Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2021, 8(7):242-246

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Cross-Browser Compatibility: Modern Web Development with Enhanced User Experience

Chakradhar Avinash Devarapalli

Software Developer
Email: avinashd7[at]gmail.com

Abstract The evolution of the digital world brings complexities with each new technology. The same is the case with websites deployed all over the internet. This is the building block of the Internet as the largest exchange of information happens through the websites of the World Wide Web. These websites are available for all sorts of users where they can get access from any operating system, device, or browser. So, a problem arises to keep the functionalities and appearance of these systems across all devices and browsers. Cross-Browser compatibility however is the primary concern and is directly linked with user experience. The browsers use their rendering engines where updates are pushed from time to time. Therefore, there is a need for a standard system to follow in association with possible techniques to overcome the problem of compatibility with different browsers. The appropriate techniques need to be applied while maintaining the standards established for the industry.

Keywords *Cross-Browser Compatibility, User Experience, Responsiveness, testing, browser engines*

Introduction

Cross-browser compatibility refers to the capacity of a web app or a website to render its components accurately without distortion in visuals and avoiding unwanted results from the features. The results should meet the expectations even if the browser changes between Chrome, Edge, Firefox, Safari, or any other.

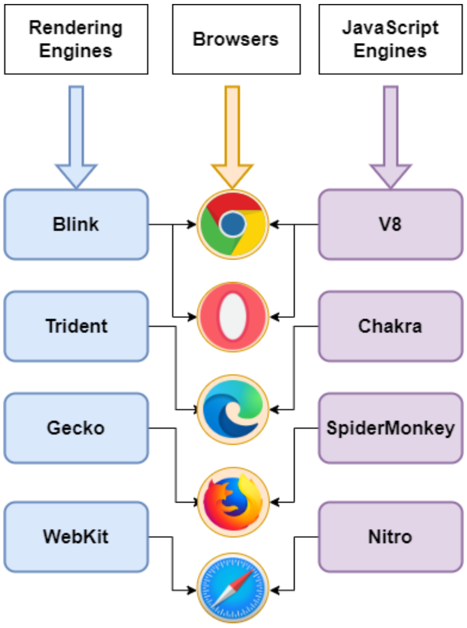


Figure 1: Browsers with Rendering and JS Engines

There are vast amounts of browsers out there and each of these has its rendering and JavaScript engines like Blink, Trident, Gecko, Webkit, V8, Chakra, SpiderMonkey, and Nito. These are responsible for interpreting and translating code into the visual experience a user has on the screens. Figure 1 above shows the engines against the selected browsers [1].

This research article is focused on resolving the cross-browser compatibility problem with the suggestion of effective measures. The challenges that are being faced in the industry are covered to possible solutions are presented against them. By applying these techniques properly, the compatibility issue would be resolved and the user experience would be enhanced. User satisfaction is directly linked with accuracy and therefore websites need to be accurate independent of the type of browser or system under the use of an individual. Therefore, a broad range of strategies are presented to resolve the browser conflicts from the user end.

In practice, web developers tend to use external libraries and frameworks to make the web development process easy but these dependencies can conflict with the browser's specifications due to different versions. Therefore appropriate usage of frameworks and libraries while maintaining the industry standards can only be helpful [2].

When different engines interpret the same code differently, inconsistencies arise. Layouts might shift, features might malfunction, browser-specific vulnerabilities can expose users to security risks, and the user experience can suffer [3]. This is why developers must navigate this diverse landscape, understanding the strengths and quirks of each engine, to ensure their creations function seamlessly across platforms. By adopting web coding standards, conducting browser testing, and leveraging cross-browser frameworks, developers can bridge the gap and deliver a consistent experience, regardless of the user's browser.

Literature Review

In the digital world with millions of devices scattered around the globe with various operating systems and browsers, cross-browser compatibility is crucial for the success of web applications and user satisfaction. W3C coding standards ensure consistent interpretation across browsers, which helps the industry to reduce conflicts in the later stages as well [4]. This research therefore focuses on the challenges that are being faced by the developers in modern web development for cross-browser compatibility and the potential conflicts that are resolved for common problems.

Different browsers utilize their dedicated rendering engines leading to inconsistencies in layout, functionality, and user experience. For instance, V8 for Chrome, Chakra for Edge, and Spider Monkey for Firefox [5], [6]. The Developers must navigate this dynamic landscape, understanding the evolving quirks and strengths of each engine.

The strategies can be used to avoid the issue of ecosystem conflicts. Cross-browser testing can assist in implementing automated and manual testing across various browsers before deployment and minimizes compatibility issues. Cross-browser frameworks like Bootstrap and Foundation streamline development help to reduce inconsistencies and promote browser-agnostic experiences [7].

