

### **Preview:**

## **US**

JavaScript is the backbone of front-end web development, and its powerful frameworks have played a significant role in its widespread adoption among developers.

However, with an increasing number of frameworks available, it can be challenging to determine the right one for your next project.

Our in-depth analysis, available at <https://www.scrumlaunch.com/blog/front-end-frameworks-to-use-in-2023> compares the pros and cons of the most popular frameworks, such as React, Angular, and Vue. It provides recommendations to help you identify the perfect match for your specific project requirements.

#scrumlaunch #software #softwaredeveloper #softwarecompany #webdevelopment  
#mobiledevelopment #graphicdesign #startupbusiness #startupsupport #ai #software

## **INDIA**

JavaScript front-end frameworks reign supreme, empowering the creation of stunning interfaces and dynamic web applications. But with a growing list of frameworks to choose from, it can be hard to decide which one to use for your next project.

In this article <https://www.scrumlaunch.com/blog/front-end-frameworks-to-use-in-2023> , we provide a comprehensive analysis of the most popular front-end frameworks, including React, Vue, Angular, and more. We'll uncover their strengths, weaknesses, and best-use scenarios, helping you identify the perfect match for your specific project requirements.

#scrumlaunch #software #softwaredeveloper #softwarecompany #webdevelopment  
#mobiledevelopment #graphicdesign #startupbusiness #startupsupport #ai #software

### **Twitter:**

JavaScript is vital for front-end web development. Its frameworks contribute to its popularity. Our analysis at <https://www.scrumlaunch.com/blog/front-end-frameworks-to-use-in-2023> compares popular frameworks like React, Angular, and Vue, offering recommendations based on your project needs.

.

## Threads:

JavaScript dominates front-end web development, and its popularity is no surprise. Discover top JS frameworks like React, Vue, and Angular in our article at <https://www.scrumlaunch.com/blog/front-end-frameworks-to-use-in-2023> . Uncover strengths, weaknesses, and best-use scenarios to find your perfect match.

Alternative titles:

Top JavaScript Trends in 2023: Which Frameworks to Choose.

Ruling the Front-end: A Comparative Study of React, Angular, and Vue JavaScript Frameworks.

An In-depth Look at React, Angular, and Vue Front-end Frameworks in the JavaScript Ecosystem.



***Front-End Frameworks to Use in 2023***

JavaScript is a constant trend in front-end development. It is the only programming language native to the web browser, meaning that the code written in JavaScript works on all devices with a modern browser. JavaScript makes creating content, updating dynamically, such as animated images, pop-ups, hiding menus, tabs, and much more possible. Other language codes can be automatically transpiled into Javascript, making it something like a basic language code. On top of it, Javascript has powerful front-end frameworks, making a significant difference in front-end development speed and quality. According to a Statista [survey](#), it is Javascript used by 65% of developers. And these developers are improving Javascript performance and expanding its functionality. Wikipedia [states](#) that JavaScript is the dominant client-side scripting language of the Web, with 98% of all websites using it for this purpose. So, JavaScript has almost no competitors for front-end web development, and it will not be replaced anytime soon.

In the JavaScript ecosystem, the list of JavaScript frameworks keeps on growing. React, Angular, Vue, Svelte, Preact, Ember, and Lit are just a few on the list. Javascript frameworks are a collection of libraries offering pre-built codes, which allow faster development time, reduce code length, and simplify maintainability.

When choosing a JavaScript framework, there are many factors to consider: their features, pros, and cons. According to the results of a [JavaScript Survey](#), React is the leader in use among front-end programmers, while Angular and Vue frameworks consistently fight for the 2nd and 3rd ladders in the list. And these top 3 have been retaining their popularity in this order since 2017. However, the picture is somewhat different regarding developers' interest and retention. React is only number 3 and 4, respectively. Vue remains in the top 4 list, while Angular falls at the bottom. Svelte and Solid frameworks share developers' retention and interest. So, let's discuss what stands behind the success of the current three leaders and analyze whether retention and interest leaders have any chance to supersede any usage leaders in the nearest future.

