

# **EVALUATION & COMPARISON OF JAVASCRIPT FRAMEWORKS USED FOR BUILDING SINGLE PAGE APPLICATION**

**NAME :** Saurabh B. Devade (10531140)

**ADDRESS:** Dublin Business School

**DEGREE:** Master in Science in Information Systems and Computing

**DATE:** 04-05-2020

# 1. INTRODUCTION

This era is an Internet era and no one will disagree with that. Nowadays everything is going online from smaller shopkeepers or fruit sellers to big shopping marts like Wallmart, Amazon, etc. Every industry is online like footwear, books, clothing, food, etc. And the role of a website has played a major role in this transition. Because to take one's business online there should be some kind of platform available and websites are one of the easiest available options out there.

## 1.1 Conventional Approach For Building Websites:

As mentioned above websites have become tremendously popular but in older times they have been through poor efficiency, poor interactivity, and poor responsiveness overall the websites were not good as websites these days are (*Mesbah and Deursen, 2007; Wang et al., 2008*). The technology used for building websites was just evolving those days. In an earlier website, on every click pages used to get reload thus reducing ease of use. The websites were used to be a multi-page interface model. So every request has to refresh the whole interface so users had to wait every time they want to retrieve some content.

## 1.2 Evolution of AJAX & SPA (SINGLE PAGE APPLICATIONS)

Thus this kind of architecture for building websites today is not recommended so there was a need for some new technology to take over this conventional method. And this new technology is called as savior **AJAX** – Asynchorunus Javascript and XML. Because of this AJAX technology, a new revolution happened in this field of web development. Sometimes it is referred to as web 2.0 (*Lin et al., 2012*). Using Ajax technology a single page application can build with ease which was highly needed. These single-page applications are composed of the individual component which can be updated, retrieved, removed independently of each other. And because of this, the need to reloading every page on user click or any action is eliminated thus the features lacking in a conventional approach like Interactivity, Responsiveness & User Satisfactions were obtained. (*Mesbah and Deursen, 2007*). Nowadays organizations and businesses are migrating websites build using an old conventional approach to this newly evolved Single Page Web Application knows as SPA based on AJAX technology. (*Mesbah and Deursen, 2007*).

## 1.3 SPA Frameworks

Javascript is the most widely known if not most commonly used programming language for the browsers and building high-end web applications. As a result of the use of javascript & AJAX, the Single Page Applications have been evolved. Also, the community for websites is increasing day by days the contribution offered by them is

huge resulting in the rise of new frameworks. Because of the increasing use of javascript and AJAX technologies and SPAs so many frameworks got created to support a new approach for building modern websites. For example Angular, ReactJS, Backbone.js, etc.

Using this modern framework helps in various methods while developing the app, it makes us write less & specific code, it improves the efficiency of the codes, removes extra boilerplate code, etc. but as every coin has two sides there are some drawbacks of using frameworks as well. Because of this while deciding the right framework while building a new application or migrating the old application to the new application it becomes a headache for developers. As there are pros and cons in every framework. (Gizas, Christodoulou and Papatheodorou, 2012; Koetsier, 2016; Lawrence, 2017)

So in this work number of **SINGLE PAGE APPLICATION** javascript frameworks is chosen and their performances are measured based on various performance factors.

## 2. OBJECTIVES

Working as a Website Developer for more than two years, I came to know the importance of frameworks and how they make the developer's life easy. It is known that there is a jungle of JavaScript frameworks so choosing the best framework depending upon the objectives of a particular project is necessary.

Therefore this research compares the five most used javascript frameworks and also the plain javascript on various merits to see if these frameworks are necessary and there is a need for frameworks.

- So this research works on following objectives
- Is there a need for single-page applications and their frameworks
- How does the performance of websites are dependent on the different frameworks
- Is using plain javascript is a good option against the frameworks
- Do frameworks reduces developers work