The Essence of Compatibility

The browser can ruin the experience even if a system is equipped with rich functionalities until and unless cross-browser compatibility techniques are considered during development. The significance of a compatible web app or a website can be judged in the following aspects,

- The user experience and satisfaction majorly depends on a responsive design and compatible features.
- The market penetration can be done with a compatible website.
- A brand equity can only be established if a centralized system is available for all devices.
- A compatibility with mobile browsers is important as most of the users are on mobile devices.
- The websites can only retain traffic if they are equipped with integrally accessible features and therefore SEO will be improved.
- The maintenance will be relatively easy if a website is developed in a more centralized way by considering all browser engines.
- Global Audiences can be attracted to diverse systems.
- Competitive Advantage is available for brands who are considering users of all browsers.

Problem Statement

The research article is concerned with finding a common problem of cross-browser compatibility that hinders a better user experience. The efficient ways for developers to use modern technologies while upholding the university standards to overcome the compatibility problem [8].

Testing the compatibility issues can be the first step in finding the solution. The common problems faced by the developers are,

- **Rendering Differences:** The interpretation of HTML, CSS, and JavaScript code varies across each browser, resulting in variations in the display of elements and the execution of functions. These disparities create bugs in function execution and layout inconsistencies of visual elements in different browsers.
- **Discrepancies in CSS Rendering:** Divergence in interpretations and implementations of CSS styles by different browsers can lead to visual irregularities, affecting font sizes, colors, spacing, and layout. The result is then the bursting of the website design, producing a poor user experience.
- **Browser Bugs:** Some browser bugs can happen during the rendering or execution of some critical functionality of a website in a browser on a specific device. Overcoming this problem requires a lot of testing and the implementation of debugging tools to make the website run perfectly on these browsers.

Resolving Conflicts

To avoid the problem of browser compatibility, a developer can adopt some solutions. The common compatibility issues can be resolved with the following approaches.

Responsiveness

The developer needs to consider the diverse screen sizes on the user end and should implement the design accordingly. In this way, the system would be adjustable to all the screen sizes. Furthermore, the orientation of the device should also be kept in mind as a user can access the web app in both landscape and portrait mode. In this way, the output would remain the same independent of the size of the screen from the user end. One way to resolve the issue of responsiveness is the use of media queries available in CSS.

Cross-Browser Frameworks

The availability of frameworks for cross-browser compatibility makes the lives of both the developers and the testing team easier. The appearance and functionalities across different browsers can be kept the same by utilizing frameworks like Bootstrap and Foundation. The earlier is good at appearance integrity and the other is more helpful in feature integrity [9].

