JavaScript Frameworks

A Comparison Of Titans

Philip Esmailzade
philipesmailzade@gmail.com

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

ABSTRACT

There are several JavaScript frameworks to choose from, but which one is the right one? This essay researches three of the biggest JavaScript frameworks right now; Angular, React and Vue.js. Focus is on analyzing the frameworks installation process, structure, syntax and documentation while answering questions with conclusions drawn from that information.

While Vue was declared best in performance, no framework came out as a clear winner. With different strengths and weaknesses in some situations, all three frameworks have their purpose in the JavaScript ecosystem. Depending on the users experience, time and needs a personal choice has to be made in the end.

CONTENTS

1 INTRODUCTION

Getting into JavaScript frameworks can be a mess. There is a whole jungle of frameworks to choose from, all with different strengths and weaknesses. Choosing the right one can be tricky, which is what this essay will help with. This essay's goal is to compare three popular frameworks that have distinguished themselves from the others; Angular, React and Vue.js.[1]

The essay will put these frameworks in the spotlight to figure out what makes them attractive to use. Every framework shares common categories which will be researched. These include:

- Background information
- The installation process
- The structure of the file system
- Languages and syntax used within the framework, focusing on components and the template views.
- Documentation

These points do a good job on giving a quick rundown on what the specific framework is all about in an objective and fair viewpoint. This information will be the basis of the conclusions drawn about the research questions provided in the next section.

[1] Github, Front-end JavaScript frameworks

2 RESEARCH QUESTIONS

- Are the frameworks easy to install?
- Do the frameworks have steep learning curves?
- What differentiates the frameworks apart?
- When should the frameworks be used?
- Are the frameworks trending up or downwards in popularity?

3 METHOD & LITERATURE REVIEW

This essay was written in a linear way. Research questions was the first thing defined. These questions laid the foundation of what the focus of this essay would be. Then the research began, starting with Angular, going over to React and finishing with Vue. The results were then drawn from the analysis section and written down with a summary put in the conclusion.

The sources used consist mostly of information retrieved from the three frameworks own websites which is then analyzed and results in drawn conclusions. These websites are the bread and butter of learning the framework, which is why I used them so heavily in this essay. The information I'm analyzing from these websites is mainly guides and tutorials. This means that there does not exist any subjective opinions, only objective facts about how the framework operates and is used. This makes it easy to write a text that accurately portraits a framework in its rightful state.

Apart from the homepage of the frameworks, statistics has been retrieved from Stack overflow and Github which accurately tells us trends in forms of issues and user satisfaction with the frameworks. I have also used sources that retrieves information directly from these resources.

Wikipedia was used in some cases to get background information about the author of the frameworks, who maintains it and when the framework was released.

4 ANALYSIS

4.1 Angular

4.1.1 Background

Angular is a framework designed to help developers create modern applications with complex requirements in simple ways. It is maintained by the Angular team at Google and released in September 2016 as a rewrite from the predecessor Angularjs written by the same team.[2] Angular applications structure consists of a MVC-like pattern where components act as the controller / model and templates as the view[3]. Angular is written in typescript which is a superset of JavaScript.

4.1.2 Installation

Installing Angular is a fast and efficient process. First of, Angular requires the user to have Node.js and npm installed. Angular offers a zip download but the easiest way is to clone angular from Git and install it with npm. Angular also offer a quick start seed for beginners which includes a playground to ease the user into learning the architecture. If the user were to use Angular in a professional environment, generating a project through the Angular command line tool would be preferable.

Installing the quick start seed is as simple as the following four commands:[4]

```
git clone https://github.com/angular/quickstart.git quickstart
cd quickstart
npm install
npm start
```

