



# **iJRASET**

International Journal For Research in  
Applied Science and Engineering Technology



---

# **INTERNATIONAL JOURNAL FOR RESEARCH**

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume: 7      Issue: V      Month of publication: May 2019**

**DOI: <https://doi.org/10.22214/ijraset.2019.5283>**

**[www.ijraset.com](http://www.ijraset.com)**

**Call:  08813907089**

**E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)**



# Single Page Web Application Technologies

Ankush Ravindra Gochke<sup>1</sup>, Mr. Prakash Kene<sup>2</sup>

<sup>1</sup>MCA Department, Modern College Of Engineering, Pune

<sup>2</sup>Assistant Prof., MCA Department, Modern College Of Engineering, Pune.

**Abstract:** *The primary objective of this thesis is to evaluate the web as a platform for applications rather than as a traditional document platform. Also, being a requirement for modern web applications, the suitability and challenges of the single page architecture will be assessed. We know that HTML was designed document-based web in mind, rendering certain problems in creating application-style web pages. In this thesis, I examine what these problems are and what practical means there are to overcome them [1].*

- 1) *What are the main challenges and advantages of building and distributing web applications?*
- 2) *How does single page architecture solve or relate to these challenges and advantages?*
- 3) *What practical means do we have to build single page applications?*

*This thesis provides a comprehensive overview of current state of web technologies, such as HTML5, CSS3 and JavaScript. I evaluate the suitability of those technologies for building applications that might replace native applications. Also, I examine concepts and techniques that are not commonly utilized in traditional web sites but solve relevant problems in applications. One of the motivators behind single page architecture is the need to enhance the user experience in the web, thus usability of web applications is also studied. I review what performance and User Interface (UI) related problems appear in web applications and how those are solved.*

## I. INTRODUCTION

JavaScript is a customer side scripting language which is utilized for giving unique traits and highlights for the site pages. JS was produced for supporting the program with highlight of offbeat correspondence, controlling the program and for client collaboration with the site page parts. A HTML page is only a static page. JavaScript can make the site increasingly intelligent and client amicability, JavaScript helps simple route of the site and to gives intelligence. Based on JavaScript there are numerous systems created and furthermore included numerous new highlights for server side scripting. JavaScript has developed such a great amount in web industry and on the off chance that you take a gander at the current sites there are no web advances that are not utilizing JavaScript. JavaScript is a dynamic scripting language, it train the program to make changes to page components in the wake of stacking a page. JavaScript is anything but difficult to learn: you simply need a program and a content proofreader to start composing and running your code.

A single-page application (SPA) is a web application or web site that collaborates with the client by powerfully changing the present page instead of stacking whole new pages from a server. This methodology maintains a strategic distance from intrusion of the user experience between progressive pages, influencing the application to act increasingly like a desktop application. In a SPA, either all vital code – HTML, JavaScript, and CSS – is recovered with a solitary page load, or the proper assets are dynamically loaded and added to the page as important, for the most part in light of client activities. The page does not reload anytime all the while, nor controls exchange to another page, in spite of the fact that the location hash or the HTML5 History API can be utilized to give the discernment and safety of isolated legitimate pages in the application. Collaboration with the single page application regularly includes dynamic correspondence with the web server behind the scenes.

## II. LITERATURE REVIEW

There are various techniques available that enable the browser to retain a single page even when the application requires server communication.

### A. *Java Script frameworks*

- 1) *Angular JS:* Since its presentation by Google, it has ended up being the most utilized and most conveyed system for significant single page application improvement structures on the web. With a lofty expectation to absorb information and MVC design, this is plainly one of only a handful couple of ground-breaking structures on the web. The huge measure of inherent usefulness



enables clients to control and structure their application for about each sort of condition or assignment they could consider. It diminishes code as well as enables applications to be effectively kept up and scaled after some time and changing models.

"AngularJS improves the front-end advancement experience." - By Webydo.

AngularJS is a basic framework for dynamic web applications. With AngularJS, planners can utilize HTML as the layout language and it takes into consideration the augmentation of HTML's sentence structure to pass on the application's parts easily. Rakish makes a big deal about the code you would somehow or another need to compose totally excess.

Regardless of the way that AngularJS is usually identified with SPA, you can utilize Angular to fabricate any sort of application, exploiting highlights like Two-way official, templating, RESTful API taking care of, modularization, AJAX taking care of, reliance infusion, and so on.

