Solent University

Faculty of Business, Law and Digital Technologies

Solent University Southampton

Faculty of Computing

Academic year of submission

Mohammad Sarker

Author

10152420

“The title of your report”

Progress Report

COM616 AE1

Tutor: Name

Date of submission: Month Year

Contents

[1. Introduction 3](#_Toc150532036)

[1.1 Background 3](#_Toc150532037)

[1.2 Problem Statement 3](#_Toc150532038)

[1.3 Aims and Objectives 4](#_Toc150532039)

[1.4 Research Question 4](#_Toc150532040)

[2. Project Evaluation (Methods) 5](#_Toc150532041)

[3. Project Progress 7](#_Toc150532042)

[4. Project Management 9](#_Toc150532043)

[4.1 Assessment & Discussion of Risks/Contingency Planning: 9](#_Toc150532044)

[4.2 Selection and Timing Including Appropriate Project Milestones: 9](#_Toc150532045)

[5. What Next? 11](#_Toc150532046)

[References List 13](#_Toc150532047)

[Bibliography 14](#_Toc150532048)

# Introduction

## 1.1 Background

JavaScript tools and frameworks have changed web development over the past decade, making possible dynamic and responsive user interfaces. These include React, a prominent JavaScript library for designing user interfaces. React prioritizes the view layer above Angular and Vue. This concentrated approach helps React create engaging and visually beautiful user interfaces, making it a popular choice among developers (Ikkala et al., 2022). React's feature is its virtual DOM (Document Object Model), which streamlines rendering by updating only the essential interface components when data changes. React apps load faster and are more responsive with this rendering method. React is now famous for developers who want to build interactive web apps with smooth user experiences (Thakkar, 2020).

Web development technologies are chosen based on their strengths, weaknesses, and compatibility with project needs. Unlike Angular, React lacks built-in features, requiring external libraries or tools to fulfil particular functions. Despite this, React's simplicity, versatility, and dynamic ecosystem keep it popular (Rane et al., 2021). ReactDOMServer handles server-side rendering efficiently. Server-side rendering is essential for SEO and web application loading performance. React speeds up user experience by rendering and transmitting HTML on the server to the client (Meredova, 2023). ReactDOMServer helps achieve this goal, making React a good choice for SEO-focused and fast-loading projects.

## 1.2 Problem Statement

Online developers have many options for designing interactive and efficient online apps due to the rapid advancement of web technologies. React is a popular library in this diversified JavaScript world for its view layer focus and speedy virtual DOM implementation. Despite its benefits, developers must choose the right technology for their projects. React's advantages and disadvantages compared to Angular and Vue are still debated. The issue is Understanding React's suitability for constructing high-performance, scalable, and stable online applications (Wernersson and Sjölund, 2023). More comparison research is needed to ensure developers can choose React over alternative frameworks. It must be thoroughly investigated to understand how ReactDOMServer's server-side rendering affects web application performance, notably initial loading times and SEO. This study will compare React to popular JavaScript frameworks to fill this gap (Kaluža et al., 2018). React's web development advantages and disadvantages and performance differences are examined. Developers can also learn about ReactDOMServer server-side rendering from the study. Web development technologies evolve with application needs. Developers must match tools and technology to project goals. React must be compared to Angular and Vue (Kabeyi, 2019). Developers may choose the optimal solution for their use cases by comparing performance and React's advantages and disadvantages.

## 1.3 Aims and Objectives

The aim of the proposed study is to assess React's capabilities for developing high-performance, scalable, and maintainable web apps.

1. To compare the performance of React to other popular JavaScript frameworks, such as Angular and Vue.
2. To identify the advantages and disadvantages of using React for building web applications.
3. To explore the use of ReactDOMServer for server-side rendering in React applications.

## 1.4 Research Question

* How does React compare to other popular JavaScript frameworks in terms of performance, scalability, and maintainability for building high-performance web applications?

