

**Research Question:**

**Front end: React vs Vue vs Angular?**

## Table of Contents

Relevant terms .....	2
Introduction.....	3
What are Angular, React.js & Vue?.....	4
Popularity, Community, and ecosystem .....	4
Philosophy comparison.....	6
Syntax comparison.....	8
Choices and results .....	9
Chosen research methods.....	9
Results.....	9
Conclusion and recommendation.....	9

## Relevant terms

Term	Definition
MVC	The <b>Model-View-Controller (MVC)</b> framework is an architectural pattern that separates an application into three main logical components Model, View, and Controller. ( <i>MVC Framework Tutorial for Beginners: What is, Architecture &amp; Example</i> , 2021)
Router	For a Single-page application in the application layer, a router is a library that decides what web page is presented by a given URL. This middleware module is used for all URL functions, as these are given a path to a file that is rendered to open the next page. ( <i>Routers - MDN Web Docs Glossary: Definitions of Web-related terms / MDN</i> , 2021)
Client-side validation	Before submitting data to the server, it is important to ensure all required form controls are filled out, in the correct format. This is called client-side form validation and helps ensure data submitted matches the requirements set forth in the various form controls. This article leads you through basic concepts and examples of client-side form validation. ( <i>Client-side form validation - Learn web development / MDN</i> , 2021)
DOM	The Document Object Model (DOM) is an application programming interface (API) for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated. ( <i>What is the Document Object Model?</i> , 2021)
HTTP client	Most front-end applications need to communicate with a server over the HTTP protocol, to download or upload data and access other back-end services. ( <i>Angular</i> , 2021)

## Introduction

Frontend frameworks and libraries are, in most cases, written in JavaScript and are for organizing the functionality, interactivity of your website. (*History of front-end frameworks - LogRocket Blog*, 2018) There are three frameworks for building web applications that every frontend developer has heard about: React, Vue.js, and Angular. Those who have heard of these tools, have also asked themselves, which one should I use?

The debate of React vs Angular and recently Vue.js, is an age-old one that recommences every year as each come out with an improved and updated version of their software. This document aims to introduce, explain, and compare each of these front-end frameworks to showcase the use cases and comparison between each, with the goal of outlining one that would be the best for the withdrive project, had we known of this document beforehand.

In order to answer the main question, this being: **Which of the aforementioned frameworks is best fit for my project?** This document will first be introducing each framework, providing some context and background behind each of the projects. These are the following sub questions that will aid in the explanation and expiration of the main question:

- I. What is the Popularity, Community, and ecosystem of each of the frameworks?
- II. What are the differences in features in each framework's features?
- III. How does the learning curve and syntax vary amongst the frameworks?

A score will be awarded from 1-10 for each framework, for each section. This is done so that later all scores can be added up. This is a quantitative way to decide which one is best fit.

## What are Angular, React.js & Vue?

Angular, React and Vue are all highly popular JavaScript libraries and frameworks that help developers build complex, reactive, and modern user interfaces for the web. Front-end frameworks declare rules on how developers should structure their HTML, CSS, and JavaScript, to make the application easier to develop. In this manner, frameworks are opinionated—meaning they prefer you to make certain decisions for ease of development. (Maximilian Schwarzmüller, 2020) Whilst they can all be used to achieve the same or similar goal, their implementation and adaptation varies. As will be explored and explained throughout.

Even though *React JS* is a library not a framework, henceforth it will be referred to as a framework as well, for clarity.

### I What is the Popularity, Community, and ecosystem of each of the frameworks?

#### Popularity and demand

Using survey results from a developer survey conducted by stack overflow taken in 2019. It the largest and most comprehensive survey of over 90,000 people who code around the world. We may use results found to present a picture over what developers around the world prefer to use. (Stack Overflow Developer Survey 2019, 2019)

