TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Undoubtedly the use of the internet is growing day by day because it is available for cheaper these days and easily available. With the increase in the use of the internet the importance and use of websites have also increased. *(Khurana and Kumar, 2019)* Smallest to smallest information is also available on the internet and websites are the only sources of that information. Organizations and companies are hugely relied on the web to deliver their products and services to the customers and they have succeeded. *(Lawrence, 2017)* Not just this but the technologies available for building the websites are also evolving. The technologies used for building websites include HTML, JavaScript, CSS, etc. But to full fill, the current market needs the traditional approach to build websites that are not considered good enough in terms of its efficiency, performance, etc*(Mesbah and Deursen, 2007; Wang et al., 2008)*. So to overcome this the new approach of the web has evolved. This approach is called Single Page Applications*(Molin, 2016)*. Using this approach building websites has become much easier not just that but websites have become more responsive and efficient and resource savings.

1.1 Conventional Approach

As discussed above the websites built in older times were not so advanced. Most of the websites were built using server-side rendering technology. In the server-side rendering, all the data that needs to be shown on the browser was already fed to the HTML page and that page is sent to the client-side. So all the content will be visible in one go at the browser. So if the size of the webpage is large it would take significant time on the browser to display the content. Also if one has to update only the specific part on the webpage, the whole page has to reload, which means all the data will again be fetched from the server and displayed on the browser. But it has a major disadvantage that even the updated data will also be downloaded again with the static data and that is wastage of resources. Static data like headers, footers logo, etc doesn't need to be downloaded again. This increases network latency and interfaces complexity. This pattern is also called the request-wait pattern. *(Mesbah and Deursen, 2007; Wang et al., 2008)*

Also, while writing the code, code repetition should be avoided as much as possible. But in the older approach, the content which should be displayed on all the pages has to be written separately which is bad practice. *(Moreno and Robles, 2015)* To overcome issues of Reloading the entire page on every interaction, wastage of resources, increased loading time, responsiveness, catching capabilities, debugging in the browser, etc. new technologies were introduced called a single page application SPA.

1.2 Single Page Applications & AJAX

Single-page applications are composed of several different independent components that can be updated or replaced dynamically so that the enter web page is not required to reload on every interaction of the user with the webpage. Sing page application runs inside the browser instead of fetching data every time from the server. Because of this, the websites have become more interactive and dynamics and innovations are being added day by day. Also these days the user-friendly ness of websites cannot be ignored. *(Davila and Navon, 2015; Voutilainen, 2017)*

Single-page applications are heavily dependent on the ajax. Ajax is Asynchronous JavaScript And XML. As the core technology of the web, AJAX has got more and more attention. Because as mentioned earlier the quality of user interactivity cannot be ignored in modern-day websites so the rich client-side technology called ajax evolved. AJAX is nothing new but just a new way of using the old standards. All the work of fetching data from the server in runtime and reducing the server side and client side time is done by the ajax and it helps to improve the user

experience. Traditionally XML was the choice of data format for the data to be transferred but Nowadays JSON is chosen. *(Smith, no date; Lin et al., 2012; Jadhav et al., 2015)*

1.3 JavaScript Frameworks

The existence of javascript frameworks is since 1995, and it is one of the most widely known and used programming language for front end web development. Still, JavaScript is growing and many new features are available now to enhance the user experience. With the evolution of Javascript language for web development, many JavaScript frameworks are also coming out to support web development. Frameworks are nothing but a set predefined code and functionality which user can use and enhance them according to their needs. Writing everything from a scratch is time-consuming therefore the use of frameworks comes in handy. Also, managing and maintaining code is as important as developing it. So architecture provided by frameworks is also important in this place. A typical JSF will provide functionality like DOM traversal and manipulation, AJAX manipulation managing layout and inserting effects, MVC architecture, URL redirecting, Routing, Stage management, Session Management, etc.

1.4 Rationale

As discussed above to improve the web development to the next level there is a jungle of JavaScript frameworks are available. Sometimes developers may choose more than one framework to fulfill the requirement of applications, so there is always a conflict of which framework to choose. *(Graziotin and Abrahamsson, 2013)* Also, it's not just about choosing the right framework but there are so many performance factors to be considered while choosing the framework. One of the important aspects of this framework is its performance in the browser. There are many more factors that may be considered while choosing the frameworks it includes factors like maintainability, validity, Lines of code, Active community. Therefore this research work will be done to compare the latest Single Page Application frameworks and how they perform in the browser. So based on performance and need developers can choose the framework. This work will mostly try to evaluate the browser-specific performance of the number of different Javascript frameworks. The chosen JavaScript frameworks and performance parameters will be discussed in-depth in the next chapters.

1.5 Research Question

This research will mostly try to evaluate the browser-specific performance of javascript frameworks on the browser. Because if your browser is taking more than 3 seconds to show a particular website then there are high chances of users navigating to some other website. So the websites display speed is one of the important factors. So based on the above criteria following research questions have been formed.

1. How to evaluate the JavaScript frameworks for Single Page Applications (SPA)?

1.6 My Interest In The Topic

Having worked as a full-stack developer for more than two years, I realized the importance of frameworks and how they can play a major role in developing any kind of software. Not just in JavaScript but any other language like Java, PHP, Python, etc.

