




Rendering performance comparison of react, vue, next, and nuxt

Mangapul Siahaan¹, Ryan Kenidy²

^{1,2}Computer Science Faculty ,Universitas Internasional Batam, Indonesia

ARTICLE INFO	ABSTRACT
<p><i>Article history:</i> Received Oct 14, 2023 Revised Oct 18, 2023 Accepted Oct 24, 2023</p>	<p>This article presents a comprehensive analysis of the rendering performance of four popular JavaScript frameworks: React, Vue, Next.js, and Nuxt.js. In an increasingly competitive web development landscape, performance optimization is paramount. By exploring various benchmarks, profiling tools, and real-world use cases, this study delves into how these frameworks handle rendering, component updates, and server-side rendering (SSR). The results offer developers valuable insights into choosing the most suitable framework for their projects, with a focus on delivering optimal user experiences and efficient web applications. The results conducted with WebPageTest, PageSpeedInsight, and GTMetrix show that Vue is the fastest in the average result tested with Webpage Test with 0.9s compared to Nuxt with 2.2s. For PageSpeed Insight, React outperforms other frameworks with 2.8s compared to the highest which is Next with 3.9s. In GTMetrix test, Next turns out to be the fastest framework with 0.8s, while React is the slowest with 1.4s.</p>
<p><i>Keywords:</i> Framework; GTMetrix; PageSpeed Insight; Performance; Rendering; Webpage Test.</p>	<p><i>This is an open access article under the CC BY-NC license.</i></p> 

Corresponding Author:

Mangapul Siahaan
Computer Science Faculty,
Universitas Internasional Batam,
Baloi-Sei Ladi, Jl. Gajah Mada, Kec. Sekupang, Kota Batam, Kepulauan Riau 29426 Indonesia
Email: mangapul.siahaan@uib.ac.id

1. INTRODUCTION

Website development nowadays is basically divided into 2 teams. Those 2 teams are front-end, back-end (Axtell et al., 2020). Front-end is usually known as client-side and is mainly all about the design of a web. According to article of (Dutonde et al., 2022), The technologies used for developing front-end are HTML, CSS, and Javascript. Meanwhile, back-end development also known as server-side, means developing the database and sending those data to front-end (Madurapperuma et al., 2022). The datas are then converted into beautiful ui with the help of front-end. Without back-end, the interactivity by the user will be limited because they cannot maintain the data inputted into web. This type of static website only allow user to view data generated from the front-end developer. The dynamic website however, allow user to manipulate the data. Nowadays, with the help of Node.js, development of server-side is made possible and easier than before (Herron, 2020)

Research done by (Cincović & Punt, 2020) titled “Comparison: Angular vs. React vs. Vue. Which framework is the best choice?”. This research aims to discover the

difference of popular javascript frameworks such as Angular, React, and dan Vue. The measured parameters are popularity, performance, community support, learning curve, migrations, and flexibility. From the research, it shows that React and Vue is more lightweight and fast compared to Angular.

(Ollila et al., 2022) do research with title of “Modern Web Frameworks: A Comparison of Rendering Performance.” In the article, author conducted rendering strategy between Angular, React, Vue, Svelte, and Blazor. The results show there are significant difference depending on different rendering strategy used. The difference is found on several similar tasks that is run by the framework. The best rendering strategy is the one with less input size dan optimized with compile-time optimization and reactive programming models.

Another study was also conducted by (Boczkowski & Pańczyk, 2020) with the title "Comparison of the performance of tools for creating a SPA application interface - React and Vue.js." The study involved the development of two applications with similar functions using React and Vue. The results indicated differences between the two in displaying complex interfaces. React performed better in cases involving smaller-sized components, while Vue achieved better results in displays with larger elements.

Another study titled "Comparative analysis of React, Next, and Gatsby programming frameworks for creating SPA applications" was also conducted by (Świątkowski & Ścibior, 2022). The study aimed to determine which technology is superior for building the UI portion of an application. The research was carried out using React, Next, and Gatsby to build the same content. To evaluate performance differences, the researchers used web development tools and the React Library. Based on the results of the study, React was found to be the most efficient in displaying pages with a large amount of data.

