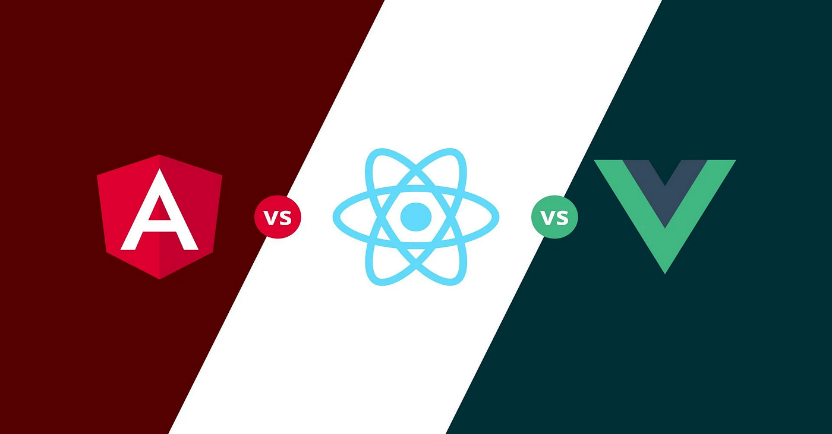
|  |
| --- |
| JavaScript Framework |
| Billy Hofland  DB03  2023/2024 |



Research Report

Research Report

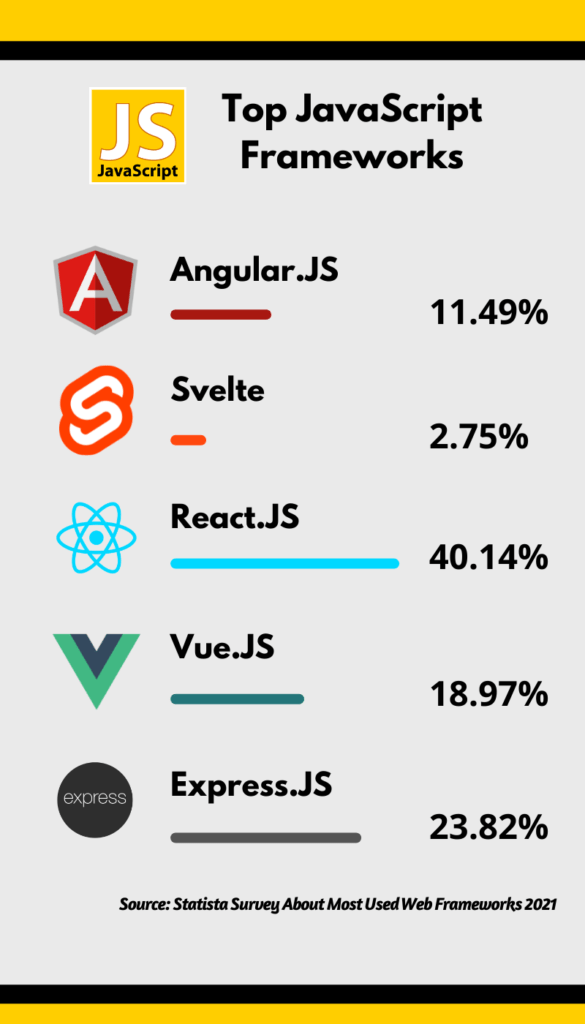


Table of contents

[Introduction 3](#_Toc147496723)

[Problem description 3](#_Toc147496724)

[Main question: 3](#_Toc147496725)

[Sub-questions: 3](#_Toc147496726)

[Results 5](#_Toc147496727)

[Sub-question 1: 5](#_Toc147496728)

[Sub-question 2: 5](#_Toc147496729)

[Sub-question 3: 5](#_Toc147496730)

[Sub-question 4: 5](#_Toc147496731)

[Resolution 5](#_Toc147496732)

[Conclusion: 5](#_Toc147496733)

[Recommendation: 5](#_Toc147496734)

[References 6](#_Toc147496735)

[References 6](#_Toc147496736)

[Version history 7](#_Toc147496737)

# Introduction

## Problem description

JavaScript frameworks are essential in the construction of web applications. They provide developers with structured ways to build applications by offering code snippets, functions, and modules that you can reuse and customize. There are a lot of JavaScript frameworks that offer different features specific for the needs of your project. Hence, it is important to analyze, compare and select the most suitable framework that aligns with the goals and needs of the project.   
  
Because of the rapidly evolving amount of JavaScript frameworks, it is a challenge for developers and companies to select the most appropriate one for their specific needs. The choice of a framework can impact the efficiency, maintainability, and performance of web applications.

## Main question:

Which JavaScript framework is most suitable for developing an online education platform?