So the main hypothesis to be tested is, using various javascript frameworks can we build a robust, scalable single-page web application. Also how the performance of this application differs in the browser in the same environment. The amount of code required to bootstrap the application so it can be maintained in the future will be tested.

The reason for choosing the above topic is because I am very much interested in software development and I would like to pursue this field as my career. Therefore, I wanted to test the different frameworks as I have some experience working with them. Also in my taught masters, I had the subject related to the Front-End web development and Android development where I have worked with the JavaScript, JQuery, Angular 8, and Ionic framework using Angular 8 and hence I wanted to focus on this specific subject area.

1.7 Roadmap for the dissertation

This dissertation is carried out in the following steps.

- Chapter 2 the details about the previous research have been discussed. Their methodologies, frameworks used by them, and metrics used by them are discussed.
- The next chapter is the methodology, in this chapter how this research will be carried out, which frameworks will be chosen what metrics will be chosen is discussed.
- The next chapter will be the development and results. Here the development of the application using different frameworks will be discussed and after the development, their performance will be computed. After the development,
- the next chapter is evaluation and discussion where all the findings and results will be discussed.
- And the last chapter will be the conclusion and future scope.

CHP 1. INTRODUCTION	• This chapter introduces the Background infromation, Need for research and hypothesis to be tested.
CHP 2. LITERATURE REVIEW	• This chapter discusses about the previous works and their findings
CHP 3 . METHODOLOGY	• In this section how this dissertation will be carried out is discusses
CHP 4. DEVELOPMENTS & FINDINGS	• In this particular chapter the development of software will be done and results will be finded.
CHP 5 . DISCUSSION & EVALUATION	• All the findings and results will be discussed here
CHP 6. CONCLUSION & FUTURE SCOPE	• Consclusion and future work will be mentioned in this chapter.

CHAPTER 2: LITERATURE REVIEW & RELATED WORK

This work is all about comparing the different latest javascript frameworks and evaluating them, which is nothing but benchmarking. So this chapter will discuss the background information on what is benchmarking and its importance in the field of technology, What are the frameworks and their importance

2.1 Benchmarking

According to the various sources, there are different ways of defining the benchmarks but more or less what is mean is the same. Here are some definitions of benchmarking by various sources, the International Organization for Standardization and the International Electrotechnical Commission has defined benchmark as *"A standard against which results can be measured or assessed"* in the same way IEEE defines benchmarking as *"A standard against which measurements or comparison can be made"* According to to the Bouckaert, Philips & Wallander the computer benchmarking can be defined as computer benchmarking is "*The act of measuring and evaluating computational performance, networking, protocols, device, and networks, under reference conditions, relative to a reference evaluation*" *(Bouckaert et al., 2011).*

Now that definition of benchmarking is clear, let's see what are the computer benchmarking and their types and understand their importance.

Benchmarking is not a new concept. Benchmarking has been around for more than decades. It's being used for comparing various platforms, different tools, different technologies, and their performances. It finds out differences between different components of the same families and tries to evaluate them. *(Lawrence, 2017).* Benchmarking tools are nothing but the solutions which are developed to automate the process to evaluate the different metrics of different application in the customer environment. Benchmarking comes in handy for the organizations or individuals while choosing a particular tool or technology for their own needs, so while choosing that they can compare the different aspects of solutions instead of choosing one solution every time thinking of it as an only correct solution. Benchmarking can also be helpful for the ones who are developing a new solution, so they can compare their result with the old solutions and get an idea of their solution to improving so overall it will help in standardization. *(Ratanaworabhan, Livshits and Zorn, 2010).*

There are two major categories of benchmarks i.e micro-benchmarks & macro benchmark. Using micro-benchmarks a very small part of the portion of the application is evaluated or analyzed on the other hand in the macro benchmark are designed to analyze the large and complex system on a bigger scale. The type of benchmarking used for this work will the microbenchmark because we will be evaluating a few parameters of an entire web application. *(Seltzer, Krinsky, Smith, Xiaolan Zhang, Harvard Uni).* Now that we have a sufficient understanding of what benchmarking is and what is its importance. In the next section, let's understand what are Javascript frameworks and its importance.

2.2 Frameworks & Javascript Frameworks

In the introduction section, the concept of frameworks and javascript frameworks is discussed briefly. But in this section, we will try to get into a deeper level and try to understand what are they and their importance.

2.2.1 Frameworks

Frameworks can be defined in many different ways. According to Ralph E. Johnson, *"Framework is nothing but a reusable system build to use in all part of software which is represented by*

predefined abstract classes and the way their instances interact"(Johnson, 1997). Further, he gives the second definition as *"Framework is a skeleton of software which can be modified the way developers want to for satisfying the needs according to the software" (Johnson, 1997).* These two definitions are not different, there phrasing is different but what they mean is the same. The first definitions talk about the way it works while the second one talks about its structural aspect. Moreover, both definitions simplify the difficulty of defining frameworks. *(Johnson, 1997).* Therefore it can be said that frameworks are build to allow developers to solve the problems that frameworks are capable of. *(Schmidt and Buschmann, 2003).*

