OPEN ACCESS

Key Words
Front-end framework, website performance, usability, user experience

Corresponding Author
Hubert Daryanto Cen,
Department of Bina Nusantara, School Of Computer Science, University Jakarta Barat, Indonesia
hubert.cen@binus.ac.id

Received: 10 May 2024
Accepted: 29 June 2024
Published: 2 July 2024

Citation: Hubert Daryanto Cen and Pualam Dipa Nusantara, 2024. Enhancing User Interface Comprehensive Evaluation: Front-End Development Frameworks and Best Practices. Asian J. Inf. Technol., 22: 1-10, doi: 10.36478/makajit.2024.1.1.10

Enhancing User Interface Comprehensive Evaluation: Front-End Development Frameworks and Best Practices

[1]Hubert Daryanto Cen and [2]Pualam Dipa Nusantara
[1]Department of Bina Nusantara, School Of Computer Science, University Jakarta Barat, Indonesia
[2]Department of Bina Nusantara, University Jakarta Barat, School Of Computer Science, Indonesia

Abstract

This research compares the performance of Angular, Vue React front-end frameworks in developing a weather application, evaluating metrics such as load times, rendering efficiency overall responsiveness. Quantitative analysis, including metrics like Time to First Byte (TTFB), First Contentful Paint (FCP) Start Render, is complemented by qualitative insights from developer usability surveys. The findings indicate that Angular delivers the best user experience, Vue excels in performance and efficiency React provides the highest satisfaction levels. Angular demonstrates the strongest Pearson correlation between user experience and satisfaction (0.487**) and records the fastest First Contentful Paint (FCP) at 0.870 seconds. Vue shows the strongest correlation between user experience and performance and efficiency (0.446**) and achieves the fastest Time to First Byte (TTFB) at 0.179 seconds. React exhibits the strongest correlation between performance and efficiency with satisfaction (0.304**) and reports the fastest Start Render metric at 0.800 seconds. This study underscores the importance of considering diverse frameworks, metrics deployment environments to refine the understanding of front-end framework performance.

INTRODUCTION

These days, web apps are an integral part of everyone's life. Without question, apps are sweeping over the business world. Mobile and innovative technologies have had a huge impact on client behavior. Companies are eagerly looking forward to creating an online presence as the worldwide influence of mobility keeps growing and changing industries[1]. Front-end development and user interface (UI) are closely associated with one another, as the front-end is part of the development process that concerns the display and direct interaction with the user. The user interface is essential to any interaction between the user and the system. As a result, the interface should be designed to feel comfortable, understandable simple to utilize. Designing a consistent user interface is considered essential, as it affects the user's degree of comfort in managing the system[2].The user interface is acknowledged as a crucial component in software projects, with approximately 48% of the project's effort allocated to designing and implementing this essential element[3].

To open up new technical paths worldwide, developers and researchers must consider how to optimize web front-end development technology. The developing technology also meets the needs of consumers for a more seamless internet experience and surfing capabilities. Developers should put a lot of effort into optimizing the web front-end technologies that are currently in use. Selecting the appropriate front-end framework is one of the most crucial steps in building a website. JavaScript frameworks are used to provide scalability and interactivity to websites. They play a significant role in front-end web development today and support the use of tried-and-true current technologies by technical developers[4].

There is no universally accepted definition of a framework. However, it is beneficial to provide an overview of different definitions for orientation. The Cambridge Dictionary defines a framework as a supporting structure around which something can be built, a system of rules, ideas, or beliefs that is used to plan or decide something. The primary purpose of using a framework is to enhance efficiency and productivity by providing a fundamental structure and standardized libraries, thereby allowing developers to focus on the business logic without needing to build everything from scratch[5].

React, Vue. js Angular are the three front-end frameworks that will be in the lead and show significant levels of popularity by 2024.React remains the top choice for front-end developers thanks to its simplicity and outstanding performance. The widespread use of React by large enterprises and its rich ecosystem makes it a top choice in front-end framework surveys and reports. Vue. js is known for its ease of use and flexible integration. As a progressive framework, it allows for gradual adoption based on project needs. Angular, developed by Google, is commonly used for enterprise applications, as evidenced by its adoption in many large enterprises and its comprehensive features supporting complex application development[6].

In the current digital world, where software programs are essential to almost all facets of daily life, it is crucial to make sure they operate at their best. Regardless of the underlying complexity of the systems they engage with, users have learned to demand flawless experiences that meet their needs. Consequently, performance testing has become an essential discipline in software development, offering insights into the behavior of programs under varied loads and conditions[7].Users' use of the website and its applications will be influenced by their quality. In this modern era of commodity competitiveness, web testing is very important[8].