Let's start with the absolute leader - **React**. React is an open-source library created by Facebook for developing dynamic UIs using reusable components. Front-end developers have used it in companies like Facebook, Instagram, Netflix, Airbnb, BBC, Pinterest, UberEats, PayPal, Walmart, Tesla, and many more. Since its release in 2013, it has gained tremendous popularity, a good base, multiple tools, and broad community support. Being a library rather than a framework, it offers more flexibility to users as it does not impose any restrictions or conventions on the application design. It can be supplemented with other libraries. The React community is gigantic, with more than 196k stars on GitHub. It is suitable for MVP, small and medium projects. Here are its main features:

**React offers reusable components.** React allows developers to create reusable components framed in a tree-like structure. Reusable components are part of what makes React so popular. Reusable components and templates make the development time shorter, allowing developers to update specific parts of pages without changing the entire design. The larger components can be divided into smaller ones. It makes the code very easy to maintain.

**React offers JSX coding style.** React allows for using JSX coding style, also called JavaScript XML, a markup syntax similar to HTML. Inexperienced developers may find it challenging to learn JSX syntax. Yet, JSX is not a requirement for using React. At the same time, experienced developers say using JSX makes the development and debugging process more manageable. The declarative style of React also makes debugging simple and easy.

**React offers one-way data binding.** React also uses Flux, an application architecture that supports one-way data binding. One-way binding binds the model's value to the view and does not have an additional watcher to determine if the user has changed the value in the view. The principal advantage of one-way data binding is that it gives better control over the app data.

**React has its own virtual DOM.** DOM keeps the virtual copy of a user interface in memory and allows it to update the page whenever there is a change in the data component caused either by the user or by the system. Specifically, virtual DOM is capable of comparing the previous and next states of the components. The real DOM renders the components whose states were changed. Virtual dom was actually invented to prevent whole code rendering when just part of the code changed.

**React offers React Native for cross-platform development.** React Native is a custom renderer for React. It uses native components instead of web components like React as building blocks. React Native by Facebook was a big turnaround. It was the first tool that assisted developers in creating a cost-efficient cross-platform mobile application.

**React has a low learning curve.** React is not a complete framework, so its learning curve is relatively low. It is considered one of the easiest to learn and ideal for beginners or less experienced developers. Anyone having even basic knowledge of programming can quickly learn React. It offers a [Getting Started](#) guide. However, React is constantly evolving. Therefore, developers must keep relearning the new features, while the documentation may be updated not that quickly.

So, React is mature enough and has excellent community support, given its long existence. It offers an easier learning curve. Being a library rather than a framework makes its usage more limited, but when bundled with other libraries, it becomes powerful. The ability to integrate it with other frameworks seamlessly gives it an excellent advantage for those who would like some flexibility in their code. The job market for React also looks fantastic. Thus, it is a good choice for someone getting started with front-end JavaScript frameworks, both for startups and developers. React is not going to give up its leading positions shortly.

The second JS framework usage leader is **Angular**. Developed by Google, Angular is an open-source TypeScript-based framework. With a collection of well-integrated libraries and tools, the framework allows developers to build, test, and modify the code. First released in 2010 as AngularJS or Angular 1, it was an open-source framework that used HTML as its template language. Version 2 of this framework and all other versions are also known as Angular (without the JS). It was released in 2016 and changed the template language to

TypeScript. It is Angular version 2 that captured immense popularity amongst developers. Currently, the Angular community has more than 84k stars on GitHub. Angular has introduced features like dependency injection that supports testing and inbuilt attributes, also known as directives. It also introduced RxJS, a reactive programming library that handles asynchronous data with multiple events. Angular is constantly improving. With each new release, Angular adds new features and optimizations. Angular is used by many companies like Deutsche Bank, IBM, Freelancer, Gmail, Forbes, PayPal, YouTube, eBay, BMW, LEGO, etc. Being a fully-fledged framework, unlike React, Angular is preferable and valuable in developing large-scale enterprise-based applications. Here are its main features:

**Angular organizes code into different modules.** Angular applications are modular, and Angular has its own modularity system called NgModules. These modules are small pieces of code with independent program functionalities. The modules available in Angular include root, core, shared, and feature modules. While a small application might have only one NgModule, most applications have many more feature modules. Angular modules enable developers to improve the organization and structure of the application, which, in turn, enhances code reusability and testability.