##### Most Loved, Dreaded, and Wanted Web Frameworks

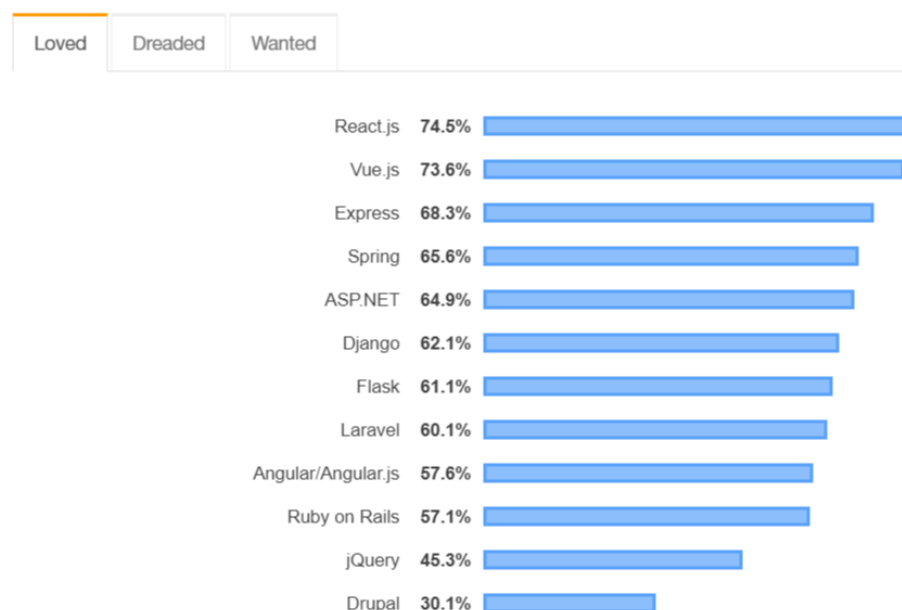


Figure 1- Most Loved Framework Chart (Stack Overflow Developer Survey 2019, 2019)

The results of this survey would suggest that developers tend to enjoy working with React.js and Vue.js as they both are at the top of the charts of most “loved” web frameworks. Angular on the other hand is placed lower the list. Potentially indicative that the prior two are easier to work with.

Another quantitative indicator that may be used is the npm downloads chart. npm is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them and manages dependency conflicts intelligently. (*npm / npm Docs*, 2021)

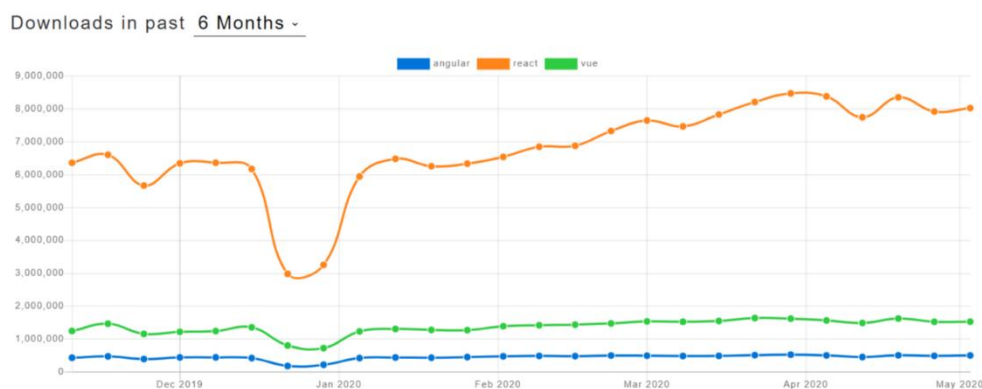


Figure 2- NPM trends downloads and GitHub usage (angular vs react vs Vue / npm trends, 2020)

React.js is significantly ahead of its competitors in this respect. Vue.js’s and React.js’s rating is significantly higher than Angular.js’s. (*Musienko*, 2021) From this we can deduce that React is downloaded a lot more than the other two and this could be indicative that its learning curve is more adapted for beginners. This will be further explored in a later section.

## Community and ecosystem

When choosing what framework, you want to learn and use, an important factor that should be considered is whether the framework you are using is still being developed and maintained. As you don't want to be using obsolete and outdated technologies, due to security and extendibility reasons. (Sawant *et al.*, 2018) This however is not an issue for these frameworks as they are all still currently being updated and maintained.

- Angular is the oldest of the three frameworks, first showing up in 2010. It is currently being developed and maintained by Google. Angular has extensive and thorough documentation available. (Angular, 2021)
- React is developed by Meta (Facebook), since 2013. It has grown massively in popularity and amassed a big community. Due to the sheer popularity of React, as mentioned before there are many resources available online for learning the library. Documentation is also available on the react website. (Getting Started – React, 2021)
- Vue is the youngest when compared to the other two frameworks but has grown a lot in popularity. A lot of documentation available, however due to a lot of the developer base being non-English speaking, it's therefore harder to find documentation and solutions to issues in English. (Aris Pattakos, 2020)

When looking at the Popularity, Community, and ecosystem of each of the frameworks, I would award: 8 points to React, 6 points to Vue and 6 points to Angular. Due to the sheer popularity of React, it means that there are many **different** resources for the library, and this is a very important thing when choosing which library to work with as sometimes one solution doesn't work for you. If more people use something, there is more probability that your problem is not unique.