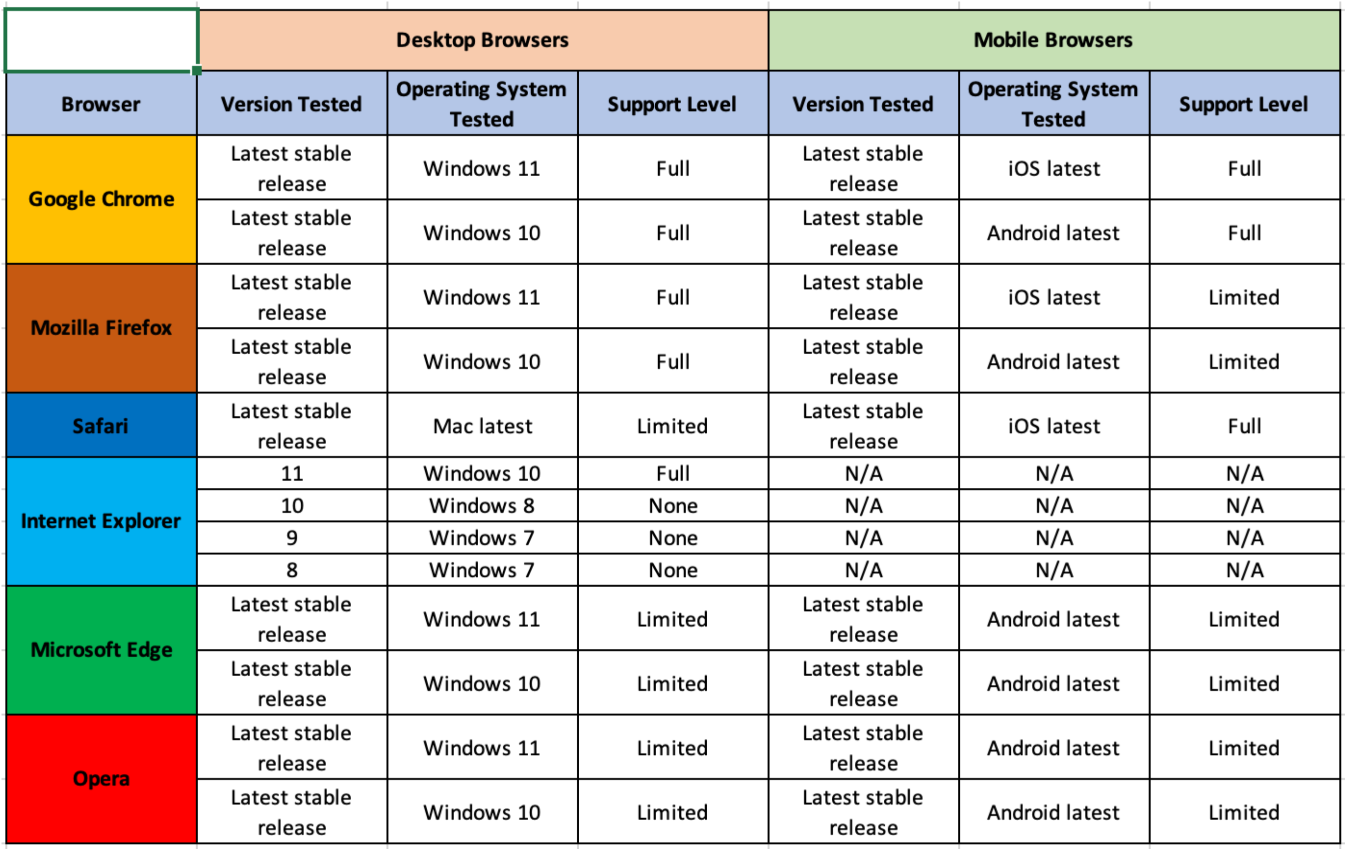
The pre-built components and stylesheets are provided by keeping the compatibility standards to help avoid browser inconsistencies. The major advantage is that it reduces time and complexities from the development end. The manual handling of responsiveness with media queries can be overturned with pre-built components.

Feature Detection

Alternative solutions can be provided to users with browsers that lack support for most of the features. This can be done by detecting the features that are unsupported and then integrating polyfills or fallback mechanisms. In this way, there will be no need for major browser updates from the user end [10].

Cross-Browser Testing

The cross-browser testing needs to be applied to visual appearance, functionalities of individual features, accessibility, and performance on different browsers, devices, and operating systems. This testing approach across various devices, and browsers like Chrome, Edge, Firefox, and Safari is crucial and helps to identify compatibility issues most users may encounter [6]. The browser-specific issues and bugs can therefore be minimized and developers can save time and resources during the development as well as maintenance phases. A cross-browser testing sample matrix looks like the one shown below in Figure 2,



	Desktop Browsers			Mobile Browsers		
Browser	Version Tested	Operating System Tested	Support Level	Version Tested	Operating System Tested	Support Level
Google Chrome	Latest stable release	Windows 11	Full	Latest stable release	iOS latest	Full
	Latest stable release	Windows 10	Full	Latest stable release	Android latest	Full
Mozilla Firefox	Latest stable release	Windows 11	Full	Latest stable release	iOS latest	Limited
	Latest stable release	Windows 10	Full	Latest stable release	Android latest	Limited
Safari	Latest stable release	Mac latest	Limited	Latest stable release	iOS latest	Full
Internet Explorer	11	Windows 10	Full	N/A	N/A	N/A
	10	Windows 8	None	N/A	N/A	N/A
	9	Windows 7	None	N/A	N/A	N/A
	8	Windows 7	None	N/A	N/A	N/A
Microsoft Edge	Latest stable release	Windows 11	Limited	Latest stable release	Android latest	Limited
	Latest stable release	Windows 10	Limited	Latest stable release	Android latest	Limited
Opera	Latest stable release	Windows 11	Limited	Latest stable release	Android latest	Limited
	Latest stable release	Windows 10	Limited	Latest stable release	Android latest	Limited

Figure 2: A matrix of sample test runs for Cross-Browser Compatibility

The tools can help in testing cross-browser functionality testing. Manual testing across numerous browsers and devices can be tedious and time-consuming. There are a lot of automation tools and IA-powered tools like task runners, Sauce Labs, and Lambda Test. Selenium, Cypress, Playwright, and Puppeteer are some of the most popular frameworks for automated testing, which can help identify and address compatibility issues efficiently [11]. While Selenium reigns supreme in community support, Cypress offers flaky test reduction, but limited language support. Evaluate each option based on your specific needs and team skillset for the optimal testing solution.

