STM JOURNALS

Recent Trends in
Programming Languages

ISSN: 2455-1821
Volume 12, Issue 2, 2025
May–August
DOI (Journal): 10.37591/RTPL

https://journals.stmjournals.com/rtpl

Review RTPL

Evolution of CSS Frameworks: From Bootstrap to Utility-First Design

Riddhi Nahar*

Abstract

CSS frameworks have become integral to modern web development, offering developers predefined styles and components that simplify the design process, ensure visual consistency, and accelerate deployment. Over the years, these frameworks have evolved significantly: from traditional component-based architectures, such as those seen in Bootstrap and Foundation, to more flexible utility-first methodologies like Tailwind CSS. This study explores the evolution of CSS frameworks by highlighting the contrasting approaches of these widely used tools. The study provides a detailed examination of each framework's architecture, strengths, limitations, and overall applicability. Key focus areas include development speed, customization capabilities, learning curve, performance efficiency, and the effectiveness of responsive design. Additionally, a comparative analysis is conducted using measurable parameters such as framework size, popularity, browser compatibility, and support for mobile-first design. By comparing traditional and utility-first approaches, this research aims to guide developers in making informed decisions when selecting a CSS framework that best meets the needs of their specific project. Whether building a scalable enterprise-level application or a lightweight personal website, understanding the trade-offs between these methodologies is essential. The study emphasizes aligning framework selection with both current development trends and project-specific requirements.

Keywords: CSS frameworks, responsive design, bootstrap, tailwind CSS, Utility-First CSS, web development

INTRODUCTION

CSS Frameworks have played a very significant role in the history of web development, providing a huge range of options for developers to create beautiful, yet somewhat standardized web pages. Over the last couple of years, a few frameworks have come into the picture, helping in styling the content of the web. Earlier, Bootstrap or Foundation ruled the day providing either numerous complete UI components or responsive design capability. However, recent years have seen the emergence of something new: utility-first CSS, as seen in frameworks like Tailwind CSS.

Utility-first CSS frameworks equip developers with pre-made atomic CSS classes, allowing for a very granular way of applying a style on any element. Rapid development and customization for set projects can happen using these frameworks without the headache of writing a lot of custom CSS. Some of the main highlight points about utility-first frameworks are:

- *Atomic classes:* They allow for single-property definition of a CSS property (e.g. padding, margin and colour) to amend styles for an element specifically.
- *Preconfigured grid systems:* Utility-first ones often come with preconfigured grid systems to help with layout chores with ease.

***Author for Correspondence**
Riddhi Nahar
E-mail: riddhinahar028@gmail.com

Student, Department of Computer Science Engineering, Rajasthan College of Engineering for Women, Jaipur, Rajasthan, India

Received Date: April 23, 2025
Accepted Date: May 13, 2025
Published Date: June 13, 2025

Citation: Riddhi Nahar. Evolution of CSS Frameworks: From Bootstrap to Utility-First Design. Recent Trends in Programming Languages. 2025; 12(2): 40–48p.

- *Rapid prototyping:* Ready styles allow developers to design layouts quickly and test them, shortening the time for their whole development process.

While bootstrap and foundation are both traditional component-based frameworks, they have given us two methods of going about using them. We can either use their predefined UI components or customize them using our own custom CSS. This way is even not as time consuming as what they have created as utility-first approaches.

LITERATURE REVIEW

CSS Frameworks: A Historical Perspective

No doubt, it is indeed possible to develop such contents in an entire web page using HTML. There is always time and energy to build a web page from scratch. Speed optimizers and lazy plugins in the form of CSS frameworks have personally improved the speeds with which a web page can be formed. They have greatly increased productivity in project management by saving time that would have been spent closely observing CSS frameworks for holding bona fide sources that make it easy for developers [1]. When pages are built using CSS frameworks; the pages will therefore be cross-browser compatible; hence a good thing is that documentation in them gives quite clear guidelines on installation and usage which makes it easy for developers [2]. Some become vastly browser-friendly due to their modularity; however, the complete body of constantly refreshing grand lobbies in open-source and numerous free CSS frameworks is just another thriving, evolving change [3, 4]. The efficiency it causes in web development effectively serves to improve, indirectly, the performance of the website while keeping a current but futuristic design flavour in the aesthetics of a web-based model view. Most CSS frameworks are available free of charge; hence quality and performance improve as knowledge and experiences are shared among developers [3]. Moreover, frameworks can compete with customized or even infinitely expanded experiences.

