iJRASET
International Journal For Research in
Applied Science and Engineering Technology

Volume: 10 Issue: VII Month of publication: July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45260

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

Comparative Analysis on Front-End Frameworks for Web Applications

Rishi Vyas
Electronics and Communication Engineering, Ujjain Engineering College, Ujjain, Indore Road, Ujjain - 456010 (M.P.)

Abstract: An essential step in developing any program or app is choosing the appropriate front-end framework or library. Front-end Web development sounds similar to the JavaScript framework. Both of these choices are available for web development requirements. Vue, React, and Angular all fall under the umbrella of JavaScript frameworks. Due to the extensive spectrum of issues developers encounter daily, the industry offers a wide diversity. With the help of many accessible frameworks, a web application may be constructed as intended while considering all practical considerations. The advantages and disadvantages of the fundamental elements and distinctive features of frameworks are discussed in this paper.

Additionally, it offers a thorough analysis of the research on front-end frameworks. This study gives an overview of the front-end frameworks discovered in the literature, outlining the essential components of these frameworks using a systematic literature review as methodology. The three most popular frameworks, Vue.js, Angular, and React, were examined for the necessary features.

Keywords: Library, Frameworks, Single Page Applications, Multi Page Applications, Data Binding, DOM Manipulation, MVC Architecture

I. INTRODUCTION

Web applications today play a significant role in everyone's lives. Apps are undoubtedly taking over the corporate sector, from online ticket booking to meal ordering and cab booking to online purchasing. Client behavior has significantly changed as a result of technology innovation and mobility. Businesses are excitedly anticipating establishing an online presence in 2022 as the impact of mobility continues to grow and transform industries globally. Every frontend developer has heard about three frameworks for building web applications: React, Vue.js, and Angular. The principles and best - practice for project development are defined by frameworks, which also simplify many aspects for us, such as code structure and maintenance, by acting as a skeleton for our applications. Technically, all three frontend components are nothing more than lines of code that the browser subsequently renders to produce an attractive web page. A software framework (frontend or backend) contains standardized pre-written code, which facilitates and speeds up the development of specific functionality.

Angular is a fully-fledged front-end framework, React is a UI library, and Vue.js is a progressive framework. They can be used interchangeably to build front-end applications, but they're not 100 percent the same, so it makes sense to compare them and understand their differences. Each framework is component-based and allows the rapid creation of UI features. However, they all have different structures and architecture.

Since thousands of apps are available, developing an application is not just a complex job; you need a particular strategy, considerable preparation, and access to leading technology to design an enterprise-level app. The paper aims to demonstrate how well-suited the frameworks are for creating web applications and identifying essential qualities that allow for more individualized web application development.

II. ASSOCIATED CONCEPTS

The development of web apps requires a practical framework. JavaScript frameworks are developing at a breakneck pace; today, we have frequently updated versions of Angular, React.js, and Vue.js. These technology stacks have been replacing pure server-side technologies. We have slowly progressed from server-side applications to more distributed applications where a part of the job is done on the client-side and a position on the server-side. Now we have transitioned to single-page applications or SPA, and all of these technologies are based purely on JavaScript. As a result, they are lightweight and aren't dependent on third-party run-time tools. React is a front-end JavaScript library, while Angular and Vue are JavaScript frameworks. Although the terms library and frameworks are used interchangeably, they are significantly different. A library is a set of classes and functions, and users have complete control to call them. Frameworks, on the other hand, are like models. They have limited choices and come with a blueprint of their own. As a result, the user doesn't have much control.

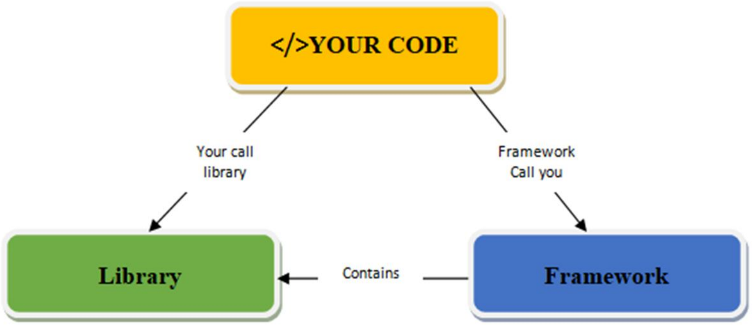


Fig 1. Functioning of Library and Framework

Thus, the critical difference between a library and a framework is "Inversion of Control," commonly called IoC. We exercise control when we use a library's methods. The power is reversed in the framework, i.e., the framework makes the call. The program defines a skeleton and its features to fill the frame. So, JavaScript, an interpreter language where code is executed line by line, makes all these tools very efficient. These are avoided, making these tools quick because transforming the code into an intermediate code would call for a run time environment and cause a delay.[1]

React is used to develop web and mobile applications, and single-page applications utilize Angular; we can also set the hybrid applications using the stack, mainly because it is purely based on HTML, CSS, JS, and jQuery. Vue is also used to develop single-page applications and user interfaces. SPA is a modern approach to creating web applications used primarily by applications with less content. A single-page app only reloads the information the user needs. In a multi-page app, all the pages the user has requested and the whole web page content are updated. The server loads the complete page when the user runs a SPA. Later, just the essential data is transferred in the form of JSON files in response to the request.