# Project Evaluation (Methods)

***2.1 Strategy for gathering literature Review:***

A thorough literature review has assessed React's viability and compared it to other major JavaScript frameworks. The first stage was a thorough analysis of web development literature, scholarly publications, and research papers on React, Angular, and Vue from credible magazines and conferences. Google Scholar, Elsevier, Scopus, IEEE, and Science Direct databases were used. Literature review keywords included "React," "Angular," "Vue," "web development frameworks," "virtual DOM," "server-side rendering," and "JavaScript performance." These terms were chosen to focus the research on leading JavaScript frameworks' comparative analysis, performance evaluation, and web development applications (Marx-Raacz Von Hidvég, 2022). This exhaustive review revealed each framework's strengths, shortcomings, and distinctive traits, forming the study's foundation. Developer community polls were also done to assess sentiment and preferences for various frameworks. Experienced professionals and industry experts were surveyed. These surveys yielded quantitative and qualitative data on problems, performance expectations, and use-case scenarios. This multimodal approach ensured a full grasp of React and its equivalents' practical ramifications and user experiences. React, Angular, and Vue web apps were also audited (Paleyes et al., 2022). The audits sought to discover performance, scalability, and maintainability issues. The study examined real-world applications to understand developer problems and how framework decisions affect end-user experience.

***2.2 Evaluation and Organisation of collected Information:***

The information gathered was subjected to a comprehensive examination procedure to guarantee its relevance, correctness, and dependability. Each source was evaluated critically based on the publication's trustworthiness, the author's reputation, and the research methodology used. The results of the literature research, questionnaires, and audits were methodically compiled and categorized to allow for a systematic analysis (Horsman, 2018). The organised data was then evaluated and reviewed in light of acceptable guidelines, legal and ethical issues, and industry best practices. This assessment aided in the identification of trends, patterns, and emerging practices in the web development industry. Furthermore, the approaches, tools, frameworks, and platforms employed in the surveyed applications were assessed to comprehend their impact on overall performance and user satisfaction. Technical and design procedures were documented during the testing phase, including developing React's server-side rendering via ReactDOMServer. This testing enabled a hands-on examination of React's capabilities, providing valuable insights into its effectiveness in real-world circumstances (Abu-Rumman et al., 2022). To create relevant conclusions from the surveys, statistical tests such as regression and co-relation will be performed, and SPSS will be used to run these tests resiliently. Furthermore, test results, performance metrics, and user input were captured and examined to generate relevant findings.

# Project Progress

The first phase of this research comprised a thorough review of scholarly articles, research papers, and existing literature on React, Angular, and Vue. This comprehensive literature analysis laid the groundwork for understanding these frameworks' essential principles, strengths, and shortcomings. Prior research gave critical theoretical insights into React's virtual DOM, its server-side rendering capabilities enabled by ReactDOMServer, and its performance metrics compared to competing frameworks (Meredova, 2023). The literature review has helped lead the investigation of practical implementations and uses of these technologies in web development.

Hands-on experimentation with React's server-side rendering (SSR) technique using ReactDOMServer has been a crucial milestone in our research quest. This procedure thoroughly examines SSR implementation, particularly emphasising improving initial loading speeds and search engine optimisation (SEO) for online apps. The complications of SSR implementation were recorded using systematic technical procedures, highlighting encountered challenges and their associated resolves (Emara et al., 2021). Preliminary test results indicate promising improvements in application loading speed and SEO optimisation. However, it is critical to highlight that these findings are preliminary and rigorous testing is presently being conducted to evaluate the consistency and dependability of these benefits across varied scenarios.

Concurrently, continued efforts are oriented toward developing survey questions and systematically collecting relevant data for audits. The surveys, now in the questionnaire creation stage, are methodically designed to elicit nuanced developer feedback. These insights cover web development framework preferences, problems, and expectations. Concurrently, meticulous information gathering for audits is underway (Appelbaum et al., 2018). This phase thoroughly examines existing web apps built with React, Angular, and Vue. This research intends to validate theoretical knowledge with empirical evidence by connecting with the developer community and critically reviewing real-world applications, offering a complete understanding of these frameworks' performance and usability elements (Brieger et al., 2020).

Looking ahead, the study's trajectory includes the completion of surveys, the execution of extensive audits, and the improvement of SSR trials. Integrating theoretical knowledge, empirical insights from surveys and audits, and actual experimentation results will aid in synthesising comprehensive findings about React's viability in building high-performance, scalable, and maintainable online applications (Bahraini et al., 2020). This research initiative aims to significantly contribute to web development by helping developers make well-informed framework selections.

