Analysis and Comparison of Different Frontend Frameworks

Suryaansh Rathinam[1][0000-0002-0370-8763]

[1] Manipal Institute of Technology, Manipal, Karnataka, India

Abstract

A frontend framework is pre-written code that provides the architecture for a project, along with certain features to help with the development process. With the availability of many frameworks, each with its own features and advantages, it is important to understand the difference between the various frameworks and make the right choice of framework for a particular project. This paper elaborates on the features of three popular frameworks: React, Angular and Vue, draws a comparison among them, and provides guidelines on how to make the choice of a suitable framework for a project.

Keywords: Frameworks, React, Angular, Vue, JavaScript

1 Introduction

Over the past couple of decades, the web has grown exponentially, and with it has increased the size and complexity of various web applications and services provided over the web. This has led to developers needing higher-level tools and technologies which cater to the development of complex web applications, handling more user traffic.

To cater to this increasing need, various frontend web-development frameworks such as ReactJS, AngularJS, VueJS, and NextJS, to name a few, were developed. While all these frameworks find their base in JavaScript, each offers developers a different set of features, structures, and options.

The objectives of this paper are as follows-

- To analyze various popular frontend frameworks that have come into use over the last couple of years.
- To do a qualitative comparison between the features and structures of these frameworks.
- To break-down some of the large-scale applications built using particular frameworks.
- To assess how to choose between the vast array of frameworks available.

2 Research Questions

This paper aims to tackle the following questions:

1. What are the different features provided by the various frontend frameworks?
2. How can one choose between various frameworks for a web application?

3 Contributions

1. A detailed comparison of various frontend frameworks' features, functionality, and structure.
2. Case study analysis of large-scale applications built using particular frontend frameworks
3. Methodology and suggestions in making choices of frameworks for particular projects.

4 Literature Review

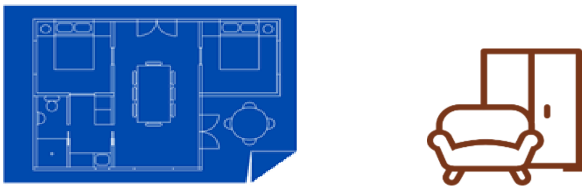
Over the past couple of decades, web development, along with its associated technologies, has continuously changed. With the advent of the Web in the 1990s, early web pages, built primarily using HTML and CSS, were text-based websites that did not have to deal with much traffic. As time passed and the size of the web expanded rapidly, so did the need for more complex web applications to provide services to the users.

To simply work for developers and facilitate faster and more efficient development of websites, JavaScript-based frameworks started emerging in the 2000s. The first of these was jQuery [1], launched in 2006. jQuery is referred to as a fast, lightweight, JavaScript library, that uses an API to make tasks like document traversal, manipulation, and event handling simpler. [1] It is referred to as a library and not a framework; for this paper, we will consider JavaScript libraries such as jQuery and ReactJS alongside JavaScript frameworks.

While subtle, the distinction between a library and a framework is essential to understand. A library consists of a reusable package of predefine functions, objects, and methods that the developer can use in the project as and when deemed fit. Some of the popular JavaScript libraries are jQuery, ReactJS, and D3.js.

On the other hand, a JavaScript framework is a pre-written code that provides an architecture for the project. It provides a standardized structure that developers can follow and extend based on the framework's features and the developer's needs. Some of the popular JavaScript frameworks are AngularJS and VueJS.

While there has not been extensive research comparing the various available frameworks, the works of [2] and [3] serves as a basis for me to build upon further in this paper.



Framework: Blueprint of house

Library: Furniture inside the house

Fig. 1. Depiction of difference between framework and library

After jQuery for developed in 2006, the upcoming years saw the release of many new frontend frameworks and libraries, which helped ease the development process.