AngularJS is based on the conviction that decisive programming ought to be utilized to make UIs and interface programming segments, while basic writing computer programs is more qualified to characterize an application's business rationale. The structure adjusts and stretches out conventional HTML to show dynamic substance through two-way information restricting that takes into consideration the programmed synchronization of models and perspectives. Accordingly, AngularJS de-stresses unequivocal Document Object Model (DOM) control with the objective of improving testability and execution.

AngularJS's plan objectives include: to decouple DOM control from application rationale. The trouble of this is drastically influenced by the manner in which the code is organized. to decouple the customer side of an application from the server side. This permits advancement work to advance in parallel and takes into consideration reuse of the two sides. to give structure to the voyage of structure an application: from planning the UI, through composition the business rationale, to testing.

AngularJS actualizes the MVC example to isolate introduction, information, and rationale parts. Utilizing reliance infusion, Angular brings generally server-side administrations, for example, see subordinate controllers, to customer side web applications. Thus, a significant part of the weight on the server can be decreased.

The advantages of AngularJS are:

- a) Angular JS provides capability to create Single Page Application in a very clean and maintainable way.
- b) Angular JS provides data binding capability to HTML. Thus, it gives user a rich and responsive experience.
- c) Angular JS code is unit testable.
- d) Angular JS uses dependency injection and make use of separation of concerns.
- e) Angular JS provides reusable components.
- f) With Angular JS, the developers can achieve more functionality with short code
- g) In Angular JS, views are pure html pages, and controllers written in JavaScript do the business processing

#### B. EmberJS -

EmberJs is an improved and updated variant of SproutCore. Constructed basically for expansive scale applications it is a rarity indeed utilized for single page application advancement assignments. By grasping web parts and a model view controller-based design, the system is one of only a handful not many that is ES6 objection. With ideas, for example, courses, layouts, models, perspectives, controls, and parts the structure gives the important basis to an application. The formats are worked with Handlebars which itself is a standout amongst the most acknowledges layout structures.

Ember.js is an open-source JavaScript web structure, in light of the Model- view- ViewModel (MVVM) design. It enables engineers to make scalable by consolidating common idioms and best practices into the structure.

Coal is utilized on numerous prominent sites, including Discourse, Groupon, LinkedIn, Vine, Live Nation, Nordstrom, Twitch.tv and Chipotle. Albeit fundamentally thought about a system for the web, it is additionally conceivable to construct work area and portable applications in Ember. The most outstanding case of an Ember work area application is Apple Music, a highlight of the iTunes desktop application.

- 1) Ember.js is an open source JavaScript framework under MIT license.
- 2) It provides the new binding syntax using the HTMLBars template engine which is a superset of the Handlebars templating engine.
- 3) It provides the Glimmer rendering engine to increase the rendering speed.
- 4) It provides the Command Line Interface utility that integrates Ember patterns into development process and focuses easily on the developer productivity.
- 5) It supports data binding to create the link between two properties and when one property changes, the other property will get upgraded with the new value



- 6) Ember.js is used for creating reusable and maintainable JavaScript web applications.
- 7) Ember.js has HTML and CSS at the core of the development model.
- 8) It provides the instance initializers.
- 9) The routes are core features of the Ember.js which are used for managing the URL's.
- 10) Ember.js provides Ember Inspector tool for debugging Ember applications.
- 11) Ember.js uses templates that help to automatically update the model, if the content of applications gets changed.

### C. Backbone

The heritage single page application improvement system with less than 800 lines of code is as yet the software engineer's most loved for most applications as it gives the important adaptability through its MVC display while additionally supporting accumulations that take into account varieties to program design. By altogether diminishing information attach ups to the DOM, Backbone enables designers to make auxiliary code that can be effectively adjusted to reflect changes in the DOM. Alongside the Underscore library, Backbone figures out how to deal with even the most unpredictable application calm nimbly.

Backbone.js offers structure to web applications by giving models key-esteem official and custom occasions, accumulations with a rich API of enumerable capacities, sees with revelatory occasion taking care of, and associates everything to your current API over a RESTful JSON interface.

Backbone.js is a JavaScript library with a RESTful JSON interface and depends on the Model– view– moderator (MVP) application plan worldview. Spine is known for being lightweight, as its solitary hard reliance is on one JavaScript library, Underscore.js, in addition to jQuery for utilization of the full library. It is intended for creating single-page web applications, and for keeping different pieces of web applications (for example various customers and the server) synchronized. Spine was made by Jeremy Ashkenas, who is likewise known for CoffeeScript and Underscore.js.