4.1.3 Structure of file system

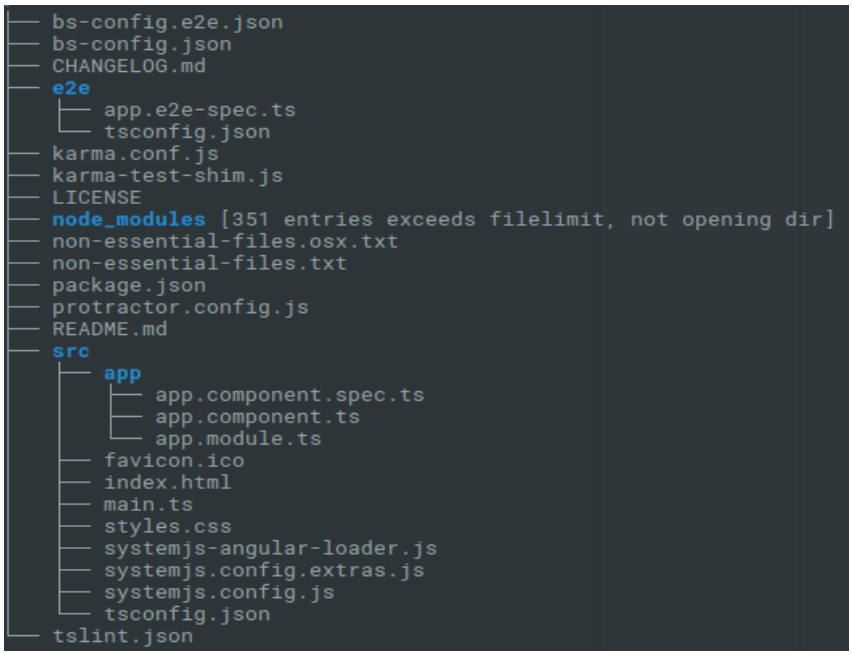
The structure made by the installation is simple at first glance. Angular installs a lot of files but most of them is safe to completely ignore by the user. The folders interesting to the user is the folder called src/ and e2e/.[5]

The e2e folder handles all the end to end tests written. It is not of interest to a beginner but may be visited frequently by the more experienced user.[5]

The src folder is where the source code of the project lays. This is where the files used for the application can be found. The root houses an index file and a bootstrap file. The app folder includes the components and views used to make things happen.[5]

---

2 Wikipedia, Angular (application platform)
3 Angular, Template Syntax
4 Angular, Setup for local development
5 Angular, Anatomy of the Setup Project



4.1.4 Syntax

Angular is unique in the way that the user is free to choose to write their code in either typescript or JavaScript. The former transpiles into the later at compilation which makes the choice entirely subjective. Typescript is a big reason to choose Angular. It is a good option when working in large groups because of its tooling. Typescript is efficient when it comes to autocompletion and refactoring which makes working in a group reliable. It may take a while to learn the language but the profit of writing a statically-typed language with the same pros as ES6 Javascript can be something to consider.[6]

The templates are an important part of Angular. They are written in HTML but with an "Angularized" twist. This is where the users get to see the changes of their application visually.

Interpolation is a common and simple way to display data in a template. The object property between the curly braces gets replaced by the actual value of the property in a string format.[7]

Interpolation is done with the double curly braces.[7]

 My name is {{person.Name}}

Angular also provides the functionality to use loops and conditional statements in markup which is a powerful tool in displaying specific data simple and efficient. Structural directives are distinguished with a *.

*ngIf is used to show or hide elements from the DOM.

<div *ngIf="showDiv">

*ngFor is used to iterate through objects and arrays.

<li *ngFor="let item of list">

[6] Victor Savkin, Angular: Why Typescript?, 2016 – 07 - 22
[7] Carlos Menezes, Angular 2 – A quick intro about template syntax, 2016 – 04 - 12

*ngSwitch is used to switch template depending on which statement being true.

```
<div [ngSwitch]=”menu”
        <div *ngSwitchWhen=”0”> </div>
        <div *ngSwitchWhen=”1”> </div>
</div>
```

These are all familiar directives to a programmer. The difference is that you now can use them in the context of a HTML template.

Event binding is a way to bind an event to a specific element which is a simple way to make applications functional and interactive. The syntax for event binding are distinguished with parenthesis followed by the statement. A package of events is bundled into the angular/common package which is easily imported.[7]

```
<div (click)=”makeSomethingHappen()”>
```

Property binding is distinguished by the square brackets []. Property binding is used to easily transfer data from the component to the class.[7]

```
<p [style.color]=”red”>Test</p>
```

4.1.5 Documentation

Angular’s homepage is an extensive resource for information. It provides a beginner’s guide to getting started with the framework and detailed tutorials explaining every step thoroughly. There is also sections about fundamentals in the framework that is good to learn when taking a deep dive into the framework as well as a section about certain techniques worth discussing such as safety and deployment.