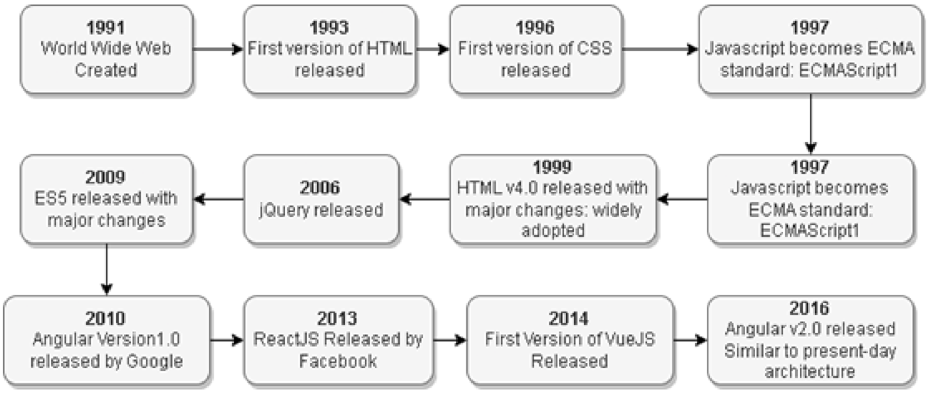

1991
World Wide Web Created
1993
First version of HTML released
1996
First version of CSS released
1997
Javascript becomes ECMA standard: ECMAScript1
1997
Javascript becomes ECMA standard: ECMAScript1
1999
HTML v4.0 released with major changes: widely adopted
2006
jQuery released
2009
ES5 released with major changes
2010
Angular Version1.0 released by Google
2013
ReactJS Released by Facebook
2014
First Version of VueJS Released
2016
Angular v2.0 released Similar to present-day architecture

Fig. 2. Advancement in Web-Development over the years

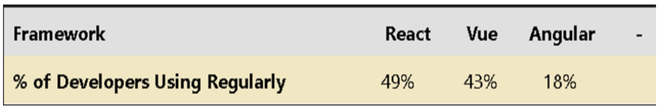
5 Frameworks Analysis

5.1 Methodology

For the purpose of analysis and further comparison, this paper will be focusing on three frameworks in particular namely React, Angular and Vue. While apart from these frameworks there are many more JavaScript frameworks used for frontend development, such as jQuery, Svelte, Ember etc., the popularity of React, Angular and Vue far supersedes that of the other available frameworks [Table 1]. Due to their vast popularity these frameworks have been used for the development of many large-scale applications. Hence, in this section we will focus on each of the individual frameworks, their features and functionality.

According to a survey by JetBrains in 2021 [2], comparing the popularity of JavaScript Frameworks among developers.

Table 1. Popularity of Frameworks among developers



Framework	React	Vue	Angular	-
% of Developers Using Regularly	49%	43%	18%	

5.2 React JS

ReactJS is a JavaScript library developed and maintained by Facebook (now Meta), used to create interactive user-interfaces. [3] React's wide popularity is primarily due to its improved performance with the help of the use of the virtual DOM over the actual DOM, ease of learning, and reusable components.

Virtual DOM vs. Actual DOM:

. The Document Object Model (DOM) is a programming interface for web documents. The DOM represents the document as nodes and objects so that programming languages can interact with the page. [4] It further represents web pages in a tree structure, with the nodes containing objects that developers can modify using JavaScript. Hence it is a representation of the user-interface of the Web Application. Any time an object of the DOM is updated, the updated component, along with all its children, gets re-rendered. This re-rendering of the UI makes the DOM slow, as the UI must be re-rendered with every DOM update.

React handles this by making use of the Virtual DOM (VDOM). The VDOM is a virtual representation of the Real DOM, where React first updates any state changes and the affected components, then compares the VDOM obtained with the snapshot of the VDOM before the update occurs. This helps React figure out which objects have been changed, and only those objects are updated in the Real DOM; instead of the entire Real DOM being updated, only the, affected objects are updated.

This greatly helps React improve the performance and memory utilization of the DOM

.

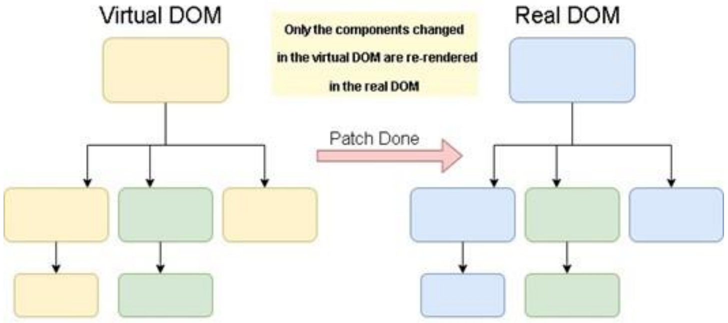

Virtual DOM
Only the components changed in the virtual DOM are re-rendered in the real DOM
Real DOM
Patch Done