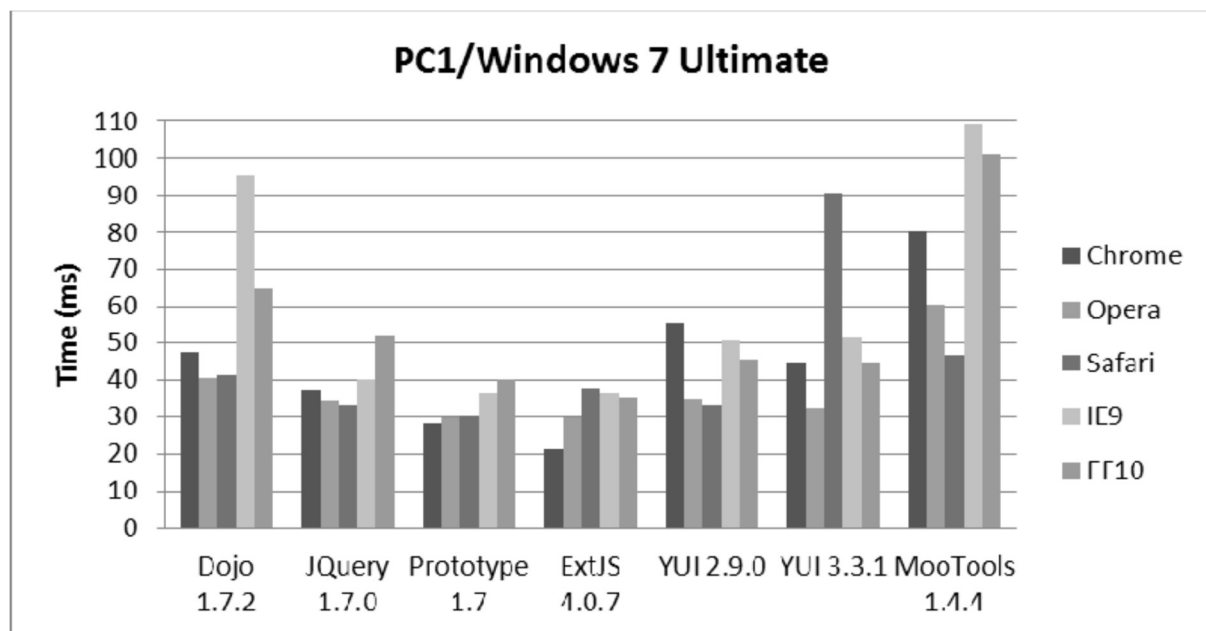
## 3. RESEARCH QUESTIONS

1. Are Javascript Frameworks useful for building Single Page Applications?
2. How is the performance of different Javascript frameworks against each other?
3. Is plain Javascript is efficient than its frameworks?
4. Do frameworks help in writing better code?

## 4. LITERATURE REVIEW

Since JavaScript and its frameworks are still evolving to achieve new heights in the field of web application development, there is still a lot of work to be done to improve it more. But some people have contributed to the community by comparing and benchmarking the few of the frameworks based on certain criteria.

In the year 2012 Gizas, Christodoulou, and Papatheodorou conducted the diverse quality, performance and validations test on the frameworks like Dojo 1.7.2, ExtJS 4.0.7, jQuery 1.7.0, MooTools 1.4.4, Prototype 1.7 & YUI2 2.9.0. The performance tests were performed by using tools like JSMeter ([jmeter.info](http://jmeter.info)), Cloc ([cloc.sourceforge.net](http://cloc.sourceforge.net)), and Understand ([scitools.com](http://scitools.com)). there were different metrics of performances like size metrics, maintainable metrics & complexity metrics and they observed the following measures. (Gizas, Christodoulou and Papatheodorou, 2012).



Later on in the year 2017, a student from Dublin Institute of Technology worked on benchmarking various JSF as a thesis project. The approach used in this work is using a TODOMVC application. This application is an open-source application on GITHUB. This application is build using all the available javascript. This application can be used for frameworks of your choice and different performance result can be obtained. In this work, the following metrics were observed for different javascript frameworks. Lines of Code, Cyclomatic Complexity, Halstead Complexity, Maintainability Index, Database metrics, Page-load/render-time, Speed index, Render, and frame measurements. (Lawrence, 2017). The Similar approach was used by the ERIC MOLIN for his degree project in 2016. He used the same TODOMVC app to examine the various performance parameters. (Molin, 2016). In 2015 Again same TODOMVC app was used by Hans Findel Davila as a thesis for his master's degree. The frameworks chosen by him are Angular, Backbone, Ember, Marionette and React. The various task

was performed using that application like inserting and deleting items from the TODO list. Add 1000 tasks to the To-Do list, Delete all the tasks one by one, Incremental behavior to add 5000 tasks to the list these are some example of the tasks performed to evaluate the performances.(*Davila and Navon, 2015*).

## 5. METHODOLOGY

### 5.1 TOOLS

As mentioned above we are comparing the Different SPA application frameworks with each other as well as with plain javascript we also need some kind of application to perform these specific tasks.

So for this purpose, we have chosen to develop a Simple E-commerce cart application where basic CRUD operations will be performed. We won't be choosing a TODOMVC application because the developers who created it do not give any support now and they have stopped working on it 4 years ago. So choosing this app to execute various performance tests won't be a good idea. Hence, the custom tool for executing the performance check will be created.