# Project Management

## 4.1 Assessment & Discussion of Risks/Contingency Planning:

A comprehensive assessment of potential hazards and contingency planning is critical in the closing stages of this project to ensure its successful conclusion. One of the critical hazards is the availability and dependability of audit data sources. To mitigate this, the researcher continually investigates different sources and develops backup strategies, such as widening the pool of applications under consideration (Alkaissy et al., 2022). Another significant danger is the ever-changing nature of web development technologies. To solve this, the project plan contains regular updates and changes, allowing new frameworks or approaches to be flexible. There is also a danger linked with the accuracy of survey responses. Robust survey design strategies, such as clear and brief questions, are being used to improve the data quality, reducing the margin for misinterpretation. There are also contingency plans for unexpected delays caused by technological difficulties or unforeseen circumstances. If necessary, the researcher is ready to devote additional resources and extend the project deadline to ensure that the quality of the research is maintained. Researcher have created regular communication channels, allowing for the rapid identification and resolution of any difficulties that may develop during the final stages (Cooper and Sommer, 2018). Furthermore, continual monitoring of project progress enables early discovery of potential dangers, permitting proactive decision-making to prevent negative consequences.

## 4.2 Selection and Timing Including Appropriate Project Milestones:

The timely and planned scheduling of the project's tasks is essential to its practical administration. The 400 hours allotted for this project have been wisely divided across several activities, guaranteeing a methodical and balanced approach. The following table outlines the key project milestones, illustrating the breakdown of hours dedicated to each task:

|  |  |  |
| --- | --- | --- |
| **Milestone** | **Hours Allocated** | **Status** |
| Literature Review | 100 | Achieved |
| Survey Development | 50 | Under Progress |
| Audit Information Gathering | 70 | Under Progress |
| Server-Side Rendering Testing | 80 | Partially Achieved |
| Data Analysis and Synthesis | 60 | Not Started |
| Report Writing and Finalization | 40 | Not Started |

These milestones form the project's backbone, directing the researcher through the research process. The primary phases are the literature review, survey development, and audit information gathering, which lay the platform for further in-depth studies. Significant time is allotted to comprehensive testing of server-side rendering and meticulous data analysis, reflecting the complexity and criticality of these jobs. The latter steps involve synthesizing the collected data into a thorough report, ensuring the findings are presented coherently and conclusively. The project will move forward quickly if these milestones are met, as well as continual risk assessment and contingency preparation. The researcher hopes to produce a rigorous and insightful analysis by painstakingly handling the project's complexities, significantly contributing to web development frameworks.

The second part of the project involves a strategic and systematic implementation approach, building on the progress accomplished in the previous stages. The implementation plan thoroughly addresses key study objectives by leveraging insights from the literature analysis, ongoing surveys, and the early stages of server-side rendering experiments.

# What Next?

**1) Completion of the survey and data analysis:**

The current survey development will be conducted thoroughly, ensuring the questions are transparent, objective, and targeted to elicit nuanced insights from the developer community. Once completed, the surveys will be distributed to participants, including new developers, experienced professionals, and industry experts. The collected replies will be rigorously analyzed using statistical tools such as regression and correlation, which will be facilitated by software such as SPSS. This in-depth examination reveals patterns, preferences, and issues developers encounter while working with React and other frameworks while offering significant qualitative and quantitative data.

**2) Extensive Audit Analysis:**

Concurrently, the audit information-gathering procedure will be accelerated. Real-world applications built with React, Angular, and Vue will be scrutinized for performance bottlenecks, scalability concerns, and user experience issues. The researcher will get significant insights into the practical ramifications of framework choices through in-depth code reviews and usability testing. The findings of these audits will be reviewed alongside the survey results, adding empirical evidence to the research.

**3) Advanced Rendering Optimisation on the Server:**

The server-side rendering testing phase will be expanded, emphasising sophisticated optimization approaches. This will entail experimenting with various ReactDOMServer setups, caching techniques, and rendering strategies. The research attempts to maximise efficiency improvements in initial loading times and SEO optimization by fine-tuning the server-side rendering process. A comparison of these improvements between React, Angular, and Vue will provide a more nuanced knowledge of each framework's capabilities in real-world circumstances.

**4) Analysis of Comparative Performance:**

The comparative performance analysis is at the heart of the project. Extensive performance testing will be performed, with elements like rendering speed, memory utilization, and responsiveness measured across various scenarios and application complexities. The results will be rigorously studied and compared, demonstrating React's strengths and weaknesses compared to its competitors, Angular and Vue. This analysis will be based on real-world data obtained through surveys, audits, and hands-on testing, confirming its practicality.