Fig. 3. Use of Virtual DOM to handle updates to the DOM

In addition to the performance optimization offered by React, it is also preferred by developers (Table 1) because it is extremely easy for new developers familiar only with HTML and Vanilla JS (refers to the use of JavaScript without any of its libraries), to adapt to ReactJS. This is because ReactJS allows developers to write code in JSX, a syntactic extension to JavaScript, which allows HTML code to be embedded along with the JavaScript functions [3].

Lastly, Meta (Facebook) also provides browser extensions to simplify the process of debugging React Applications. It allows developers to inspect component hierarchies in React and record the web application's performance. [5]

5.3 Angular

Angular is a popular JavaScript frontend framework developed by Google. Angular 1 (also known popularly as AngularJS) was the version released by Google in 2010; however, in 2016, Google released Angular 2, completely rewriting the original version released [6]. While over the years, AngularJS was popular due to a variety of its features, as of January 2022, Google has officially announced the end of support for AngularJS [7]. Hence, for this paper, we shall primarily focus on Angular2 and its subsequent versions, which are currently in wide use among developers, the features offered, and how it differed from its predecessor, AngularJS.

Angular is a development platform built with the help of Typescript, which includes a component-based framework, a collection of libraries, and a set of developer tools to build and test applications [8]. Some of the features of Angular that impact its use among developers are as follows:

Compared to React, which uses a Virtual DOM as discussed previously, Angular directly interacts with the Real DOM, updating the entire DOM tree when any change to the user interface occurs.

Architecture

. While AngularJS supported MVC and MVVM Architecture, Angular2, taking inspiration from React, shifted to a Component-based Architecture. In the Angular Framework, Applications consist of Angular Components organized into NgModules. All the components and modules in Angular are classes with associated decorators that provide Angular with the required metadata.

Components have associated templates that define views that Angular can access and modify based on the app’s logic. The components also utilize services, providing functionality to the application that may not directly be related to the view; these services are injected into the components and dependencies. The components of an application consist of many views arranged hierarchically.

The template associated with a component works to combine HTML with the Angular markup. Templates have associated directives that provide the program logic. Angular evaluates the program logic and resolves the template's binding (Event and Property Binding). [9]

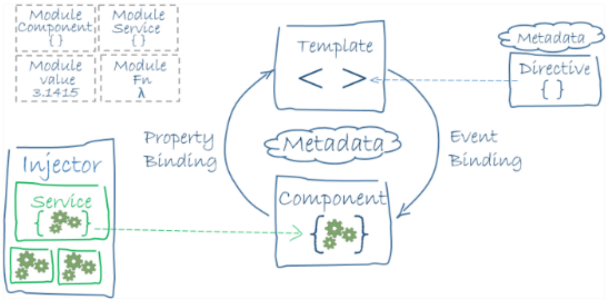


Fig. 4. Angular Architecture (From [8])

Data Binding in Angular.

Data Binding refers to keeping the UI of the page up to date based on the state of the web application. [8] Angular provides three categories of Data Binding:

- One-way from data source to view target
- One-way from view target to data source
- Two-way