The research conducted by (Vianto & Siahaan, 2022) measures the performance of websites built with React, Angular, SolidJS, Vue, and Svelte. The researchers used the waterfall method to build the websites that would be tested. The parameters measured were (i) First Contentful Paint, (ii) Largest Contentful Paint, (iii) Time to Interactive, (iv) Speed Index, (v) Total Blocking Time, (vi) Cumulative Layout Shift. The website built for this research is a weather website using the OpenWeather API. The tool used for the research is Google Lighthouse. Vue is the framework with the best performance in this study.

Based on the previous research, the researchers conducted a performance comparison of front-end frameworks with both client-side rendering (CSR) and server-side rendering (SSR). The frameworks compared included React, Vue, Next, and Nuxt. The research is about building a website that displayed a list of movies based on data retrieved from the TMdB API. Unlike the previous research that was tested with Google Lighthouse or web development tools, the researchers used three testing tools: WebPage Test, PageSpeed Insight, and GTMetrix (Hossain et al., 2021). The parameters measured included (i) Load Time, (ii) First Byte, (iii) Start Render, (iv) Speed Index, (v) Cumulative Layout Shift, (vi) First Contentful Paint, (vii) Largest Contentful Paint, (viii) Total Blocking Time, and (ix) Time To Interactive. The methodology employed was ADDIE. Thus, this research aids to find the rendering performance of React, Vue, Next, and Nuxt. The result is intended to benefit web developers particularly as an insight into the selection and learning of Javascript Front-end and Fullstack Framework.

In summary, our study contributes to the existing body of knowledge by offering a comprehensive analysis of rendering performance across these popular frameworks, with a specific focus on CSR and SSR, thus addressing several gaps in previous research. Moreover, it adds practical implications for web developers seeking to optimize their web development projects.

2. RESEARCH METHOD

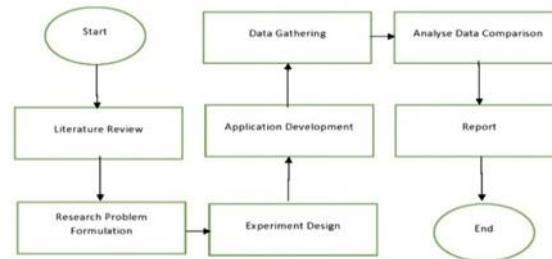


Figure 1. Research Flow

This research started with the process of literature review. Researchers will read several literatures and gain an insight of how to begin the experiment. Research problem formulation is the process that come after literature review. It is done by specifying problems in society that is concerning enough be researched. After having the problem specified, researcher will begin to design the experiment scope. With the experiment scope, researcher will be able to know type of data to measure to compare between the advantages and disadvantages of React.js, Next.js, Nuxt.js and Vue.js. Then, researcher will start to develop the application. This process includes designing, prototyping, and programming with 4 different frameworks to build a movie list web application. When all of the web applications have been developed, researchers will continue with the process of data gathering. The datas gathered are then analysed and after the step of analysis is done. Researchers will write a report for the experiment results.



Figure 2. ADDIE Model (Aklani & Jacky, 2022)

The model used in this research is ADDIE model based on the reference (Aklani & Jacky, 2022). This model helps researcher to develop the application required for experiment. There are 5 steps in ADDIE model (Spatiotti et al., 2022), the step for this research is as follows:

a. Analysis

In the analysis phase, researchers will identify and select the performance metrics to be used in the comparative analysis of rendering performance. Researchers will also gather the movie data provided by TMDb API.

b. Design

In this process, researchers will design required UI / UX design using Figma.

c. Develop

Develop is the process where researchers start to build the program needed for experiment.

d. Implementation

After development process, researchers will deploy and test the program in this implementation process.

e. Evaluation

Finally, researcher begin to collect the data and compare them with metrics stated in research method section.