The most important section of Angular’s website is the api section. This is where developers will spend most of their time searching for framework specific help. This includes docs about core packages included in Angular such as animations, routers, http and testing.

The footer of Angular’s page links to gitter and stackoverflow which houses impressive numbers. The gitter channel has over 13000[8] people joined and there are over 75000[9] questions tagged with Angular at Stackoverflow.

[8] Gitter, angular/angular
[9] Stackoverflow, Tagged questions Angular

4.2 React

4.2.1 Background

React is a javascript framework designed to allow users to build user interfaces capable of handling data with speed, efficiency and scalability. The applications created are dynamically loaded without having to refresh the page. It was initially released in March 2013 and is maintained by primarily Facebook and Instagram.[10]

Compared to other javascript frameworks, React does not adhere to the MVC pattern. Instead, React is focused on the view in MVC. The core purpose is to create reusable interface components to enhance and help create powerful views , much like directives in Angular.[11]

4.2.2 Installation

Installing React can either be smooth as sail or a hassle depending on what the user wants out of it. The “Create React App” is the fastest and easiest way to get right into developing applications. The create-react-app is basically a module used to create react projects without having to configure build options like babel and webpack. If the user is starting from scratch, the Create React App is the best way to go. Node.js and npm is a requirement.

Four commands is all it takes to get started:[12]

```
npm install -g create-react-app
create-react-app my-app

cd my-app
npm start
```

The more troublesome way of installing React is when trying to implement React into an already existing project. This is a process which requires a package manager like npm to install React, a bundler to optimize the apps performance and a compiler to enable JSX and ES6 features in your project.[12]

4.2.3 Structure of filesystem

Create React Apps file structure is impressively minimal. Index.html and index.js are the only two files that cannot be deleted or renamed. The former contains the template view and the latter contains the javascript entry point.[13]

---

10 Wikipedia, React(JavaScript library)
11 Eric Simons, What exactly is React?, October 2017
12 React, Installation
13 Github, create-react-app README.md

```
my-app/
  README.md
  node_modules/
  package.json
  public/
    index.html
    favicon.ico
  src/
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
```

4.2.4 Syntax

React uses an extension to javascript called JSX. With JSX, users can define react elements for rendering to the scene with a syntax similar to html.

```
const element = <h1>Hello, world!</h1>;
```

Similar to interpolation in Angular, users can embedd javascript directly in JSX with the help of curly braces.[14]

React elements function the same as a regular JavaScript objects. They work similar when used in conjunction with conditional statements and loops which means being able to be returned, used as an argument and assigned to a variable.[12]

Components in React are created by defining a function. The only rules are that the function must have an input as argument and return a React element. The component can then be used in JSX by writing the component name in the same syntax as an html tag.

```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}

const element = <Welcome name="Sara" />;
```

Event binding is done similarly to HTML and Angular. The event followed by the statement that the user wants to execute.[15]

```
<button onClick={activateLasers}>
```

Overall, using React with JSX provides an easier experience and learning curve to the framework. Programmers with JavaScript backgrounds will have no problem to get into React and its syntax. The fact that JSX is optional and can be expressed in regular JavaScript and newer ES6 features as well makes the framework attractive to all kinds of developers.

4.2.5 Documentation

The React homepage is the first place a beginner should go to learn all about the framework. It houses a quick start section with guides ranging from installation help, both beginner and advanced techniques about the framework as well as how to get into the mindset of building apps with React. The page also has an entire tutorial where users can follow along a guide

14 React, Introducing JSX
15 React, Handling Events

explaining how to create a simple tic tac toe game. There is also an api reference section which is handy to fully immerse yourself in the frameworks details.

There is a discussion forum available for conversation around best practices, versions and the like. Asking for help with code is mainly done on Stackoverflow though, which currently houses over 60000[16] questions asked.

4.3 Vue.js

4.3.1 Background

Vue.js is a front end JavaScript framework released by Evan You in February 2014.[17] Vue.js was designed to be a lightweight framework focused on the view part of an application that could be integrated into the system incrementally.[18] Vue.js is similar to React more so than Angular because of this.