We can all agree that how technologies have become competitive & challenging over the years. Therefore, there are certain characteristics that any tool or frameworks should possess to sustain in this highly competitive market. These characteristics are as follows

1. **Affordability:** Affordability states that the total ownership costs of software acquisition and should not be very high.
2. **Extensibility:** It should be compatible with the new updated to address new requirements so it will be adopted in the market even for new requirements.
3. **Flexibility:** Should support the emerging technologies
4. **Portability:** It should not be bounded to a specific environment like OS. It should be multipurpose and platform or environment independent.
5. **Reliability:** To make sure that built applications are robust and tolerant of future faults.
6. **Scalability:** The application should be able to handle a large number of requests or clients simultaneously.
7. **Trustworthiness:** To ensure integrity confidentiality and availability in distributes systems. *(Schmidt and Buschmann, 2003)*

2.2.2 JavaScript, JavaScript Frameworks & SPA frameworks

JavaScript was designed by Brendan Eich at Netscape. It is an object-oriented programming language focused on nonprogrammers to extend the support for client-side code execution. It does not concept of classes and does not support encapsulation. It does not even have structured programming like other programming languages like JAVA, C, C++. JavaScript believes in flexibility. No one can deny the success of JavaScript, if we talk about the numbers 97 out of 100 websites use JavaScript as their client-side scripting. Sometimes this language is also referred to as general-purpose programming language. *(Richards et al., 2010).* Initially, it was named LiveScript but later it got renamed to JavaScript fun sun and Netscape started shipping JavaScript with the Netscape browser in Dec 1995. JAVA was the only language that used to run in the browser, but it was very heavy to execute, so JavaScript came as an alternative. Also, it was targeted for the less experienced developers. *(Voutilainen, 2017).* JavaScript may look like other languages by syntactically for eg C, C++, or JAVA but it is a loosely typed language. For example, while defining any variable in javascript you don't have to type the data type like Integer or String you can just type 'var' and there you go, your variable is defined.

The popularity of JavaScript has grown over the years with the development of the web. The scripting of webpages has become more complex due to the evolution of a technology called AJAX as the webpages have become more complex, unlike static pages in older times. Web giants such as Amazon, Facebook, Gmail contains a significant amount of JavaScript code. Web Apps have become more popular because they don't need any additional information software mechanism they are OS independent i.e they can work on any platform like Windows, Android, Linux all they need is one browser with an active internet connection. *(Ratanaworabhan, Livshits and Zorn, 2010).*

When ajax evolved and with the help of jQuery it became easy to update certain parts of the webpage dynamically. Because of jQuery, it became easy to manipulate the DOM and update the data fetched from the server. These pages were interactive but it is nothing like a single page application that exists today. In jQuery, one has to find every element on the DOM to manipulate it with the help of either CSS class or with the element id or with the element name. It was not that efficient. *(Voutilainen, 2017)*

Therefore, the concept of client-side javascript frameworks became popular around 2009. JSF helps in binding the HTML page with the javascript code with the help of data binding so we don't have to do this explicitly as we do in jQuery. Also, JSF's make the processing of fetching data from the server and updating it in DOM a lot easier. It takes care of all the routing needed in the application. Also, it helps in managing the code structure, plus the separation of codes this all makes the frameworks very important. These frameworks are called SPA frameworks. The definition of SPA is already mentioned in the introduction section here we will see what are the basic attributes of the SPA's.

1. **Components**: The page to be displayed is divided into different parts while developing called components, but while rendering it shows all as one page.
2. **Web interface**: The interaction between a user and a web server.
3. **Update**: It is possible to update, delete, or replace one component with another component dynamically.
4. **User action**: User can interact with the page by doing any action with the use of an input-output device, for some action to take place.

2.3 Previous work

Few attempts have been made to evaluate the single page application frameworks in the past. Every work has different ways and metrics to be evaluated. The choice of framework is also different from work to work. This work is the progression of past works.

One of the most recent studies done by *(Lawrence, 2017)* for evaluation of the single page application frameworks. For his work, he has chosen a reference TODO MVC application developed and maintained by Addy Osmani & Sindre Sorhus called TodoMvc. This application has been developed in every latest JavaScript from that time (2017). According to him, any other application could be used but that particular application is maintained by the expert developers so that the app has been chosen. Also, different applications may yield different results. Todo application has an **input field** to add the items in the list, there is also functionality to mark items as completed or uncompleted. Then there is a **task list** which is shown on the webpage each item is editable as well. It also has the **footer section** where items can be sorted based on their status like completed, uncompleted, or all, etc. Frameworks selected by him are BackboneJS, ReactJS & AngularJS.

To perform the benchmarking test there is 3 task to be executed. In the first task, 100 items will be added to the todo list, then the second task is to mark all added task as completed task in which all the 100 tasks added in the previous step will be marked as completed and in the final task, all the completed task will be removed. After the completion of all the tasks, the benchmark application will show the visual report of the performances of all the selected frameworks in this case BackboneJS, ReactJS & AngularJS.

The figure shown below depicts the performance results generated by the Todo MVC application.

All the tests were executed 25 times in his work to ensure the validity of work in three different browsers i.e Google Chrome, Mozilla Firefox & Microsoft Edge.

