# 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 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 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 evolution for 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.

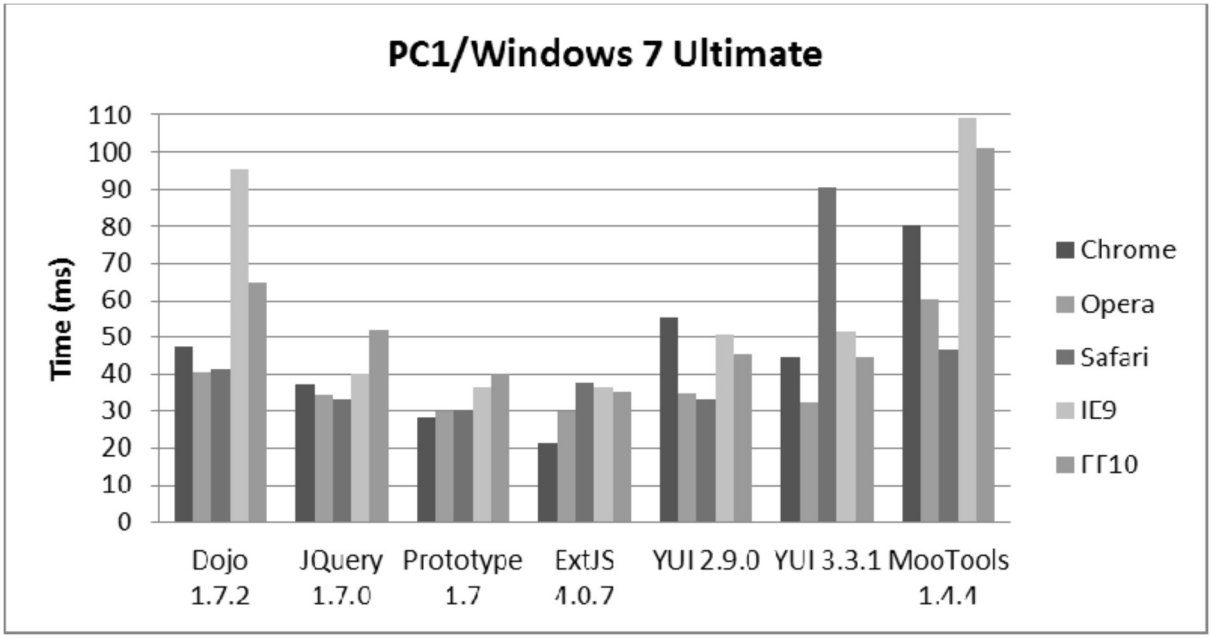
Moreover, in the Microsoft Edge browser, we can see the significant rise in time take for execution for all the frameworks. But here as well BackboneJS outperformed any other framework. BackboneJS took 233 ms and React-JSX took 1953 ms. much higher

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