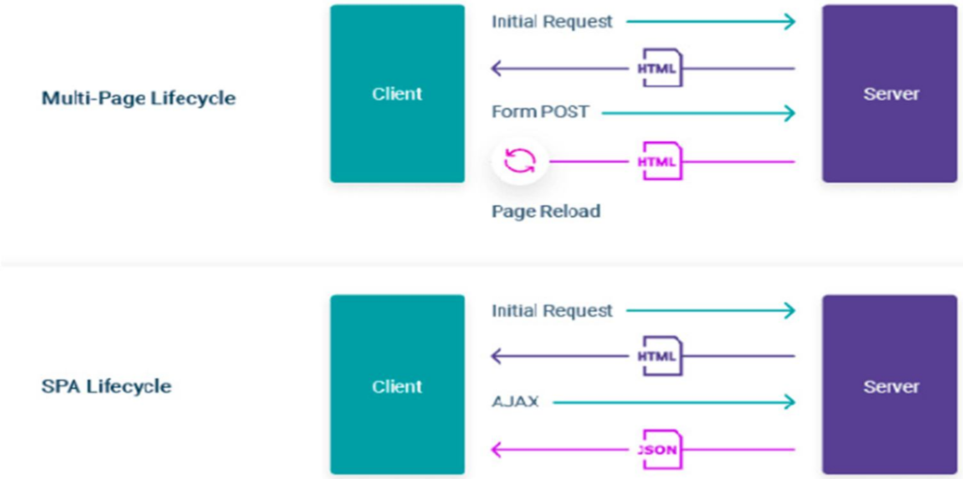


Fig 2. Lifecycle of SPA and MPA

Source : https://lvivity.com/single-page-app-vs-multi-page-app

SPA is a browser-based application that avoids the need for page refreshing while in use. SPAs are all about serving an outstanding UX by trying to imitate a "natural" environment in the browser — no page reloads, no extra wait time. It is just one web page that is visited, loads all other content using JavaScript — which heavily depends on SPA, requests the markup and data independently, and renders pages straight in the browser. Single-page websites assist in keeping the user in a pleasant digital environment where content is delivered in an understandable, clear, and logical manner. Multiple-page applications work in a "traditional" way. Every change, e.g., display or submit data back to server requests rendering a new page from the server in the browser. These applications are extensive and more prominent than SPAs because they need to be. These apps contain several levels of UI because of the content; as a result, the solution improves, enabling the user to refresh certain areas of the application. On the other hand, it adds more complexity and is more challenging to develop than a single-page application.[2]

Popular JavaScript tools and frameworks like Angular, React, and Vue assists programmers in creating intricate, responsive, and contemporary web user interfaces. We may even create native mobile apps for mobile devices with the aid of Angular, React, and Vue with the help of additional frameworks like React Native, Ionic (with Angular or with React), or Native Script. Angular is the oldest of the lot. Google created it, and it was published in 2010 then. React is an open-source framework developed by Facebook and released in 2013, and Vue.js is the youngest of the three. It was released in 2014 and designed by ex-Google employee Evan You.

III. POSSIBLE COMPARISON DIMENSION

A. Trends And Popularity

JavaScript is a programming language that offers a wide range of functionality and many development environments. There are compelling arguments for sticking with the crowd when it comes to programming frameworks and libraries:

1) Renowned frameworks are less likely to vanish (i.e., the core team will probably maintain them in the future).
2) Popular frameworks tend to have significant communities that produce complementary libraries and tooling.
3) It is easy to find more help (tutorials, forum threads, docs) on popular libraries and frameworks.

React, Angular, and Vue being the most popular frameworks, were analyzed considering the survey report of Stack Overflow. React leads the market, with 35.9% of developers using it. However, Angular is surpassing the popularity of Vue.js. Angular is being chosen by 25.1% of developers, while Vue is utilized by 17.3% of developers.

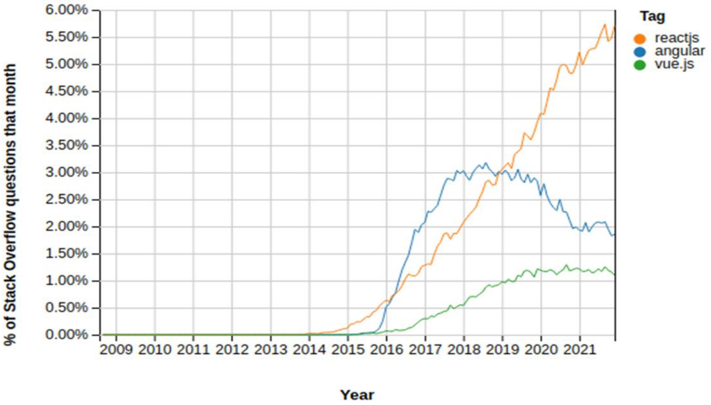


Fig 3. Stack overflow trends
Source : https://insights.stackoverflow.com/trends

Look at the demand represented in Google Trends for the last five years. The blue, red, and yellow lines represent React, Angular, and Vue.js.