- 1) BackboneJS provides various building blocks such as models, views, events, routers and collections for assembling the client side web applications.
- 2) When a model changes, it automatically updates the HTML of your application.
- 3) BackboneJS is a simple library that helps in separating business and user interface logic.
- 4) It is free and open source library and contains over 100 available extensions.
- 5) It acts like a backbone for your project and helps to organize your code.
- 6) It manages the data model which includes the user data and displays that data at the server side with the same format written at the client side.
- 7) BackboneJS has a soft dependency with jQuery and a hard dependency with Underscore.js.
- 8) It allows to create client side web applications or mobile applications in a wellstructured and an organized form.

### D. React JS

In figuring, React (otherwise called React.js or ReactJS) is a JavaScript library for structure UIs. It is kept up by Facebook and a network of individual designers and organizations.

Respond can be utilized as a base in the improvement of single-page or versatile applications. Complex React applications generally require the utilization of extra libraries for state the board, directing, and connection with an API.

Respond energizes the production of reusable UI parts, which present information that changes after some time. Loads of individuals use React as the V in MVC. Respond abstracts away the DOM from you, offering a less difficult programming model and better execution. Respond can likewise render on the server utilizing Node, and it can control local applications utilizing React Native. Respond executes single direction receptive information stream, which lessens the standard and is simpler to reason about than customary information official.

## III. FINDINGS

### A. Why ReactJS is the better option ?

Well, There are various frameworks and libraries out there. Choosing the right framework and right library is very important and is solely dependent on the purpose of your app which is to be developed.

ReactJS is very simple and unique. It is easy to learn and has a wide range of unique features which makes it unique.

While different advances, for example, Angular were accessible when Facebook created ReactJS, most engineers were compelled to complete a great deal of coding. Engineers utilizing different structures have the test of adjusting on most codes notwithstanding



while creating parts that changed much of the time. What they needed was a structure that could enable them to separate complex parts and reuse the codes to finish their ventures quicker.

ReactJS gave the arrangement that designers were searching for. It utilizes JSX (a one of a kind language structure that permits HTML cites just as HTML label sentence structure application for rendering explicit subcomponents) This is exceptionally useful in advancing development of machine-clear codes and in the meantime aggravating segments into a solitary time irrefutable document. Today, ReactJS has turned out to be exceptionally prominent in view of its additional straightforwardness and adaptability. Numerous individuals are notwithstanding alluding to it as the eventual fate of web improvement. It is evaluated that in excess of 1,300 designers and more than 94,000 destinations use ReactJS.

Some portion of this tremendous prevalence originates from the way that top organizations, for example, Facebook, PayPal, Uber, Instagram, and Airbnb use it to illuminate UI related issues. This believability has drawn many individuals to the system.

Organizations rushed to get a handle on this innovation as a result of the straightforwardness it offered to designers; that is, the capacity to learn React in absolute minimum time. Code reusability with problem free expansion/change of functionalities in the current framework implied designating altogether less time and spending plan on improvement and building bigger groups, individually.

Top organizations are genuinely captivated by React's abilities on a business point of view. The rundown of organizations using React underway is there to demonstrate it, just to give some examples: Facebook, Instagram, Netflix, Whatsapp, Salesforce, Uber, The New York Times, CNN, Dropbox, DailyMotion, IMDB, Venmo, and Reddit are the significant association ones among the 100+ other medium to vast scale organizations.