Vue.js has gained traction the past years because of its ability to be powerful while staying lightweight and simple. No need to learn typescript or JSX, plain JavaScript is all you need.

4.3.2 Installation

Installing vue.js could not be easier if the goal is to get up and running immediately. Vue provides either a downloadable version or a CDN that the user can include easy and fast in their index file.[19]

<script src="https://cdn.jsdelivr.net/npm/vue"></script>

Vue also offers a CLI tool to install a production ready build used for bigger scale projects. It takes a little more time setup but offers a wide array of builds that may suit your application.[19]

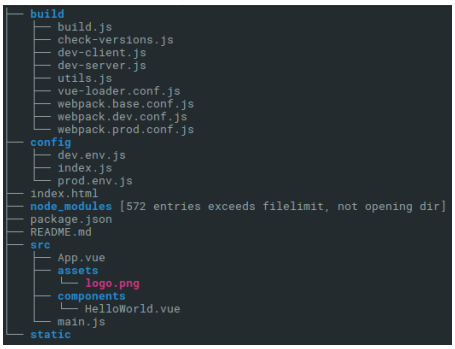
```
$ npm install --global vue-cli
$ vue init webpack my-project
$ cd my-project
$ npm install
$ npm run dev
```

4.3.3 Structure of filesystem

As said in the installation section, Vue can be either installed or just imported with a script tag from a CDN. This means that Vue is flexible in the projects it is implemented. Vue can work in a project with a single file, which makes the file structure hard to analyze. The file structure when installing a bare bone project with the cli is similar to other frameworks. A big folder with dependencies, a config folder and of course the src/ folder. We can see the simplicity in the src/ folder with only an App.vue file and a HelloWorld.vue component.

---

16 Stackoverflow, Tagged questions reactjs
17 Wikipedia, Vue.js
18 Vue.js, Introduction
19 Vue.js, Installation



4.3.4 Syntax

Vue does not force you to use any JavaScript extensions or the like when writing code. Vue supports JSX and even typescript but it is in no way required to use when utilizing Vue. This makes development in Vue familiar due to the basic knowledge needed to create something. It also makes it easier for already existing projects to implement Vue in its ecosystem without having to write additional code to accommodate.[20]

Template syntax is crucial in a framework like Vue and thankfully it is very easy to pick up. Interpolation is done like the other two mentioned frameworks by having {{ }} curly braces with for example a property in it. Interpolation in Vue supports JavaScript expressions as well.

Message: {{ msg }}

Directives are used to manipulate the DOM. Conditional statements can be done with the "v-if" directive and for loops with "v-for".[21]

Now you see me [21]

<li v-for="item in items">[21]

Directives also includes event binding with examples like bind and on. These types of directives can take arguments to specify what is going to happen. [21]

<a v-bind:href="url"> ...

<a v-on:click="doSomething"> ...

An important factor why to use Vue is components. Like React, components can be created to reuse code in a simple and reusable way. Scalability is a factor that makes these frameworks relevant, and components are the key factor to scale. [20]

```
Vue.component('my-component', {
  template: '<div>A custom component!</div>'
})


<div id="example">
```

[20] Vue.js, Comparison with other frameworks
[21] Vue.js, Template syntax

```
  <my-component></my-component>
</div>
```

Created components are used by custom tags created by Vue. This is an easy concept to grasp and makes the learning curve significantly easier.

4.3.5 Documentation

Vue.js provides an extensive website in terms of both installation, introduction and advanced techniques for the framework. Angular and React both had educational pages and Vue is no exception. The introduction is well written and easy to follow with code examples for everything. The api page is on par with other frameworks and there is also small example programs for the user to learn from. What differentiates Vue's website is the comprehensive style guide.
The style guide is a best practice guide on how to write code with Vue. They use a ranking from A to D to explain the importance of using said practice. Vue also houses over 10000[22] questions asked on Stackoverflow.

[22] Stackoverflow, tagged questions vuejs

5 RESULTS

Are the frameworks easy to install?