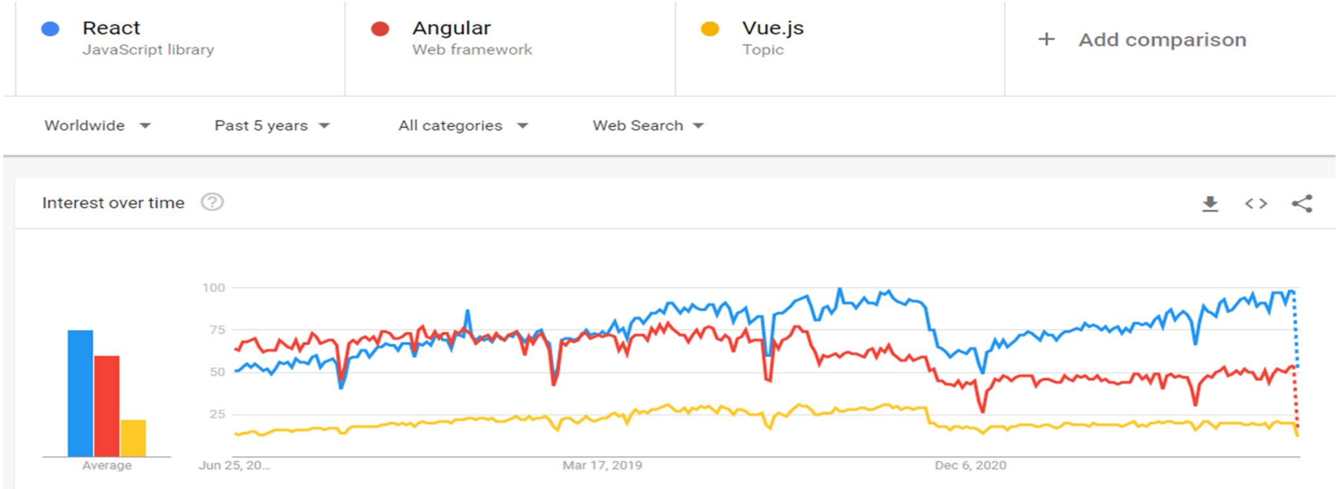


Fig 4. Google trends
Source : https://trends.google.com/trends

The chart shows a slight variation in the quantity of React and Angular inquiries from 2015 to 2016. Then, we observe that their difference grew over a bit time. The demand was balanced by the middle of 2017, at which point React started to grow and approach Angular's requirements. Although the Vue.js framework was still not well-known, it gradually gained its market share among frameworks, indicating the potential for future expansion. In the last years, React has surpassed Angular and presented evident popularity among developers, meaning that React is the most usable front-end framework on the market, followed by Angular.

Overall, React, and Angular develop organically with relatively the same dynamics. If we attempt to predict how popular these front-end frameworks will be, React shows a good trend, whereas Angular shows a minor fall. The demand for Vue.js development services is still unclear, but according to its specific convenient structure, it will grow as well, maybe just a bit less than the main frameworks.

According to the NPM trends, React, Angular, and Vue.js is the most downloaded JavaScript frameworks, but the battle between these frameworks will continue in 2022. If the downloads of all three web development frameworks are estimated from the last two years, undoubtedly, React is leading while Vue.js downloads prove to be higher than Angular.