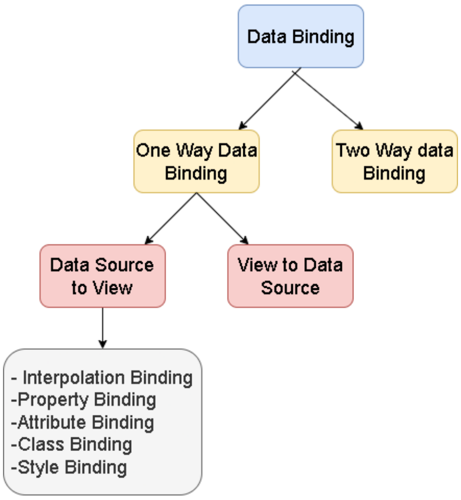


Fig. 5. Types of Data Binding

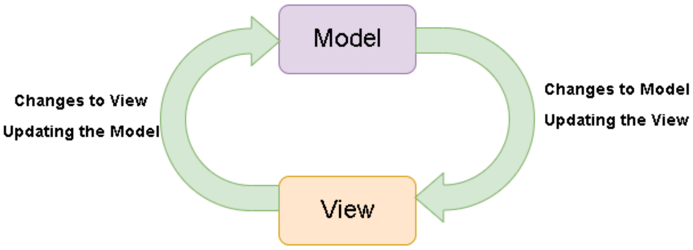


Fig. 6. Two-Way Data Binding

Furthermore, Angular also provides inbuilt support to set up Unit Testing and Integration Testing with the help of the Jasmine testing Framework, an open-source behavior-driven development framework for testing JavaScript code [10] , and Karma as the task-runner for tests.

Changes From AngularJS to Angular2+

. While the upgrade from AngularJS to Angular2 and then subsequent updates to Angular2 have resulted in many changes in the frameworks, some of the significant differences between the versions are:

- While AngularJS is a JavaScript-based framework, Angular is built using TypeScript, which is statically typed and helps ensure fewer errors and better code understandability.
- Applications made using Angular are also optimized for mobile browsers, whereas AngularJS provides no support for mobile devices.
- While AngularJS has an MVC based architecture, the architecture of Angular is primarily based on Components consisting of templates and directives.
- AngularJS relies on third party tools for testing where as Angular comes support for unit and integration testing based on Jasmine and Karma.

5.4 Vue

VueJS is a JavaScript framework for building a user interface built upon HTML, CSS, and JavaScript. [11] Of all the frameworks analyzed in this paper, Vue is the newest, having been created by Evan You in 2014. Even despite its comparatively late emergence, due to the robust set of features offered by it, it has grown to give competition to similar frameworks like React and Angular.

Architecture

. VueJS is based on the MVVM (Model – View – ViewModel) architecture which consists of the following three components:

- Model: Represents the data access and business logic of the application
- View: It represents the UI of the application that the user interacts with.

- View Model: It consists of the logic of the view layer, linking it to the model layer, and processing the interactions between the two.

VueJS primarily focuses on implementing the functionality of the ViewModel layer.

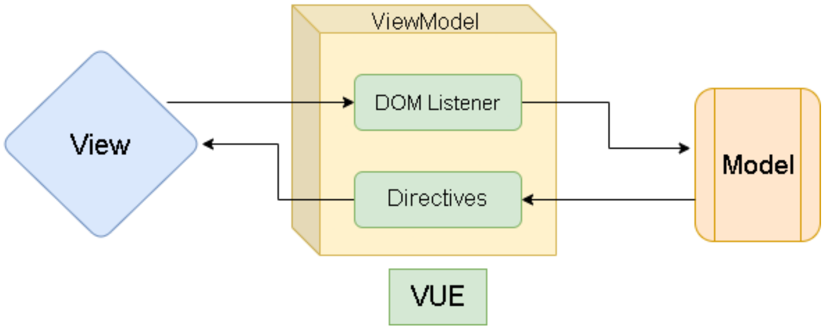


Fig. 7. VueJS Architecture

VueJS follows a component-based structure similar to both React and Angular. Components help split the user interface into simple, easy-to-understand, and reusable pieces of code. This modularization process helps improve the development and maintainability of web applications.

Since Vue was initially developed by Evan You to take the best features of Angular and make a lightweight custom tool for development, Vue contains some of the primary functions and features of Angular such as templates, views, and directives, as discussed previously.

Rendering Mechanisms.

Similar to React JS, Vue, too, uses the Virtual DOM to handle rendering and re-rendering of the user interface, hence improving the performance and memory utilization of the web applications.

VueJS templates are first compiled into render functions that return the VDOM trees. Then, on runtime, the VDOM tree is mounted on the actual DOM, which renders the web application. The render functions are generated to track the dependencies, and in case of any changes, an updated VDOM tree is created. The updated VDOM tree is compared with the older version, and only the particular components that were changed are updated in the actual DOM.