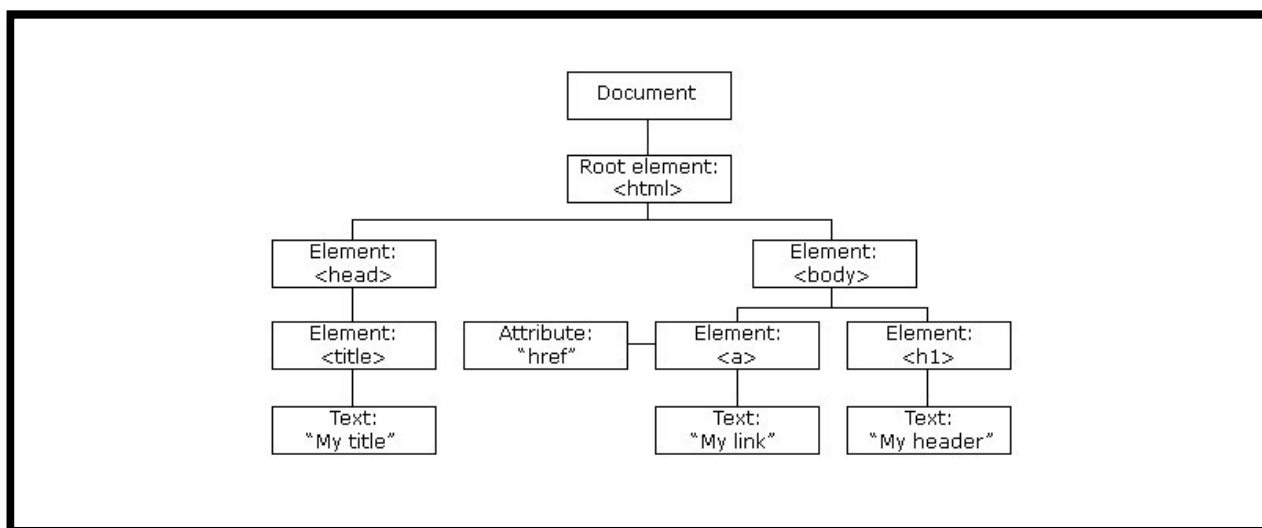


Fig 1.1

One of the truly cool pieces of React is the virtual DOM. Ordinarily, when you build up an application that has a great deal of client communication and information refreshes, you need to painstakingly think about how your application structure is going to affect execution. Indeed, even with quick customer stages and JavaScript motors, broad DOM control can be an act bottle-neck and even outcome in an irritating client experience. More terrible, on the grounds that the DOM is tree-organized, basic changes at the best dimension can make colossal swells the UI.

Respond tackles this by utilizing a virtual DOM. This is, as the name suggests, a virtual portrayal of the DOM. Any new view changes are first performed on the virtual DOM, which lives in memory and not on your screen. An effective calculation at that point decides the progressions made to the virtual DOM to recognize the progressions that should be made to the genuine DOM. It at that point decides the best method to roll out these improvements and afterward applies just those progressions to the genuine DOM. This ensures a base refresh time to the genuine DOM, giving higher execution and a cleaner client experience all around.

The genuine procedure is more confounded than that and you can peruse progressively about how React recognizes the progressions made to the virtual DOM and makes the rundown of changes that should be made to the real DOM. For more data, read the page on Reconciliation on the React site.

### B. Simple component

```
import React from 'react';
class App extends React.Component {
  render() {
    return (
      <div>
        <Header/>
        <Content/>
      </div>
    );
  }
}
class Header extends React.Component {
  render() {
    return (
      <div>
        <h1>Header</h1>
      </div>
    );
  }
}
class Content extends React.Component {
  render() {
    return (
      <div>
        <h2>Content</h2>
        <p>The content text!!!</p>
      </div>
    );
  }
}
export default App;
```



Fig 1.2

From fig1.2, we can see the normal 2 lines of html. Somebody can say that its just 3 text lines. But, when you look at the code, you will see that, there are multiple components in a single page. Every component is nothing but another div. The main div (App) contains 2 tags. Header and Footer. Both tags represent a different component. And when the main div renders, all the divs renders too. This is the main working of react js. Only one html file is rendered which will contain a main div or a component which will contain all other components. Which div is to be rendered when solely depend on the coding conditions.

### C. Nothing is perfect !

1) *High Pace of Development:* This hindrance are appropriately depicted by designers Michael Jackson and Ryan Florence on Modern Web: "in the event that you didn't see we're driving a vehicle here with two punctured tires, the hood just flew up before the windshield, and we do not understand what's happening any longer!" nature always shows signs of change, and engineers should normally relearn the better approaches for getting things done. Everything is advancing, and a few engineers are not happy with staying aware of such a pace.



- 2) *Poor Documentation*: The issue with documentation follows back to steady arrivals of new devices. Extraordinary and new libraries like Redux and Reflux are promising to quicken crafted by a library or improve the whole React biological system. Toward the end, engineers battle with coordinating these instruments with ReactJS. A few individuals from the network feel that React innovations are refreshing and quickening so quick that there is no opportunity to compose legitimate guidance. To fathom this, designers compose their own documentation for explicit devices utilized by them in current ventures.
- 3) *'HTML in my JavaScript!' – JSX as a barrier*: ReactJS utilizes JSX. It's a sentence structure augmentation, which permits blending HTML with JavaScript. JSX has its own advantages (for example, shielding code from infusions), yet a few individuals from the improvement network consider JSX to be a genuine detriment. Engineers and planners grumble about JSX's multifaceted nature and ensuing steep expectation to absorb information.
- 4) *Additional SEO Hassle*: There have been worries that Google and other web crawlers can't list or ineffectively list dynamic website pages with customer side rendering. These worries haven't been completely demonstrated and there are exposing materials around. Google itself affirmed in 2014 that their crawlers are fit for perusing dynamic substance. In this way, we aren't going to state that your ReactJS application won't be ordered by Google. It's 2018 all things considered.