After running tests in the chrome browser, the BackboneJS showed the least amount of execution time of 157 ms while React-JSX took the maximum time for execution i.e 904 ms. Angularjs and React-no-JSX take 346 ms and 554 ms respectively.

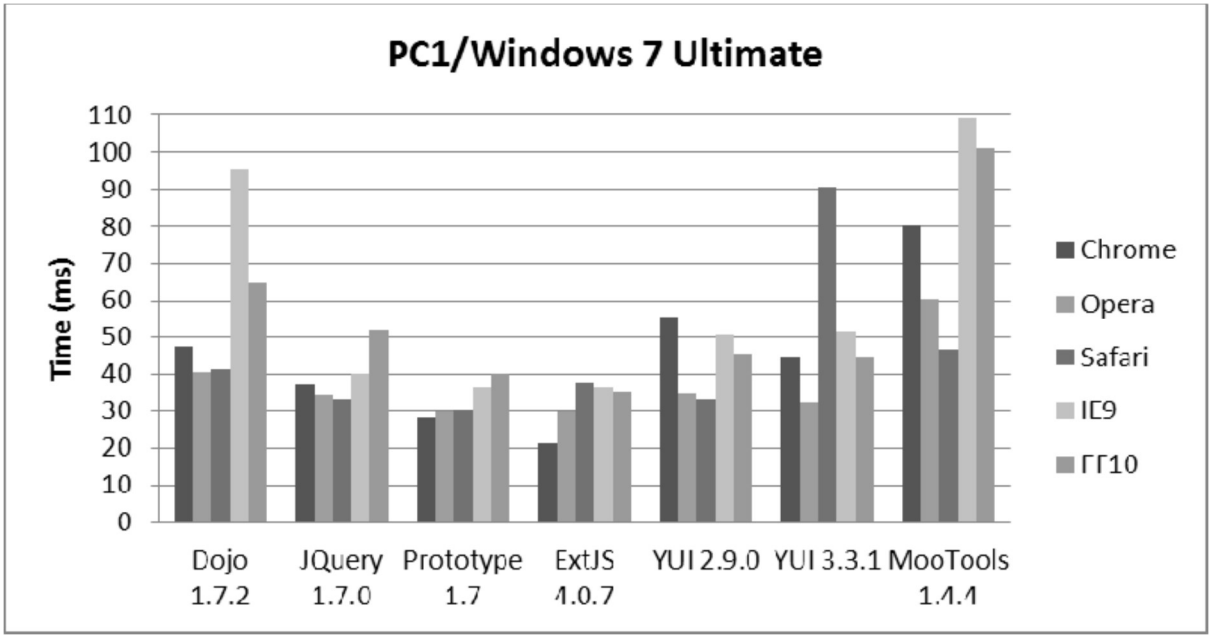
Moreover, in the Microsoft Edge browser, we can see the significant rise in time take for execution for all the frameworks. But here as well BackboneJS outperformed any other framework. BackboneJS took 233 ms and React-JSX took 1953 ms. much higher

And the same for the Mozilla browser.BackboneJS wins here as well with 266 ms. So Overall he concluded that BackboneJS is the most efficient framework in terms of the execution while React-JSX takes much more time for task execution comparatively. Also, all the frameworks have more execution time in the Microsoft Edge browser compared to the other browser. *(Lawrence, 2017)* .

A very similar approach was used by *(Davila and Navon, 2015)*. This work also used the same TodoMvc application by Addy Osmani & Sindre Sorhus. But the selection of frameworks was different. Frameworks chosen in this work are Angular, Backbone, Ember, Marionette and React. And very similar tasks mentioned in previous works were performed and results were obtained and evaluated.

According to *(Gizas, Christodoulou, and Papatheodorou, 2012)* quality of framework is one of the most important factors when it comes to benchmarking. Therefore in his work size, complexity, and maintainability of the framework have given more importance. The parameters measures are **Size Metrics**: lines of code(LOC), the number of statements and number of comment lines, and ration between the lines of code and comment lines. The next metric is **Cyclometric complexity**: Which is McCabe's cyclomatic complexity, branches, and depths. In **Maintainability metric:** Halstead metric and maintainability index is measured.

From his result, he found out that YUI3 3.4.1 frameworks take the most number of lines to build the application with the most number of comment lines i.e 12210 & 9624 respectively. Where jquery takes the moderate amount of lines 7252. If we talk about the Cyclomatic complexity in his work YUI3 3.4.1 framework has more number of functions i.e 28 whose complexity is greater than 20 & DOJO 1.7.2 being the framework with the least number of functions whose complexity is greater than 20. Also, several performance tests were run on the five different browsers including Chrome, Safari, Opera, IE9, and firefox. In those tests, he has mentioned some issues revealed by the performance test that is as follows. IE8 has shown the very big execution time for the frameworks MooTools, YUI2, and YUI3.


PC1/Windows 7 Ultimate
Time (ms)
110
100
90
80
70
60
50
40
30
20
10
0
Dojo 1.7.2
JQuery 1.7.0
Prototype 1.7
ExtJS 4.0.7
YUI 2.9.0
YUI 3.3.1
MooTools 1.4.1
Chrome
Opera
Safari
IE9
IE10

CHAPTER 3: METHODOLOGY

This chapter provides an overview of the research methodology and methods used to carry out this artifact. This section describes the selection of frameworks, selection of parameters, an application designed for computing the performances, different browsers used, tools and techniques used for obtaining the various results.