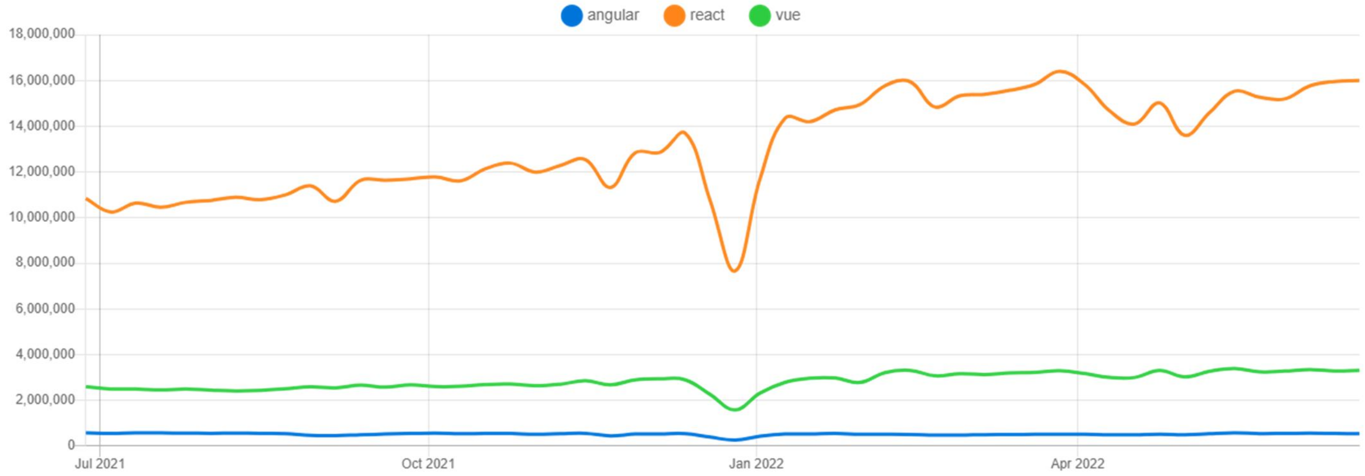


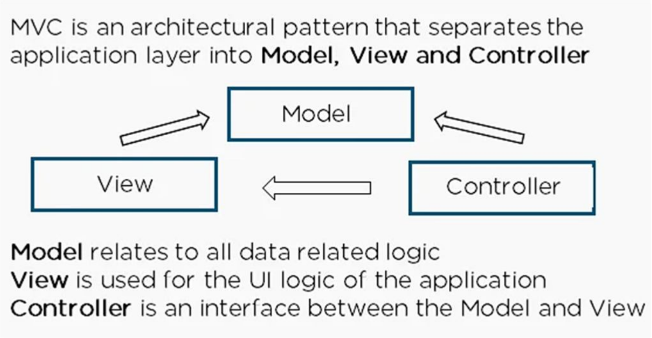
Fig 5. NPM Trends

Source : https://npmtrends.com/angular-vs-react-vs-vue

B. Architecture Involved

Angular has the full-fledged Model-View-Controller (MVC) architecture. Angular is based on MVC, making it super easy to modify and upgrade personal views without completely rewriting the other ideas. In Angular, we do not need to go through packages and then select from those. Instead, one can start working directly without worrying about adding functionalities via different packages.

MVC is an object-oriented design principle that mandates the separation of control, meaning one thing can do only one thing. It is the traditional way of separating even the most complex UIs into three views – Business Logic, UI Logic, and Input Logic. The execution is simple. [3]


MVC is an architectural pattern that separates the application layer into **Model, View and Controller**

Model relates to all data related logic
View is used for the UI logic of the application
Controller is an interface between the Model and View

Fig 6. MVC Architecture

Source : https://www.simplilearn.com/react-vs-angular-vs-vue-article

So, the model deals with entity classes, the data, and business logic. In contrast, the view deals with the representation and UI logic of the application where the user interacts with the web application. By interpreting user input as an event, the controller serves as a bridge between the View and Model and drives the functionality of the two. Any changes can be easily factored in and isolated, making testing much more accessible.

React represents the MVC framework's View component. React is incredibly versatile and can be integrated with several packages that have been created for it, which aids in the development of a whole application. It is thus a benefit in this situation. At the same time, Vue is focused primarily on the view and model layer of the MVC framework. Still, the packages developed for Vue are lesser than that of React due to its lower popularity, making it challenging to select packages that provide functionality relevant to a particular scenario.

C. Features

1) Angular

Angular is the "biggest" framework of the three. It is sometimes even called a "platform" rather than a framework. Because Angular comes with support for a variety of features. It helps

- Controlling the UI
- Reacting to user input
- Validating user input in forms
- Routing, state management
- Sending Ajax HTTP requests
- Providing offline support & PWA capabilities
- Ensure end-to-end testing, building application
- Managing multiple applications and connecting them

The best thing about Angular is its constant updates. Angular launches an update every six months, and the new versions build upon the older ones. Take the Angular 11 update, for example, eliminating all the previous version's bugs. We must monitor updates because a significant update may impact the code. However, Google waits another six months after an update before pulling the previous tools, providing a whole year to change code if the need arises.

All frameworks aim to make it easier for developers to build reactive, complex user interfaces. Nevertheless, Angular gives us the complete set of tools for that. It provides DOM manipulation support, and In addition, there is an official CLI that helps create and manage Angular projects, keep them up-to-date, add dependencies, and even deployments!

At its core, Angular is about building reusable user interface components. We can then control it with Angular and combine it with other members to create an entire user interface from those Angular-controlled components.[4]

2) React

This open-source JavaScript library has become quite the rage for developing interactive web and mobile apps. React library helps developers with the following features -

- Code Reusability
- Ease-of-use
- Customizable - Control is the primary area where the library and framework diverge. React has an advantage over Angular in this area because it is very customizable. Unlike Angular, which does not allow much modification, the user is in control and incorporates the needed library parts.
- React uses a virtual DOM that renders nodes when needed to maximize efficiency.
- Another benefit of this framework is server-side rendering for some implementations, such as content-focused applications.
- Offers tree shaking and bundling support to reduce the user's resource load.
- SEO Friendly - The clean and simple code of React makes the page load time of a website almost negligible. Moreover, page load time is critical as the bounce rate shoots up to 38% if the load time is more than 5 seconds!
- React functions by rendering content to the DOM and controlling it efficiently. It is also all about components and building user interfaces from parts. Additionally, it provides us with all the "tools" necessary to specify what should be rendered in which manner and under what conditions.