3. RESULTS AND DISCUSSIONS

3.1 Analysis

There are 3 tools used for testing each rendering performance metric. (1) Webpage Test is tool used for Assessing the website's speed, user-friendliness, and robustness (Abdulla et al., 2023). (2) GTMetrix, an automated software testing and provides suggestions to enhance website management quality (Dawis & Setiawan, 2022). (3) PageSpeed Insight, a performance testing tool, evaluates various aspects of a website's performance to provide valuable insights and recommendations for optimization (Muna et al., 2023).

Table 1. Performance Metrics

Metrics	Description	Tools		
		WebPage Test	PageSpeed Insight	GTMetrix
Load Time	Measure the website loading time for a fully rendered website (De Munk & Malavolta, 2021)	✓		✓
First Byte	Measure time taken to receive the first byte (Eslamdoust, 2022)	✓		✓
Start Render	The first content being rendered that can be seen by user even if it's background (Król, 2018)	✓		✓
Speed Index	Measure the speed for the content of the page to be filled (De Munk et al., 2022)	✓	✓	✓
Cumulative Layout Shift	Measure the load time when website element shift until it finishes (Mokhtari et al., n.d., 2022)	✓	✓	✓
First Contentful Paint	Time taken for the first element to be rendered (Tengriano et al., 2022)	✓	✓	✓
Largest Contentful Paint	Time taken for the largest element to be rendered (Justus & Jaeger, 2022)	✓	✓	✓
Total Blocking Time	Amount of time used for blocking the element to be rendered (Hossain et al., 2021)	✓	✓	✓
Time to Interactive	Total time for a website to be fully interactive (Namoun et al., 2020)	✓		✓

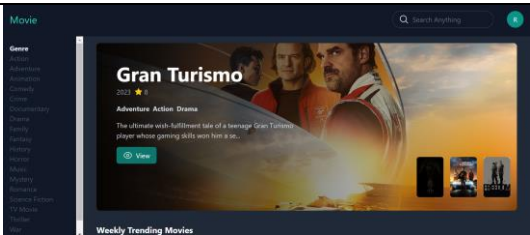



Performance testing is a way to organize, carry out, monitor, and assess a system's outcomes under specific circumstances to identify the predicted behaviors of determining software under these circumstances (Martins et al., 2020). Based on the meaning above, performance testing is conducted considering the importance of performance to user. By having a better performance, a system will give user a better

experience (Hamzah et al., 2022). The metrics tested in this research can be viewed at table 1 (Hossain et al., 2021).

3.2 Design

During this phase, researchers design the UI using Figma to be converted by code. The design of each framework can be seen on Table 2.

Table 2. Movie Web by Framework

Framework	Link	UI
React.js	https://ryan-react-movie-app.netlify.app/	
Vue.js	https://ryan-vue-movie-app.netlify.app/	
Next.js	https://ryan-next-movie-app.netlify.app/	
Nuxt.js	https://ryan-nuxt-movie-app.netlify.app/	

3.3 Develop

After finish designing, the next step is to convert the design with code. This process involves coding with syntax by each framework.