It depends on what the framework is going to be used for. All three frameworks are easy to install if the user only wants a quick start. To get a production ready development project up and running, it gets harder. Vue.js is the easiest to quick start because of the simple script tag users can include with a link to the CDN[19]. React and Angular is not that far behind though with React having the Create-React-App[12] and angular with its 4-command installation[4]. Angular wins when having to setup a professional development build because you already get everything with the normal installation. React and Vue.js are trickier due to the time it takes to enable ES6 features with Babel and Webpack / Browserify to bundle code together.

Do the frameworks have steep learning curves?

Angular is the hardest of the three to learn. Angular is a framework working with a MVC like structure using typescript. This means that everything from components, directives and modules are going to be written in typescript.[6] It is possible to write everything in JavaScript, but it is not recommended and would defeat the purpose of using Angular. The learning curve of Angular includes learning Typescript, getting familiar with the framework which includes a heavy weight library compared to React and Vue.js and not being able to incrementally implement it to existing projects.

React only focus on the view part of MVC[11] which makes it easier to implement into existing projects. JSX is an extension to JavaScript that is used to make it easier for developers to write and read code intended for React. It has similarities to markup language which makes it easy to learn. [14]

Vue.js has the flattest learning curve of the three. It supports typescript and JSX but is not the default language to code in.[14] All you need in Vue.js is basic understanding of JavaScript. It is the lightest framework of the three[14] and also the easiest to pick up due to the limited knowledge you need. Plain JavaScript is used to create reusable components to be used in the template that houses some framework specific syntax.

The template syntax of the three frameworks are similar with interpolation, event binding and the like working in the same way but is written with slight differences.

What differentiates the frameworks apart?

Due to Angular, React and Vue all being JavaScript frameworks there tends to be a lot of similarities between them. Especially React and Vue.js share functionality and mindsets together, for example a virtual DOM and focusing on the core library.[23] Therefor the differences that can make or break the decision of using a framework are small.

Angular is a complete package right out of the box. Installation is straight forward. Common libraries needed are already bundled in the installation. Typescript is a powerful language with Angular. The style guide on how to build apps is straight forward and structured. Developing in Angular is like coding on rails once you get the hang of it (Which takes time), and whether you like it or not is completely subjective.

React and Vue on the other hand are more flexible in terms of what packages to install and how to approach the structure of developing an app. They are inherently also easier to pick up and create simple things due to how light weight they are. Vue can brag about performing better than both React and Angular[24] as well as having the flattest learning curve of them. The way the template is written in React and Vue is one of the small factors that can decide which framework to choose. React uses JSX while Vue uses regular HTML markup with framework specific syntax mixed in. JSX affects the learning curve of React which widens the gap between the simplicity of React and Vue.

When should the frameworks be used?

What is implied in the results of the previous question is that the user ultimately must decide what framework to use depending on the circumstances of the application in mind and to what degree of experience the user has. The learning curve goes from Angular having the steepest and Vue having the flattest with React somewhere near Vue in between.