React facilitates the creation of visually appealing user interfaces. Its markup syntax closely resembles HTML, making it straightforward to use and implement. One important component that greatly improves overall application performance is the Virtual DOM, which does away with the necessity for page reloading. React's JavaScript foundation gives it access to NPM, a package manager that makes managing and installing external dependencies easier. Class component life cycles can be customized thanks to React's life cycle functions. React is one of the most widely used frameworks for developing applications and user interfaces, thus it stands to reason that demand for its features will increase in the near future. Its value in assisting companies in achieving their objectives strengthens its position in the market and guarantees its continued relevance over time[9].

Vue is a forward-thinking framework utilized in the development of user interfaces. In contrast to traditional monolithic frameworks, Vue is carefully designed to support gradual adoption. The view layer is the only thing that the core library prioritizes, which makes it easy to understand and combine with other libraries or ongoing applications. Certain properties are produced when Vue is instantiated. Vue creates getters and setters for each data property (a particular property of the Vue object) when it is defined. These JavaScript functions are linked to a JavaScript object and are intended to dynamically detect modifications to a given property and return a computed value[10].

Angular is a comprehensive, component-based, open-source framework that adheres strictly to the Model-View-Controller (MVC) architecture. Because Type script is used in its construction, single-page applications (SPAs) may be created. Angular gives precise instructions on how to build apps and includes comprehensive documentation on how to render the Document Object Model (DOM). Applications in Angular are organized as trees of components. These components offer functionality to the user interface (UI) by using dependency injection, pipelines, directives services. The logical building blocks of huge programs and modules are used to arrange the functionality of the program. An Angular application is made robust by connecting many components[11].

Performance is critical for providing a great user experience (UX) since it directly influences how users interact with a product or website. Performance testing of a web-based application is conducted to evaluate its performance under high traffic conditions. The goal is to ensure the website can maintain optimal efficiency, respond promptly remain highly accessible to users. In the current digital era, users anticipate a flawless user experience and speedy responses from websites. To make sure the website lives up to these expectations, performance testing is therefore crucial[12].

Researchers conducted a review of previous studies on comparison of front-end framework for web development. The study was conducted by (Kaluža and Vukelic, 2018) is titled Comparison of Front-End Frameworks for Web Applications Development. This study examined the most optimized framework for developing Multi Page Applications (MPAs) and Single Page Applications (SPAs). The findings suggest that the Vue. js framework is suitable for creating both MPA and SPA apps due to its high and comparable performance in both categories. React can also be used for building MPA and SPA web applications, as it demonstrates comparable results, albeit at lower values compared to Vue. js. The analysis indicates that Angular is not suitable for creating MPAs but is currently the top framework for developing SPA apps[13]. Researchers also conducted a review of several previous studies on measuring Web Performance, there was some study related to the research that the researchers conducted. The first study was conducted by(Maila-Maila *et al.,* 2019) is titled Evaluation of Open Source Software for Testing Performance of Web Applications. This study was conducted based on the criteria outlined in the ISO/IEC 25023 standard and the software testing process recommended by ISTQB. Jmeter emerged as the top-performing tool, meeting 80% of the selection criteria[14].The second study was conducted by(Bello Bada, 2021) is titled Performance Optimization of Web-Based Application. This study was conducted based on assesing the system's performance before and after implementing optimization, which included loas tests with 3000, 5000 10000 concurrent users. The outcomes were then compared to determine the effectiveness of the performance optimization techniques applied[15].The third study was conducted by(Alam and Dewi, 2022) is titled Performance Testing Analysis of Bandung Tanginas Application with JMeter. This study was conducted based on employing load testing to evaluate the application's performance. The analysis will utilize the results from the load testing experiments in the given scenarios. According to the study's findings, the bandungtanginas. id application is most suitable for users aged thirty to fifty[16].The fourth study was conducted by (Muriyatmoko and Musthafa, 2022) is titled Website Performance Testing Using Speed Testing Model: A Case of Reputable Indonesian Journals. This study was conducted based on analyzing the performance of journal websites in Indonesia accredited by SINTA 1. The evaluation employs parameters from Gtmetrix tools and utilizes descriptive statistical methods for calculation. This research hypothesis concerning the reliability of website speed testing has been experimentally tested and validated[17].The fifth study was conducted by (Indrianto, 2023) is titled Performance Testing on Web Information System Using JMeter and Blaze meter.

This study was conducted based on the performance of an Information System Website utilizing two tools, specifically Apache JMeter and Blaze meter. The tests included 50 and 100 users, with a 10-second ramp-up period a single loop. The findings indicate that the system performs well on these modules, exhibiting a stable average response time, higher throughput reduced deviation[11].