**5) Report Synthesis and Documentation:**

Concurrent with the research efforts, detailed documentation outlining each phase's methodology, tools, and findings will be kept. This documentation will clearly record the study process, allowing for replication and peer evaluation. The collected data, survey insights, audit findings, and performance analysis results will be combined into a cohesive and insightful study report. This paper will comprehensively assess React's appropriateness for high-performance online applications and provide valuable advice for developers and stakeholders.

# References List

Abu-Rumman, A., Al Shraah, A., Al-Madi, F. and Alfalah, T., 2022. The impact of quality framework application on patients’ satisfaction. *International Journal of Human Rights in Healthcare*, *15*(2), pp.151-165.

Alkaissy, M., Arashpour, M., Li, H., Alaghmand, S. and Nezamian, A., 2022. Quantitative analysis of safety risks and relationship with delayed project completion times. *Risk analysis*, *42*(3), pp.580-591.

Appelbaum, D.A., Kogan, A. and Vasarhelyi, M.A., 2018. Analytical procedures in external auditing: A comprehensive literature survey and framework for external audit analytics. *Journal of Accounting Literature*, *40*(1), pp.83-101.

Bahraini, N.H., Matarazzo, B.B., Barry, C.N., Post, E.P., Forster, J.E., Dollar, K.M., Dobscha, S.K. and Brenner, L.A., 2020. Protocol: examining the effectiveness of an adaptive implementation intervention to improve uptake of the VA suicide risk identification strategy: a sequential multiple assignment randomized trial. *Implementation science*, *15*, pp.1-12.

Brieger, E., Arghode, V. and McLean, G., 2020. Connecting theory and practice: reviewing six learning theories to inform online instruction. *European Journal of Training and Development*, *44*(4/5), pp.321-339.

Cooper, R.G. and Sommer, A.F., 2018. Agile–Stage-Gate for Manufacturers: Changing the Way New Products Are Developed Integrating Agile project management methods into a Stage-Gate system offers both opportunities and challenges. *Research-Technology Management*, *61*(2), pp.17-26.

Emara, M., Hutchins, N.M., Grover, S., Snyder, C. and Biswas, G., 2021. Examining Student Regulation of Collaborative, Computational, Problem-Solving Processes in Open-Ended Learning Environments. *Journal of Learning Analytics*, *8*(1), pp.49-74.

Horsman, G., 2018. Framework for Reliable Experimental Design (FRED): A research framework to ensure the dependable interpretation of digital data for digital forensics. *Computers & Security*, *73*, pp.294-306.

Ikkala, E., Hyvönen, E., Rantala, H. and Koho, M., 2022. Sampo-UI: A full stack JavaScript framework for developing semantic portal user interfaces. *Semantic Web*, *13*(1), pp.69-84.

Kabeyi, M.J.B., 2019. Evolution of project management, monitoring and evaluation, with historical events and projects that have shaped the development of project management as a profession. *Int J Sci Res*, *8*(12), pp.63-79.

Kaluža, M., Troskot, K. and Vukelić, B., 2018. USPOREDBA FRONT END FRAMEWORKA ZA IZRADU WEB-APLIKACIJA. *Zbornik Veleučilišta u Rijeci*, *6*(1), pp.261-282.

Marx-Raacz Von Hidvég, T., 2022. Are the frameworks good enough?: A study of performance implications of JavaScript framework choice through load-and stress-testing Angular, Vue, React and Svelte.

Meredova, A., 2023. Comparison of Server-Side Rendering Capabilities of React and Vue.

Paleyes, A., Urma, R.G. and Lawrence, N.D., 2022. Challenges in deploying machine learning: a survey of case studies. *ACM Computing Surveys*, *55*(6), pp.1-29.

Rane, S.B., Potdar, P.R. and Rane, S., 2021. Development of Project Risk Management framework based on Industry 4.0 technologies. *Benchmarking: An International Journal*, *28*(5), pp.1451-1481.

Thakkar, M., 2020. Building React Apps with Server-Side Rendering. *Use React*.

Wernersson, D. and Sjölund, V., 2023. Choosing a Rendering Framework: A Comparative Evaluation of Modern JavaScript Rendering Frameworks.

# Bibliography

**Appendix A – Draft Literature Survey**