3.1 Application Design

This research is about computing the performance of various single page application frameworks. So to compare this framework there is a need for web applications built using these frameworks. As discussed in the literature review researcher *(Lawrence, 2017) & (Davila and Navon, 2015)* have used the TODO MVC app created by Addy Osmani & Sindre Sorhus. This is an open-source project built for computing the performance of different frameworks. At the time of their research, this application was well maintained and updated regularly. So it made sense to use that application to save time. The first approach was to use the same app. But this particular app is not being updated and maintained by its developers. In addition to this, most of the frameworks used in that app have updated versions now. So the first approach is eliminated.

So for this study, a new application will be developed using the choice of frameworks. The frameworks chosen will be discussed in a later section of this chapter.

For this study, The Netflix Movie app will be created. As our aim is not to build the app but to study the performance app one smaller portion of the app will be created. Also, the UI is not important in this specific condition so, the app with basic UI & UX is developed. Also, the TODO MVC app hasn't implemented the whole UI.

Netflix movie app will have one route "/home" on that route. All the movies and shows of Netflix will be displayed along with their Title, Year Released, Duration, Title, Genre, Rating & with a little description of that movies or shows. The database is taken from Kaggle.com.

Further, this app also will have search functionality. Using this feature visitors can search the movie or shows based on Title, Show cast, Director. Genre, Rating, etc. The clear search feature will also be there.

3.2 Chosen Frameworks

A lot has been said in Chapter 1 & Chapter 2 about the Single Page Application frameworks. This section will talk about the frameworks chosen for this artifact and the reason behind it. Four frameworks are selected for building the Application and they are Angular 8, ReactJS, VueJS & EmberJS. While developing the latest version of all the frameworks are used. While selecting the framework, for similarity the frameworks with the CLI tool are given preference. CLI tool stands for a command-line interface. It was the measured source of input in all Linux based systems in the 70s–80s. *(Jayakody et al., 2017)* The reason for using command-line tools while developing a web application is to get bootstrap with the application quickly. In very few commands it setups the project directory and it also helps in maintaining the code in the future. And in the next one command application will be running in the browser. It also helps in adding external dependencies to the project easily. *(Farwell, 2017).*

Frameworks	**First Release**	**Latest Stable Release**
Angular 8	14 Sept 2016	12 Aug 2020

ReactJS	29 May 2013	19 March 2020
EmberJs	8 December 2011	4 May 2020
Vue.js	February 2014	13 December 2019

Let's see the frameworks selected in more detail.

3.2.1 Angular 8: Angular is a google's JavaScript framework and it was created by two google developer Misko Hevery and Adam Abrons. When it was created it was based on plain javascript and it was named as AngularJS. At that time majority of websites developed were Multi-Page website and the cons of those have been already discussed. jQuery library was famous at that because using ajax with jQuery is much easier than the plain javascript and it was easy to manipulate the DOM with jQuery. But AngularJS went one step ahead of jQuery because of features like two-way data binding, route handling, support for the external library, etc.*(Studiengang Bachelor et al., no date).* The reason behind the creation of it was to extend the HTML capabilities with the help of custom directives made by angular. The next update of the angular framework was different from its last version. Which was called Angular 2? The version used in this framework is Angular 8.0. The major difference was the AngularJS version was based on JavaScript while all Angular 2+ versions are based on the typescript.

The structure of the angular framework is component-based like many other frameworks. Components are the main building blocks of the frameworks. They are used excessively throughout the application. Each component can be associated with the separate HTML sometimes referred to as template & CSS file. All the business logic, data manipulation, arithmetic, and mathematical operations are written inside the components.

Between all the components there is one main component called app-root component which helps in initializing the entire application. The components are hierarchically like parent and child. And it is possible to pass the states between parent and child components.

Data binding is also one of the most important features provided by angular frameworks. Data Binding is the connection between actual data and the front UI of the application. For showing the data fetched from the server or any other source on the UI we have to establish a connection between the template and the components. Angular has different ways of a binding component variable to the template.

Property Binding: In property Binding the direction of the flow of data is from component to template.

Event Binding: Event binding is the opposite of the property binding. Data always flow from the template to the component.

Two Way Data Binding: Two-way data binding is the most important, if not most important then one of the most used data bindings throughout the application. Angular keeps track of every variable using watchers. As soon as the data changes in input filed the values in components also get changes and vice versa.

Routing provided by angular is a mentionable topic. Routing plays an important role in any website. Without proper nesting of routes, the app won't be considered as a good app. While generating the app using the command, angular ask whether the routing module should be included or not. If selected yes it will add a routing as a separate module and we can add as many as routes we want and we can nest one route inside the another.