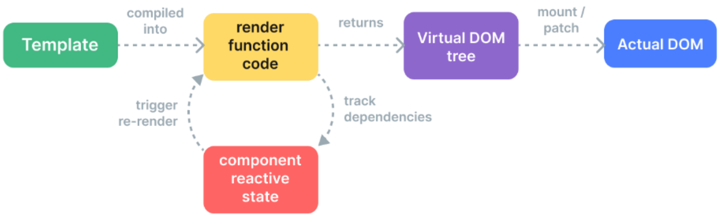


Fig. 8. Rendering pipeline of VueJS using Virtual DOM (From [11])

Overall, the popularity of VueJS has grown over the last couple of years. Despite not being backed by a large corporation like Meta for React and Google for Angular, it has still been able to give them competition and is now widely adopted among developers. Vue JS is maintained primarily by its creator Evan You, and a large open-source community, leading to many libraries such as Vuex, Element, and the Vue CLI being developed. Vue has worked to combine the rich functionality of Angular with the lightweight, fast and easy learning curve of ReactJS.

6 Case Studies.

In this section, we will analyze some of the large-scale applications built using the frontend frameworks discussed previously and the advantages of using the particular framework.

6.1 React: Facebook

As ReactJS was developed at Facebook and is maintained by developers at Facebook and a large community of developers, it is no surprise that the Facebook website is developed using ReactJS and the mobile app using React Native.

Facebook is a social networking platform founded in 2004, with over 2.9 billion users now. It allows users to connect with other individuals, share updates and photos of their lives, and direct messaging services. It has also grown to offer additional services such as live streaming, a marketplace, and news feeds.

React is now among the most popular frameworks used among developers. Facebook has been using React to develop their web application as it provides efficiency and optimization with the Virtual DOM and easy SEO optimization, among other features discussed in Section 5.1. Furthermore, since React is developed by Facebook and then used for building their applications, it is easier to fix any issues or work on additional functionality required in React or the Facebook Web Application.

6.2 Angular: Firebase

Since Angular is developed and maintained by Google, many of the internal products used by Google and many public-facing services use Angular as part of their technology; one of the most prominent services of Google built using Angular is Firebase.

Firebase is a Backend-as-a-Solution (BaaS), providing developers with the ability to build and manage their applications' backend; it provides services such as a real-time database, authentication, cloud messaging, and hosting, which developers can easily integrate into their existing applications.

Most of the reasons for Angular’s adoption have been discussed in section 5.2; in addition to its various features, such as in-built testing, two-way data binding, and dependency injection, it offers high-end functionality to developers, as there is no dependency on any third-party libraries and packages, as is with the case React and Vue. Instead, all the functionality needed by developers comes covered within the framework.

6.3 Vue: GitLab

VueJS which was created by Evan You and released in 2014, has fast risen to popularity in the developer community, and has come to be adopted in the development of many large-scale popular applications, such as GitLab.

GitLab is a DevOps software package that offers remote access to GitHub repositories, along with features that simplify the software development life cycle, such as collaborative development, Continuous Integration- Continuous Deployment (CI/CD) pipelines, bug-tracking, and code review.

The primary reason for adopting VueJS for the development of GitLab has been its simplicity and ease of use. [12] Its easy-to-understand source code and documentation, coupled with the robust set of features provided by it, further help to simplify the development process. Hence VueJS provides developers with a balance between the structure and simplicity of the framework and enables them to implement the same functionality and features required with less code. [12]

7 Results and Discussion

7.1 Comparison

After an elaborate discussion of the features offered by the various frameworks, this section works on a side-by-side comparison of these frameworks.

Fig 10. Below depicts that over the last 5 years, the popularity of React, and hence the number of its downloads, has been increasing at a rapid pace, while for Angular and Vue, the number of downloads have been increasing, but at a much slower pace compared to that of React. This correlates with the data in Table 1, pertaining to the current popularity of the Framework.

