**Cross Browser Compatibility Issue**

**What is “Cross-browser compatibility?”**

Cross-browser compatibility ultimately has very little to do with what a web site looks like, and a lot more to do with how it functions. It also has relatively little to do with browsers, and perhaps could better be explained as multiple user-agent compatibility.

“Compatibility” (in this context) is not a term which means “looks and behaves identically” — instead, it may be better described as “performs equivalently under alternative conditions.” But developers and designers tend to most immediately seize upon appearance as the guiding line for cross-browser compatibility.

**5 Most Common Browser Compatibility Issues**

**1) Incorrect (or no) DOCTYPE**

The so called Doctype should always be the very first line in your html. It looks something like this:

<! DOCTYPE html>

This simple one-line of code can make all the difference between a cleanly rendered website and a strange rendering. This is especially true for Internet Explorer as it will fall back to it’s own Quirksmode where it interprets quite a few web standards entirely different. So if some browsers acts up in any way check your Doctype first!

### **2) No CSS Reset**

Did you know that every browser comes with a different set of internal, basic CSS styles which apply if the current website does not override them? Yep every browser will render a page differently even if it lacks any CSS styling of their own. To avoid having any side effects, you can use a so called CSS reset style sheet in your page to make sure that every browsers starts rendering with the same basic set of rules. A few commonly used reset style sheets are:

* [Normalize.css](https://github.com/necolas/normalize.css/blob/master/normalize.css)
* [HTML5Reset](http://html5reset.org/)
* [Eric Meyers CSS Reset](http://meyerweb.com/eric/tools/css/reset/)

Make sure to have one of these added as the first style sheet to your page to ensure a proper CSS reset. Of course many popular front-end frameworks like [Twitter Bootstrapped](http://getbootstrap.com/2.3.2/) already include a CSS reset so there is no need to add a second one.

### **3) Vendor Specific CSS Styles**

If a browser vendor implements a new CSS functionality they will often hide it behind a so called vendor specific CSS style. After the style has been established the vendor will often remove the vendor specific version or add a modified version without the vendor prefix. Let’s look at the opacity style for Mozilla Firefox (-moz) and without a vendor specific prefix:

-moz-opacity: 0.6;

opacity: 0.6;

So to make sure your code works in all browsers you’ll need to add the unprefixed version alongside all prefixed one’s to make sure it get’s picked up by all browsers. But how many prefixes are there:

* -ms for Microsoft (Internet Explorer)
* -moz for Mozilla Foundation (Firefox)
* -o for Opera Software
* -webkit  for Safari and Chrome

Therefor a full declaration of opacity should look like this:

Test{

-moz-opacity: 0.6;

-ms-opacity: 0.6;

-webkit-opacity: 0.6

Opacity :0.6;

}

### **4) Lack of Valid HTML / CSS**

Different browsers interpret HTML and CSS differently, and some are more forgiving than others.  For example it could happen that you forgot to close one <div> in your code. Now maybe Chrome and Firefox will just add the missing </div> and it will render correctly. Other browsers may not be so forgiving and the rendering will look different without so much as a hint on what might be wrong. And finding the missing closing tab manually can be quite frustrating, especially on a larger web project.

Luckily you can automatically validate your code using the W3C Validators for [HTML](https://validator.w3.org/) and [CSS](https://jigsaw.w3.org/css-validator/).

### **5) Outdated Browser Detection**

Are you still using older JavaScript code in your application? Trying to [detect the browser rather than features](http://jibbering.com/faq/notes/detect-browser/)? Unfortunately many older browser detection scripts out there will break if the used version of a browsers is unknown. So if your code does not run in more modern browsers check if maybe your browser detection is outdated and does not handle the more current browsers that well. Or even better remove the browser detection and detect features instead! You can use the handy library [Modernizer](http://modernizr.com/) for this.

**Solution**

A good way to test your site on cross browser is to use cross browser automation tools. Here are some useful tools.

**Adobe Browserlab**

Adobe Browserlab is a web hosted service which allows designers to test designs on multiple browsers and operating systems. In this, system are included a zoom function and measurement tools like rulers and guides, and the ability to move quickly to specific areas of a screenshot. Browser sets allow designers to customize, edit, and save a combination of browsers to test.

**Netrenderer**

IE NetRenderer is free browsers compatibility tester. With IE NetRenderer you can check how a website is rendered by different versions of Internet Explorer.

**Browsera**

Browsera is meant to be used as a testing tool, not just a screenshot tool, and works at the site level instead of the page level. Browsera crawl the pages on your site and actually test them for layout inconsistencies and scripting errors.

**Litmusapp**

Litmus shows you screenshots of your websites as they look across all major web browsers. After you finished testing, you can get a full compatibility report ready for review by your clients.

**Browsrcamp**

Browsrcamp allows you to test the compatibility of your design with Mac OS X browsers. You can check compatibility of your design with Safari. Simply add your website URL and you will get your website screenshot.

**CONCLUSION**

As the technology is changing rapidly, the application designers believe to give the best features and functionalities to the applications. But most of the applications are not stand alone. Users have different platforms and browsers. Designers cannot assume that their application will run fine and display and work for all the browsers without cross browser testing. Hence to avoid loss of business and reputation it is very important to pay attention to cross browser issues. Indeed it is hard to find out compatibility of your application for each existing browser. Hence the help of cross browser automation tools is indeed a good way in this respect. Only then users will be able to get unbiased what the applications can deliver them.