browsers, BrowserCam is probably our only option. For everyone else, it would be recommended either BrowserSeal or Multi-Browser Viewer; both come with standalone browser versions that are extremely important for testing. Unfortunately, both of them are Windows only, so Mac users will probably have to go with BrowserLab or BrowserCam. If automatic testing is important to you, then the BrowserSeal automation edition is our best choice.

5. CONCLUSION

Development of the websites enforces some entirely new challenges in the world of software quality & reliability and the process of testing. Within minutes of going live, worldwide audiences of a website make its quality and reliability crucial factors in its success. The impact of quality remains is in the mind of the Website user. A poor quality Website, one with many broken pages and faulty images, with Cgi-Bin error messages, etc., may cost a lot in poor customer relations, lost corporate image, and even in lost sales revenue. Unhappy users will quickly depart for a different site; and, they probably won't leave with a good impression.

Therefore, Web applications developers, and Website quality assurance managers need comprehensive systematic approach and automated testing tools of Websites that meet their specific needs. In web performance testing web site functionality on different web browsers, operating systems and different hardware platforms is checked for software, hardware memory leakage errors. The above explained technical approach mechanized web testing and the automated testing tools of websites offers the potential to meet quality and reliability challenges. As one year ago, there were almost no good options for testing cross-browser compatibility of websites. Lately, there have been a lot of newcomers to the browser testing world, some of which offer truly excellent services and actually make this stuff pretty easy.

6. REFERENCES AND IMPORTANT LINKS

[1] http://www.smashingmagazine.com/2010/06/04/cross-browser-testing-a-detailed-review-of-tools-and-services/

[2] http://www.softwaretestinghelp.com/web-application-testing/

[3] http://www.soft.com/eValid/Technology/White.Papers/wpaper.testing.pdf

[4] http://www.cigital.com/

[5] http://www.softwaretestinghelp.com/how-can-a-web-site-be-tested/

[6] http://freelancefolder.com/7-fresh-and-simple-ways-to-test-cross-browser-compatibility/

[7] http://www.softwaretestinghelp.com/what-is-client-server-and-web-based-testing-and-how-to-test-these-applications/

[8] http://www.lc7inc.com/quality-assurance/cross-browser-compatibility.