

Comparison of Utility-First CSS Framework

Nandan S^{1*}, Usha Sree R², Priyanka Mohan³

^{1,2,3}Dayananda Sagar Academy of Technology and Management, Karnataka, India

*Email: nandan7198@gmail.com

Abstract

Utility-first CSS frameworks have revolutionized web development by offering predefined utility classes that streamline the design process and reduce the need for custom CSS. However, selecting the right framework can be challenging due to the variety of available options. This paper addresses the problem of choosing between two of the leading utility-first CSS frameworks Tailwind CSS and Tachyons by providing a comparative analysis based on key factors such as size, load speed, flexibility, ease of use, and community support. The objective of this research is to identify the strengths and weaknesses of both frameworks, helping developers make informed decisions based on project needs. Our methodology involved testing load speeds using Locust for performance analysis, reviewing community support through GitHub repositories and forums, and assessing the flexibility and ease of use through practical development tasks. The results revealed that while both frameworks are robust, Tachyons excels in performance and simplicity due to its smaller size, whereas Tailwind CSS offers greater customization and flexibility, making it more suitable for complex projects. The novelty of this research lies in its direct comparison of utility-first frameworks, highlighting how developer preferences and project requirements play a crucial role in framework selection. In summary, this study provides valuable insights for developers looking to optimize web development workflows by selecting the most appropriate CSS framework based on specific project goals.

Keywords

CSS Frameworks, Locust, Tachyons, Tailwind CSS, Utility-first CSS

Introduction

CSS frameworks are essential tools in modern web development, streamlining the design process and ensuring consistency across web applications. Among the various types of CSS frameworks available, utility-first frameworks like Tachyons and Tailwind CSS have gained significant popularity due to their distinct approaches to styling and their emphasis on flexibility and efficiency.

Tachyons is a functional CSS framework known for its ability to create fast-loading, easily maintainable websites. This framework prioritizes minimalism by offering a set of single-purpose

Submission: 16 October 2024; **Acceptance:** 5 November 2024



Copyright: © 2024. All the authors listed in this paper. The distribution, reproduction, and any other usage of the content of this paper is permitted, with credit given to all the author(s) and copyright owner(s) in accordance to common academic practice. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license, as stated in the web [site: https://creativecommons.org/licenses/by/4.0/](https://creativecommons.org/licenses/by/4.0/)

utility classes that are directly applied to HTML elements. By focusing on simplicity, Tachyons reduces the need for custom CSS, which can often lead to bloated and hard-to-maintain codebases. The emphasis on performance and code simplicity allows developers to quickly prototype and build web applications without getting bogged down in complex styling rules. The minimalist approach also contributes to faster load times, as the CSS file sizes are typically smaller, leading to improved overall performance and user experience (Fiala, 2019).

In contrast, Tailwind CSS is another utility-first framework that provides extensive control over design elements such as layout, typography, color, and spacing. Tailwind CSS stands out for its high level of customization, enabling developers to extend the built-in styles or make modifications to meet specific design needs. This flexibility makes it possible to create highly tailored and unique user interfaces without sacrificing consistency. Tailwind's comprehensive set of utility classes allows for rapid development and iteration, as styles can be adjusted directly in the HTML, leading to a more efficient workflow. Additionally, Tailwind's configuration file offers a centralized way to manage design tokens and theme settings, further enhancing its adaptability and ease of use (Klimm, 2021).

Literature Survey

The landscape of CSS frameworks has been extensively studied, revealing various insights into their effectiveness and suitability for different development scenarios. Tools like Locust are commonly used to analyze performance and load capabilities of web technologies. Locust, in particular, is employed to simulate real-world usage and measure how different frameworks handle load under various conditions (Shrivastava et al., 2020). This tool is instrumental in assessing the real-time performance of frameworks like Tailwind CSS and Tachyons, providing empirical data on their efficiency.