The principal objective of this research is to compare a weather application utilizing three distinct front-end frameworks (React, Vue Angular) and to undertake a comprehensive performance assessment of each. By implementing an identical application across these frameworks, this study aims to evaluate the ease of development, coding practices learning curves inherent to each framework. Furthermore, the research will rigorously measure performance metrics, including load times, rendering efficiency overall responsiveness, with the intent to identify which framework offers the most optimal performance for typical weather application functionalities. Additionally, the study seeks to provide recommendations on which framework excels in terms of user experience, performance and efficiency user satisfaction. It is recommended that future researchers expand upon this study by exploring additional frameworks, incorporating more diverse performance metrics considering the impact of various deployment environments to further refine the understanding of front-end framework performance.

MATERIALS AND METHODS

The research starts with the Define Research Objective, which aims to specify the scope of the study and define particular performance parameters to compare (e.g., load time, rendering speed scalability). Next, relevant performance metrics and best practices for measuring them are identified previous research and performance benchmarks comparing React, Vue Angular are surveyed in the Literature Review. In the Implement Testing Script and Perform Performance Testing stage, scripts are written to automate performance tests to ensure consistency and repeatability initial tests are run to identify any immediate issues or discrepancies. Data collection and analysis include collecting information on important performance parameters like rendering speed and load time, then use statistical techniques to analyze the information and find significant differences by Combining Analysis(quantitative and qualitative). Lastly, the Conclusion presents a brief and clear summary of the findings along with suggestions for which framework works best for specific use cases or under particular situations.

Weather forecasting is predicting the climate conditions of the air which can change from put to put and time to time. So fundamentally, it may be a complex handle that tests the utilization of science and advancement to predict the climatic circumstances at a given time. The expectation of climate makes a difference us adjusts our day-to-day life. Numerous parameters influence the determining of climate like atmospheric temperature, pressure, wind speed, air humidity so on. The reason of weather forecasting is to supply people and organizations with data they can utilize to decrease climate-related misfortunes and develop the community benefits, which incorporate well-being and property security, public health and safety support financial thriving and a standard of living. not them[18].

Quantitative analysis, which entails the measurement and examination of numerical data, is essential in evaluating the performance and efficiency of JavaScript frameworks. In this realm, performance metrics serve as critical tools for objective assessment. Important indicators include Time to First Byte (TTFB), which measures the duration from a user’s request to the receipt of the first byte of data; First Contentful Paint (FCP), which marks the time at which the first piece of content is displayed and Speed Index, which gauges the speed at which content is visually rendered during the page load. Additionally, Largest Contentful Paint (LCP) records the time taken for the largest visible content element to render, while Cumulative Layout Shift (CLS) tracks visual stability by summing individual layout shift scores. Total Blocking Time (TBT) measures the time the main thread is blocked, hindering user interactions. Lastly, Page Weight indicates the total size of the web page.

Usability Testing is a parameter of measuring how user-friendly the application is. To find bugs in the program before it goes live, a team of testers is usually involved. The program may be mobile or web-based. The main aim of usability testing is to guarantee that the program is manageable, adaptable user-friendly[19].

Usability and user experience alone cannot determine whether a product meets users' needs and goals or ensures a good user experience and loyalty. We suggest that the relationship between user experience and usability can be understood through three classifications. (1) Usability is part of the user experience. (2) Usability is the measure of user experience (3) usability and user experience complement each other. This perspective explains the relationship between usability and user experience[20]. This concludes the user experience, with usability concentrating on functionality and user experience highlighting the emotional responses to product aspects. It views user satisfaction as a quality metric and aims to enhance the overall user experience. In essence, it pertains to efforts that utilize three key attributes for measuring how effectively users can utilize the software to accomplish a specific set of tasks. These attributes include efficiency, effectiveness user satisfaction[21].

Qualitative analysis through usability testing surveys can give significant experiences into developers' encounters with website performance. Such surveys can include questions regarding the method of reasoning behind framework determination, user experience, effect on performance best practices adopted by developers when utilizing the framework. For instance, a survey may inquire approximately the reasons developers select a specific framework, whether due to performance, documentation, or other factors. Furthermore, the overview can reveal how the chosen framework affects website performance and the best practices utilized to improve performance. Theoretical foundations supporting this approach include Grounded Theory, which helps in creating hypotheses based on the subjective information collected User-Centered Design (UCD), which ensures that user feedback is leveraged to improve website performance.