Writing with a Historical Background in CSS Frameworks

It is important to mention that developing a web page contents is possible with entirely HTML. There is always time and effort involved to develop a web page from the scratch. Speed optimizers and lazy plugins, like CSS frameworks, have also developed web pages within much shorter duration and have hiked productivity in project management. These have saved project time by reducing the time spent on closely observing CSS frameworks that provide strong sources for making it easy for the developers [1]. When pages are built using CSS frameworks, the pages are cross-browser compatible; therefore, an extra good thing is that documentation in them gives quite clear guidelines on installation and usage which makes it easy for developers [2]. Some of them become extremely browser-friendly due to their extensibility yet entire set of constantly updating libraries in open-source environments and many CSS frameworks thrive and adapt changes as time goes [3]. It brings efficiency in web development and serves relatively well in the performance of the website while keeping a current design flavour with yet a futuristic design mood in the aesthetics part. Most CSS frameworks are available for free; hence better quality and performance is given as sharing their knowledge and experiences from among developers is easy [3]. Also, group frameworks give a lot more space for practically infinite customization. There are static pages, dynamic pages, and responsive pages. Static pages are very easy to create as well as maintain. There are not any tools, meaning they are hardly ever changed. In contrast, dynamic websites generate real-time content for an end-user information query and respond with specific results for request capture. It stores database for further competition and optimization. Responsive sites use a percentage-based layout as opposed to fixed measurements to construct the sites, rendering them easily adjustable on any device across a number of screen sizes [5].

Component-based Frameworks: Bootstrap and Foundation

A project that was once stately in the using of Twitter has now become a divinity in web developing because of its supreme ease of usage along with an all-encompassing feature set [2]. It still is the most popular library for HTML, CSS, and JS when it comes to any web designing and developing project. It

is rich in features; of these, responsiveness certainly deserves a mention [3]. An individual framework for creating responsive websites, or websites whose elements shift according to the user's screen width or the width of the browser. It is very well documented along with the complete set of out-of-the-box components and also SASS and LESS features; the entire design actually becomes a fun experience [5]. More than ready-to-use built-in components, it gives more options of customization to the user and gives wider scope to the developer for customizing according to his needs [5]. There are also working JavaScript components dedicated to dynamic web pages that require less of the SASS LESS preprocessor and allow much more freedom for custom styling for their components. The implementation of Bootstrap, while that of Foundation illustrating both their basic component structure and styling approach.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">

<div class="container text-center mt-4">

 <h3 class="fw-bold">Hello Bootstrap</h3>

 <button class="btn btn-primary mt-2">Click Me</button>

</div>
```

At ZURB is yet another very famous open-source framework based on components: Foundation was very well known for its flexibility as well as advanced features; it offered the possibility of either integrating legacy feature sets or fusing them with a project's newer features or specifications. The Foundation framework uses a SASS preprocessor, and utilizes preset variables of CSS. The syntax is itself taken with a bit of pinch and hence easy to code from the start. It was designed as mobile-first; the web pages were developed for mobile and tablet screen and later reconfigured accordingly for larger screen applications. Hence, it makes the second, most in-demand and popular framework in the world [6]. The professionals, who rather would prefer working from scratch with design files, adopt it because it does not come along with UI components. It serves through Ruby, SASS, and SCSS to achieve that. This framework then is quite capable of dealing with training projects as well as business domains with larger outreach compared to any other framework [7].

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/foundation-
sites/dist/css/foundation.min.css">

<div class="grid-container text-center">

 <h3>Hello Foundation</h3>

 <button class="button primary">Click Me</button>

</div>
```