In any case, despite everything you need to do some testing to guarantee that your application makes a mate out of Google as there were issues revealed by certain clients. Website design enhancement experts suggest running your React applications through Fetch as Google apparatus to adapt better how the crawlers experience them.

While this isn't a big problem, SEO may add up to your development effort. You can learn more about React apps SEO testing here.

#### IV. CONCLUSION

In the event that there's an exercise learned here, it's that viewpoint that guides systems. Its structure, its engineering, even the issue it is attempting to comprehend pursues from this point of view and sets a tone. From that point, a network assembles around this viewpoint and catalyzes its endeavors, and after somewhat of a period, another system is conceived.

A structure is just an exemplification of certain examples, coordination of certain innovations, and source code to help make our web applications less demanding to manufacture and keep up. In the event that you are an individual designer, the best exhortation we can offer is to invest some energy with the same number of structures you may think would work for you. On the off chance that you are a business chief or draftsman attempting to settle on a choice, recollect that an element list is only one part of a choice, and now and then more isn't better. Test yourself or your group to investigate a system, on the whole, begin with a rundown of what is critical to you and your association, particularly those things that rise above specialized highlights.

#### REFERENCES

- [1] Misra, S., Cafer, F.: Estimating Quality of JavaScript. The International Arab Journal of Information Technology 9(November), 535–543 (2012)
- [2] Nisos Technologies, Top 10 frameworks for single page application development <https://nisostech.com/top-10-frameworks-for-single-page-application-development/>
- [3] Sneha Ambulkar, Angular JS, International Journal of Scientific & Engineering Research, Volume 7, Issue 2, February-2016 ISSN 2229-5518
- [4] Webydo, Practical Uses of AngularJS: Create a Single Page Application (SPA) or a Website Menu in an Instant <https://www.awwwards.com/practical-uses-of-angularjs-create-a-single-page-application-spa-or-a-website-menu-in-an-instant.html>
- [5] Thinkwik, Why ReactJS is gaining so much popularity these days <https://medium.com/@thinkwik/why-reactjs-is-gaining-so-much-popularity-these-days-c3aa686ec0b3>
- [6] Hamza Mahmood, Advantages of Developing Modern Web apps with React.js <https://medium.com/@hamzamaahmood/advantages-of-developing-modern-web-apps-with-react-js-8504c571db71>
- [7] The Good and the Bad of ReactJS and React Native <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-reactjs-and-react-native/>
- [8] John Willoughby, The Top 5 Benefits of React that Make Life Better <https://www.telerik.com/blogs/5-benefits-of-reactjs-to-brighten-a-cloudy-day>
- [9] Jason Hoffmann, The question of “why?” in single-page application frameworks <https://blog.logrocket.com/the-question-of-why-in-single-page-application-frameworks-91383446d0f5>
- [10] Kit Kelly, Web Frameworks:Conclusions <https://www.sitepen.com/blog/web-frameworks-conclusions/>
- [11] Misra, S., Cafer, F.: Estimating Quality of JavaScript. The International Arab Journal of Information Technology 9(November), 535–543 (2012)
- [12] Naimul Islam Naim, ReactJS: An Open Source JavaScript Library for Front-end Development [https://www.theseus.fi/bitstream/handle/10024/130495/Final\\_Year\\_Thesis.pdf?sequence=1&isAllowed=y](https://www.theseus.fi/bitstream/handle/10024/130495/Final_Year_Thesis.pdf?sequence=1&isAllowed=y)
- [13] Graziotin, D., Abrahamsson, P.: Making sense out of a jungle of javascript frameworks - towards a practitionerfriendly comparative analysis. In: Product-Focused Software Process Improvement - 14th International Conference, PROFES 2013, Paphos, Cyprus, June 12-14, 2013. Proceedings, pp. 334–337 (2013). DOI 10.1007/978-3-642-39259-7\_28





10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)