Combining quantitative and qualitative analyses involves using both numerical data and non-numerical insights to evaluate and compare JavaScript frameworks. This all-encompassing method offers a thorough comprehension of the advantages and disadvantages of every structure. Page load times and the quantity of network requests are two examples of quantifiable, verifiable facts that quantitative analysis provides to complement qualitative research. In the meanwhile, by providing insights into user experience, usability other subjective elements that cannot be quantified explicitly, qualitative analysis supports quantitative analysis. The evaluation is made more comprehensive and precise by combining these two approaches, which facilitates better decision-making when choosing the framework that best suits the requirements of the project.

When measuring the performance of a web page, there are a few measurements utilized to analyze the time and way in which the page loads. These requirements give a comprehensive view of the net page execution and its effect on client involvement. This research combines numerical data with non-numerical insights using both quantitative and qualitative analysis to assess and compare different JavaScript frameworks. For the quantitative analysis, this research uses a performance testing tool known as Web Page Test, which is used to conduct performance testing on a website. For the qualitative analysis, this research employs surveys for developers.

RESULTS AND DISCUSSIONS

This section presents the findings from the web page performance tests conducted on three front-end frameworks: React, Vue Angular. The performance evaluation utilized web page test tools to systematically measure the speed, efficiency stability of each framework. Additionally, the analysis of data collected from questionnaires distributed to respondents is discussed. This data was meticulously processed using SPSS software, incorporating validity tests, descriptive statistics correlation analyses.

Web Page Performance Test

Web Page Test is a robusttool used for conducting comprehensive performance testing of web pages. It allows users to analyze various aspects of a website's performance, including load times, rendering resource usage, across different browsers and connection speeds. By simulating real-world conditions, Web Page Test provides valuable insights into how a website performs under different scenarios, enabling developers to optimize their sites for better user experiences.

The page performance metrics provide insights into loading efficiency of React Framework. Time to First Byte (0.264s) measures server responsiveness. Start Render (0.800s) and First Contentful Paint (0.918s) indicate when visual content first appears. The Speed Index (1.898s) reflects overall loading speed. Largest Contentful Paint (0.918s) marks the loading time of the largest element. Cumulative Layout Shift (0) shows layout stability. Total Blocking Time (0.000s) indicates no main thread blocking. Page Weight (2,788 KB) affects loading time, especially on slower connections.

The page performance metrics provide insights into loading efficiency of Vue Framework. Time to First Byte (0.179s) measures server responsiveness. Start Render (0.900s) and First Contentful Paint (0.885s) indicate when visual content first appears. The Speed Index (1.993s) reflects overall loading speed. Largest Contentful Paint (0.885s) marks the loading time of the largest element. Cumulative Layout Shift (0) shows layout stability. Total Blocking Time (0.000s) indicates no main thread blocking. Page Weight (2,758 KB) affects loading time, especially on slower connections.

The page performance metrics provide insights into loading efficiency of Angular Framework. Time to First Byte (0.210s) measures server responsiveness. Start Render (0.900s) and First Contentful Paint (0.870s) indicate when visual content first appears. The Speed Index (2.087s) reflects overall loading speed. Largest Contentful Paint (0.870s) marks the loading time of the largest element. Cumulative Layout Shift (0) shows layout stability. Total Blocking Time (0.000s) indicates no main thread blocking. Page Weight (2,823 KB) affects loading time, especially on slower connections.

Validity Test: Validity testing is the method of deciding the extent to which a measurement instrument, such as a questionnaire or test, truly measures what it is gathered to measure. Validity refers to how much an instrument can be depended upon to deliver substantial or precise information. Validity is decided by the significant and suitable interpretation of the information obtained from the measuring instrument as a result of the analyses. Researchers commonly use the Bivariate Pearson correlation (Pearson Product Moment) as a technique for testing validity. This method involves correlating each item's score with the total score, which is the sum of all items. If the calculated r = r table (two-tailed test with sig. 0.05), then the instrument or items are considered valid[22].

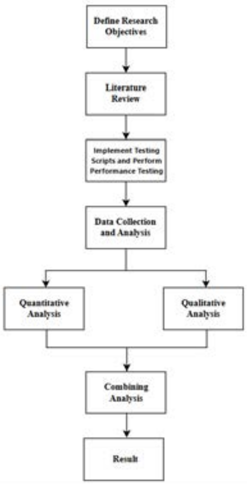
Define Research Objectives
Literature Review
Implement Testing Scripts and Perform Performance Testing
Data Collection and Analysis
Quantitative Analysis
Qualitative Analysis
Combining Analysis
Result