3.2.2 ReactJS: ReactJS is maintained and developed by Facebook Inc. It was found in 2003 by Facebook's Jordan Walke as open-source software and since then it has gained a lot of attraction by developers. React is nothing but 'V' in the MVC architecture or It can be said that It is used for just creating user interfaces. *(Lawrence, 2017)*.The primary object of the react is to enhance the performances of the applications. It was mainly focused to address the performance issues of the web applications. The most famous use case of ReactJS is Instagram & Whatsapp. Most often react is mistaken as a framework but it just a library to build a UI component. The difference between library & Framework is that framework help you decide the entire application, your applications are entirely dependent on that single framework it makes developer's job easy while in case of the library, you can just import it and it will you help you build that particular smaller portion of the website. *(Voutilainen, 2017).*

Virtual DOM is the main reason why react is gaining more importance over frameworks like Angular our VUE. *(Javeed, 2019)*. With the help of virtual DOM react decide whether the component should be reloaded or not based on the current state of the component and the changes that have occurred.

One way data flow is used in the ReactJS which helps in controlling the flow of data within the application which improves the stability and to detect the changes occurred.

To re-render the component, state and properties play a major role in deciding that. When there is a transfer of properties or states from parent to child the react DOM compares them with the previously stored value and if there is a difference between the new value and previous value then only it re-renders the component. *(Javeed, 2019).*

Components are the major core of the ReactJS library like Angular. To transfer certain states of the component to the view and display it is the main purpose. The component can be written in two ways i.e one as a function and one as an ES6 class. *(Le, 2020).*

Some important features of ReactJS

- SEO Friendly: ReactJS pages are SEO friendly since the main aim of ReactJS is a performance improvement, rendering of pages is fast in ReactJS therefore they are SEO friendly. *(John, 2010)*
- Testing: It is easy to do testing with the reacts. *(John, 2010)*
- Code Stability: The data flow is unidirectional which keeps the hierarchy very stable. Even if there are some changes in the child component it does not affect the parent. *(John, 2010)*
- Performance: Because of the virtual DOM and Server Side rendering performance of the react app is very fast compared to the others. *(John, 2010)*

3.2.3 Vue.js: Vue.js was introduced in the market (2014) year later of ReactJS's release. Vue.js came out in February 2014 and it was developed by the Evan You who is the former developer of google where he worked a lot on the AngularJS framework. Vue.js is considered a progressive web application framework and it relies on the principle of the Model-View-ViewModel (MVVM) principle. This framework can be used for smaller as well very high and Single Page Applications as well. One measure feature of Vue.js is its ability to scale the applications.

Vue.js is entirely an open-source project build by the single developer as a hobby initially and not by any large funding company.

Like angular and react Vue.js is also built over the components.

Data Binding: Vue.js gives two options to bind the data between the component and its template. One-way data binding and Two-way data binding. *(Shen, Sun and Li, 2018; Song, Zhang and Xie, 2019)* Most often two-way data-binding will be used to synchronize the changes. i.e whatever the changes are being done on the view will always be in sync with the component and vice versa. But it doesn't happen in the one way data-binding. *(Studiengang Bachelor et al., no date)*

One-way: v-bind

Two-way: v-model

Directives: Similar to angular Vue.js also makes use of directive. Directives in Vue.js are prefixed by v-. Directives can be used for data binding, property binding, event handling, and more.

Examples of Directive :

- <p @click="someFunction"> ... </p>
- <button v-on:click="getData"> ... </button>

3.2.4 Ember.js: Ember.js has released in December 2011 by the Ember core team under the MIT license as the open-source project. The original author of the framework is Yehuda Katz. Like the applications mentioned above, it allows developers to build scalable single-page applications. Many Popular sites like Yahoo, Groupon, Discourse make use of Ember.js. *(Shrestha, 2015)*

A router is the most important and powerful concept of the Ember.js. It emphasizes the importance of the URL in managing the state of the application. According to Tom Dale one of the lead dale of Ember.js, when it comes to the latest web application URLs are not just the way to uniquely locate the pages on the server but they serve the more purpose to the web application. *(Dale, 2012)*

Model: Every route is associated with the different model which contains the data associated with the state of that route. This model can be updated or retrieved by the server using AJAX technology and based on that view can be updated very easily.

Controllers: A controller acts as a bridge between the model and its associated view. All the business logic like form handling, data manipulation, or mathematical applications can be written inside the controllers.

Now we are aware of the frameworks which will be used in this study, lets see the various parameters which will be computed and evaluated to analyze the performance of these frameworks.

3.3 Performance Parameters & Methods To Compute Those

The main objective of this work is to calculate the speed indexing of the web application in the browser. So in this work, only the parameters which reflect performances in time are given the importance. Also, a few parameters associated with the code are taken into consideration.

3.3.1 Lines of codes: Lines of code is one of the most important and oldest metrics around there. Iw was used for the first time in the year 1960 in economic, social studies & productivity it came out as an effective metric. In the era of assembly languages, this metric was quite simple but as soon as languages starting evaluated and it started moving toward a structural

programming approach like C language, this concept started getting complex. Therefore later this metric got standardized and two counting methods *(Rentrop et al., 2006; Lawrence, 2017)*.i.e SLOC & LLOC. SLOC is the physical lines of code whereas LLOC is the logical lines of code. SLOC gives the measure of the number of physical lines in the code excluding the comments and LLOC gives the measure of the number of executable statements within the code. *(Park, 1992).*

Method to count LOC: We will be using two tools to count the Physical lines and And source code lines. Tools used are sloc and cloc and these tools are available on npm as well as on Github. *(Dodds, 2020; Kohlhase, 2020)* We just need to simply write npm install command and then these tools will get installed on the machine. And after the successful installation of these tools, we can execute the commands to count the LOC metric. The reason for using two different tools is for greater accuracy.