Al-Salmi offers a comprehensive comparison of CSS frameworks, focusing on their relative strengths and weaknesses across different scenarios (Al Salmi, 2023). His research emphasizes the impact of framework choice on development efficiency, code maintainability, and performance. Al Salmi's findings underscore the importance of understanding each framework's unique features and trade-offs, which directly informs our comparative analysis of Tailwind CSS and Tachyons.

Ardito discusses user interface (UI) development in modern web applications and highlights the role of utility-first frameworks in enhancing design workflows (Ardito, 2021). Ardito's work illustrates how frameworks like Tailwind CSS can streamline UI development by offering a rich set of utilities and customization options. His research provides a backdrop for evaluating how these frameworks contribute to user experience and design efficiency.

Mohd conducted a comparative analysis of various CSS and JavaScript frameworks, including utility-first frameworks. Their study provides valuable insights into how different frameworks impact development practices and efficiency. The authors' discussion on utility-first frameworks adds context to our analysis by highlighting their advantages in terms of development speed and ease of customization (Mohd et al., 2022).

Methodology

This document clearly describes the method and tools used. The tools and techniques are clearly analyzed. We employed load speed tests to identify Locust. By discussing the specific features and capabilities that make Locust popular among users, we can further explore its effectiveness as a load testing tool. Additionally, I inspected documentation and community forums, as well as browsed repositories on GitHub to judge the flexibility, ease of use, and the degree of community support. The following are the comparison parameters:

- **Size**

Size refers to the file size of the CSS framework, impacting the initial load time of a web page. Smaller file sizes generally lead to faster load times, enhancing user experience, especially on mobile devices or slower connections. Tailwind CSS has a file size of approximately 46.2KB (minified and compressed) by default, as shown in Figure 1, but this can vary based on customization. Extensive customization through the configuration file (`tailwind.config.js`) can increase the file size. On the other hand, Tachyons, which focuses on essential utility classes and has a smaller default file size of around 14KB (minified and compressed), contributes to a leaner CSS footprint. In this analysis, researchers found that while Tailwind CSS offers greater flexibility, it may come at the potential cost of increased file size, whereas Tachyons' smaller size is advantageous for performance-critical applications.

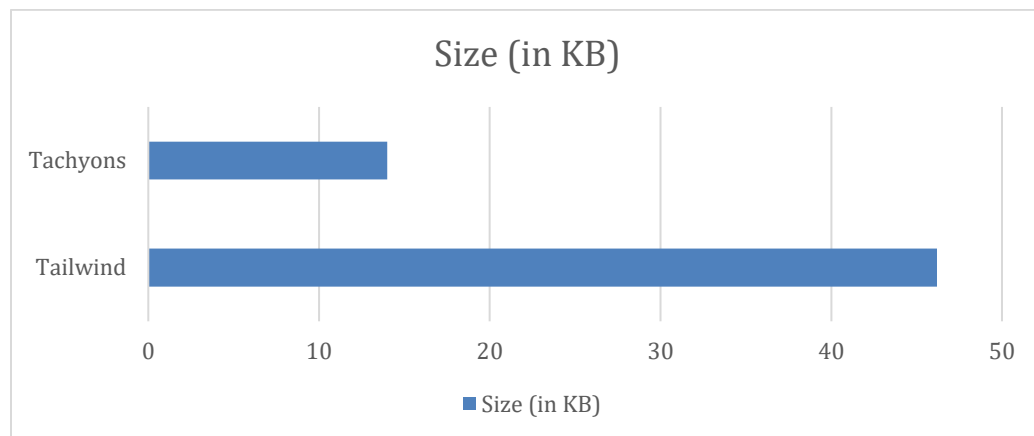


Figure 1. Comparison of Size: Tachyons vs. Tailwind CSS

- **Load Speed**

Load speed measures how quickly a web page becomes interactive, which improves user satisfaction and can positively affect search engine rankings. CSS file size, HTTP requests, and framework efficiency influence this. Tailwind CSS optimizes with performance in mind, using tools like PurgeCSS to remove unused styles, thereby maintaining quick load times with minimal performance impact. Tachyons, with its smaller file size, ensures faster load times and a reduced CSS footprint. Load speed tests show that Tachyons performs exceptionally well, often demonstrating minimal average response time differences compared to Tailwind CSS. Sometimes Tailwind CSS was faster, and other times Tachyons was faster, indicating similar response times under equal conditions. Both frameworks perform well in terms of load

speed, especially when optimized, so the choice between them depends more on factors like ease of use and flexibility.