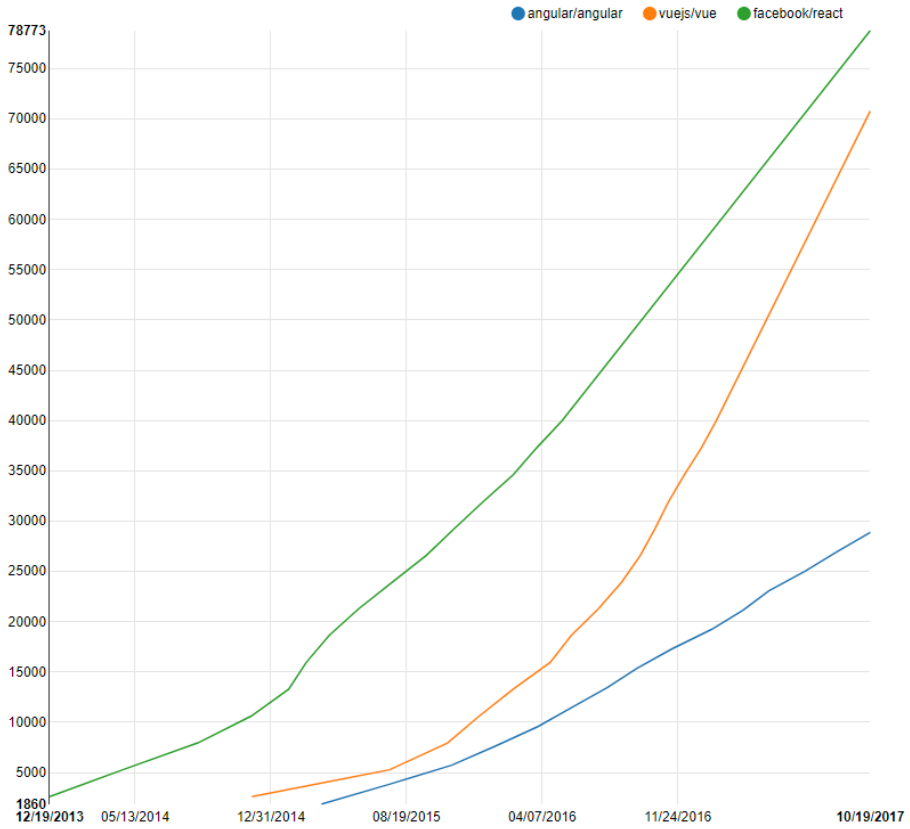
A complete beginner creating an app would be having more problems with Angular than the other two, while a team of experienced developers with a background in object oriented programming certainly would welcome Angular opposed to the others.

---

[23] Vue.js Comparison with other frameworks

[24] Stefan Krause, Results for js web frameworks benchmark round 4

Are the frameworks trending up or downwards in popularity?



If we measure popularity purely by stars given on Github we can see that React(78773 stars) leads by a small margin next to Vue(70000~ stars) while Angular(28000~ stars) drags its feet behind the other two. The numbers are deceiving towards Angular whose previous version has its separate repository with 57000~ stars. The low numbers are justified though with Angular having 2120 issues open compared to Reacts 393 and Vue's 63. [25]

All of the three frameworks are booming in the industry right now and are clearly on the rise. Vue with the steepest curve of the three in terms of Github stars is the competitor best suited to fight against React and Angular. Other than trends and general user satisfaction analysis it is impossible to know which one will be the safest to bet on in the future.

[25] Npmtrends, react vs vue vs @angular/core

6 CONCLUSION

Angular, React and Vue are three frameworks booming in the JavaScript world. They each offer something unique and appeal to different demographics. The trend upwards does not seem to stop for a while and there has not been a better time as now to pick one up. Choosing one is a hard process depending on what the user wants to accomplish and ultimately it comes down preference.

Vue wins the raw performance war as well as being the lightest of them all out of the box. With that said, Angular and React is not far from Vue. Performance tests should not deter users when picking one of these frameworks. What is more important is to think about what framework fits your idea. Do you have the time to learn both a new language and framework? Are extra libraries needed or is basic core functionality enough for the project? Is the framework going to be applied to an existing or a new project? Those are only a few of the questions you need to answer to make a calculated decision on what framework to use.

Angular is the framework users will appreciate the most if their background comes from object oriented programming because of typescript. It is also the framework with the strictest style guide on how to do things. This also means that documentation is extensive which helps new users getting into the framework easier.

React and Vue are both lighter than Angular but offers more flexibility in how apps are written, what languages to use and in implementing the framework in existing projects. The important distinction between React and Vue is syntax and template coding. Writing templates in JSX(React) or markup(Vue) is personal preference and it is recommended to research to the depth to know what fits you the best. Vue is perfect for the beginner developer who only has basic knowledge in JavaScript and markup language while React is somewhere between Angular and Vue in terms of the learning curve because of having to pick up JSX in order to be proficient.

At the end of the day, which one you pick is of lesser significance. All three of them are excellent choices and the best one will be the one that lets you express yourself naturally in code and enjoy yourself.

7 FUTURE WORK

This essay is a surface scratching study on three different frameworks to help the reader understand what they are about and the difference between them. Future work would include a more in depth look into each framework spending more time researching performance, testing and statistics. It would appeal more to an advanced user who already knows about the frameworks but wants to know more about the inner structure, design philosophies and demographic it adheres to.