However, It does not have built-in functionality for form validation. It does not ship an HTTP client or come with a router (for rendering various components in response to URL changes). Although not for all circumstances, it has some state management support built-in. Additionally, it lacks any additional unique features and is unquestionably "slimmer" than Angular. We must rely on the (arguably quite active) React community for those features.[5]

3) *VUE*

Vue is a systematic framework which kind of sits between React and Angular. It is not as "big" as Angular but includes more features than React.

- We have built-in state management with Vue, and it also comes with a router. However, neither HTTP client functionality nor form validation is included.
- Third-party simulations have demonstrated that Vue outperforms Angular and React when manipulating the rows and columns of tables. It is because Vue, unlike Angular, uses a virtual DOM that is lighter than the one used by React.
- Debugging in Vue runs parallel to the coding process. The UI can be visualized when code takes the form while it is written, thus helping developers to debug their code easily.
- It is a tiny framework of only 21KB, so download and startup take next to no time. Even the code in Vue is short and straightforward, which allows the time-efficient development of applications and templates.

The foundation of Vue is all about creating user interfaces by integrating reusable components, just like Angular and React. Nevertheless, beyond that, it gives a bit more than React and a bit less than Angular.[6]

D. *Performance*

Performance is a very crucial feature. The moment we build an HTML view and render it into the browser, this view to the user is just HTML tags. However, the code is converted into a balanced tree structure if we look at the internal working. This representation of code in the form of a tree data structure is called DOM.

The W3C designed the DOM API. Document Object Model, or DOM, is an acronym that describes how a document may be viewed and modified; it transforms HTML code into a balanced tree structure. DOM contains a node for each UI element present in the web document. It is beneficial as it allows web developers to modify content through JavaScript. Additionally, the fact that it is significantly structured aids in our ability to select specific targets and makes working with the entire body of code much more superficial. The DOM is updated each time the state of the application changes so that it is reflected in the user interface.

This B-Tree algorithm is high-speed, so when the user makes any changes to the UI, the changes are reflected immediately. This is facilitated by the B Tree algorithm, so as a result, the performance is improved. Thus, updating the tree here is not a costly operation; indeed, we have many algorithms on trees to make the updates fast. What is proving to be expensive is that the changed element and its children must be rendered anew to refresh the UI of our page every time the DOM is updated. This way, the DOM and UI components must be updated whenever a member is altered. [7]

These things occur when we write code in the JavaScript file or the console:

- The browser parses the HTML to find the node associated with this id. It eliminates the specific element's child element.
- Adds the "updated value" to the element (DOM).
- Updates the parent and child nodes' CSS values.
- Refresh the design.
- Finally, traverse the tree and draw it on the browser's screen.

Thus, DOM manipulation plays a vital role in the performance analysis of the frameworks. There are two different kinds of performance on which the frameworks are assessed:

1) *Startup Performance*: It signifies how fast does the web app (web page) load and become interactive (= usable by the user)? The size of the produced code bundle, consisting of the input code and the framework code combined, significantly impacts startup performance. These factors give us an idea of which frameworks are feasible when a faster initial load time is needed. Also, the overall bundle size, i.e., the code generated and uploaded to the deployment server, matters most here. In small application comparisons, Angular tends to be a bit bigger than Vue and React apps, with Vue being smaller than React. For more prominent apps, all three frameworks should produce roughly equally-sized code bundles - assuming the optimization techniques are implemented like "Lazy Loading," which all three frameworks support.

2) *Runtime Performance*: It is concerned with how fast the app is up and running (i.e., how well does it re-render, updates the UI, reacts to user input, and other tasks.)? The internals of the chosen framework and how it handles DOM manipulation and updating have the most significant impact on runtime performance. The Virtual DOM, maintained in memory by React, is a simplified version of the Real DOM. For every object in the original DOM, there is an object in React Virtual DOM. It is the same, but it does not have the power to change the document's layout directly. Virtual DOM manipulation occurs rapidly compared to physical DOM manipulation since no images are produced on the screen. As a result, the virtual DOM rather than the real DOM is updated first whenever the state of our application changes. A new Virtual DOM tree is constructed each time the state of any element changes. The differences between the new Virtual DOM tree and the old One are then noted. The optimal technique to make these changes to the real DOM is then determined. Only the changed items will now be displayed once more on the page. [7]

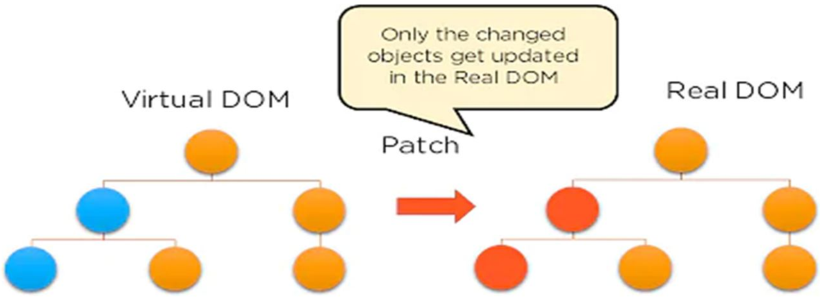


Fig 7. DOM tree updated
Source : https://www.simplilearn.com/react-vs-angular-vs-vue-article