## II What are the differences in features in each framework's features?

When picking what framework to use for a project, it is important to make the selection based on what you want your web application to do. Each one of them has pros and cons and is better adapted to certain types of projects, as each has unique aspects. Due to the sheer number of features included in each, four **core** features that are useful for my use case are namely: **MVC architecture, DOM, Router, Client-side validation, and HTTP client.** A breakdown of each is discussed below:

*Angular* - Angular comes packed with support for many useful things. This includes of the MVC architectural pattern, its components act as the controller/model and its templates as the view which allows developers to work on both internally (*Angular, 2021*), controlling the UI, reacting to user input, validating user input in forms, routing, state management sending AJAX HTTP requests (*Maximilian Schwarzmüller, 2020*). All these features that Angular provides the developer with is with the hopes to make developing without the need for additional libraries. (*Angular, 2021*)

*React* – React natively does **not** include built-in form validation support, or router, and no HTTP client. React also does not have tools to natively include the MVC pattern, React only adheres to the view element of MVC. To supplement these missing features, one must use other libraries which has caveats, which include the dependency on third party tools which may be abandoned at any moment. Also, there may be potential incompatibility between certain external libraries. (*Maximilian Schwarzmüller, 2020*)

*Vue* - Vue is a framework which kind of sits between React and Angular. It's not as "big" as Angular but it definitely includes more features than React does. Vue does give you built-in state management and it also ships with a built-in router. It does, however, not include form validation or HTTP client functionalities. (*Maximilian Schwarzmüller, 2020*) Vue – is the same phonetically in English as *view*, and it corresponds to the traditional (MVC) architecture. The core library of Vue.js focuses the view layer by default. (*The Good and the Bad of Vue.js Framework Programming, 2019*)

Just like Angular and React, the core of Vue is combining reusable components to create a user interface. It's vaguely in between the two.



From this it may seem that Angular should be the first choice as it has in-built tools for almost everything you would need for a project. But doesn't have to be the case. It entirely depends on the needs of a said project and the better choice may change on the type of project and the projects specific needs. As mentioned, before you may supplement missing features in Vue and React using community made libraries. This does mean however that compatibility may become an issue, and the need for up-to-date support of each of the additional projects.

When looking at the differences in features of each of the frameworks, I would award: 10 points to Angular, 8 points to Vue and 5 points to React. As out of the box Angular just has the most features from the get-go and doesn't rely on third party tools as much as React does.

### III How does the learning curve and syntax vary amongst the frameworks?

The learning curve, for Angular is quite steep, Angular requires you to learn associated concepts like TypeScript and MVC. (Shaumik Daityari, 2019) It also is a framework with so many features it may be daunting to pick what to learn and in what order, and if its all relevant to your actual project.

React offers a Getting Started guide that should help one set up React in about an hour. The documentation is thorough and complete. (Getting Started – React, 2021) React is just a library. This makes the learning curve of the core framework not so steep but depends on the path you take with additional functionality by implementation of additional libraries. (Shaumik Daityari, 2019)

Vue has a fairly shallow learning curve, it is quite easy to develop and learn it, but this is also a blessing and a curse as Vue allows for bad code, which will be detrimental for long term development and testing. (Shaumik Daityari, 2019)

When it comes to syntax and structuring your code, it's a matter of personal preference. Some like to use TypeScript while others prefer to use Javascript. As mentioned before Angular is the only one of the three that only uses Typescript, whilst Vue supports both. (*docs— Vue.js*, 2021)

When looking at the learning curves and syntax of each of the frameworks, I would award: 10 points to React, 8 points to Vue and 5 points to Angular. As React has good docs available as well as many online resources.

## Choices and results

### Chosen research methods

A lot of this document relied on analysis and review of articles and studies written by others. This was important, as when conducting research, one should look for resources compiled by individuals that are highly skilled and knowledgeable in their field. To combat affirmation bias, a source must be looked at critically and with the use of other resources, either enforcing for contradiction of a statement/s.