**Angular offers Cross-Platform development.** Angular offers modern web development features to build apps for all platforms simultaneously. Code can be reused to be implemented in SPAs and PWAs, web apps, mobile projects, and all modern web browsers.

**Angular has Model-View-Controller (MVC) architecture.** MVC is an architectural design that helps simplify the creation of huge applications by splitting the framework into three major conceptual components: Model, View, and Controllers. Each of these components is built to handle specific development aspects of an application which offers excellent scalability. MVC is popular as it isolates the application logic from the user interface layer and supports the separation of concerns. Thus, MVC supports rapid and parallel development, makes it much easier to debug problems, and allows seamless integrations of advanced features.

**Angular is two-way data binding.** The bi-directional data flow enables Angular to connect DOM to the model data through the controller. It allows for real-time synchronization between the model and view, ensuring they keep up with each other, with any changes to the model immediately reflected on the view.

**Angular has a Command Line Interface (CLI).** The tool allows users to type in commands that can produce immediate results. It also offers a neat coding feature of TypeScript. As this interface type needs much less memory (RAM), it helps reduce the application size of Angular, offering an enhanced development experience. The tool can be used directly in a command shell or indirectly through an interactive UI like Angular Console.

**Angular has a steep learning curve.** Considering it is a complete solution, the framework is not beginner-friendly. It also offers an [Introduction guide](#) and a great deal of documentation. Developers are never helpless in the Angular community. Still, learning

Angular is quite challenging for beginners as they should learn associated concepts like TypeScript and MVC. It explains why Angular is not among the leaders when it comes to developers' interest. But developers continue to use Angular, despite the rise of React, as it offers a comprehensive solution that React doesn't. It means that developers don't need to piece together different libraries to get a complete solution – they can just use Angular, as it includes everything required to build a complete web application, including a UI, state management, and an architectural system.

So, Angular is the oldest framework among all and the most mature. It is not the "next big thing". It is just a big thing. It is a full package, versatile, well-maintained, and performs well when building apps for different platforms at once. Angular is the best option for developing large-scale enterprise applications. However, the learning curve is steep, and development concepts in Angular may put off new developers. Angular is a good choice for companies with large teams and developers already using TypeScript.

And the third JS framework usage leader is **Vue.js**. Released in 2014, Vue is an open-source JavaScript framework, also known as the "Progressive JavaScript Framework". With 200k stars on Github, Vue.js has attained much popularity among developers in a relatively short time. Its popularity is primarily due to its ease of use and scalability - it can quickly scale between a library and a full-fledged framework. It allows the creation of interactive UI elements that are lightweight and easy to adapt, with a possibility for small projects to scale up to larger ones. Its main aim was to improve the features of front-end frameworks like Angular and React. For instance, it has the virtual DOM and templating syntax used in React and also the two-way binding and directives used in Angular. It has a simple structure with its own command-line interface (CLI) for generating code, developing, and deploying an app. Vue 3.0 version brought some long-awaited implementations. It has full support for TypeScript, a new build toolchain utilizing Vite, and improved management of state through Pinia. Vue's internal structure has been transformed into a series of individual modules. The new architecture provides better support and allows end-users to reduce execution time by up to half. The Composition API built into Vue 3.0 makes organizing and reusing code easier, which is a must for large projects. Vue is more suitable for small and medium-sized applications. However, Vue 3.0 makes the framework attractive not only for small and medium-sized but also for large projects because there is no longer a performance penalty, and readability has been improved. The framework is not yet so popular among market giants, just like its competitors, but is used in production by companies such as Alibaba, Gitlab, Adobe, Tencent, Xiaomi, Grammarly, Netflix, Reuters, and 9Gag. The main features of Vue.js are:

**Vue suggests component-based programming.** The framework doesn't impose a strict routine but allows developers to decide what they want to create. Vue permits developers to structure Web components and custom HTML elements that can be used and reused in web apps and web pages.