The Rise of Utility-First CSS: Tailwind CSS

Tailwind CSS describes new level redirect concerning paradigm shift in the design of CSS framework. Instead of offering finished components, it provides almost an entire library of utility classes for one to use right in the HTML. It has almost infinite flexibility and control over styling. The developer can design according to their specifications without any pre-defined component. Speed-wise, Tailwind CSS gives options to the developer minimalist speed on the site. Tailwind does well when used by proficiently using professional developers. For its compatibility, it turns to be a very easy framework for web designing..


```
<script src="https://cdn.tailwindcss.com"></script>

<div class="p-4 bg-blue-100 text-center">

 <h3 class="text-lg font-semibold">Hello Tailwind</h3>

 <button class="mt-2 px-4 py-2 bg-blue-500 text-white rounded">Click Me</button>

</div>
```

Skilled developers use Tailwind CSS, as it gives flexibility, while Bootstrap, made more for beginners, reduces development time by providing ready-to-go templates and built-in components [8]. Furthermore, because of not requiring extensive programming knowledge, Bootstrap allows easy understanding and learning [9].

Skeleton

This next contender is Skeleton. Skeleton is lightweight and very easy to code with [9]. It has a small functionality, thus is very suitable for small projects. The minimalist grid-based structure of the Skeleton framework highlights its lightweight design approach.

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/skeleton/2.0.4/skeleton.min.css">

<div class="container">

 <h3>Hello Skeleton</h3>

 <button class="button-primary">Click Me</button>

</div>
```

Skeleton has been designed with mobile devices first. It consists of a 12-column fluid grid system that has potential rows and columns [9]. Skeleton's support targets device resolutions starting from 960 pixels and above. It works in modern browsers; SASS and LESS are supported [10].

Bulma

Bulma is an open-source framework. It is very lean and light, not comparable even to Skeleton [10]. With it, you are able to create front-end designs based on its ready-made components. You can design a site with Bulma responsiveness. The use of Bulma's utility classes and modular components helps to achieve a clean and responsive layout.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.4/css/bulma.min.css">

<div class="section has-text-centered">

 <h3 class="title is-4">Hello Bulma</h3>

 <button class="button is-primary">Click Me</button>

</div>
```


METHODOLOGIES USED

This research used comparative analysis as its method of inquiry in order to examine the CSS Framework. Comparative analysis is a method of inquiry in which two or more subjects are compared and contrasted to establish the similarities and differences between them and to identify patterns. In other words, to point out the similarities and differences, to better understand cases. It helps to compare the advantages, disadvantages, features, and implementation of the selected CSS frameworks, and draws inferences and makes informed decisions. Selection of a CSS framework was based on the purpose of the research. This research was conducted to assess the performance and features of the CSS frameworks to help developers identify the best-suited frameworks for their projects. Analysing the available frameworks and researching to find the best available frameworks for web page design and to compare those frameworks is one of the most valuable time and contribution by developers. This research will assist web developers to save their time and effort by presenting a complete analysis and comparison of the best popular CSS frameworks and assist developers with identifying the most appropriate CSS framework for their design project [4]. Researching and analysing the most ideal frameworks to build web pages can be a developer's most precious time and energy. But this research is important because it assists web developers in saving time and effort by giving an in-depth review and comparison of the most used CSS frameworks, enabling them to determine the most appropriate CSS framework for their particular projects. The aim of this study is to thoroughly analyse the advantages and disadvantages of different CSS frameworks based on their main features. By comparing and discussing the most common frameworks, along with their pros and cons, developers can make educated choices about which framework fits best with the needs of their project and with their own experience. CSS is crucial in assisting developers in styling HTML pages and formatting them correctly.

TAILWIND CSS IMPLEMENTATION

In order to illustrate the benefits of utility-first CSS, we implemented a sample web page using Tailwind CSS. It contains a navigation bar, a hero section, and a content section. Utilizing Tailwind's utility classes, we can quickly style elements on the page with fine-grained control over typography, spacing, and colours.

This implementation demonstrates the possibility of making visually appealing and responsive web pages with minimal CSS code using Tailwind CSS (Figure 1). The utility-first approach makes it easier for developers to quickly prototype and iterate on designs instead of relying on pre-built components.