Fig. 1: Framework of thought

Table 1. Survey Questions

User Experience (X)	The framework is easy to learn and use
	Available documentation and resources are helpful
	The framework's user community is very supportive
Performance and Effiency (Y)	The framework provides good performance in the applications I develop
	The framework enables rapid development
	The framework has a small bundle size
Satisfaction (Z)	I prefer to use this framework for my future front-end projects
	The framework provides the most satisfying overall development experience

Table 2. React Framework Test Result

Metrix	Value
Time to First Byte	0.264s
Start Render	0.800s
First Contextful Paint	0.918s
Speed Index	1.898s
Largest Contextful Paint	0.918s
Cumulative Layout Shift	0
Total Blocking Time	0.000s
Page Weight	2,788 KB

Table 3. Vue Framework Test Result

Metrix	Value
Time to First Byte	0.179s
Start Render	0.900s
First Contextful Paint	0.885s
Speed Index	1.993s
Largest Contextful Paint	0.885s
Cumulative Layout Shift	0
Total Blocking Time	0.000s
Page Weight	2,758 KB

Table 4. Angular Framework Test Result

Metrix	Value
Time to First Byte	0.210s
Start Render	0.900s
First Contextful Paint	0.870s
Speed Index	2.087s
Largest Contextful Paint	0.870s
Cumulative Layout Shift	0
Total Blocking Time	0.000s
Page Weight	2,823 KB

Table 5. Combine Framework Test Result

Metrix	React	Vue	Angular
Time to First Byte	0.264s	0.179s	0.210s
Start Render	0.800s	0.900s	0.900s
First Contextful Paint	0.918s	0.885s	0.870s
Speed Index	1.898s	1.993s	2.087s
Largest Contextful Paint	0.918s	0.885s	0.870s
Cumulative Layout Shift	0	0	0
Total Blocking Time	0.000s	0.000s	0.000s
Page Weight	2,788 KB	2,758 KB	2,823 KB

Table 6. R Table

N	Two-Way Test Significance (2-tailed)				
	0.1	**0.05**	**0.02**	**0.01**	**0.001**
100	0.1638	0.1946	0.2301	0.254	0.3211
101	0.163	0.1937	0.229	0.2528	0.3196
102	0.1622	0.1927	0.2279	0.2515	0.3181
103	0.1614	0.1918	0.2268	0.2504	0.3166
104	0.1606	0.1909	0.2257	0.2492	0.3152
105	0.1599	0.19	0.2247	0.248	0.3137
106	0.1591	0.1891	0.2236	0.2468	0.3123
107	0.1584	0.1882	0.2226	0.2458	0.3109
108	0.1576	0.1874	0.2216	0.2446	0.3095
109	0.1569	0.1865	0.2206	0.2436	0.3082

Table 7. React Framework Validity Test

		Total
X1	Pearson Correlation	.440**
	N	109
X2	Pearson Correlation	.542**
	N	109
X3	Pearson Correlation	.400**
	N	109
Y1	Pearson Correlation	.604**
	N	109
Y2	Pearson Correlation	.640**
	N	109
Y3	Pearson Correlation	.477**
	N	109
Z1	Pearson Correlation	.592**
	N	109
Z2	Pearson Correlation	.586**
	N	109

Table 8. Vue Framework Validity Test

		Total
X1	Pearson Correlation	.559**
	N	105
X2	Pearson Correlation	.650**
	N	105
X3	Pearson Correlation	.596**
	N	105
Y1	Pearson Correlation	.323**
	N	105
Y2	Pearson Correlation	.489**
	N	105
Y3	Pearson Correlation	.596**
	N	105
Z1	Pearson Correlation	.376**
	N	105
Z2	Pearson Correlation	.608**
	N	105

Table 9. Angular Framework Validity Test

		Total
X1	Pearson Correlation	.508**
	N	100
X2	Pearson Correlation	.513**
	N	100
X3	Pearson Correlation	.606**
	N	100
Y1	Pearson Correlation	.236*
	N	100
Y2	Pearson Correlation	.302**
	N	100
Y3	Pearson Correlation	.551**
	N	100
Z1	Pearson Correlation	.693**
	N	100
Z2	Pearson Correlation	.560**
	N	100

Table 10. React Framework Descriptive Statistics

	N	Mean	Mode
X1	109	4.25	5
X2	109	4.31	5
X3	109	4.23	5
Y1	109	3.97	5
Y2	109	3.87	4
Y3	109	4.06	5
Z1	109	4.10	5
Z2	109	3.86	4

Table 11. Vue Framework Descriptive Statistics

	N	Mean	Mode
X1	105	3.77	4
X2	105	4.03	4
X3	105	3.77	4
Y1	105	3.93	4
Y2	105	4.23	5
Y3	105	3.88	4
Z1	105	4.05	4
Z2	105	3.93	4