**Vue offers Model-View-ViewModel (MVVM) Architecture.** In MVC, the controller is the entry point to the app, while in MVVM, the view is the entry point to the app. It dramatically enhances UI by simplifying user interface event-driven programming. MVVM separates the



user interface from the app logic and makes the code event-driven. So when the UI gets outdated, for instance, programmers don't have to wait for a redesign of the whole app but just plug in a new template.

**Vue allows for cross-platform development.** Vue makes it easy for developers to build top-rated mobile apps with excellent features such as cross-platform compatibility. The ecosystem of Vue.js is vivid. By leveraging Quasar Framework, NativeScript, Ionic, Framework7, Vuetify, and other mobile UI frameworks, Vue.js is also used for building apps for Android and iOS.

**Vue has its own Virtual DOM.** Vue uses virtual DOM, which is also used by React frameworks. The changes are not made to the DOM. Instead, a replica of the DOM is created, which is present in the form of JavaScript data structures. Whenever any changes are to be made, they are made to the JavaScript data structures, and the latter is compared with the original data structure. The final changes are then updated to the real DOM. It is good in terms of optimization, as it is less expensive, and the changes can be made faster.

**Vue supports two-way bindings using V-model.** This functionality is called reactive two-way data binding, which means if you change the input value, the bound data will be changed. The v-model directive makes two-way binding between a form input and app state very easy to implement. It keeps the app data in sync with the DOM automatically.

**Vue has a low learning curve.** According to the developers, Vue.js is one of the easiest frameworks to learn. Any developer can master it in a short time. To begin learning, a developer needs to know JavaScript and HTML. Besides, VueJS has excellent [documentation](#) that is very helpful for beginners. But, for now, there is a lack of support due to a relatively smaller community.

**Vue also has a Command-line interface (CLI),** which allows running Vue commands in the terminal. It is used for the development and installation of the Vue framework core libraries, as well as third-party plugins. It reduces the time of initial project setup and scaffolding to a minimum.

VueJS is a developing framework. It is the youngest among the usage leaders. Unlike its competitors, it is not backed by any tech giant but is entirely supported by sponsors and its open-source community. There are fewer job opportunities with VueJS compared to React and Angular. But the demand for Vue.js has picked up relatively quickly and is still gaining popularity. It is because Vue stands out as a simple and straightforward framework that is easy to learn and use. VueJS has taken the good parts from Angular and React and is much faster compared to them because of its lightweight library. With these benefits, Vue can challenge the dominance of React and Angular in 2023.

And finally let's review the retention and interest leaders Svelte and Solid to learn if any of them is going to be the next big thing.

**Svelte** has been the number one framework attracting front-end programmers' interest since 2019. It is a free and open-source front-end compiler rather than a framework or library for building fast reactive web apps with less code. It was released in 2016 and is maintained by the Svelte core team members. Currently, it has more than 62k stars on GitHub. Google backs Vue.js and Angular, React is supported by Facebook, while Svelte has no significant support yet. It compiles HTML templates to an ideal JavaScript at a run time, avoiding virtual DOM. Instead, it uses a custom JavaScript Virtual Machine specifically designed to build user interfaces. It is the main difference between Svelte and other frameworks, making it significantly faster than other frameworks like React and Angular and even faster than Vue. It is compatible with native HTML code and doesn't impose a strict structure, resulting in smaller application sizes. Despite being young, such well-known companies use Svelte for their websites: The New York Times, Rakuten, Phillips, Absolute Web, Cashfree, Razorpay, and Godaddy. And the interest in the framework is expected to rise. Svelte main features are:

**Svelte doesn't use a virtual DOM.** Being a compiler, it does not require a download of any additional libraries to the browser to run the Svelte code. It compiles code into tiny, pure Vanilla JS. As a result, the code works much faster from the beginning, making a potential app product much lighter and more user-friendly. It is the first and most notable difference that Svelte has presented to the world of coding. - it reduces the size of the files significantly and offers lightweight programs for websites and web apps.