- **Flexibility:**

Flexibility refers to the framework's ability to adapt to specific project needs, including customization and the availability of plugins and tools. Tailwind CSS offers extensive configuration options for colors, spacing, typography, and more. It is supported by a rich ecosystem of plugins for additional functionality. In contrast, Tachyons provides core utility classes that cover essential styling needs with limited customization, promoting a maintainable codebase but limiting extensive customization. Analysis suggests that Tailwind CSS offers more flexibility for complex projects, while Tachyons is more suitable for simpler, performance-focused projects.

- **Ease of Use**

Ease of use encompasses the learning curve, documentation quality, and overall developer experience, impacting productivity and development time. Tailwind CSS has a steeper learning curve due to its extensive utility classes and configuration options, but it also provides comprehensive guides, examples, and an active community for support. Tachyons, on the other hand, has an easier learning curve for beginners with its straightforward functional approach and concise, effective documentation, though it has fewer resources compared to Tailwind CSS. Consequently, Tachyons is more accessible for beginners, while Tailwind CSS offers a richer set of tools for experienced developers.

- **Community Support**

Community support indicates a framework's long-term viability, with a strong community providing resources, plugins, themes, and support. Tailwind CSS benefits from a large and active community, offering extensive resources and frequent updates, with numerous plugins, themes, and tools contributed by the community. In contrast, Tachyons has a dedicated but smaller community that provides regular updates and contributions, focusing on minimalism. While Tailwind CSS benefits from a larger and more active community that offers more resources and support, Tachyons provides strong support for those who prefer a minimalistic approach.

Results and Discussion

Tachyons' smaller default file size makes it more suitable for projects with stringent performance requirements. On the other hand, both frameworks perform equally well in terms of load speed. In terms of flexibility, Tailwind CSS provides a wider range of customization options, which makes it more suitable for complex projects. While Tachyons is easier for beginners, Tailwind CSS offers a more comprehensive set of tools for experienced developers. Tailwind CSS has a larger and more active community, offering more resources and support. Figure 2 shows the weekly downloads of two npm packages, Tachyons and Tailwind CSS, from June 2022 to May 2023. Tachyons has consistently low downloads, peaking at around 52,000. In contrast, Tailwind CSS shows a steady rise, reaching over 5.5 million downloads in May 2023.