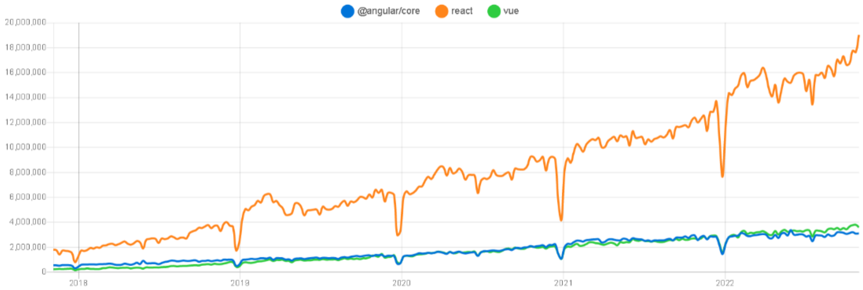
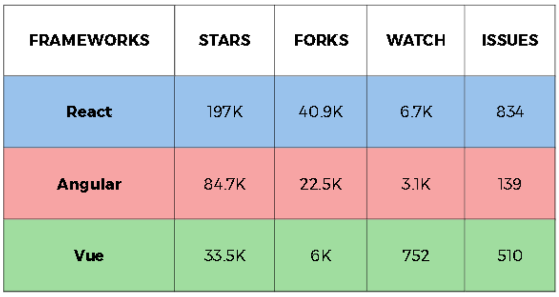


Fig. 9. Comparison of NPM Downloads of the 3 Frameworks over the last five years [13]

Table 2 below depicts the comparison of the GitHub repositories of the three frameworks, which is the open-source popularity of these frameworks, as we have seen in Fig 10 as well, the React GitHub repository has the most number of stars, forks and watchers, followed by that of Angular and then Vue. The comparatively low number of interactions on Vue’s repository can be due to its recent emergence in 2014, compared to that other two frameworks.

Table 2. Comparison of GitHub Repositories of the 3 Frameworks



FRAMEWORKS	STARS	FORKS	WATCH	ISSUES
React	197K	40.9K	6.7K	834
Angular	84.7K	22.5K	3.1K	139
Vue	33.5K	6K	752	510

The tables Table 3-5 given below, draw a side-by-side comparison of the differences in the features of Angular, React and Vue, taking two of the frameworks at a time, based on the discussions on these frameworks in Section 5.

Table 3. Comparison of Angular and React

Angular	React
With a large core library, it reduces any dependencies on external libraries.	It has a small core library with external libraries to provide additional functionality.
It has a steeper learning curve.	It provides an easy learning curve for developers familiar with JS and HTML.

Based on Typescript.	Based on ES6 Syntax and JSX

Table 4. Comparison of Vue and React

Vue	React
Easier to work with for beginners not familiar with JavaScript concepts.	Easy to work with for developers familiar with functional JavaScript
It handles the mutation of data using data objects.	It handles the mutation of data using state objects.
Supports two-way data binding	Supports one-way data binding

Table 5. Comparison of Angular and Vue

Angular	Vue
It is a heavy framework suitable for large enterprise-scale applications.	Lightweight framework, suitable for small-scale applications
It has a steep learning curve.	Easy to work with for beginners not familiar with JavaScript concepts.

7.2 Suggestions

Based on previous discussions, it is evident that for any developer starting work on a project, the dilemma of choosing the suitable framework from a variety of options, each with its own set of features and structures, is always present. Hence, to decide on a framework based on a project's requirements and the developer's capabilities, the following can be kept in mind.

Suppose the project is one of your first projects, or the developer is relatively inexperienced. In that case, it might be preferable to opt for Vue.js or React, as they are much easier to grasp and start working with compared to Angular, which is more challenging to learn.

If the application to be built is a large, enterprise-scale application, then it is preferable to work with Angular as it is suited for such applications. In contrast, if the application is meant to be a small-scale application with no plans of scaling up, VueJS would be the preferred choice as it is lightweight and, at the same time, provides many of the features that Angular provides.

Lastly, a developer’s familiarity with TypeScript can prove advantageous, as using TypeScript helps debug code, resulting in cleaner, easier-to-understand code. Since