Table 12. Angular Framework Descriptive Statistics

	N	Mean	Mode
X1	100	3.96	4
X2	100	3.94	4
X3	100	3.69	4
Y1	100	4.19	5
Y2	100	4.09	5
Y3	100	3.78	4
Z1	100	4.03	5
Z2	100	3.54	4

Table 13. Pearson’s correlation coefficient

Correlation Coefficient	Interpretation
0.00-0.10	Negligible correlation
0.10-0.39	Weak correlation
0.40-0.69	Moderate correlation
0.70-0.89	Strong correlation
0.90-1.00	Very strong correlation

Table 14. React Framework Correlation Analysis

		X	Y	Z
X	Pearson Correlation	1	.237*	.141
	Sig (2 tailed)		.013	.145
	N	109	109	109
Y	Pearson Correlation	.237*	1	.304**
	Sig (2 tailed)	.013		.001
	N	109	109	109
Z	Pearson Correlation	.141	.304**	1
	Sig (2 tailed)	.145	.001	
	N	109	109	109

Table 15. Vue Framework Correlation Analysis

		X	Y	Z
X	Pearson Correlation	1	.446**	.363**
	Sig (2 tailed)		< .001	< .001
	N	105	105	105
Y	Pearson Correlation	.446**	1	.130
	Sig (2 tailed)	< .001		.185
	N	105	105	105
Z	Pearson Correlation	.363**	.130	1
	Sig (2 tailed)	< .001	.185	
	N	105	105	105

Table 16. Angular Framework Correlation Analysis

		X	Y	Z
X	Pearson Correlation	1	.342**	.487**
	Sig (2 tailed)		< .001	< .001
	N	100	100	100
Y	Pearson Correlation	.342**	1	.166
	Sig (2 tailed)	< .001		.099
	N	100	100	100
Z	Pearson Correlation	.487**	.166	1
	Sig (2 tailed)	< .001	.099	
	N	100	100	100

Table 17. Combining Framework Correlation Analysis

		React	Vue	Angular			
		Y	Z	Y	Z	Y	Z
X	P	.237*	.141	.446**	.363**	.342**	.487**
	Sig	.013	.145	< .001	< .001	< .001	< .001
	N	109	109	105	105	100	100
Y	P		.304**		.130		.166
	Sig		.001		.185		.099
	N		109		105		100

Based on Table 6, the minimum Pearson Correlation value is 0.1865, as it uses 109 respondents (N) with a significance level of 0.05. All Pearson correlation values for each item are above 0.1865. This is indicated by the * or ** in the Total column of the output table. Therefore, these 8 questionnaire items are considered valid.

Based on Table 6, the minimum Pearson Correlation value is 0.19, as it uses 105 respondents (N) with a significance level of 0.05. All Pearson correlation values for each item are above 0.19. This is indicated by the * or ** in the Total column of the output table. Therefore, these 8 questionnaire items are considered valid.

Based on Table 6, the minimum Pearson Correlation value is 0.1946, as it uses 100 respondents (N) with a significance level of 0.05. All Pearson correlation values for each item are above 0.1946. This is indicated by the * or ** in the Total column of the output table. Therefore, these 8 questionnaire items are considered valid.

Descriptive Statistics: Descriptive statistics refer to the use of statistical methods to summarize and describe the main features of a collection of data. Descriptive statistics are the kind of information presented in just a few words to describe the basic features of the data in a study such as the mean and mode. Mean is the mathematical average value of a set of data. It may be computed by dividing the total number of observations by their sum. While mode is the value that appears most often in a set of data, it is the data with the highest frequency.

The descriptive statistics output for the React Framework, as shown in Table 10, presents the analysis of responses to eight questionnaire questions. Each question was answered by 109 respondents. The mean values for X1 to Z2 are 4.25, 4.31, 4.23, 3.97, 3.87, 4.06, 4.10 3.86, respectively. The mode, which represents the most frequently occurring response, is 5 for most questions, reflecting a strong consensus. However, for questions Y2 and Z2, the mode is 4.

The descriptive statistics output for the Vue Frame work, as shown in Table 11, presents the analysis of responses to eight questionnaire questions. Each question was answered by 105 respondents. The mean values for X1 to Z2 are 3.77, 4.03, 3.77, 3.93, 3.23, 3.88, 4.05 3.93, respectively. The mode, which represents the most frequently occurring response, is 4 for most questions, reflecting a strong consensus. However, for questions Y2, the mode is 5.