3.4 Implementation

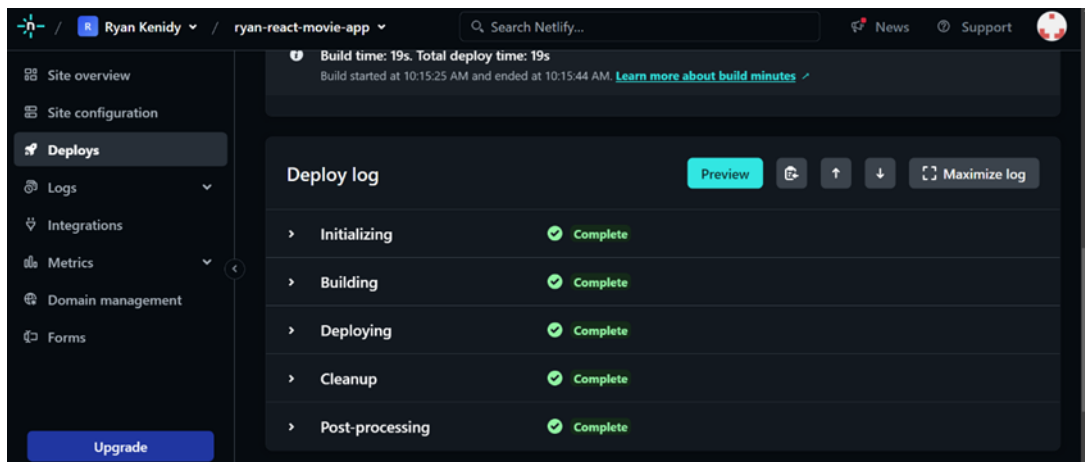


Figure 3. Netlify Deployment and Testing

The final coding version is deployed and tested in this phase. The codes were published to github code repository. All of the webs are deployed to netlify hosting platform and the link of each framework is displayed at table 2.

3.5 Evaluation

Based on the websites above, the rendering performance test was conducted and the results are counted in the second unit and can be viewed in table 3 – 6 below.

Table 3. WebPage Test Rendering Performance Result

Metrics	React	Vue	Next	Nuxt
LT	0.4 s	0.3 s	1.3 s	1.5 s
FB	0.1 s	0.1 s	0.7 s	0.7 s
SR	2.6 s	0.3 s	1 s	0.9 s
SI	3.2 s	0.3 s	4 s	5 s
CLS	0 s	0 s	0.4 s	0 s
FCP	2.6 s	0.3 s	1.2 s	0.8 s
LCP	6.4 s	7.1 s	5 s	7.3 s
TBT	0 s	0 s	0.3 s	0.1 s
TTI	2.5 s	0.5 s	4 s	4 s
Average	1.9 s	0.9 s	1.5 s	2.2 s

The result tested with WebPage Test shows all required metrics. Based on the result table, we can conclude that Vue is the fastest to fully load the website with the lowest load time of 0.3s compared to Nuxt with 1.5s. Vue also ends up as the best for speed index means the speed of the content to be filled is the fastest with 0.3s. While Nuxt is again the slowest of all frameworks for speed index. For average of all metrics, vue has the fastest rendering performance with 0.9s compared to Nuxt with 2.2s

Table 4. PageSpeed Insight Rendering Performance Result

Metrics	React	Vue	Next	Nuxt
LT	-	-	-	-
FB	-	-	-	-
SR	-	-	-	-
SI	2.2 s	2.1 s	5.4 s	4.8 s
CLS	0.1 s	0.1 s	0.7 s	0 s
FCP	1.5 s	1.5 s	1.7 s	1.5 s
LCP	11.3 s	13.7 s	11.8 s	7.7 s
TBT	0 s	0 s	0.3 s	1.6 s
TTI	-	-	-	-
Average	2.8 s	3.4 s	3.9 s	3.1 s

Using Pagespeed insight as a testing tool, the result does not complete all metrics. For speed index metrics, react is the fastest with 2.2 compared to Next with 5.4. The result is different compared to webpage test result, this can be highly due to network latency and tools testing algorithm. In average, react performs the best with 2.8s while Next is the worst with 3.9s.

Table 5. GTMetrix Rendering Performance Result

Metrics	React	Vue	Next	Nuxt
LT	0.3 s	0.3 s	0.4 s	0.3 s
FB	0.2 s	0.1 s	0.1 s	0.1 s
SR	1.9 s	0.3 s	0.3 s	0.2 s
SI	2.2 s	2.9 s	1.7 s	2.2 s
CLS	0 s	0 s	0 s	0 s
FCP	1.9 s	0.3 s	0.3 s	0.2 s
LCP	2.6 s	3.1 s	2.1 s	3.3 s
TBT	0.5 s	0.4 s	0.6 s	0.6 s
TTI	3 s	3.2 s	2.1 s	3.5 s
Average	1.4 s	1.1 s	0.8 s	1.1 s

In GTMetrix test results, all metrics are complete. From total blocking time, Nuxt and Next have higher value as fullstack framework with server-side rendering compared to React and Vue which means there are time when it is blocked before rendering. However, even with total blocking time, both Next and Nuxt is faster to render the first byte compared to their front-end only framework (React and Vue). Other metrics also show that they are faster than React and Vue. This can be viewed in the average with Next and Next with 0.8 and 1.1 compared to React and Vue with 1.4 and 1.1

