Research Paper on Frontend and Backend Frameworks

-Rares Petrisor-

# Topic of this research paper

This paper is created in order to assess what frameworks I should use for my IP project. This paper was written using the DOT framework, and it’s scope is for me to find out more about these frameworks and assess which one I should use for my project. The Library method will be used in order to conduct this research and achieve a conclusion, mainly the literature study method.

## The main question this research aims to answer:

## What is a Framework? How does it help in the process of development?

In software development, a framework is a piece of software that provides generic functionality that can be accessed with user-written code. It provides a way to build applications on top of it, while simplifying the development process by providing useful utility functions. This means that choosing to use a framework is really important since it simplifies the amount of work you have to do, by building on top of what other already have, while also providing a layer of organization and a way of working to the developer.

*This information was pulled from* [*https://en.wikipedia.org/wiki/Software\_framework*](https://en.wikipedia.org/wiki/Software_framework)*. This researching technique falls under the Library – Literature Study DOT research framework.*

The following subquestions will be supportive questions asked in order to provide an accurate answer for the main question this document aims to answer.

## Why is Javascript so popular nowadays with Web Development Frameworks?

There are 3 languages most Web Developers use in their day to day work: HTML, CSS and Javascript.

HTML is a Markdown language, providing you with the ability to place things on screen. CSS is a really powerful design language that allows you to choose and customize how you want everything to look.

And what can be debated as the most important out of these 3 is Javascript, a very powerful programming language that is used to write all the logic behind the websites.

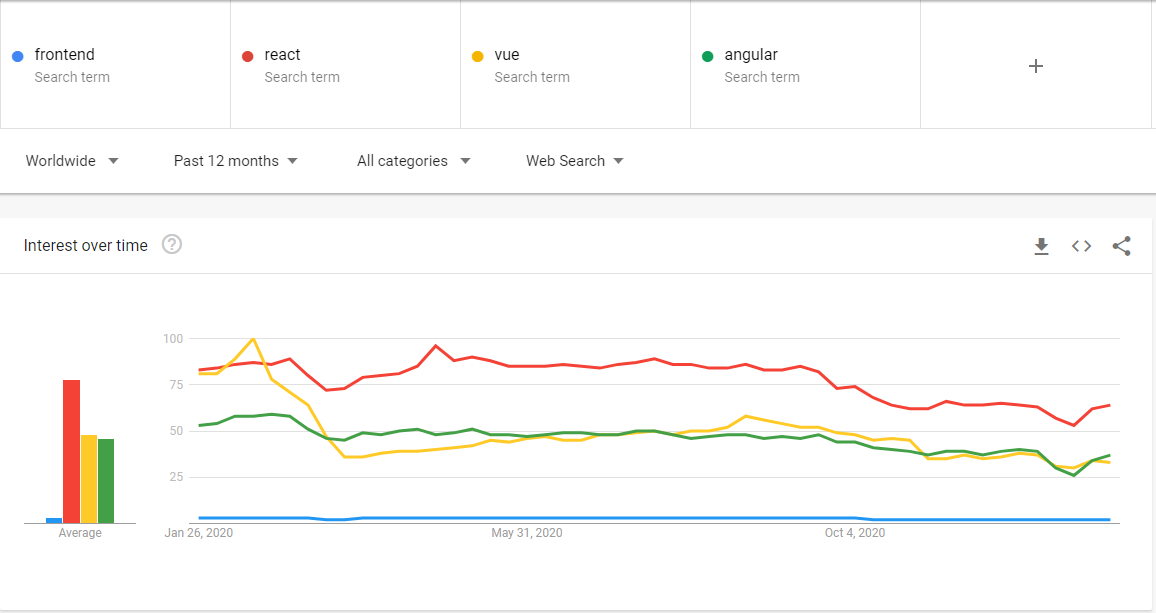
One of the most important things that makes Javascript special is it’s ability to support for both functional and object-oriented programming. Another interesting advantage Javascript has is that it natively runs on all platforms, mobiles devices but also PC. It can also run both client-sided and server-sided which makes it even more attractive to developers, no matter their experience with programming.

So it seems the answer is Javascript Frameworks are so popular because of the various advantages the language has and the flexibility and versatility it supports.

## What are the most popular Frontend Frameworks and which should I choose?

Sorting by job descriptions and what companies look most after,in the past 2 years, the 3 most popular frontend frameworks are : ReactJS, AngularJS and VueJS.

And this checks out in the community as well. These are the 3 most googled after frontend frameworks, as can be seen in the below figure.



I will now go over each of these 3 and mention some pros and cons for each of them.

