React vs. its competitors: a Systematic Literature Review

Jakub Błażejowski

Introduction

According to sources like the Stack Overflow 2023 developer survey or the State of JavaScript survey, React remains the most popular library for web application front end creation, with its reported usage being almost double of that achieved by its main competitors — Angular and Vue. It is that much more surprising, considering the fact that Vue and other, newer frameworks, like Solid or Svelte, are steadily gaining developers' interest as stated by the same sources. Because of that, finding out why React maintains its position, is an important problem. One method that could potentially help to explain that, would be to compare the other libraries to React.

Background

React is a JavaScript library used to create the front end part of web application. It was developed by Facebook and its first version was published in 2013. Besides React, there are many other frameworks and libraries which serve this purpose, while sometimes also providing other functionalities. Among them, there are Vue, Angular, Preact, Ember, Svelte and Solid.

Aim and research questions

The purpose of this work was to find out the current state of knowledge on what comparisons had already been done, what were their results and how were they conducted. With that knowledge, it became possible to tell if the answer to the problem already exists and how to prepare an experiment that achieves that. The research was done in the form of a Systematic Literature Review to make sure that all available data on the topic has been analyzed.

RQ1

How do React's alternatives compare to it?

RQ2

How to compare front end libraries?

Method

To make sure that the Systematic Literature Review (SLR) provided complete and reproducible results, it was conducted according to the following protocol.

Acceptance criteria

The following criteria were used to preliminarily filter out the works which weren't relevant to the research questions.

Date range
: Years 2013–2022 — the start date marks the date React, which is the main focus of this work, was first published. Because of the ever-changing nature of technology, the newer articles proved to be more useful.

Language
: English — it was assumed that all important and impactful publications eventually get published in English.

Publication type
: Articles and conference papers — to make sure only the newest data makes it into the analysis.

Selection steps

During the selection phase, articles, which are supposed to represent the state of knowledge, got selected. The process was split into several phases. Every phase was concluded with a manual filtration, which, based on the title and the abstract, eliminated entries which were not related to the study.

Database-driven selection

The first step in the selection process was performed using a reference and citation database. The choice of the database was done with the help of the supervisor, who suggested Web of Science (WoS) as the database which would provide most useful results. For this step, a following query was used.

```
(TS=(comparison) OR TS=(effectiveness) OR TS=(efficiency)
    OR TS=(speed) OR TS=(performance)) AND
(TS=(Svelte) OR TS=(Solid) OR TS=(React)) AND
TS=(JavaScript) AND
WC=(Computer Science, Software Engineering) AND
PY=(2013-2022)
```

The query was executed on 2023-05-05 and it generated 29 results. Previous iterations, which did not include the frameworks or the programming language specification, returned thousands of results from fields unrelated to the study, so those restrictions were added. After the manual selection, only 5 articles remained.

Snowballing

This step was planned as a mean to generate more results in case the previous step did not provide satisfying results. It could have been repeated multiple times, where each iteration would have involved previously unvisited references and citing articles of papers from the previous phase. In the end, it got executed one time, generating 167 results (50 citing articles and 117 references), out of which 10 got selected (3 citing articles and 7 references).

Data extraction table

A data extraction table was a tool used to gather information, which could help answer the research questions, from the articles returned from the selection procedure. It comprised the following columns:

Tools used
JavaScript libraries compared in the article.
Characteristics
Characteristics that could be used to compare JavaScript libraries.
Methodology
The procedure and tools that were used in order to analyse the libraries.
Results
Results of the comparisons.
Validity threats
Anything that may make the article's results less viable.

Results analysis

In this section, an attempt was made to answer the research questions, based on the results of the Systematic Literature Review.

RQ1: How do React's alternatives compare to it?

There are libraries which are in many ways better than React, but it still is a solid option and probably will be actively used in the future.

Performance-wise, applications designed with React are worse than those made with, for example, Svelte or Vue, but the difference is mostly negligible, especially in smaller applications. Frameworks like Angular or Blazor perform significantly worse, which was one of the reasons for React's broad adoption, but they are still worth considering because of the bigger set of out-of-the-box functionalities like testing support or routing. For React, it is easy to learn and find answers to questions, because of a big and active community. It is also easy to find experienced developers. This isn't at all the case for Angular and Blazor, with Vue being the closest competitor and Svelte and other newer frameworks having an active community, but a significantly smaller number of experienced developers.

RQ2: How to compare front end libraries?

Library comparison should ideally consist of as many characteristics as possible, as long as the characteristics are relevant to every compared library. The areas to prioritize are: web application performance, the ecosystem and the available documentation.

Performance is by far the most often considered characteristic. It is usually done with Google Lighthouse tool, but it can also be done manually, which can sometimes prove beneficial. Google's tool, while very useful, is not able to generate some metrics, such as the Script Execution Time. (full render cycle without the rendering part). In component-based libraries, it can also be worthwhile to group the measurements by, for example: operation type (create, update, delete), component type (static, dynamic), part of the component tree that is operated on (tree root, tree element) or the size of the component tree. Sometimes the functionalities provided by the library, or availability of complementary libraries, can be an important factor that improves both the developer's and end-user's experience. Depending on the size of the application and time available, the learning curve becomes relevant alongside of availability of learning resources and community quality. Other properties, which could be useful in case the compared libraries prove to be very similar, are: build size, build time, rebuild time and startup time.