While counting the LOC we will count only the amount of code developers has to write and not the auto-generated code by the CLI.

Command used – sloc src & cloc src where src is the directory from which we want to count the LOC.

```
C:\Local Disk D\STUDY\Thesis\CODE\FRONT-END\thesis-netflix-angular8\angular>sloc src

---------- Result ------------

            Physical :  533
              Source :  389
             Comment :  82
 Single-line comment :  17
       Block comment :  65
               Mixed :  5
 Empty block comment :  0
               Empty :  67
               To Do :  0

Number of files read :  22

--------------------------


C:\Local Disk D\STUDY\Thesis\CODE\FRONT-END\thesis-netflix-angular8\angular>cloc src
      21 text files.
      21 unique files.
       8 files ignored.

github.com/AlDanial/cloc v 1.86  T=0.04 s (490.2 files/s, 12417.7 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
TypeScript                      13             55             74            208
JavaScript                       2              0              0             75
CSS                              3              7              2             63
HTML                             3              4              1             43
-------------------------------------------------------------------------------
SUM:                            21             66             77            389
-------------------------------------------------------------------------------

C:\Local Disk D\STUDY\Thesis\CODE\FRONT-END\thesis-netflix-angular8\angular>
```

While building the website, the speed taken by the webpage matters a lot. Faster pages are more efficient and they provide a much better user experience. According to kissmetrics infographics, one-fourth of the user we move to the other website if the website they are opening takes more than 3 seconds*(sean, 2011)*. In the case of mobile users, this ratio is even higher. The same study states that 73% of users have faced the use of a slower page load. According to Walmart the if the load time increases by even 1s, then the conversion rate is decreased by 2%. Also, the SEO of webpages should be good so they can rank higher according to the Google ranking algorithm. Now Google has included the page speed as their criteria that means the pages with the higher speed will be ranked higher when someone searches for something on google*(sean, 2011)*. Therefore obtaining the performances of the webpage has become more important. So the next 5 parameters will talk about the factors related to the performance of the webpage. *(Guard, 2020)*

The tools used for computing these parameters are google's lighthouse tool. This open-source tool can be used for monitoring the performances of the web application as well as, the load times, their speed index, etc. Is also gives the score for SEO. *(Google, 2020; Saif, Lung and Matrawy, 2020)*

3.3.2 First Contentful Paint (FCP):

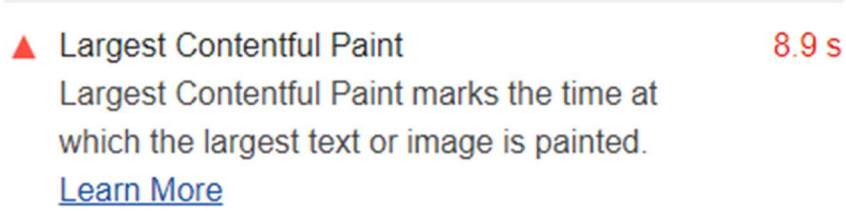
When the user navigates to the particular URL then it takes some time for the browser to fetch data from the server and display it in the browser. FCP measures the amount of time taken by the browser to display the first content of the DOM. It may be anything for example any image, heading, paragraph, etc. FCP score is a comparison of FCP of your website with real-time websites based on data from the HTTP archive. *(Rahman and Ikbal, no date; Budiman et al., 2018; Rome et al., 2019)*

3.3.3 Speed Index :

Speed index measures the time taken for the browser to visually display the data on the webpage during the page load. In simple words, the speed index is how many milliseconds it takes to show the visible parts of the webpage on the website. *(Gao, Dey and Ahammad, 2017; da Hora et al., 2018; Furtak and Wittie, 2019).* For capturing the speed index lighthouse captures the video of webpage loading and calculates the visual progression. Then lighthouse uses the node.js Speedline index module to generate the report. *(Irish, 2020)*

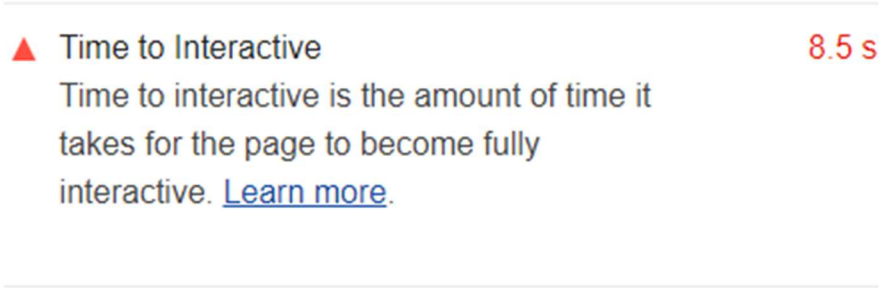
3.3.4 Larget Contentful paint (LCP):

One of the main factors while calculating the performances of the browser is LCP. LCP measures the time taken to display the largest element of the webpage. This helps in an approximation of when the main content of the page will be visible to the user. The available older metrics load or DOMContentLoaded are not useful sometimes because they don't correspond to the important content of the webpage. *(Walton, 2019)*