CASE STUDIES OF CSS FRAMEWORK ADOPTION

The practice was reading through some case studies of various sites and how each could have adopted CSS frameworks differently for purposes other than what they were meant to carry out.

```
<template>
  <input
    class="border-gray-300 focus:border-indigo-300 focus:ring focus:ring-indigo-200
     focus:ring-opacity-50 rounded-md shadow-sm"
    :value="value"
    @input="$emit('input', $event.target.value)"
    :placeholder="placeholder"
    ref="input"/>
</template>
```

Figure 1. Tailwind CSS implementation.

Bootstrap: Spotify

Bootstrap comes completely integrated with the currently leading online music application for instance as a marketing tool for its website. With this part of Bootstrap, Spotify defines itself to incorporate as many of established widely used components and their cribbing by itself. The fact that it will have great user experience, irrespective of device used because of Bootstrap's responsiveness, and UI components guarantees it.

Foundation: Mozilla Developer Network (MDN)

Mozilla has powered the site known as the 'developer network' via Foundation, the open web standards company, before making its recent happenings which are integrated as Foundation is equally attached to very well-structured semantic HTML and, more importantly, accessibility. As MDN organizes the page layout well, it is for developers with Foundation's flexible grid and Mobile First approach.

Tailwind CSS: Refactoring UI

The website provides samples and lessons about UI design. This particular site is built with Tailwind CSS. The utility-first approach enables maximum design flexibility without relying much on the pre-built UI components, hence allowing Refactoring UI to use Tailwind for building style in the most flexible way. Such flexibility will provide developers with freedom in building and maintaining a creative user interface, consequently propagating performance efficiency.

Bulma: Behind Laravel's Clean Look

The official Laravel website, home to one of the world's most popular PHP frameworks, is built using Bulma. Laravel is known for its emphasis on simplicity and elegance, and Bulma complements this perfectly. Built on a lightweight Flexbox foundation, Bulma offers clean, pre-designed components and a mobile-first layout that helps the Laravel site maintain a polished, professional look while ensuring a smooth user experience.

Skeleton: Ideal for Portfolios and Landing Pages

When it comes to small-scale projects like personal portfolios or simple landing pages, Skeleton is a go-to choice. It is a minimalist, lightweight CSS framework that focuses only on the essentials, i.e., basic grid layout and typography. This stripped-down approach makes it perfect for quick-loading, no-fuss websites, which is why it is especially popular with small businesses and solo developers.

These case studies highlight just how flexible CSS frameworks can be. Whether you are building something large and complex, like Spotify or the Mozilla Developer Network, or something lightweight and focused like a portfolio page, there is a framework suited to your needs. It all comes down to finding the right balance of scalability, customizability, and ease of use for your specific project.

COMPARATIVE ANALYSIS

We contrast five most widely used CSS frameworks: Bootstrap, Foundation, Tailwind CSS, Bulma, and Skeleton, in our research. The analysis is done according to a number of key features, including idea or methodology, popularity, size of the framework, browser support, responsiveness, flexibility for customizing, and ease of learning. This comparison assists in bringing out the positives of every framework and facilitating the decision-making of a developer in using any of them in various stages of web development, whether large projects or light designs.

Popularity and Support

The Bootstrap has a very large community, sufficient documentation, and lots of resources that are ready to go. Therefore, it is ideal for beginners and projects that need quick solutions. Foundation has a smaller but stronger community that lends life to projects that focus on semantic HTML and accessibility. Tailwind CSS enjoys a rapidly expanding community and good documentation, appealing to developers seeking granularity and customization.

Framework Size

These two are moderately sized frameworks and will, one time or another, affect your website's loading time. With a customizable Tailwind CSS, the developer can add only the necessary utility classes, leading to reduced file size and improved performance.

Browser Compatibility

These frameworks offer good support to browsers assuring compatibility across modern browsers.

Responsive Design

Bootstrap, Foundation and Tailwind CSS have all top-of-the-line responsive designs, which allow developers to craft website designs that not only look perfect but also respond to the resolution and device change.