The descriptive statistics output for the Angular Frame work, as shown in Table 12, presents the analysis of responses to eight questionnaire questions. Each question was answered by 100 respondents. The mean values for X1 to Z2 are 3.96, 3.94, 3.69, 4.19, 4.09, 3.78, 4.03 3.54, respectively. The mode, which represents the most frequently occurring response, is 4 for most questions, reflecting a strong consensus. However, for questions Y1, Y2 and Z1, the mode is 5.

Correlation Analysis: Pearson correlation analysis describes the degree of link between variables. However, it does not clarify which variable is the cause and which is the effect. The examination of the relationship between two variables is termed simple correlation, while correlation involving more than two variables can be partial or multiple. A Pearson correlation coefficient (r) of 1 signifies a perfect positive linear relationship, -1 indicates a perfect negative linear relationship 0 denotes no linear relationship. Based on the questionnaire, the variables are divided into three categories: user experience (X), performance and efficiency (Y) satisfaction (Z). Here, we aim to examine the relationships among these variables.

The Pearson correlation analysis presented in Table 14 shows the relationships among three variables : X, Y Z. The variabels X and Y have a positive correlation with a value of r = 0.237 and p = 0.013, indication a weak correlation and significant relationship. The variables X and Z have a possitive but non-significant correlation with a value of r = 0.141 and p = 0.145,indicating weak correlation between them. Then, the variables Y and Z also have a positive correltaion, with a value of r = 0.304 and p =0.001, indicating a moderate and significant relationship.

The Pearson correlation analysis presented in Table 15 shows the relationships among three variables : X, Y Z. The variabels X and Y have a positive correlation with a value of r = 0.446 and p = < 0.001, indication a moderate correlation and significant relationship. The variables X and Z have a possitive and significant correlation with a value of r = 0.363 and p =< 0..001,indicating weak correlation between them. Then, the variables Y and Z also have a positive correltaion, with a value of r = 0.130 and p =0.185, indicating a weak and non-significant relationship.

The Pearson correlation analysis presented in Table 16 shows the relationships among three variables : X, Y Z. The variabels X and Y have a positive correlation with a value of r = 0.342 and p = < 0.001, indication a weak correlation and significant relationship. The variables X and Z have a possitive and significant correlation with a value of r = 0.487 and p = < 0.001,indicating moderate correlation between them. Then, the variables Y and Z also have a positive correltaion, with a value of r = 0.166 and p =0.099, indicating a weak and non-significant relationship.

Based on Table 17, conclusions can be drawn from Pearson's correlation analysis between the variables of user experience (X), efficiency and performance (Y) satisfaction (Z) with the three front-end frameworks (React, Vue, Angular). React has the weakest correlation between variables, indicating that factors such as user experience do not have a strong relationship with efficiency and performance or user satisfaction. This means that in React development, user experience does not directly affect performance and efficiency or satisfaction of the framework. Results from the Web Page Test also confirm these findings, showing that Vue and Angular have better performance (Time to First Byte and First Contentful Paint) compared to React, consistent with a stronger correlation between these variables. In Vue, there is also a weak relationship between performance and efficiency with satisfaction, as evidenced by the Pearson correlation results supported by the Web Page Test findings (Start Render). Therefore, the relationship between Pearson correlation results and Web Page Test performance indicates that frameworks with a stronger correlation between variables provide better performance compared to frameworks with a weaker correlation.

CONCLUSION

After conducting a comprehensive analysis using Web Page Test and Pearson correlation, it can be concluded that each of the three front-end frameworks has its own strengths and weaknesses across three key aspects: user experience, performance and efficiency satisfaction. Angular is the most suitable framework for achieving the best user experience, as shown by the strongest Pearson correlation between user experience and satisfaction which is 0.487**, along with the fastest First Contentful Paint (FCP) metric recorded at 0.870 seconds from Web Page Test. For optimal performance and efficiency, Vue stands out. This is evident from the strongest Pearson correlation between user experience and performance and efficiency which is 0.446**, supported by the fastest Time to First Byte (TTFB) metric, which is 0.179 seconds according to Web Page Test. Lastly, React is the preferred framework for the highest satisfaction levels. This is indicated by the strongest Pearson correlation between performance and efficiency with satisfaction which is 0.304**, reinforced by the fastest Start Render metric, which is 0.800 seconds as reported by Web Page Test. In summary, the choice of framework can be guided by the specific aspect prioritized: Angular for user experience, Vue for performance and efficiency React for overall satisfaction.

Recommendations for future research include conducting a larger-scale study with a wider variety of websites and frameworks. It is also important to investigate the impact of other factors, such as website design and content, on user experience, performance and efficiency satisfaction. Additionally, developing a more comprehensive model for selecting the best front-end framework for a given website would be beneficial. We hope this study has been helpful in providing insights into the performance of different front-end frameworks we believe that our findings can be used to improve the user experience of websites and applications.