At any given moment, React maintains two Virtual DOMs, one of which is the updated Virtual DOM and the other simply the pre-update version of the updated Virtual DOM. Now it analyses the Virtual DOM before and after the update to determine the changes, such as which components are in the DOM. "Diffing" is the process of contrasting the current Virtual DOM tree with the previous one. React only updates those items on the real DOM once it determines that. The real DOM is updated by React using batch updates. It simply means that batches of changes to the actual DOM are transmitted rather than single update every time a component’s state changes. React accomplishes this most effectively by guaranteeing that the Real DOM receives batch updates to re-render the UI, which is the most expensive component of the process. Reconciliation is the term used to describe the entire process of converting changes to the actual DOM. The performance is greatly enhanced as a result.

Angular, on the other hand, uses real DOM. When rendering the web application, Angular is slower than React and Vue because it uses Real DOM rather than a Virtual DOM. This means that Angular updates the entire DOM tree structure whenever an object changes.

Virtual DOM (VDOM) is a React concept that Vue has incorporated. The rendering time is faster than Angular and comparable to React since Vue updates only those objects modified in the "actual" DOM.

Since no time is spent making unnecessary changes to the entire tree React is extremely fast. On the other hand, accurate DOM updating is time-consuming, risky, and error-prone. Vue also ensures quicker and bug-free performance.

E. *Data Binding*

The process that links the application's user interface (UI) to the models is called data binding. Using Data Binding, the user can manipulate the elements present on the website using the browser. Therefore, whenever some variable changes, that particular change must be reflected in the Document Object Model or the DOM. The incremental display of a webpage makes data binding convenient when pages contain an extensive amount of data.

The relationship between the components and the DOM is specified via data binding in Angular. Data binding is a feature of all Angular versions, from AngularJS to the most recent Angular 9 version. Two-way data binding, which Angular supports, enables the application to transfer data in both directions—from the components to the templates and vice versa. By doing this, the application's models and views are guaranteed to be constantly synced. Setting the element property and listening to element change events are two actions that two-way data binding will carry out. [8]

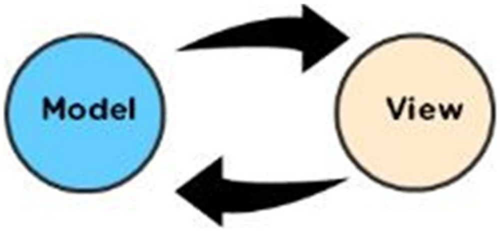


Fig 8. Two way data binding
Source : https://www.simplilearn.com/react-vs-angular-vs-vue-article

AngularJS templates work differently. The template is first compiled on the browser using the uncompiled HTML and any extra markup or directives—the process of compilation results in a live view. Any modifications to the view are promptly updated in the model, and model modifications are transmitted to the view. The model serves as the sole source of truth for the application's state, substantially streamlining the developer's programming interface. The view can be assumed simply as an instant projection of the model. This ensures that the framework can connect the DOM to the Model data with the help of the controller. Hence, in Angular, any changes made to the view are immediately visible in the model part of the architecture, and this approach adversely affects the performance.

ReactJS renders components to the user interface, and the component's logic contains the data seen in the view. (UI). The connection between the data displayed in the view and the component's logic is called data binding in ReactJS.
One-way Data Binding: One-way data binding is used in ReactJS. One of the requirements listed below can be met in one-way data-binding:

- *Component to View:* Any modification to the component's data would be mirrored.
- *View to Component:* Any changes to the view would be mirrored in the component's data.

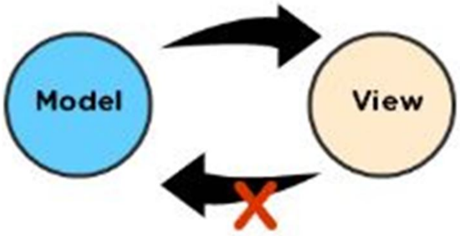


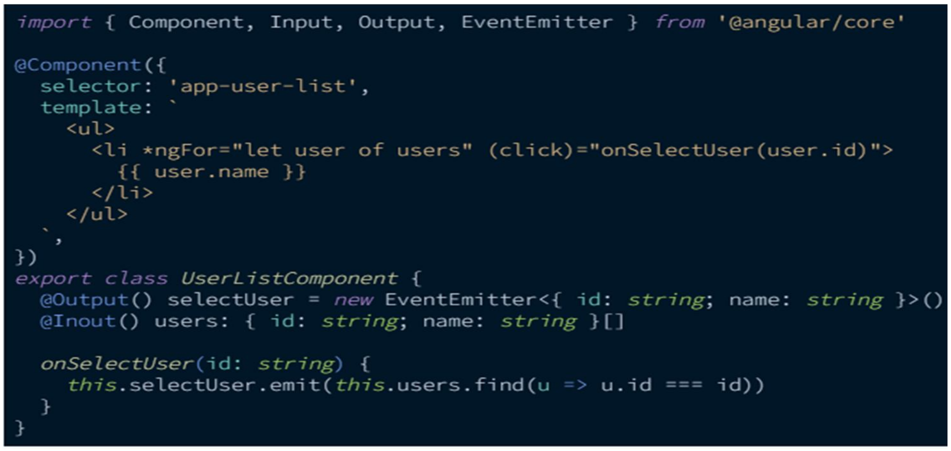
Fig 9. One way data binding
Source : https://www.simplilearn.com/react-vs-angular-vs-vue-article