Regular Updates

The system should be maintained regularly and the updates need to be pushed. The reason is to stay updated with the browser updates. The earlier dependencies and framework versions get deprecated over time and chances of feature failure are high in that case. Regular testing approaches can be applied to the system to stay connected with what needs to be changed before feature failure.

Research Findings

The problem of compatibility with a website with different browser once solved with the help of suggested ways can bring the following benefits to the industry as well as for users,

- **Expanded Audience Reach:** A compatible website on different browsers guarantees accessibility to a wide range of users, independent of their platforms.
- **Enhanced User Experience:** Eliminating compatibility issues fosters user satisfaction and engagement.
- **Improved Diversity:** Addressing browser-specific vulnerabilities would help target a diverse range of devices across the globe with any screen size, resolution, or orientation mode.

Apart from these, most of the resolutions presented are helpful in other technologies like mobile development and desktop applications. The functionalities, continuation in design, accessibility, and performance can be improved.

Future Development

The exploration of compatibility is a continuous focal point for industries based on its complexity and the benefits it offers after overcoming diverse problems. The methods to eradicate this problem are still evolving and it is expected that the improvement in server-side rendering can be helpful to avoid more than half the chances of this problem if used properly.

Conclusion

User satisfaction is directly linked with the unified systems where a system can be accessed on multiple platforms with integral results and features. The established web standards and their continuous evolution can help the industry provide unified solutions to targeted users without impacting the results. The general web user is not concerned with the background activities and only requires a standard system independent of the type of browser being used to access a particular web page. Therefore, the industry should be equipped with modern solutions to the problems that arise regarding the compatibility of browsers with a diverse range of features offered in a website.

The current solution needs the developer to adhere to university standards, consider user requirements, and apply conventional practices to make the system succeed in the market. The internet has an aggregation of diverse audiences around the world and they have multiple options to access a web page with the use of different browsers, and one can't predict a particular browser to target users which is why there is a need to apply techniques to make it generalize for all browsers being used.

References

[1] A. Grosskurth and M. Godfrey, "A reference architecture for Web browsers," in *21st IEEE International Conference on Software Maintenance (ICSM'05)*, Budapest, Hungary, Sep. 2005.

[2] P. Andrew, B. Julia, S. Tatiana and V. Kateryna, "Development of the system to provide cross-browser compatibility of web application," in *IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Kyiv, UKraine, May. 2018.

[3] D. Ramdas, "CROSS BROWSER COMPATIBILITY ISSUES," San Diego State University, 2015.

[4] S. L. Henry, "Introduction to Web Accessibility," W3-Web Accessibility Initiative, Feb. 2005. [Online]. Available: https://www.w3.org/WAI/fundamentals/accessibility-intro/#:~:text=The%20W3C%20Web%20Accessibility%20Initiative,standard%3A%20ISO%2F%20IEC%2040500.. [Accessed 21 May. 2021].

[5] W. M. Watanabe, F. C. Paes and D. Silva, "Towards Cross-Browser Incompatibilities Detection: A Systematic Literature Review," *International Journal of Software Engineering & Applications (IJSEA)*, vol. 10, no. 6, Nov. 2019.

[6] N. S. Leandro, A. M. Maximiliano, L. G. Cristina and I. Emanuel, "A Systematic Literature Review in Cross-browser Testing," *Journal of Computer Science & Technology*, vol. 18, no. 1, pp. 18-27, Apr. 2018.

[7] A. C. and A. Padmanabhan, "Real User Monitoring Benchmark Report," devopedia, Jul. 2020.

[8] O. Sharma and J. Ochin, "Cross-Browser Incompatibility: Reasons and Solutions," *International Journal of Software Engineering & Applications (IJSEA)*, vol. 2, no. 3, Jul. 2011.

[9] I. Zuzak, M. Ivankovic and I. Budiselic, "A Classification Framework for Web Browser Cross-Context Communication," in *School of Electrical Engineering and Computing, University of Zagreb, Croatia, (2) Google Inc.*, Zurich, Switzerland, Aug. 2011.

[10] A. Murolo, F. Stutz, M. Husmann and M. C. Norrie, "Improved Developer Support for the Detection of Cross-Browser Incompatibilities," in *International Conference of Web Engineering*, Jun. 2017.

[11] B. Gruenbaum, "Puppeteer, Selenium, Playwright, Cypress – how to choose?," 04 Jun. 2020. [Online]. Available: https://www.testim.io/blog/puppeteer-selenium-playwright-cypress-how-to-choose/. [Accessed 21 May. 2021].