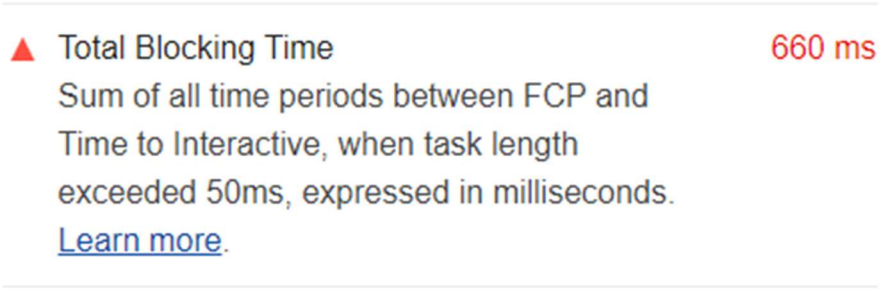
3.3.5 Time To Interactive (TTI):

The webpage is not interactive as soon as it loads the content of the web page. It takes some time for it to become interactive. Interactive means buttons to become clickable or input fields to become typeable etc. This metric is important because content visibility should not be optimized at the cost of TTI. This leads to very poor user experience and there are high chances of users navigating away from such websites. *(Saif, Lung and Matrawy, 2020)*



3.3.6 Total Blocking Time (TBT) :

TBT gives the value of time the pages are blocked from responding to any user events. Events like keyboard inputs, or mouse inputs or touch inputs. Lighthouse derives this value from summing all the blocked portion of long tasks between FCP and TTI. Long tasks are very heavy and they keep the main thread very busy at the cost of interactivity and this is not a good practice. Some times long task makes the pages irresponsive as well. *(Osmani, 2019).*



3.3.7 Time Taken To Render The Real-Time Data:

This study is all about Single Page Applications. In single page application, real-time data i.e data fetched from the server plays a major role. Because in the same URL different data has to keep updating. That's the whole point of the single-page applications. So this custom test is build to test how different frameworks work with the dynamic data. So to test this search functionality has been implemented.

In this test, the user can search the movies of his interest. He can search by title, cast, director, genre, rating, etc. Once the user clicks on the search button this query will be sent to the server using ajax and based on that search query all the movies related to that search query will be returned from the database.

We make use of **performance.now()** method. This method returns the one **DOMHighResTimeStamp** which is accurate up to 5 microseconds. The returned value represents the time elapsed since the time origin. This functionality is available in web workers.

```
searchSHows() {
  console.log("START", performance.now());
  const jsonBody = JSON.stringify({ title: this.seacrhQuery });
  const config = {
    method: "post",
    url: "http://localhost:8080/netflix/searchSHows",
    headers: {
      "Content-Type": "application/json",
    },
    data: jsonBody,
  };
  axios(config)
    .then((response) => {
      this.shows = response.data.resultSet;
    })
    .catch((err) => {
      // Manage the state of the application if the request
      // has failed
      console.log(err);
    });
},
```

As shown in the above code snippet when the user enters the required query in the input field shown on the webpage and then he clicks on the search button this function gets executed. The first line of this function is **performance.now().** Here the time origin is started and value is logged to the console. This function calls the API written in the backend. After successfully fetching the data the result set is fed to **this.shows** variable. Now our **this. shows** variable has the array of objects containing all the shows based on the user's search criteria. As soon this variable gets the new value the framework detects the change has happened and it updates the template with this newly received data.

```
<div class="flex-container">
  <div v-for="show in shows" class="show-item" :key="show.netflixShowId">
    <div v-if="show.netflixShowId==6229">
      <p v-bind:style="{ 'background-color': calculateRenderingTime() }">LAST</p>
    </div>
    <div class="item-header-container">
      <h3>{{show.title}}</h3>
    </div>

    <p>
      <font-awesome-icon icon="calendar-alt" class="icons" />
      Year : {{show.yearRelased}}
    </p>
    <p>
      <font-awesome-icon icon="clock" class="icons" />
      Duration : {{show.duration}}
    </p>
    <p>
      <font-awesome-icon icon="film" class="icons" />
      Category : {{show.listedIn}}
    </p>
    <p>
      <font-awesome-icon icon="video" class="icons" />
      Type : {{show.type}}
    </p>
    <p>
      <font-awesome-icon icon="comments" class="icons" />
      Description : {{show.description}}
    </p>
  </div>
</div>
```

Since this is the vue.js template the list is rendered using the **v-for** directive. When the template is rendering the list and it reaches the last element on the template it fires the **calculateRenderingTime()** method. When this function will be called exactly and what is **6229** in the above figure is explained in the findings section.

```
calculateRenderingTime() {
  var t1 = performance.now();
  console.log("END==>", t1);
  return "";
},
```

In This **calculateRenderingTime()** function, again performance.now() method is called. And if we take the difference between start performance time and end performance time, The time taken to make the ajax call and render the template will be obtained.

This is how the time is taken to render the real-time data is calculated. This parameter will be understood more clearly in the findings section.

So this is all about the test we are going to perform on the applications built using all the frameworks