REFERENCES

Angular

1. Github, “Front-end JavaScript frameworks” Available at: https://github.com/collections/front-end-javascript-frameworks [Accessed: 2017-11-09]

2. Wikipedia,"Angular (Application platform)" Available at: https://en.wikipedia.org/wiki/Angular_(application_platform) [Accessed: 2017-10-07]

3. Angular,"Template syntax". Available at: https://angular.io/guide/template-syntax [Accessed: 2017 – 10 - 07]

4. Angular,"Setup for local develop". Available at: https://angular.io/setup [Accessed: 2017 – 10 - 07]

5. Angular,"Anatomy of the setup project". Available at: https://angular.io/guide/setup-systemjs-anatomy [Accessed: 2017 – 10 - 07]

6. Savkin, Victor. 2016-07-22. "Why typescript?" Available at: https://vsavkin.com/writing-angular-2-in-typescript-1fa77c78d8e8 [Accessed: 2017-10-07]

7. Menezes, Carlos. 2016 – 04 – 12 "Angular 2 – A quick intro about template syntax" Available at: https://medium.com/front-end-hacking/angular-2-a-quick-intro-about-template-syntax-121f9b160a64 [Accessed: 2017-10-07]

8. Gitter, "angular/angular" Available at: https://gitter.im/angular/angular [Accessed: 2017-10-07]

9. Stackoverflow,"Tagged questions" Available at: https://stackoverflow.com/questions/tagged/angular [Accessed: 2017-10-07]

Angular, "Architecture Overview". Available at: https://angular.io/guide/architecture [Accessed: 2017 – 10 - 07]

Angular,"What is Angular?". Available at: https://angular.io/docs [Accessed: 2017 – 10 - 07]

Angular,"Cheat sheet". Available at: https://angular.io/guide/cheatshee [Accessed: 2017 – 10 - 07]

Angular/angular, "CLI tool for angular". Available at: https://github.com/angular/angular-cli [Accessed: 2017 – 10 - 07]

React

10. Wikipedia,"React (Javascript library)" Available at: https://en.wikipedia.org/wiki/React_(JavaScript_library) [Accessed: 2017-10-15]

11. Simons, Eric. October 2017. "What exactly is React?" Available at: https://thinkster.io/tutorials/what-exactly-is-react [Accessed: 2017-10-15]

12. React, "Installation" Available at: https://reactjs.org/docs/installation.html [Accessed: 2017-10-15]

14. React, "Introducing JSX" Available at: https://reactjs.org/docs/introducing-jsx.htm [Accessed: 2017-10-15]

15. React, "Handling Events" Available at: https://reactjs.org/docs/handling-events.html [Accessed: 2017-10-15]

16. Stackoverflow,"Tagged questions" Available at: https://stackoverflow.com/questions/tagged/reactjs [Accessed: 2017-10-15]

Vue.js

17. Wikipedia, "Vue.js" https://en.wikipedia.org/wiki/Vue.js [Accessed: 2017-10-17]

18. Vue.js, "Introduction" https://vuejs.org/v2/guide/ [Accessed: 2017-10-17]

19. Vue.js, "Installation" https://vuejs.org/v2/guide/installation.html [Accessed: 2017-10-17]

20. Vue.js, "Comparison with other frameworks" https://vuejs.org/v2/guide/comparison.html [Accessed: 2017-11-09]

21. Vue.js, "Template syntax" https://vuejs.org/v2/guide/syntax.html [Accessed: 2017-10-17]

22. Stackoverflow, "Tagged questions" https://stackoverflow.com/questions/tagged/vue.js [Accessed: 2017-10-17]

Vue.js, "List rendering" https://vuejs.org/v2/guide/list.html [Accessed: 2017-10-17]

Others

24. Krause,Stefan."Results for js web frameworks benchmark – round 4" Available at: http://stefankrause.net/js-frameworks-benchmark4/webdriver-ts/table.html [Accessed: 2017-10-17]

25. Npmtrends, "react vs vue vs @angular/core" Available at: http://www.npmtrends.com/react-vs-vue-vs-@angular/core [Accessed: 2017-10-17]

Qian,Tim."StarHistory" Available at: http://www.timqian.com/star-history/#facebook/react&angular/angular&vuejs/vue [Accessed: 2017-10-17]