Table 6. Average of Webpage Test, PageSpeed and GTMetrix

Metrics	React	Vue	Next	Nuxt
LT	0.3 s	0.3 s	0.8 s	0.9 s
FB	0.1 s	0.1 s	0.4 s	0.4 s
SR	2.2 s	0.3 s	0.6 s	0.5 s
SI	2.5 s	1.7 s	3.7 s	4 s
CLS	0 s	0 s	0 s	0 s
FCP	2 s	0.7 s	1 s	0.8 s
LCP	6.7 s	7.9 s	6.3 s	6.1 s
TBT	0.1 s	0.1 s	0.4 s	0.7 s
TTI	2.7 s	1.8 s	3 s	3.7 s
Average	1.8 s	1.4 s	1.9 s	1.9 s

Based on average of all WebPage Test, PageSpeed Insight, and GTMetrix, Vue is still the fastest framework even compared to its full stack framework Nuxt. Next and Nuxt are slower compared to a fully front-end framework. However, Nuxt and Next is faster in rendering the first and largest component compared to React and Vue.

3.6 Discussion

The performance test results showcase the strengths and weaknesses of React, Vue, Next, and Nuxt frameworks. In terms of WebPage Test metrics, Vue and React stand out as top performers with rapid Load Times of 0.3s and 0.4s, respectively, highlighting their ability to swiftly load web content. Vue's and React's quick First Byte response times of 0.1s further demonstrate their efficiency in initiating data transmission. Vue excels in Start Render and Speed Index, indicating its prowess in rapidly displaying content and ensuring a responsive user experience. In contrast, Next and Nuxt exhibit slightly longer times in these aspects, which may affect the user experience, especially in scenarios requiring real-time data rendering. Vue and React also impress with minimal Cumulative Layout Shift (CLS) values, providing a visually stable user experience.

PageSpeed Insight metrics align with the findings from WebPage Test, where Vue and React maintain their strong performance with low Speed Index values, indicative of a smooth user experience. Cumulative Layout Shift (CLS) metrics in PageSpeed Insight favor Vue and React, ensuring a visually stable user experience. Vue's and React's swift content delivery is evident in First Contentful Paint (FCP) and Largest Contentful Paint (LCP) metrics. Total Blocking Time (TBT) metrics reveal Vue and React as top performers in minimizing unresponsiveness, and Time To Interactive (TTI) metrics position them as leaders in providing quick interactivity.

GTMetrix metrics echo the trends observed in previous tests, highlighting Vue and React as top performers in Load Time, Start Render, and Speed Index, emphasizing their ability to provide a swift and responsive user experience. Cumulative Layout Shift (CLS) values continue to favor Vue and React, while First Contentful Paint (FCP) and Largest Contentful Paint (LCP) metrics underscore Vue's and React's proficiency in rapidly displaying content. Total Blocking Time (TBT) metrics highlight Vue's and React's strengths in minimizing unresponsiveness, and Time To Interactive (TTI) metrics again position them as leaders in providing quick interactivity. The average values across the GTMetrix metrics reaffirm Vue and React as solid choices for web development projects.

In summary, Vue and React consistently emerge as strong performers across various performance metrics, including load times, content rendering, and interactivity. Their efficient performance enhances user satisfaction and underscores the importance of making informed framework choices tailored to the unique needs of a web development project. While Next and Nuxt provide reasonable performance, Vue and React are particularly well-suited for projects that demand speed and responsiveness, ensuring a smoother user experience and ultimately impacting the success of web applications.