The data can be taken from the view and push it to the model. The view adjusts whenever the model state is updated. Thus, it makes react modular and fast and improves its performance.[8]
Vue uses reactive two-way data binding. It automatically selects the proper update mechanism for the element based on the input type.

F. Code syntax

1) Angular

Angular projects use Typescript, a superset (i.e., an addition) to JavaScript. Although Typescript cannot be run in a browser, Angular projects come with tools that will internally translate Typescript code into JavaScript that can be used in browsers. Being a statically typed language, Typescript, we must provide the variable's type. Typescript is friendlier for developers with an object-oriented programming (OOP) background.
This is how a typical Angular code sample would seem:



```
import { Component, Input, Output, EventEmitter } from '@angular/core'

@Component({
  selector: 'app-user-list',
  template: `
    <ul>
      <li *ngFor="let user of users" (click)="onSelectUser(user.id)">
        {{ user.name }}
      </li>
    </ul>
  `,
})
export class UserListComponent {
  @Output() selectUser = new EventEmitter<{ id: string; name: string }>()
  @Inout() users: { id: string; name: string }[]

  onSelectUser(id: string) {
    this.selectUser.emit(this.users.find(u => u.id === id))
  }
}
```

Fig 10. Angular code snippet

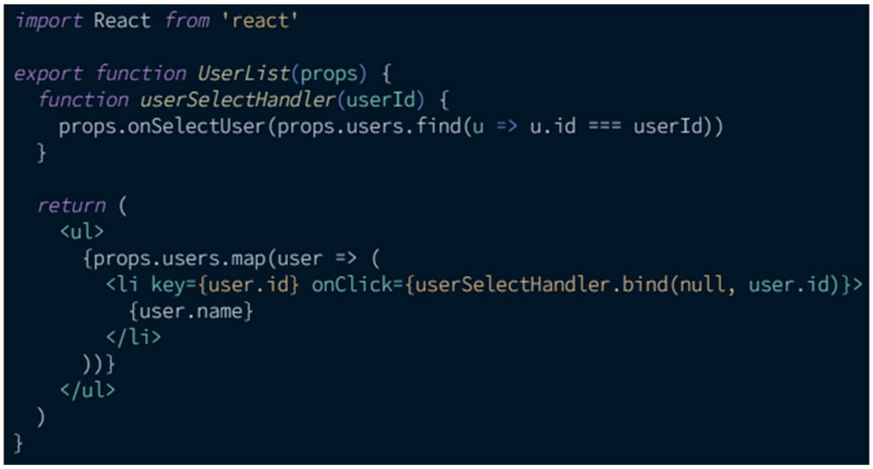
This is a reasonably trivial component (UserListComponent) that uses property binding (@Input ()) to expose data to be set "from outside" (i.e., from inside the component, this component will be used in), event binding (@Output ()) to emit events and a directive - ngFor - to render a list of data. We can see that Angular uses a Typescript feature called "Decorators" (@Component) to attach extra data to standard classes (UserListComponent). We can write code like this, and Angular, behind the scenes, takes care of manipulating and "connecting" the real DOM.

So a developer never has to write any code that directly creates or removes elements to/ from the DOM. We can instead create Angular components with the syntax shown above to let Angular do the heavy lifting. We can define inputs (properties) and outputs (events) of components and also can manage some component-specific or app-wide states with the help of Angular (not visible in the above snippet).

To run various tests, Angular uses Jasmine. The Jasmine framework enables several functions to create multiple test scenarios.

2) *React*

React combines UI templates and JavaScript logic, a concept that no framework had implemented before. The result is called the JSX. React uses a component that contains both the markup and logic in the same file. React only requires the knowledge of JavaScript.



```
import React from 'react'

export function UserList(props) {
  function userSelectHandler(userId) {
    props.onSelectUser(props.users.find(u => u.id === userId))
  }

  return (
    <ul>
      {props.users.map(user => (
        <li key={user.id} onClick={userSelectHandler.bind(null, user.id)}>
          {user.name}
        </li>
      ))}
    </ul>
  )
}
```