Customization

Bootstrap has pretty good customization as SASS variable and component overriding. Foundation supports the most important kind of deep customization, helping developers to achieve the design on which they rely. Tailwind CSS offers unparalleled customization through its utility classes, enabling developers to create unique and highly customized designs.

Learning Curve

Bootstrap has a relatively easy learning curve due to its extensive documentation and pre-built components. Foundation has a moderate learning curve, requiring developers to understand its underlying principles and customization options. Tailwind CSS also has a moderate learning curve, requiring developers to familiarize themselves with its utility classes and configuration options.

FINDINGS AND INTERPRETATION

From CSS framework to today, it has changed so dynamically and alive to developing contemporary webs. Component-based frameworks, such as Bootstrap and Foundation, have very well supported the quickly rapid development and the maintain design consistency of UI elements that are in place but have their own disadvantages such as file size and a rather unfortunate degree of customization flexibility.

In comparison to component-based frameworks, utility-first frameworks like Tailwind CSS provide that less restrictive view; rather, they would give a way for designers to produce a totally personalized design without going through the already ready-to-use component. Besides that, extra learning curve is a downside of this philosophy, but far away from the positive side is better performance gains and code cleanliness by not having excess styles. Hence, the popularity of Tailwind CSS right now is becoming more like a whirlwind in the industry because there came a time that it is favourable for flexible build and modular design.

Bulma and Skeleton; with their different advantages and disadvantages, Bulma aptly simplifies front-end development through the modern Flexbox system while still lightweight and easy to use, whereas Skeleton is a love child of minimalism and strictness. Their target site is that of small projects and incredibly fast-loadable pages, barring everything else in the world of styling.

Now any CSS framework you will choose will depend on the requirements of the project and what has been the acquaintance of the developer with it. Naturally, Bootstrap remains most beginner-friendly, as support is often very broad-ideal for fast development and enormous applications. Foundation, however, is best for businesses and professional applications, for a very powerful focus on accessibility and semantic HTML. Tailwind CSS appeals to those aiming for total customization and optimization, while Bulma and Skeleton offer lighter frameworks aimed at projects where speed and simplicity reign supreme. As shown in Table 1, Bootstrap and Tailwind CSS lead in popularity and responsive design support, making them preferred choices for projects where these features are prioritized.

Table 1. Comparative analysis of key features of five popular CSS frameworks.

Features	Bootstrap	Foundation	Tailwind CSS	Bulma	Skeleton
Approach	Component-Based	Component-Based	Utility-First	Component-Based	Minimalist
Popularity	Very High	Moderate	Growing	Moderate	Niche Usage
Framework Size	Moderate	Moderate	Customizable	Lightweight	Extremely Lightweight
Browser Support	Excellent	Excellent	Excellent	Excellent	Good
Responsive Design	Excellent	Excellent	Excellent	Excellent	Basic Support
Customization	Moderate	High	Very high	Moderate	Low
Learning Curve	Easy	Moderate	Moderate	Easy	Very Easy

No single CSS framework is best for every circumstance, as these findings show. Different frameworks have strong points that are meant for specific project requirements, performance considerations, and user preferences.

CONCLUSION

From the comparative analysis of Bootstrap, Foundation, Tailwind CSS, Bulma, and Skeleton, it is clear that each CSS framework prepared in different ways satisfies different project parameters and developer proficiencies.

Bootstrap, the most popular framework, follows the component-based approach, offers a more-than-moderate framework size, fantastic browser support, and an easy learning curve. So, it is very much recommended for beginners and rapid development since it constitutes a wide range of pre-built UI components. On the contrary, its customization capabilities are quite moderate when compared to other more flexible frameworks.

Foundation is a freely customizable component-based framework that supports good responsiveness, which places it among the main competitors for professional developers who significantly need design control. However, the framework has some amount of complexity that makes it slightly more challenging for beginners to use than Bootstrap.

With its utility-first orientation, Tailwind CSS affords customization opportunities without restraint on framework size that is in line with any project-specific requirements. It ensures impressive browser support and responsive design, but the moderate learning curve will see developers struggling to embrace the utility-based methodology before they can enjoy its full benefit.