*This information was pulled from* [*https://trends.google.com/trends/explore?q=frontend,react,vue,angular*](https://trends.google.com/trends/explore?q=frontend,react,vue,angular)

[*https://www.simform.com/best-frontend-frameworks/*](https://www.simform.com/best-frontend-frameworks/)

[*https://existek.com/blog/top-front-end-frameworks-2020/*](https://existek.com/blog/top-front-end-frameworks-2020/)

*This researching technique falls under the Library – Literature Study DOT research framework.*

**React**

Pros:

* Reusability of components
* Consistent and smooth performance (between mobile and PC)
* Easy to learn
* Most popular, so you can find more people in the community that have the same problems as you

Cons:

* Constant updates on the page
* JSX is harder to understand at first
* Bad state management

**Angular**

Pros:

* In-built 2 way data binding
* Less coding
* Decoupling of components
* Reusable components
* Pretty popular as well, lots of support here too

Cons:

* Harder to learn and understand fully
* Code structure and size scales harder with the size of the project

**Vue**

Pros:

* Good native documentation
* Simple syntax
* Flexibility to structure

Cons:

* Least popular out of these 3 so it’s harder to find community solutions to issues
* Language barrier with plugins and components

Now that we have the facts, I can justify why I have chosen React. The big community behind this framework allows for relatively easy solutions to common issues. It is also the most popular with companies so experience with it is even more valuable than with the others. It also provides reusable components and freedom to the way you structure things. This being said, I think ReactJS best fitted my needs for this project and for my programming knowledge from now onwards.

## What are the most popular Backend Frameworks and which should I choose?

Same criteria as for frontend can be applied to the backend frameworks, but here there is a twist. There are a lot of backend *languages* to choose from.

The five most popular backend frameworks, in the past 2 years, are : Django(Python), Laravel(PHP), .NET Core, ExpressJS(Javascript) and Spring(Java).

I will go over the pros and cons for these as well and then present my reasoning for my choice.

*This information was pulled from*

<https://morioh.com/p/d024b656ccc2>

[https://blog.back4app.com/backend-frameworks/](https://blog.back4app.com/backend-frameworks/%20)

<https://hackernoon.com/python-for-web-development-pros-and-cons-and-best-frameworks-et6d3z6h>

<https://stackshare.io/expressjs>

<https://rabbitstack.github.io/spring/spring-boot-or-not-to-spring-boot/>

*This researching technique falls under the Library – Literature Study DOT research framework.*

**Django**

Pros:

* Fast to develop, easy to use, low learning curve
* Lots of features
* High scalability
* Optimal security
* Versatile

Cons:

* High memory consumption
* Speed limitations
* Lack of true multiprocessing
* Design restrictions

**Laravel**

Pros:

* Easy to learn
* MVC framework
* Good ORM support
* Makes use of Artisan

Cons:

* Less inbuilt support than the others

**.Net Core**

Pros:

* Cross Platform
* Microsoft Support
* Simple maintenance
* Web Api support
* Scaling and Dockerization

Cons:

* Legacy
* High learning curve
* Only windows-based tools

**ExpressJS**

Pros:

* Scalability
* Same language as frontend framework
* Lots of community support
* Supports caching

Cons:

* Client request problems with several middleware
* Issues in callbacks

**Spring**

Pros:

* Simplified dependency manager
* Visibility into app internals

Cons:

* Bad started framework
* Hard to get a good grasp of
* Hard to get familiar with the Spring ecosystem

To summarize all of this, ExpressJS provides me with everything I need while also working within the same language as the frontend which is a big plus. Also, just like React, the ExpressJS community is quite large so lots of problems are easily solved.

## But what about the data?

For our final piece of research, I needed something to store data that supported an ORM design. Going with the same principles as before, the 5 most popular database choices are: Oracle, MySQL, Microsoft SQL Server, PostgreSQL and MongoDB. Here the decision was more subjective than anything and I chose MongoDB (implemented using mongoose) just because the other 4 all are using SQL and work in a more standardized way, whereas MongoDB used JSON objects to store it’s data and works directly with npm packages (mongoose) to interact with it directly with JS code.

# The results

Using the data I found, there are a lot of good choices for both Frontend and Backend Frameworks, each bringing something unique to the table. The diversity is so big that one could almost say the decision behind this choice is left to preference but that is not entirely true. There are some limitations frameworks have that you need to be aware of before deciding on which you want to stick with.

# Conclusion

To answer to initial question that brought the need for this research paper, after analyzing the data on these frameworks I have come to the conclusion that the best combination of frameworks for my project are ReactJS for frontend, because of its big community that can help solve common issues pretty easily, but also because of it being easy to learn while also providing reusable components. As for backend, the decision to choose ExpressJS was harder but in the end the big community and also the benefit of using the same language across frontend and backend sold it for me.

Literature References:

<https://morioh.com/p/d024b656ccc2>

[https://blog.back4app.com/backend-frameworks/](https://blog.back4app.com/backend-frameworks/%20)

<https://hackernoon.com/python-for-web-development-pros-and-cons-and-best-frameworks-et6d3z6h>

<https://stackshare.io/expressjs>

<https://rabbitstack.github.io/spring/spring-boot-or-not-to-spring-boot/>

<https://en.wikipedia.org/wiki/Software_framework>

<http://ictresearchmethods.nl/Methods>

<https://trends.google.com/trends/explore?q=frontend,react,vue,angular>

<https://www.simform.com/best-frontend-frameworks/>

<https://existek.com/blog/top-front-end-frameworks-2020/>