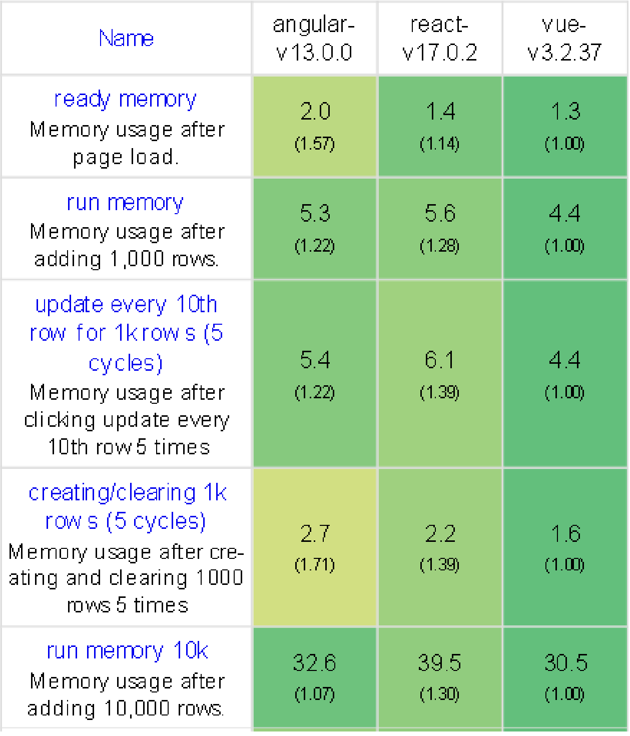
Angular is based on TypeScript, a developer familiar with TypeScript can also take advantage of the variety of functionality offered in the Angular framework. While both React and Vue also support TypeScript, only HTML and JavaScript knowledge are sufficient to get started with them.

7.3 Performance Comparison

The previous sections of this paper focused on comparing the different features and structures of the frameworks, while this section will elaborate on the quantitative performance comparison between the frameworks.

The tables below are taken from [14] in which a large, randomized table was created using various frameworks on Chrome 104 Browser, and different operations were performed on the table, the speed (in milliseconds) and memory usage (in MB) (with 95% confidence interval) for the frameworks.

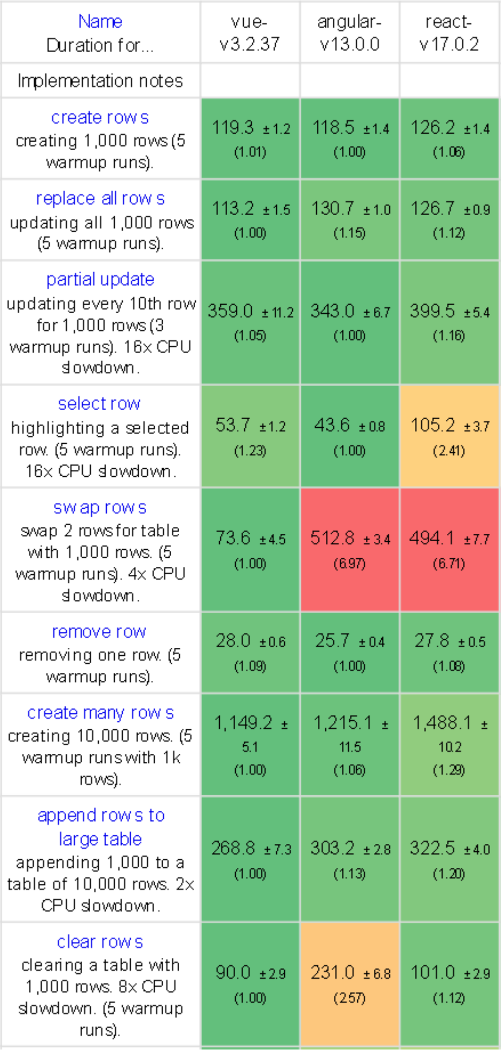
Table 6. Memory Usage for different frameworks [14]



Name	angular-v13.0.0	react-v17.0.2	vue-v3.2.37
ready memory Memory usage after page load.	2.0 (1.57)	1.4 (1.14)	1.3 (1.00)
run memory Memory usage after adding 1,000 rows.	5.3 (1.22)	5.6 (1.28)	4.4 (1.00)
update every 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	5.4 (1.22)	6.1 (1.39)	4.4 (1.00)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	2.7 (1.71)	2.2 (1.39)	1.6 (1.00)
run memory 10k Memory usage after adding 10,000 rows.	32.6 (1.07)	39.5 (1.30)	30.5 (1.00)

Table 6 above depicts the difference in memory usage of the three frameworks at various stages, while Table 7 depicts the speed of different operations being conducted on a table. As is evident from Table 7, React is slightly slower in selecting rows compared to Angular and Vue, whereas both Angular and React are considerably slower than Vue when it comes to swapping rows, and Angular is at a slight disadvantage compared to React and Vue, at clearing table. When it comes to memory usage, there is no major disparity among the three frameworks for this particular test, but the values in Table 6 and 7, vary with operating systems, browser, browser version and depending on the operations being tested.

