

DAR Report

Hubba - Front-End Technology

Version 1.1

Prepared By: Development Hell

Class: CECS 491-04

Date: December 4, 2022

Github Repository:

<https://github.com/DevelopmentHellaHell/SeniorProject>

Team Leader

Kevin Dinh

Members

Garrett Tsumaki

Bryan Tran

Jett Sonoda

Tien Nguyen

Darius Koroni

Revision History

Version	Overview	Date
1.0	Initial DAR report	12/4/2022
1.1	Quantified attributes being compared and added further justification in conclusion	12/10/2022

Decision

The following options will be considered as the front-end framework or library used for the client's UI and UX development.

1. **Angular version 14**
 - a. TypeScript-based framework developed by Google
2. **React version 18.2.0**
 - a. JavaScript-based library developed by Meta
3. **Vue.js version 3.2.45**
 - a. JavaScript-based framework developed by Evan You and community members

Analysis

	Angular	React	Vue.js
Memory transfer required on startup (x 0.5)	282.8 kB 0.5	274.6 kB 1	196.5 kB 1.5
LTS and stability	Unsupported after 18 months 1	Legacy support for older features, maximum support of 6 years 2	Legacy support for older features, maximum support of 6 years 2
Document Object Model Responsiveness for Writing	45.2 ms for creating 1k rows 2	52.8 ms for creating 1k rows 1	47.4ms for creating 1k rows 2
Storing Values Across Views and Implicit Typing (x 1.5)	Must create own service to receive event 1.5	Native support with props, allows all types with type checking 4.5	Native support with props, must declare prop type explicitly 3
Windows/Mac Compatible	Windows and Mac 1	Windows and Mac 1	Windows and Mac 1
TOTAL	6	9.5	9.5

Recommendation

Memory transfer required on startup: The amount of memory required for the front-end application to load affects the initial responsiveness of the application. As a result, we hope to limit the maximum memory required for loading the front-end to 500 kB. Since every page will be using the chosen front-end, even a minor difference could have a noticeable difference for our user base. Vue is slightly faster than React, which in itself is slightly more responsive than Angular. All three options presented are within our accepted bounds of memory usage for startup.

LTS and stability: Using a framework which allows for long term support without having to worry about deprecated features is important to our team. Although larger companies might adapt well to new features replacing old ones, our team is small and wants to focus on the stability of our application as a priority. Angular updates far too frequently for our team's liking and both React and Vue have proven to have support for older features into their latest versions.

Document Object Model Responsiveness for Changes: Making changes to the webpage will occur when updating values in a view or loading new results on a page. React performed the most poorly of the three technologies presented, while Angular and Vue performed similarly. The difference is only minor but can add up once our application is fully fledged, which is important to keep in mind when regarding the scalability and future performance of our application.

Storing Values Across Views and Implicit Typing: As a single-page application most of the work for our product will deal with passing values across different views. Angular does not provide native support for transferring values from a component in one view to a component in another view. This means the developer will have to create the service to transfer values on their own. Vue allows the passing of values, but they must be explicitly typed before passing the values. React is the most flexible of all three, allowing all values to be passed as long as the values are not modified when treated as inputs to the new view.

Windows/Mac Compatible: Although we are limiting our use of technology for browsers to Chrome, we still hope to be accessible to two of the largest markets available for computing, MacOS and Windows. All three front-end solutions allow for compatibility for these large markets.

Conclusion: We recommend React. Although Vue puts up a good case as a competitor to React, React makes it easier for developers to store and transfer values across different components in views. This is an important feature for our single-page application and the ease that React provides for it outweighs the benefit of a slightly faster read and startup speed.

References

- Borrelli, P. (2021, November 10). Angular vs. react vs. vue.js: Comparing performance. LogRocket Blog. Retrieved December 10, 2022, from <https://blog.logrocket.com/angular-vs-react-vs-vue-js-comparing-performance/>
- Dziuba, Anna. Angular vs. React vs. Vue.js – Choosing a JavaScript Framework for Your Project in 2023, Relevant Software, 22 Nov. 2022, <https://relevant.software/blog/angular-vs-react-vs-vue-js-choosing-a-javascript-framework-for-your-project/>.
- Krause, S. (2021, October 26). Results for js web frameworks benchmark - official run. Interactive results. Retrieved December 10, 2022, from <https://rawgit.com/krausest/js-framework-benchmark/master/webdriver-ts-results/table.html>
- Neagoie, Andrei. "Tech Trends Showdown 🏆: React vs. Angular vs. Vue." Zero To Mastery, 20 Dec. 2021, <https://zerotomastery.io/blog/tech-trends-showdown-react-vs-angular-vs-vue/>.
- Raval, Nihar. "React vs Vue: Comparison of the Best JavaScript Frameworks at a Glance!" *React vs Vue*, Radixweb, 25 Aug. 2022, <https://radixweb.com/blog/react-vs-vue>.