Also, all the operations will be performed on the Google Chrome browser because it is used most widely throughout the world.

### 5.2 JAVASCRIPT FRAMEWORKS

We are choosing the 5 most commonly and widely used JSF for our studies Angular, React, AngularJS, VUE, & BackboneJS

**AngularJS** - Angular was created by Google employees Misko Hevery and Adam Abrons in 2008. long ago it had been referenced to as AngularJS and developed in plain JavaScript. This was at a time when the bulk of internet sites where supported the multi-page application approach: When a user clicked on a link, the browser had to retrieve the requested HTML document from the server. counting on the web connection and therefore the responsiveness of the server, it could take a good amount of your time until the user could view the new page. Gradually user devices increased in overall performance so application logic may well be executed within the browser. This led to the approach of Single-Page Applications (SPAs).

**React** – Facebook has created this library for creating user interfaces for the webpage. In 2013 it has been published as an open-source library and since then it has gained a lot of attraction of developers and geeks throughout the world. Some of the popular users of reacting are Instagram and WhatsApp. Often people think of React as a web development framework but is a kind of library used for a variety of scenarios including native applications like (iOS & Android).

**VUE** – Vue is relatively new in this comparison even though it was released two years before the Angular 2 framework. Vue was released in the year 2014 and it is created by

the former employee of Google who has worked a lot on AngularJS. Vue is known as the progressive frameworks which can be used to build user interfaces for webpages. While it is not strictly associated with the Model-View-Viewmodel (MVVM) pattern, the design principles of Vue were partly inspired by it. Scalability is one of the merits of this framework.

**BackboneJS** - As mentioned in the official online documentation of BackboneJS, it is a JavaScript library that provides a structure for web application and it supports the MVP – Model View Present architecture. Backbone makes use of models with key-value bindings and custom events. Models can either be created, validated, destroyed, or saved to the server. Whenever a UI action causes an attribute of a model to change, the model triggers a “change” event. In other words, models manage an internal table of data attributes and triggers these change events when a piece of data within it is changed (see Figure 3). Collections help to handle a group of related models by providing helper functions to perform aggregations or computations against a list of models.

### 5.3 METRICS TO BE MEASURED –

**First Contentful Paint** - FCP measures how long it takes the browser to render the first piece of DOM content after a user navigates to your page. Images, non-white elements, and SVGs on your page are considered DOM content; anything inside an iframe isn't included.

**First Meaningful Paint** - FMP measures when the primary content of a page is visible to the user. The raw score for FMP is the time in seconds between the user initiating the page load and the page rendering the primary above-the-fold content. FMP essentially shows the timing of the paint after which the biggest above-the-fold layout change happens. Learn more about the technical details of FMP in Google's Time to First Meaningful Paint: a layout-based approach.

**Speed Index** – Speed index determines how quickly the content is displayed on the page during the loading of the page. Lighthouse tool first records the video of the page and then it measures the visual progression between the pages. Lighthouse then uses the Speedline node, js module to generate the speed index score.

**First CPU Idle** - First CPU Idle measures how long it takes a page to become minimally interactive. A page is considered minimally interactive when:

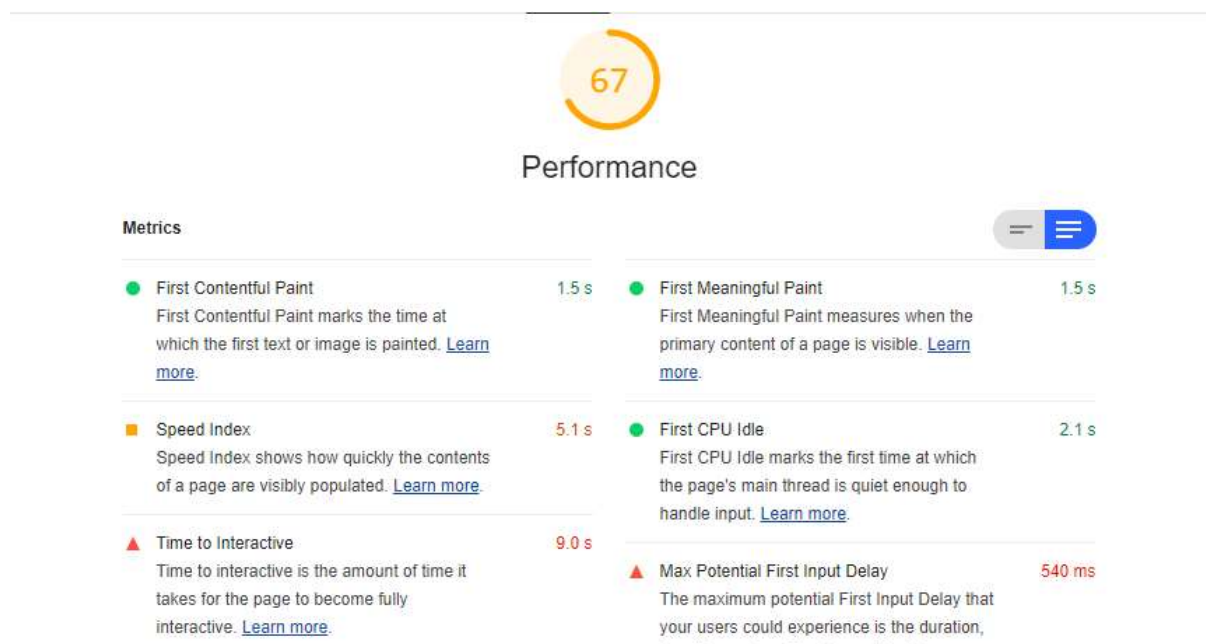
- Most - but not necessarily all - UI elements on the screen are interactive, and
- The page responds, on average, to most user input in a reasonable amount of time.