Table 7. Speed of Operations done on large randomized table



Name Duration for...	vue-v3.2.37	angular-v13.0.0	react-v17.0.2
Implementation notes			
create rows creating 1,000 rows (5 warmup runs).	119.3 ±1.2 (1.01)	118.5 ±1.4 (1.00)	126.2 ±1.4 (1.06)
replace all rows updating all 1,000 rows (5 warmup runs).	113.2 ±1.5 (1.00)	130.7 ±1.0 (1.15)	126.7 ±0.9 (1.12)
partial update updating every 10th row for 1,000 rows (3 warmup runs). 16x CPU slowdown.	359.0 ±11.2 (1.05)	343.0 ±6.7 (1.00)	399.5 ±5.4 (1.16)
select row highlighting a selected row. (5 warmup runs). 16x CPU slowdown.	53.7 ±1.2 (1.23)	43.6 ±0.8 (1.00)	105.2 ±3.7 (2.41)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4x CPU slowdown.	73.6 ±4.5 (1.00)	512.8 ±3.4 (6.97)	494.1 ±7.7 (6.71)
remove row removing one row. (5 warmup runs).	28.0 ±0.6 (1.09)	25.7 ±0.4 (1.00)	27.8 ±0.5 (1.08)
create many rows creating 10,000 rows. (5 warmup runs with 1k rows).	1,149.2 ± 5.1 (1.00)	1,215.1 ± 11.5 (1.06)	1,488.1 ± 10.2 (1.29)
append rows to large table appending 1,000 to a table of 10,000 rows. 2x CPU slowdown.	268.8 ±7.3 (1.00)	303.2 ±2.8 (1.13)	322.5 ±4.0 (1.20)
clear rows clearing a table with 1,000 rows. 8x CPU slowdown. (5 warmup runs).	90.0 ±2.9 (1.00)	231.0 ±6.8 (2.57)	101.0 ±2.9 (1.12)

8 Conclusion

This paper has analyzed in depth the importance of frontend frameworks and the features provided by three frontend frameworks: React, Angular, and Vue. It then goes on to examine some of the large-scale applications built using these frameworks and understand the reason for their use. Then a qualitative comparison is drawn between the various frameworks based on various features that they offer, and guidelines are provided to developers on selecting a particular framework for projects. Lastly, a quantitative comparison is analyzed among the three frameworks.

References

1. "Jquery," [Online]. Available: https://jquery.com/.
2. "JetBrains," [Online]. Available: https://www.jetbrains.com/lp/devecosystem-2021/javascript/. [Accessed 29 10 2022].
3. ReactJS, "React – A JavaScript library for building user interfaces," [Online]. Available: https://reactjs.org/. [Accessed 29 10 2022].
4. "Introduction to the DOM - Web APIs - MDN Web Docs," [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction. [Accessed 29 10 2022].
5. "React Developer Tools," [Online]. Available: https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi/related?hl=en. [Accessed 31 October 2022].
6. Y. Xing, "Research and Analysis of the Front-end Frameworks and Libraries in E-Business," 2019.
7. "AngularJS," [Online]. Available: https://docs.angularjs.org/. [Accessed 31 October 2022].
8. "Angular: Introduction to Angular Docs," [Online]. Available: https://angular.io/docs. [Accessed 31 October 2022].
9. "Angular: Introduction to Angular Concepts," [Online]. Available: https://angular.io/guide/architecture. [Accessed 1 November 2022].
10. "Jasmine Documentation," [Online]. Available: https://jasmine.github.io/. [Accessed 1 November 2022].
11. "Vue.js," [Online]. Available: https://vuejs.org/. [Accessed 1 November 2022].
12. "Why we chose Vue.js | GitLab," GitLab, [Online]. Available: https://about.gitlab.com/blog/2016/10/20/why-we-chose-vue/. [Accessed 6 November 2022].
13. "NPM Trends," [Online]. Available: https://npmtrends.com/@angular/core-vs-react-vs-vue. [Accessed 6 November 2022]
14. "Interactive Results," [Online]. Available: https://krausest.github.io/js-framework-benchmark/2022/table_chrome_104_windows.html. [Accessed 6 November 2022].