4. CONCLUSION

Performance definitely lead to better user satisfaction. One way of increasing performance can be started from choosing the better framework for performance. Based on interviews, Vue is faster than other frameworks. The result by doing this rendering performance comparison with WebPage Test, GTMetrix, and PageSpeed Insight, researcher find that Vue is the fastest in the average result tested with Webpage Test with 0.9s compared to Nuxt with 2.2s. For PageSpeed Insight, React outperforms other framework with 2.8s compared to the highest which is Next with 3.9s. In GTMetrix test, Next turns out to be the fastest framework with 0.8s, while React is the slowest with 1.4s. However, we recognize limitations: a narrow focus on select frameworks, a single use case, variability in testing tools, and the neglect of environmental factors. To address these, future research should explore diverse frameworks, dynamic applications, and reliable testing methods. Additionally, user-centric studies can offer practical insights. Such efforts would provide a more holistic understanding of web development framework performance, benefiting both developers and end-users in the dynamic field of web technology.

REFERENCES

- Abdulla, R., Faraj, H. A., Mohammed, K. O., & Younis, M. M. (2023). Usability Evaluation of the Top 10 Universities in Iraq Using Heuristic Methods. *Al-Mustansiriyah Journal of Science*, 34(2), 50–59. <https://doi.org/10.23851/MJS.V34I2.1234>
- Aklani, S. A., & Jacky, J. (2022). Perancangan Dan Implementasi Aplikasi Hotel Di Smk Harmoni Batam Menggunakan Framework Addie. *National Conference for Community Service Project (NaCosPro)*, 4(1), 790–794. <https://doi.org/10.37253/NACOSPRO.V4I1.7038>
- Axtell, N., Bacon, L., & Windall, G. (2020). COMPACT Web Design Approach: A Methodology and Modelling Technique for. *Neil-Axtell.Wikidot.Com*. <http://neil-axtell.wikidot.com/local--files/publications/COMPACTPaperSQMJan2007Final.pdf>
- Boczkowski, K., & Pańczyk, B. (2020). Comparison of the performance of tools for creating a SPA