For certain sections of this document, quantitative data analysis, in the form of raw data analysis was used. Namely within the section that discussed community and eco system of the frameworks.

### Results

Looking at each of the frameworks in conjunction with my personal project and needs, the following table was created:

	<b>I</b>	<b>II</b>	<b>III</b>	<b>Total</b>
<b>React</b>	8	5	10	23
<b>Vue</b>	6	8	8	22
<b>Angular</b>	6	10	5	21

The results of this table suggest that React is the winner, but it must be mentioned that the other two were not far off. With Angular being short two points, and Vue being just one point short.

## Conclusion and recommendation

To conclude, knowing that all three frameworks are being actively developed and maintained while features are added or removed as new versions come, you won't make a mistake regardless of your chosen framework. The most important thing would be to always keep up to date on the newest feature. Each framework respectively has its own pros and cons.

As per the results, React is my library of choice, There are just so many resources available, so I am able to find whatever I need with online resources. Whilst it doesn't come with a lot of features compared to Vue and Angular. The libraries that are needed for the scope of this project, are well maintained and have active teams working on them. Namely: Axios and Bootstrap.

The recommendation is that there is simply too much overlap over each of the frameworks. There is simply not one that is objectively better than another. As per the table of results, If I had done this project again, I would stick with React, as I am now more proficient with it. There are many available libraries for it. Combined with the vast number of resources that I can consult for help and guidance in the development process.

## Bibliography

*What is the Document Object Model?* (2021). W3.org. [https://www.w3.org/TR/REC-DOM-Level-1/introduction.html#:~:text=The%20Document%20Object%20Model%20\(DOM,document%20is%20accessed%20and%20manipulated.&text=Nevertheless%2C%20XML%20presents%20this%20data,used%20to%20manage%20this%20data](https://www.w3.org/TR/REC-DOM-Level-1/introduction.html#:~:text=The%20Document%20Object%20Model%20(DOM,document%20is%20accessed%20and%20manipulated.&text=Nevertheless%2C%20XML%20presents%20this%20data,used%20to%20manage%20this%20data).

*History of front-end frameworks - LogRocket Blog.* (2018, October 16). LogRocket Blog. <https://blog.logrocket.com/history-of-frontend-frameworks/>

Maximilian Schwarzmüller. (2020, March 19). *Angular vs React vs Vue - My Thoughts.* Academind.com; Academind. <https://academind.com/tutorials/angular-vs-react-vs-vue-my-thoughts#framework-backgrounds>

*Angular.* (2021). Angular.io. <https://angular.io/docs>

*React – A JavaScript library for building user interfaces.* (2021). Reactjs.org. <https://reactjs.org/>

*Routers - MDN Web Docs Glossary: Definitions of Web-related terms | MDN.* (2021, October 8). Mozilla.org. <https://developer.mozilla.org/en-US/docs/Glossary/routers>

*Client-side form validation - Learn web development | MDN.* (2021, October 8). Mozilla.org. [https://developer.mozilla.org/en-US/docs/Learn/Forms/Form\\_validation](https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation)

*Stack Overflow Developer Survey 2019.* (2019). Stack Overflow. <https://insights.stackoverflow.com/survey/2019#most-loved-dreaded-and-wanted-npm> / *npm / npm Docs.* (2021). Npmjs.com. <https://docs.npmjs.com/cli/v6/commands/npm>

Musienko, Y. (2021, September 24). *title.* Merehead; Merehead. <https://merehead.com/blog/angular-vs-react-vs-vue-best-choice-2021/>

Sawant, A. A., Huang, G., Vilen, G., Stojkovski, S., & Bacchelli, A. (2018). Why are Features Deprecated? An Investigation Into the Motivation Behind Deprecation. *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*.  
<https://doi.org/10.1109/icsme.2018.00011>

Aris Pattakos. (2020, November 26). *Angular vs React vs Vue 2021*. AThemes; aThemes.  
<https://athemes.com/guides/angular-vs-react-vs-vue/#:~:text=React%20is%20a%20UI%20library,them%20and%20understand%20their%20differences>.

*Getting Started – React*. (2021). Reactjs.org. <https://reactjs.org/docs/getting-started.html>

*The Good and the Bad of Vue.js Framework Programming*. (2019). AltexSoft.  
<https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>