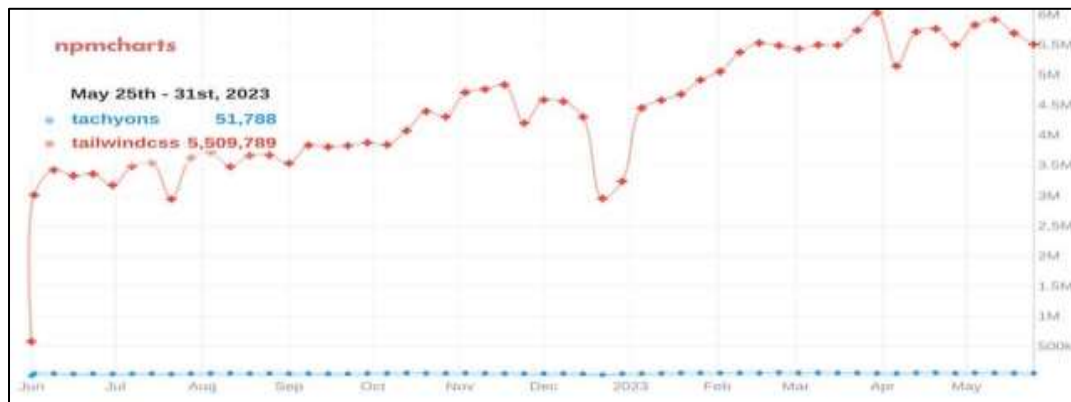


Figure 2. Comparison of Downloads: Tachyons vs. Tailwind CSS

For the implications, the findings suggest that the choice between Tailwind CSS and Tachyons depends largely on the specific needs of the project. For projects requiring extensive customization and strong community support, Tailwind CSS is the best choice. For projects that prioritize simplicity and performance, Tachyons may be more suitable. This study's limitations stem from the scope of evaluated parameters. Future research could explore additional factors such as compatibility with other tools and frameworks and real-world case studies.

Conclusion

Both Tailwind CSS and Tachyons are powerful utility-first CSS frameworks that offer unique advantages to developers. Tailwind CSS excels in flexibility by providing extensive customization options that allow developers to tailor their designs precisely to their needs. It also benefits from a large and active community, which means more resources, plugins, and support are available for those using the framework. On the other hand, Tachyons is known for its simplicity and minimalism. Its smaller default file size makes it a beneficial choice for projects where performance is a critical factor. Tachyons' straightforward approach with single-purpose utility classes ensures quick and simple styling without the need for extensive custom CSS. Future recommendations focus on developers should choose Tailwind CSS for projects that require extensive customizations and a wide array of community resources. Its flexibility and the breadth of available tools make it suitable for complex, large-scale projects where design precision is paramount. On the other hand, we recommend Tachyons for simpler projects that prioritize performance and ease of use. Its minimalistic approach allows for rapid loading times and straightforward development, making it an excellent choice for projects with stringent performance requirements or for developers who prefer a more straightforward, no-frills styling method.

Acknowledgement

The researcher did not receive any funding for this study, and the results have not been published in any other sources.

References

- Al Salmi, H. (2023). Comparative CSS frameworks. *Multi-Knowledge Electronic Comprehensive Journal for Education and Science Publications (MECSJ)*, (66). https://mecsjs.com/uplode/images/photo/hat4_.pdf
- Ardito, L. (2021). User Interface Development of a Modern Web Application. <https://webthesis.biblio.polito.it/30076/1/tesi.pdf>
- Duvander, J., & Romhagen, O. (2019). What affects the choice of a JavaScript framework: Interviews with developers. URN: <urn:nbn:se:hj:diva-46268>
- Fiala, J. (2021). Web application for a restaurant with rooms. *Theses.cz*. <http://theses.cz/id/yoy0su/>
- Khan Mohd, T., Thompson, J., Carmine, A., & Reuter, G. (2022). Comparative Analysis on Various CSS and JavaScript Frameworks. *Journal of Software*, 282–291. <https://doi.org/10.17706/jsw.17.6.282-291>
- Klimm, M. C. (2021). Design Systems For Micro Frontends-An Investigation Into The Development Of Framework-Agnostic Design Systems Using Svelte And Tailwind Css (Doctoral dissertation, Hochschulbibliothek der Technischen Hochschule Köln). https://epb.bibl.th-koeln.de/files/1666/SCIBachelor_Design_Systems_and_Micro_Frontends.pdf
- Kumar, A., & Singh, R. K. (2016). Comparative analysis of angularjs and reactjs. *International Journal of Latest Trends in Engineering and Technology*, 7(4), 225-227. <https://doi.org/10.21172/1.74.030>
- Shrivastava, S., & Prapulla, S. B. (2020). Comprehensive review of load testing tools. *International Research Journal of Engineering and Technology*, 7(3392-3395), 43. <https://www.irjet.net/archives/V7/i5/IRJET-V7I5651.pdf>
- Vyas, R. (2022). Comparative analysis on front-end frameworks for web applications. *International Journal for Research in Applied Science and Engineering Technology*, 10(7), 298-307. <https://doi.org/10.22214/ijraset.2022.45260>