## Sub-questions:

1. What are strengths and weaknesses for the top three most popular JavaScript frameworks?
2. Literature study: By looking at the documentation, I can find information about strengths and weaknesses of the most popular frameworks.
3. SWOT analysis: By conducting a SWOT analysis for each framework, identify the strengths, weakness, opportunities, and threats.
4. How readily available and comprehensive are the support and resources, including documentation, community forums, and tutorials, for the most popular frameworks?

A. Community Research: By searching for various forums and user created posts, I can find out what community is the most active for popular frameworks.

B. Survey: By conducting a survey for a large forum I can receive information about what the community uses.

1. What are the built-in security protections and vulnerabilities in top three most popular frameworks?
2. Problem analysis: Research what the most popular vulnerabilities and attacks are towards frameworks. And how different frameworks find solutions for these problems.
3. Document analysis: By researching online guides and forums, I can get a general idea of strengths and weaknesses of the most popular frameworks.
4. How easily can each framework be customized to meet the specific needs and features required for an online education platform?
5. Best good and bad practices: By viewing existing projects, I can see how developers managed to customize their needs and features with a specific framework.
6. Observation: By looking at what other people are using and how they are customizing their framework I can see if it’s a good fit for my project

# Results

## Sub-question 1:

**What are strengths and weaknesses for the top three most popular JavaScript frameworks?**

JavaScript frameworks differ in many ways, the frameworks each have their own unique strengths and weaknesses. All these popular frameworks are under active development. Meaning they regularly release new versions and community posts which can enhance your development skills. There is no best framework available, however there can be a framework that is most suitable for your personal project requirements and your experience in working with frameworks. (Ritika, 2022)

There are multiple differences between these frameworks that are of importance for my personal project, Popularity being one of them. Because I have never worked with a framework before it is decisive to pick a framework that has a high popularity with an active community. The most popular framework at this moment is React, with a large community that is actively involved in forums like stack overflow there are a lot of resources and difficulties already availible for use. (Mahajan, 2019)

Another key factor that is important for an online learning platform is user experience and engagement, prior to my project, it would be preferable if certain parts of code would be reusable. Considering that my project uses two front ends with overlapping components. “React offers a component-based architecture that divides the application’s UI into smaller, reusable components” (Laurent, 2023). This architecture provides reusability and customization and can therefore be used in different parts of the application as well as swapping components based on user preferences or behaviour. Hence for an online education platform this architecture would be ideal since users and teachers like to view components in a different way or personalize them.

**SWOT analysis for the three most popular frameworks:**

|  |  |  |
| --- | --- | --- |
| React | Positive | Negative |
| Intern | 1. Component-based architecture 2. Large active community | 1. State management is more difficult for larger projects. 2. Frequent updates may lead to version compatibility issues. |
| Extern | 1. Integration with tools like React Native 2. Improvements in developer experience | 1. React changes and updates frequently which results in more maintainability for the application. |

|  |  |  |
| --- | --- | --- |
| Angular | Positive | Negative |
| Intern | 1. Strong TypeScript support. 2. Structured architecture. | 1. Harder to learn. 2. Angular has a low performance in runtime. |
| Extern | 1. Support for large-scale applications. 2. Integration with tools like Firebase. | 1. Complexity of the framework may discourage developers. |

|  |  |  |
| --- | --- | --- |
| Vue.js | Positive | Negative |
| Intern | 1. Lightweight and flexible. 2. Easy to learn and use, for beginners. | 1. Smaller ecosystem and community compared to react and angular. 2. Less adoption in large-scale applications. |
| Extern | 1. Growing popularity and community. 2. Continued improvements in performance and tooling | 1. Depends on a single creator/maintainer |

## Sub-question 2:

To know how readable and comprehensive a community is we must know the certain attraction to a framework and how many people interact with it. One way to do this would be social media, how active is a popular framework on different social media sites and how many users interact with this framework? I have done some research and put these numbers in visual graphs. React is the framework with the most interaction.  
A graph of different colored bars

Description automatically generated

Users that are members within LinkedIn 1

A graph of different posts

Description automatically generated

Total likes in the past 3 post over different social media platforms

A graph of different colored columns

Description automatically generated with medium confidence

Total posts of the framework on all social media platforms

A graph of followers

Description automatically generated

Followers of a framework on different social media platforms

To further investigate the popularity and likings of the community me and a fellow student conducted a survey about which framework they used and posted it in a reddit community for frameworks. We posted this survey in 2 communities and this is the results we received:

A black rectangular object with a white line

Description automatically generated

Survey 1 about which framework people have worked with before in the past (<https://www.reddit.com/r/learnjavascript/comments/16ug7hz/with_which_javascript_framework_have_you_worked/>)

A black rectangular object with a white line

Description automatically generated

Survey 2 about which framework people have worked with before in the past (<https://www.reddit.com/r/webdev/comments/16ug9yd/with_which_javascript_framework_have_you_worked/?rdt=42659>)

## Sub-question 3:

JavaScript frameworks like Angular, Vue, and React are widely used for developing web applications. Because large enterprises as well as many individuals use these frameworks, they are a notable target for various types of cyberattacks. For the most popular attacks I will research whether the most popular frameworks have countermeasures for them.