**Time To Interactive** – Time to interactive measures the time taken by page to become fully interactive. A page considered to be fully interactive when

- The page displays useful content, which is measured by the First Contentful Paint,
- Event handlers are registered for most visible page elements, and
- The page responds to user interactions within 50 milliseconds.

**First Input Delay** - First Input Delay (FID) is an important, user-centric metric for measuring load responsiveness because it quantifies the experience users feel when trying to interact with unresponsive pages—a low FID helps ensure that the page is usable.

All the above-mentioned parameters are measured using the Google Audit. Google audit makes use of the Lighthouse tool and chrome dev tools to measure the performances of the webpages.



**Lines Of Code** – Lines of codes is the oldest metrics in the field of software projects. This metric was used in 1960 for the first time. This metric gives us the blank lines, comment lines, and physical lines of the source code. The tool used to measure this is an open-source project on GitHub named **cloc**.

The above-created cart application will have basic functionalities. It will have a basic ajax call to check how each framework will execute ajax call their response time and their simplicity.

Also, it will have to retrieval functionality i.e retrieving more than 100 or 1000 records and displaying it correctly on the page. And then the time taken for this feature will be counted by the custom code.

## 6. RISK ANALYSIS

**Ecommerce-App** – Since 5 frameworks along with the plane javascript have been chosen for the comparison and an app from scratch is being constructed, so because of time constraints if the app won't be able to complete using all the frameworks a simple todo application will be created.

**Tools** – If any of the tools mentioned above fails to work, then Benchmark.js library is available to perform the above operations will be used for analysis.

## 7. GANTT CHART

TASK NAME	MONTH 1	MONTH 2	MONTH 3
Planning			
Research			
Design			
Implementation			
Follow Up			

## 8. REFERENCING

Davila, H. F., and Navon, J. (2015) 'Performance of Javascript Frameworks on Web Single Page Applications (Spa)'.

Gizas, A. B., Christodoulou, S. P. and Papatheodorou, T. S. (2012) 'Comparative evaluation of JavaScript frameworks', *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web Companion*, (Cc), pp. 513–514. DOI: 10.1145/2187980.2188103.

Koetsier, J. (2016) 'Evaluation of JavaScript frameworks for the development of a web-based user interface for Vampires', p. 56. Available at: <https://esc.fnwi.uva.nl/thesis/centraal/files/f371967790.pdf>.

Lawrence, C. (2017) 'Benchmarking JavaScript Frameworks', pp. 31–33. DOI: 10.21427/D72890.

Lin, B. *et al.* (2012) 'Comparison between JSON and XML in Applications Based on



AJAX', *Proceedings - 2012 International Conference on Computer Science and Service System, CSSS 2012*. IEEE, (February 1998), pp. 1174–1177. DOI: 10.1109/CSSS.2012.297.

Mesbah, A. and Deursen, A. Van (2007) 'Migrating Multi-page Web Applications to Single-page A JAX Interfaces'.

Molin, E. (2016) 'Comparison of Single-Page Application Frameworks: A method of how to compare Single-Page Application frameworks written in JavaScript.', *Degree in Project Computer Science and Engineering*.

Wang, Q. *et al.* (2008) 'An automatic approach to reengineering common website with AJAX', *Proceedings - International Conference on Next Generation Web Services Practices, NWeSP 2008*, pp. 185–190. DOI: 10.1109/NWeSP.2008.34.