REFERENCES

1. Vyas, R., 2022. Comparative analysis on front-end frameworks for web applications. Int. J. Res. Applied Sci. Eng. Technol., 10: 298-307.
2. Pitale, A. and A. Bhumgara, 2019. Human computer interaction strategies — designing the user interface. November 27-29, 2019, IEEE, 2019 Int. Conf. Smart Syst. Inventive Technol. (ICSSIT).
3. Darmawan, I., M.S. Anwar, A. Rahmatulloh and H. Sulastri, 2022. Design thinking approach for user interface design and user experience on campus academic information systems. JOIV : Int. J. Inf. Visualization, Vol. 6 .10.30630/joiv.6.2.997.
4. Mohane, A., and K. Ghate, 2021. Understanding Web Front-end Development Technology based upon Responsiveness, Optimization and Interactive Nature. Inter. Res. J. Eng. Technol. 8: 56-72.
5. Partelow, S., 2023. What is a framework? understanding their purpose, value, development and use. J. Environ. Stud. Sci., 13: 510-519.
6. Pargaonkar, S., 2023. A comprehensive review of performance testing methodologies and best practices: Software quality engineering. Int. J. Sci. Res. (IJSR), 12: 2008-2014.
7. Yu, J., 2019. Exploration on web testing of website. J. Phys.: Conf. Ser., Vol. 1176 .10.1088/1742-6596/1176/2/022042.
8. Chen, S., U.R. Thaduri and V.K.R. Ballamudi, 2019. Front-end development in react: An overview. Eng. Int., 7: 117-126.
9. Pšenák, P. and M. Tibenský, 2020. The usage of vue js framework for web application creation. Mesterséges intelligencia, 2: 61-72.
10. Indrianto, I., 2023. Performance testing on web information system using apache jmeter and blazemeter. J. Ilmiah Ilmu Terapan Universitas Jambi, 7: 138-149.
11. Kaluža, M. and B. Vukelic, 2018. Comparison of front-end frameworks for web applications development. Zbornik Veleucilišta u Rijeci, 6: 261-282.
12. Maila-Maila, F., M. Intriago-Pazmiño and J. Ibarra-Fiallo, 2019. Evaluation of open source software for testing performance of web applications. Adv. Intell. Syst. Comput., 931: 75-82.
13. Bada, A.B., 2021. Performance optimization of web-based application. Int. J. Comput. Sci. Eng., 10: 39-45.
14. Alam, E., and F. Dewi, 2022. Performance Testing Analysis Of Bandung Tanginas Application With J Meter. Intern. J. Inno. Enter. Sys. 6: 157-166.
15. Muriyatmoko, D. and A. Musthafa, 2022. Website Performance Testing Using Speed Testing Model: A Case of Reputable Indonesian Journals. J. Ilmu Teknik dan Informatika 2: 40-45.
16. Bharti, G.K., A. Ranjan, A. Bharat and S. Yadav, 2023. Design and development of an efficient and intelligent weather forecasting app. Adv. Transdisciplinary Eng., 32: 353-358.
17. Jain, S. and P. Purandare, 2021. Study of the usability testing of e-commerce applications. J. Phys. Conf. Ser., Vol. 1964 .10.1088/1742-6596/1964/4/042059.
18. Ryu¯kyu¯, D, K. Okinawa, K. K.Si. K. Gijutsu and O .K. K .S .G. Kokuritsu, 2017. IEEE Computer Society, & Institute of Electrical and Electronics Engineers 2nd International Conference on Intelligent Informatics and BioMedical Sciences , November 24-26, 2017, Okinawa, Japan.,
19. Ahmad, N.A.N., N.I.M. Hamid and A.M. Lokman, 2021. Performing usability evaluation on multi-platform based application for efficiency, effectiveness, and satisfaction enhancement. Int. J. Interact. Mobile Technol. (iJIM), 15: 103-117.
20. SÜRÜCÜ, L. and A. MASLAKÇI, 2020. Validity and reliability in quantitative research. Bus. & Manage. Stud.: An Int. J., 8: 2694-2726.
21. Gupta, A., P. Mishra, C. Pandey, U. Singh, C. Sahu and A. Keshri, 2019. Descriptive statistics and normality tests for statistical data. Ann. Cardiac Anaesth., 22: 67-72.
22. Schober, P., C. Boer and L.A. Schwarte, 2018. Correlation coefficients: Appropriate use and interpretation. Anesthesia & Analg., 126: 1763-1768.