**Attacks: Cross-Site Scripting (XSS) and injection attacks**

XSS attacks function by injecting malicious scripts into webpages viewed by other users. These scripts can then be used to steal information and carry out actions on behalf of the user without their consent. Each framework handles this vulnerability differently.

Vue:

“Vue mitigates XSS through a feature named “escaping”. Where any information entered can be taken as a text string.” (Matias, 2022)

Angular:

“Angular uses several security features to protect applications from attacks. For example, Angular uses Content Security Policy (CSP) to prevent cross-site scripting (XSS) attacks. CSP is a security policy that helps detect and prevent XSS attacks by allowing sources of content to be loaded into a webpage.” (Srivastava, 2023)

React:

“Typically, this attack would work in traditional HTML, JS based application. But React is clever! Instead of executing the malicious code, it would interpret the code as a string”(Vindula, 2023)

the way frameworks handle OWASP top 10 really differs, as developer you must understand what security procedures were build-in the framework you are using and apply them accordingly.

**Angular**

This JavaScript framework offers an embedded safeguard against XSS attacks. "Cross-site scripting (XSS) enables attackers to inject malicious code into web pages. Such code can then, for example, steal user and login data, or perform actions that impersonate the user. This is one of the most common attacks on the web" (Acunetix, 2023). Angular introduces functionalities that assist in safeguarding against two prevalent internet security issues: cross-site request forgery (CSRF or XSRF) and cross-site script inclusion (XSSI). Although these problems are primarily addressed server-side, Angular provides utilities that facilitate easier client-side management. Angular incorporates intrinsic protection against malicious code in web applications, utilizing features such as Content Security Policy (CSP). CSP helps thwart cross-site scripting (XSS) attacks by governing which scripts are permissible to run on a webpage. Within Angular, CSP permits developers to define which external resources may be loaded, minimizing the likelihood of unauthorized script execution, and bolstering the application's overall security (Angular, 2023). Angular supplies coherent and comprehensible documentation on the application of its security features.

**React**

React does not provide inherent safeguards against Cross-Site Scripting (XSS) attacks. The onus is on developers to manually manage input sanitization or utilize third-party libraries (Bahrynovska, 2023). Furthermore, React does not inherent support for Cross-Site Request Forgery (CSRF) protection; hence, developers are tasked with instating suitable measures independently. Although Content Security Policy (CSP) is not exclusive to React, it can be imposed at the server level to alleviate specific kinds of attacks. Nonetheless, a plethora of online resources, articles, and advice is accessible, providing guidance on focal points and methods to tackle security issues within React applications.

**Vue.js**

In Vue, several vulnerability reports pertaining to Cross-Site Scripting (XSS) originate from instances where developers deliberately render unsensitized, user-supplied content as Vue templates. This practice is intrinsically hazardous and is not something Vue can mitigate. Moreover, initializing Vue on a page that includes server-rendered and user-contributed content can introduce analogous vulnerabilities. The recommended best practice advises against mounting Vue on nodes containing such content (You, 2023)."HTTP security vulnerabilities, such as cross-site request forgery (CSRF/XSRF) and cross-site script inclusion (XSSI), are primarily addressed on the backend, so they aren't a concern of Vue's. However, it's still a good idea to communicate with your backend team to learn how to best interact with their API, e.g., by submitting CSRF tokens with form submissions" (You, 2023).The official website provides warnings regarding potential risks in this area.

## Sub-question 4:

# Resolution

## Conclusion:

## Recommendation:

# References

# References

Hert. (2023, 10 5). *her*. Retrieved from hert: http://Google.com/

Laurent, L. (2023, 7 31). *AppMaster*. Retrieved from React's Component-Based Architecture: A Case Study: https://appmaster.io/blog/react-component-based-architecture

Mahajan, A. (2019, 12 05). *React vs. AngulaWr vs. Vue: The Complete Comparison*. Retrieved from Dzone: https://dzone.com/articles/react-vs-angular-vs-vue-the-complete-comparison-to

Ritika. (2022, 12 1). *Invendus*. Retrieved from What is a Framework and Why use Frameworks in Software Development?: https://invedus.com/blog/what-is-a-framework-and-why-use-frameworks-in-software-development/#:~:text=The%20type%20of%20framework%20that,your%20specific%20needs%20and%20requirements.

# Version history

|  |  |
| --- | --- |
| **When?** | **What?** |
| 21/09/2023 | Version 1.0 of research report |
| 28/09/2023 | DOT Framework for sub-questions |
| F05/10/20235 | Researching sub-questions |
| 06/10/2023 | Sub-question 2, survey conducted |
| 12/10/2023 | Sub-question 3,4 |