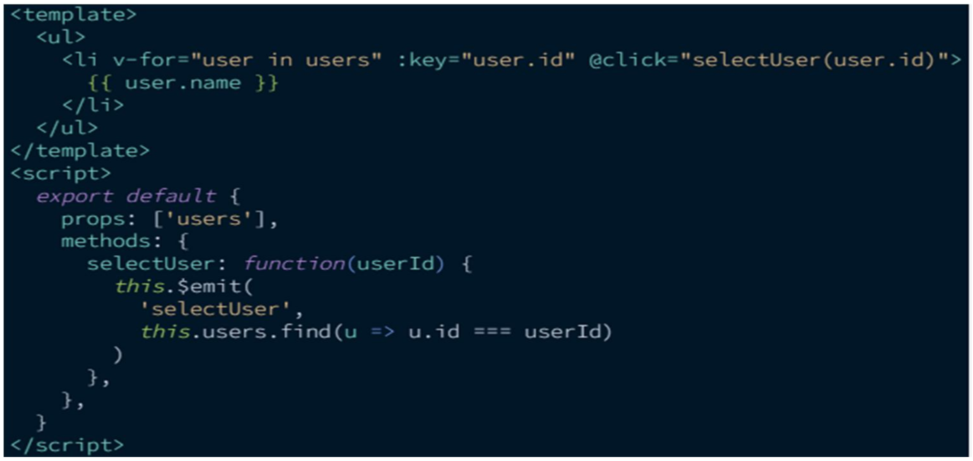
Fig 11. React code snippet

React typically uses JavaScript (though you also can use it with Typescript - for example, in my Ionic + React course), and it uses a special JavaScript "feature," which is called "JSX."JSX is not part of the JavaScript language. React projects are set up such that this "HTML in JavaScript" syntax is supported during development. Like Typescript in Angular projects, JSX is compiled into regular, browser-friendly JavaScript behind the scenes once we build the project.

Just as in the Angular snippet, we built a UserList component here - however, without decorators but instead as a simple JavaScript function that uses a concept called props to receive input and invoke events (props.onSelectUser ()).
For testing the code, React utilizes Jest. Facebook uses Jest to run tests on React, is embedded in every React library, and requires zero configurations.

3) Vue

HTML, JS, and CSS are used individually by VueJS. Because VueJS uses a template-based approach, novice developers can quickly use this programming paradigm.



```
<template>
  <ul>
    <li v-for="user in users" :key="user.id" @click="selectUser(user.id)">
      {{ user.name }}
    </li>
  </ul>
</template>
<script>
  export default {
    props: ['users'],
    methods: {
      selectUser: function(userId) {
        this.$emit(
          'selectUser',
          this.users.find(u => u.id === userId)
        )
      },
    },
  }
</script>
```

Fig 12 Vue code snippet

Vue uses regular JavaScript (optionally, we can also use Typescript) and typically utilizes "Single File Components. “It is a framework all about components, after all (just like the other two), but like Angular, it splits template (i.e., HTML) and JavaScript logic apart. Vue also supports JSX, but we can see the code as written above in most projects. This separation can be easier to grasp since we do not mix JavaScript with HTML here. Unit testing can be done with Jest or Mocha using the built-in capabilities of the Vue CLI. Writing unit tests for isolated components is simple using Vue's single-file components.

IV. CONCLUSION

This paper examined the appropriateness of front-end technologies for creating Web applications. The front-end frameworks Angular, Vue.js, and React were described. React, Angular 2, and Vue are compared in a variety of ways, including data binding, DOM manipulation, features, popularity, code syntax, and performance, among other things. The criteria for the selection of a specific framework were identified. It is safe to say that all three have their perks- Angular is robust and time-tested, React is flexible and quick, and Vue is simple and high-performing. The development of Angular, React, and Vue is all highly active. They maintain the current versions while routinely releasing new ones. It is crucial to note that Vue while having started more recently appears to be growing significantly, whereas Angular is not expanding as quickly as it formerly did.
The three frameworks can be used for web development, depending on the viability and goals of the application.

REFERENCES

[1] https://www.interviewbit.com/blog/framework-vs-library/

[2] Irudayaraj, Prathap & P., Saravanan. (2019). Evolution of the Single Page Application in the modern web application development. Innovative Food Science & Emerging Technologies. 6. 141-145.

[3] https://www.simplilearn.com/react-vs-angular-vs-vue-article

[4] W. Chansuwath and T. Senivongse, "A model-driven development of web applications using AngularJS framework," 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), 2016, pp. 1-6, doi: 10.1109/ICIS.2016.7550838.

[5] Pratik Sharad Maratkar , Pratibha Adkar "React JS - An Emerging Frontend JavaScript Library" Iconic Research And Engineering Journals Volume 4 Issue 12 2021 Page 99-102

[6] Song, Junhui & Zhang, Min & Xie, Hua. (2019). Design and Implementation of a Vue.js-Based College Teaching System. International Journal of Emerging Technologies in Learning (iJET). 14. 59. 10.3991/ijet.v14i13.10709.

[7] Xing, Yongkang & Huang, JiaPeng & Lai, YongYao. (2019). Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development. 68-72. 10.1145/3313991.3314021.

[8] https://www.geeksforgeeks.org/difference-between-one-way-binding-and-two-way-binding/