**Research report  
Javascript frame**



Student name: Rowan van der Weel

Student number: 4924576

Class: DB-04

Teacher: Tim Kurvers

Version: 1.0

Date: 21-09-2023

Table of contents

[**Introduction** 3](#_Toc146204405)

[Problem description 3](#_Toc146204406)

[Main question 3](#_Toc146204407)

[Sub questions 3](#_Toc146204408)

[**Results** 3](#_Toc146204409)

[Sub question 1 3](#_Toc146204410)

[Sub question 2 3](#_Toc146204411)

[Sub question 3 3](#_Toc146204412)

[Sub question 4 3](#_Toc146204413)

[**Resolution** 3](#_Toc146204414)

[Conclusion 3](#_Toc146204415)

[Recommendation 3](#_Toc146204416)

[**References** 3](#_Toc146204417)

[**Version History** 3](#_Toc146204418)

# **Introduction**

## Problem description

A JavaScript framework is a pre-built collection of tools and libraries that provides a structured way to develop web applications. In today’s landscape, it’s essential to use JavaScript frameworks to build the front-end of a web application. They offer pre-written code and patterns to handle common tasks, making it faster and easier for developers to create robust and efficient web applications.   
  
There are many JavaScript frameworks that exist and they keep evolving, each having their own strengths and weaknesses and none of them is the **best**. The choice of a framework is very important, as it directly impacts the efficiency, performance, maintainability and scalability of web applications is impacted.

For my project “Gamify” it was recommended to use either: Angular, React or Vue.js for the front-end. I don’t have any insight about JavaScript frameworks, so with this research report I hope to gain a good understanding of these frameworks and determine which is most suitable for my project.

## Main question

Which of the following JavaScript frameworks is the most suitable for GamifyWork: Angular, React or Vue.js?

## Sub questions

1. What are the strengths and weaknesses for the chosen JavaScript frameworks?
   1. **Document Analysis:** I’ll review the official documentation for each framework. Look for sections or documents that explicitly outline the strengths and weaknesses of the framework.
   2. **SWOT analysis:** Conduct a SWOT analysis for each framework. Identify the Strengths, Weaknesses, Opportunities, and Threats associated with using each framework. This will provide a structured framework for evaluating their respective advantages and disadvantages.
2. What is the level of community support and availability of resources for learning and troubleshooting for Angular, React, and Vue.js?
   1. **Community research:** Engage with online communities, forums, and social media groups dedicated to each of these frameworks. Observe discussions, queries, and the level of activity within these communities. Take note of the number of members, frequency of posts, and responsiveness to inquiries.
   2. **Survey:** Create a survey targeting developers who have experience with Angular, React, and Vue.js. Ask questions about their perception of community support, availability of resources, and their experiences with troubleshooting. Include questions about their participation in online communities, especially on platforms like Reddit. Ask if they find these forums helpful for learning and troubleshooting. Analyse the survey responses to gauge the level of support and availability of resources for each framework.
3. What security features and vulnerabilities exist in each framework?
   1. **Security test:** Conduct security testing on each of these frameworks. This may involve using specialized tools and techniques to identify potential vulnerabilities the effectiveness of their security features.
   2. **Literature study:** I’ll search for academic papers, articles, and blog posts that discuss the chosen JavaScript frameworks. Summarize the key findings regarding their strengths and weaknesses.
4. What is the level of flexibility and customization offered by each framework?
   1. **Best good and bad practices:** Research and compile a list of best practices for customization in each of these frameworks. These practices should highlight effective ways to extend or modify the behaviour of the framework while avoiding common pitfalls.
   2. **Observation:** Observe and analyse real-world applications or projects built using each of these frameworks. Pay attention to how developers have customized the framework to meet specific requirements. Document any notable approaches or techniques used for customization.

# **Results**

## Sub question 1

**What are the strengths and weaknesses for the chosen JavaScript frameworks?**  
All 3 of the frameworks have their own advantages and drawbacks. If you want to choose a framework, it’s recommended to pick one you like. I’ll zoom in at everyone of them and then I’ll give a conclusion.   
  
**Angular**

This framework is the most widely framework among the three, it includes support for a lot of things. It is build for mainly large and complex projects. “In its core, Angular is all about building re-usable user interface components which you then control with Angular and which you can combine with other components to build an entire user interface from those Angular-controlled components” (Schwarzmüller, 2023). Angular uses Typescript. Angular is very popular amongst developers.

**React**

React is a JavaScript library that is all about components. It’s a very small and simple library, which provides you tools to render whatever you want in your circumstances. React usually uses JavaScript (a special JavaScript feature called “JSX”). React has been downloaded the most in the last past 2 years, according to the graph above the conclusion section. React has the most fans and the job offering is extremely wide (best as a safe pick).

**Vue.js**

The last framework sits between React and Angular if you look at it’s size. Vue uses regular JavaScript. The framework is a very popular and the community keeps growing, but there are not as many job offerings as with React and Angular.

**Conclusion**

The more tools/support a framework contains, doesn’t immediately mean it’s better than the others. It simply relies on your project, if it’s an very complex project I would recommend to use Angular. However if you want to focus on simplicity and components, React would be a better option for that. Typescript includes all of the features of JavaScript and more. “TS is a typed superset of JS that compiles to plain JS. In Contrast, JavaScript is a dynamic language that doesn't support types.” If you are new to JavaScript, it would be better to learn that first and later you can evolve to Type (Olszewska, 2023). “Searching on indeed on February 21, 2022, showed the following results: React has 67,301 job offerings, Angular has 24,508 job offerings, and Vue.js has 3,857” (Vue vs React vs Angular: What Framework Would You Choose?, 2022). If you want to start simple and have a wide job offering, you should start with React.

My project is pretty simple and small, so Angular is out of the question. For GamifyWork I would choose either React or Vue, due to their suitability for smaller and less complex projects.

**Downloads per year**

A screenshot of a computer

Description automatically generated

**Vue vs React vs Angular – a comparative table**

A blue and white table with text

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **React** | **Positive** | **Negative** |
| **Intern** | * Large community. * Strength in components. | * State management complexity. * Lack of built-in features. |
| **Extern** | * Big demand for React developers. * Integration with other tools. | * Competition with other frameworks. * Fragmentation in the ecosystem. |

|  |  |  |
| --- | --- | --- |
| **Angular**  Angular JS full logo transparent PNG - StickPNG | **Positive** | **Negative** |
| **Intern** | * Full-featured. * Typescript integration. | * Steep learning curve. * Larger file sizes. |
| **Extern** | * Support for big and complex applications. * Integration with tools. | * Competition with other frameworks. * Demotivate developers to work with Angular |

|  |  |  |
| --- | --- | --- |
| **Vue.js**  A green and blue letter v  Description automatically generated | **Positive** | **Negative** |
| **Intern** | * Simple for beginners. * Flexible and progressive. | * Smaller ecosystem. * Less adoption in large and complex applications. |
| **Extern** | * Growing community. * Market adoption. | * Competition with other frameworks. * Depends on one person. |

## Sub question 2

## Sub question 3

## Sub question 4

# **Resolution**

## Conclusion

## Recommendation

# **References**

# **Version History**

|  |  |
| --- | --- |
| **When?** | **What?** |
| 21/09/2023 | First start, initialized it and finished the introduction. |
| 28/09/2023 | Chosen different methods from the DOT framework for the sub questions. |
| 5/10/2023 | Result sub question 1 finished. |