- application interface-React and Vue. js. *Journal Computer Sciences Insititute*, 14, 73–77. <https://ph.pollub.pl/index.php/jcsi/article/view/1579>
- Cincović, J., & Punt, M. (2020). Comparison: Angular vs. React vs. Vue. Which framework is the best choice? *Universidad de Belgrade*. <http://www.eventiotic.com/eventiotic/files/Papers/URL/50173409-699e-4b17-8edb-9764ecc53160.pdf>
- Dawis, A. M., & Setiawan, I. (2022). Evaluation of The Website ‘Aisiyah Surakarta of University Performance Based on Search Engine Optimization Using Automated Software Testing Gtmetrix. *International Journal of Computer and Information System (IJCIS)*, 3(1), 17–20. <https://doi.org/10.29040/IJCIS.V3I1.56>
- De Munk, O., & Malavolta, I. (2021). Measurement-based Experiments on the Mobile Web: A Systematic Mapping Study. *ACM International Conference Proceeding Series*, 191–200. <https://doi.org/10.1145/3463274.3463318>
- De Munk, O., Scoccia, G. L., & Malavolta, I. (2022). The State of the Art In Measurement-based Experiments on the Mobile Web. *Information and Software Technology*, 149, 106944. <https://doi.org/10.1016/J.INFSOF.2022.106944>
- Dutonde, P. D., Mamidwar, S. S., Korvate, M. S., Bafna, S., & Shirbhate, P. D. D. (2022). Website Developemnt Technologies: A Review. *International Journal for Research in Applied Science and Engineering Technology*, 10(1), 359–366. <https://doi.org/10.22214/ijraset.2022.39839>
- Eslamdoust, A. (2022). An overview of search engine optimization techniques. *Future Generation of Communication and Internet of Things*, 1(4), 52–57. https://fgciot.semnan.iau.ir/article_699935.html
- Hamzah, M. L., Rahmadhani, R. F., & Purwati, A. A. (2022). An Integration of Webqual 4.0, Importance Performance Analysis and Customer Satisfaction Index on E-Campus. *Journal of System and Management Sciences*, 12(3), 25–50. <http://www.aasmr.org/jsms/Vol12/JSMS June 2022/Vol.12.No.03.02.pdf>
- Herron, D. (2020). *Node. js Web Development Fifth Edition* (5th ed.). https://books.google.com/books?hl=en&lr=&id=4_30DwAAQBAJ&oi=fnd&pg=PP2&dq=node.js+%2B+backend+development&ots=D0Owe4go-q&sig=2IRO57I8Ic6f74uJIBHAEpK_FtI
- Hossain, M. T., Hassan, R., Amjad, M., & Rahman, M. A. (2021). Web Performance Analysis: An Empirical Analysis of E-Commerce Sites in Bangladesh. *International Journal of Information Engineering and Electronic Business*, 13(4), 47–54. https://www.researchgate.net/profile/Md-Abdur-Rahman-7/publication/354384505_Web_Performance_Analysis_An_Empirical_Analysis_of_E-Commerce_Sites_in_Bangladesh/links/616042a4e7993f536ca681ef/Web-Performance-Analysis-An-Empirical-Analysis-of-E-Commerce-Sites
- Justus, W., & Jaeger, B. (2022). A Brief Overview on HTTP. *Network*, 59. https://doi.org/10.2313/NET-2022-11-1_11
- Król, K. (2018). Comparative analysis of the performance of selected raster map viewers. *Geomatics, Landmanagement and Landscape*, 2, 23–32. <https://bibliotekanauki.pl/articles/100508.pdf>
- Madurapperuma, I., Shafana, M., & Sabani, M. (2022). *State-of-art frameworks for Front-end and Back-end Web Development*. <http://ir.lib.seu.ac.lk/handle/123456789/6339>
- Martins, G. L., Silveira, M. B. Da, Rodrigues, E., & Basso, F. P. (2020). Systematic Literature Review on Web Performance Testing. *Sol.Sbc.Org.Br*. <https://sol.sbc.org.br/index.php/eres/article/view/13739>
- Mokhtari, H., Saberi, M. K., Amiri, M. R., Vakilmofrad, H., & Moradi, Z. (2022). Evaluating the Speed and Performance of the Websites of Hospitals and Specialty and Super-specialty Clinics of Hamadan University of Medical Sciences by GTMetrix. *Informology*, 1(1), 57–66. <http://eprints.rclis.org/43586/>
- Muna, S. S., Nurdin, N., & Taufiq, T. (2023). Comparative Analysis of State Universities on Website Performance in Aceh Using the PIECES Method. *Journal of Informatics and Telecommunication Engineering*, 7(1), 71–83. <https://ojs.uma.ac.id/index.php/jite/article/view/9167>
- Namoun, A., Alshantqi, A., Chamudi, E., & Rahmon, M. A. (2020). Web Design Scraping: Enabling Factors, Opportunities and Research Directions. *12th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 104–109. <https://doi.org/10.1109/ICITEE49829.2020.9271770>
- Ollila, R., Mäkitalo, N., & Mikkonen, T. (2022). Modern Web Frameworks: A Comparison of Rendering Performance. *Journal of Web Engineering*, 21(3), 789–814. <https://doi.org/10.13052/jwe15>
- Spatioti, A. G., Kazanidis, I., & Pange, J. (2022). A Comparative study of the ADDIE Instructional

- Design Model in Distance Education. *Information*, 13(9), 402.
<https://doi.org/10.3390/info13090402>
- Świątkowski, A., & Ścibior, K. (2022). Comparative analysis of React, Next and Gatsby programming frameworks for creating SPA applications. *Ph.Pollub.PIA Świątkowski, K Ścibior Journal of Computer Sciences Institute*, 2022•*ph.Pollub.Pl*, 24, 224–227.
<https://ph.pollub.pl/index.php/jcsi/article/view/2972>
- Tengriano, H. A., Yunus, A., & Sudirman. (2022). Analisis Performa Website AyoMulai Menggunakan GTMetrix dan Page Speed Insights. *Jurnal Kharisma Tech*, 17(2), 199–213.
<https://jurnal.kharisma.ac.id/kharismatech/article/view/347>
- Vianto, V. O., & Siahaan, M. (2022). Comparative Analysis Study of Front-End JavaScript Frameworks Performance Using Lighthouse Tool. *Jurnal Mantik*, 6(3), 3462–3468.
<http://www.iocscience.org/ejournal/index.php/mantik/article/view/3131>