Bulma takes the same component approach as Bootstrap and Foundation but sets itself apart in lightweight structure and clean syntax based on Flexbox. It lends itself to customization and is friendly to beginners, making it a perfect pick for developers that like simplicity along with responsiveness.

Skeleton, the lightest of all, is, by all means, a minimal framework more suited for small projects. It is small in size and provides a primitive responsive grid system, which ensures effective lightweight designs. However, it lacks built-in UI components and is not as extensive as other frameworks.

Final Thoughts

The project size, performance needs, and developer's expertise determine the CSS framework:

- Bootstrap is mostly apt for speedy, effective development, specifically for novices.
- Foundation is meant for professionals who need optimal flexibility and control designing.
- Tailwind CSS is one of the best frameworks for those who consider customization over everything else and performance hits next.
- Bulma bridges ease of use and customization, marking it as an excellent alternative to Bootstrap.

- Skeleton is perfect for projects that need the simplest, lightest framework-tailored solutions.

In the end, it depends entirely on the complexity of the project, whether customization is necessary, and how well a person knows the framework.

FUTURE TRENDS IN CSS FRAMEWORKS

CSS frameworks are evolving rapidly, with several emerging trends influencing their adoption and application in modern web development. Notable developments include:

- *CSS-in-JS:* This technique integrates CSS directly within JavaScript components, enhancing modularity and offering greater flexibility in managing styles within component-based architectures.
- *CSS Modules:* By locally scoping class names, CSS Modules prevent naming collisions, thereby promoting cleaner and more maintainable codebases.
- *Web Components:* These allow the creation of encapsulated, reusable UI elements with self-contained styles and behaviour, encouraging code reuse and consistency across applications.
- *Dark Mode Support:* The increasing user preference for dark mode has led many CSS frameworks to incorporate native support for dark theme implementations.

Collectively, these trends underscore a broader industry focus on improving the modularity, maintainability, and performance of CSS in contemporary web development practices.

REFERENCES

1. Mohd TK, Thompson J, Carmine A, Reuter G. Comparative Analysis on Various CSS and JavaScript Frameworks. J Softw. 2022 Nov; 17(6): 282–91.
2. Al Salmi H. Comparative CSS frameworks. Multi-Knowledge Electronic Comprehensive Journal for Education and Science Publications (MECSJ). 2023; 66: 1–35.
3. Turan BO, Şahin K. Responsive web design and comparative analysis of development frameworks. Turkish Online J Des Art Commun. 2017 Jan 1; 7(1): 110–21.
4. Jaiswal P, Heliwal S. Competitive analysis of web development frameworks. Sustainable Communication Networks and Application: Proceedings of ICSCN 2021. 2022 Jan 17; 709–17.
5. Gupta AK, Venkatesha GG, Singh K, Shah S, Goel O, Jain S. Enhancing Cascading Style Sheets Efficiency and Performance through AI-Based Code Optimization. In 2024 IEEE 13th International Conference on System Modeling & Advancement in Research Trends (SMART). 2024 Dec 6; 306–311.
6. Andelhofs I. Improving the Developer Experience when Building Web Applications. Master's Thesis. The Netherlands: Maastricht University; 2023; 1–56. [Online]. Available from: https://documentserver.uhasselt.be/bitstream/1942/41279/1/66d1c359-2515-4f0b-9024-056d564dc167.pdf
7. Marcotte E. (2010 May 25). Responsive Web Design. [Online]. A List Apart. Available from: https://alistapart.com/article/responsive-web-design/
8. Bögözi Attila. (2023 Nov 24). The Evolution of CSS: From Styling Sheets to Modern Web Design. [Online]. Attila's Blog. Available from: https://blog.bogozi.com/evolution-of-css/
9. Simplilearn. (2022). Guide to CSS Framework. [Online]. Simplilearn. Available from: https://www.simplilearn.com/tutorials/css-tutorial/css-framework
10. Jayaraman KD, Shrivastav EA. Automated Testing Frameworks: A Case Study Using Selenium and NUnit. International Journal of Research in Humanities & Social Sciences (IJRHS). 2025; 13(1): 1–6.