Validity threats

The following factors could endanger the validity of this study's findings:

Data completeness

  There is a real possibility that important articles or other sources exist, that weren't included in this analysis. What mitigates a possible impact of this threat is that the articles, which were included in the review, presented consistent results.

Methodology difference

  Each of the analyzed articles defined its own methodology and focused on slightly different characteristics. The fact that the results were mostly similar counteracts this threat, but the comparisons should be treated separately.

Technology evolution

  Because the libraries get updated over the years, results of earlier research can quickly become inapplicable. Most analyzed papers were from recent years, which makes this a small risk.

Unknown framework versions

  Almost no papers specified which framework versions were tested, which could make the results unreliable. An assumption was made that researchers probably used the newest versions available at the time, which, combined with the fact that the papers were relatively new, mitigates this threat.

Vague methodologies

  Because the results of this work relies heavily on other people's work, when the methodologies or sources are not well-defined it puts their validity under question, which in turn propagates to this paper. Most of the encountered articles had their methodologies thoroughly described, which partially eliminates this threat.

Conclusions

This work's aim was to try to find out why React is the most-used JavaScript front end library. It was found out that one of those reasons is that React does really well in comparison with other libraries and while there are alternatives which could one day overthrow it, it probably won't happen in the nearest future. After collecting the results from many articles, Vue seemed to be the biggest competitor, that's capable of doing so. However, because all the analyzed comparisons used different methodologies and the libraries are constantly evolving, this prediction might prove to be incorrect. What was also gathered, was information on conducting quality library comparisons. Future work could focus on applying the findings to further comparisons, or potentially devising a comparison method that would eliminate the time factor.

References

Delcev, Sanja, and Drazen Draskovic. 2018. "Modern Javascript Frameworks: A Survey Study." In *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)*, 106–9. IEEE.

Helenius, Jouni. 2022. "Web Application Upgrade to New Modern Technology: Case Tarmo Volunteer Enrolment Service."

Ilievska, Frosina, and Sasho Gramatikov. 2022. "Analysis and Comparative Evaluation of Front-End Technologies for Web Application Development." In.

Ivanova, Slavina, and Georgi Georgiev. 2019. "Using Modern Web Frameworks When Developing an Education Application: A Practical Approach." In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1485–91. IEEE.

Mlynarski, Adam, and Karolina Nurzynska. 2017. "Comparative Analysis of JavaScript and Its Extensions for Web Application Optimization." In *Beyond Databases, Architectures and Structures. Towards Efficient Solutions for Data Analysis and Knowledge Representation: 13th International Conference, BDAS 2017, Ustroń, Poland, May 30-June 2, 2017, Proceedings 13*, 539–50. Springer.

Nakajima, Nozomi, Shinsuke Matsumoto, and Shinji Kusumoto. 2019. "Jact: A Playground Tool for Comparison of Javascript Frameworks." In *2019 26th Asia-Pacific Software Engineering Conference (APSEC)*, 474–81. IEEE.

Novac, Ovidiu Constantin, Damaris Emilia Madar, Cornelia Mihaela Novac, Gyöngyi Bujdosó, Mihai Oproescu, and Teofil Gal. 2021. "Comparative Study of Some Applications Made in the Angular and Vue. Js Frameworks." In *2021 16th International Conference on Engineering of Modern Electric Systems (EMES)*, 1–4. IEEE.

NV, Diniz-Junior Raimundo, Caio César L Figueiredoy, Gilson De S Russo, Marcos Roberto G Bahiense-Junior, Arbex Mateus VL, Lanier M Dos Santos, Raimundo F Da Rocha, Renan R Bezerra, and Felipe T Giuntini. 2022. "Evaluating the Performance of Web Rendering Technologies Based on JavaScript: Angular, React, and Vue." In *2022 XVLIII Latin American Computer Conference (CLEI)*, 1–9. IEEE.

Ollila, Risto et al. 2021. "A Performance Comparison of Rendering Strategies in Open Source Web Frontend Frameworks."

Ollila, Risto, Niko Mäkitalo, and Tommi Mikkonen. 2022. “Modern Web Frameworks: A Comparison of Rendering Performance.” *Journal of Web Engineering*.

Ramos, Miguel, Marco Tulio Valente, and Ricardo Terra. 2017. “AngularJS Performance: A Survey Study.” *IEEE Software* 35 (2): 72–79.

Siahaan, Mangapul, and Vincent Octarian Vianto. 2022. “Comparative Analysis Study of Front-End JavaScript Frameworks Performance Using Lighthouse Tool.” *Jurnal Mantik* 6 (3): 2462–68.

Tripon, Teodor-Dorin, Gianina Adela Gabor, and Elisa Valentina Moisi. 2021. “Angular and Svelte Frameworks: A Comparative Analysis.” In *2021 16th International Conference on Engineering of Modern Electric Systems (EMES)*, 1–4. IEEE.

Xing, YongKang, JiaPeng Huang, and YongYao Lai. 2019. “Research and Analysis of the Front-End Frameworks and Libraries in e-Business Development.” In *Proceedings of the 2019 11th International Conference on Computer and Automation Engineering*, 68–72.