**Apps created with Svelte have less code**, which means less data to download and, as a result, faster loading time. Svelte enforces this by introducing a simple format written in TypeScript. As the Svelte website mentions, writing less code takes less time and leaves more time for other things like optimization, "nice-to-have" features, etc. The code written in Svelte is usually 30-40% shorter than in React or Vue.

**Svelte has two-way data binding**, just like Angular and Vue. It combines string interpolation and event binding to react to events and output data at the same time. It ensures that the Logic and the View are always in sync.

**Svelte is also cross-platform.** [Svelte Native](#), released in 2021, allows Svelte developers to build native Android and iOS applications. The Svelte Native website states it's a "mobile application framework powered by Svelte that enables you to build mobile apps using the web framework you already know."

**Svelte has a low learning curve.** For developers having a general knowledge of HTML5, CSS, and Javascript, learning Svelte shouldn't be a problem. There's no need to know complicated libraries or state wrappers because the markup may be accessed directly. So it's easy to learn and has excellent documentation. Svelte has a [tutorial](#) with the necessary documentation to get started. Still, it is worth mentioning that finding answers to questions and solutions to bugs that may appear later in development can be problematic because of the small community.



So, Svelte is still in its early stages of development, with job offers demand for it being deficient for now. Currently, it has insufficient tools, small, although growing, community, and a lack of support for immediate answers. In fact, it started too late to stand a chance against React, Angular, or Vue.

And finally, **Solid** is a declarative JavaScript library with an API similar to React that popped up just in 2021. And the same year, it was ranked number one by front-end framework developers in terms of satisfaction. Currently, it has over 22k stars on GitHub and is sponsored by reputable companies such as Builder, Netlify, and Vercel. Like most JavaScript frameworks, Solid is built on the concept of components. The framework is declarative and doesn't use virtual DOM, and in this feature, it is similar to Svelte. This way, the compiler can optimize the initial render, and the runtime optimizes updates. In terms of performance, Solid outpaces React, Angular, Vue, and Svelte. It is the fastest among them. Solid is heavily inspired by React. And, in many ways, there are quite a few similarities. Including a similar syntax with React, support for JSX, API for Hooks, and features like Web Components and SSR. So, the main Solid features are:

**Solid doesn't use a virtual DOM.** Like Svelte, Solid describes the virtual DOM as pure overhead that only slows down the library's performance. Solid reaches incredible speeds without using a virtual DOM by compiling its templates down to real DOM nodes and wrapping updates in fine-grained reactions. This way, when state updates, only that code, depending on it, continues to run.

**Solid offers one-way data binding** like React, allowing more control over the application data.

**Solid has a low learning curve.** Just like React, it is one of the easiest to learn. With a basic understanding of HTML, CSS, and Vanilla JavaScript, it shouldn't take longer than three months for a developer to grasp Solid. Community members are also creating numerous valuable resources, [tutorials](#), and guides on the library. Still, the framework community is very limited.

So, Solid shows promise. But for now, it is still in its early stages, just like Svelte. One day this framework can be used as an alternative to React, but at the moment, it has no chance to compete with it. There's a lot of work to be done concerning community building, bringing some alternatives to React Native, etc.

A paradigm shift is what stands behind the prompt outburst of the popularity of both Svelte and Solid. JavaScript frameworks revolve around making it simpler to apply DOM updates. Svelte and Solid brought the ability to update changes on the DOM without reloading a whole element. However, it's much more likely that the better-funded frameworks like React, Angular, or Vue with larger teams incorporate some of the ideas from Svelte and Solid than the developers pivot from their favorites to these new tools. Still, as both frameworks are gaining popularity at an unprecedented rate, it is worth keeping an eye on them.

When choosing the best JS framework, you should pay attention to popularity, availability of specialists on the market, speed of stack development, and project requirements. Currently, there are three leaders React, Angular, and Vue. And these leaders will not give ground to any other framework any time soon, so the choice is mainly between these three. Based on our overall development experience, we can recommend Angular for complex and large-scale projects. If you have limited time and need a small project, you can confidently choose Vue. React is excellent in all other cases, but especially if you need high performance and the project needs to be implemented in a tight timeframe.

*Reading time* **14 min 57 sec**