

Analytical Comparison of Front-End Frameworks For Web-Based Applications

Jothiprabha A S¹, Dr Anita Sofia V S²

¹Dept of MCA

²Associate Professor, Dept of MCA

^{1, 2} PSG College of Arts & Science, Coimbatore, India

Abstract- A crucial stage in the development of any software or application involves the selection of the right front-end framework or library. Front-end web development often revolves around JavaScript frameworks, which provide various options for addressing web development needs. Among these choices, Vue, React, and Angular are all part of the JavaScript framework landscape. Utilizing these readily available frameworks, developers can effectively construct web applications while taking into account practical considerations. This paper delves into the pros and cons of core elements and distinctive features of these frameworks.

Furthermore, it provides an extensive examination of the research conducted on front-end frameworks. This research offers an insight into the front-end frameworks identified in the literature, elucidating the key elements of these frameworks through the utilization of a systematic literature review as its research methodology. The study focused on the three most widely adopted frameworks, namely Vue.js, Angular, and React, to assess their essential characteristics.

Keywords- Frameworks, React, Angular, Vue

I. INTRODUCTION

Contemporary web applications have assumed a pivotal role in the daily lives of individuals. Applications have unquestionably permeated various sectors, ranging from online ticket booking and meal ordering to ride-hailing and e-commerce. The advent of technology and the rise of mobility have substantially reshaped consumer behavior. Consequently, businesses are eagerly anticipating their online presence in 2022, recognizing the far-reaching influence of mobility on global industries.

Within the realm of web development, every frontend developer is well-acquainted with three prominent frameworks: React, Vue.js, and Angular. These frameworks not only define the fundamental principles and best practices for project development but also simplify several aspects such as code organization and maintenance, essentially serving as

the underlying structure for our applications. From a technical perspective, all three frontend components are essentially lines of code that the web browser subsequently renders to create visually appealing web pages. A software framework, whether for the frontend or backend, encompasses pre-existing, standardized code that expedites and streamlines the development of specific functionalities. Angular serves as a comprehensive front-end framework, React operates as a UI library, and Vue.js is categorized as a progressive framework. While these tools can be used interchangeably for constructing front-end applications, they are not identical in every aspect. Thus, it's valuable to compare them and grasp their distinctions. All three frameworks are rooted in component-based development, facilitating the swift creation of UI elements. Nevertheless, they exhibit varying structures and architectural principles. Given the abundance of available applications, developing a successful one entails more than just complexity; it necessitates a specific strategy, meticulous planning, and access to cutting-edge technologies for crafting enterprise-level applications. The primary objective of this paper is to assess the suitability of these frameworks for web application development and pinpoint key attributes that enable more personalized and effective web application creation.

II. ASSOCIATED CONCEPTS

Efficient web app development necessitates a pragmatic framework, and the landscape of JavaScript frameworks is evolving rapidly. Today, we witness the frequent updates and advancements of Angular, React.js, and Vue.js. These technology stacks are gradually supplanting traditional server-side technologies. We have transitioned from server-side applications to more distributed ones, where some tasks are executed on the client-side and others on the server-side. Presently, we have embraced single-page applications (SPAs), and all these technologies are firmly rooted in JavaScript, rendering them lightweight and free from reliance on third-party runtime tools. It is worth noting that React is a JavaScript library for front-end development, whereas Angular and Vue are full-fledged JavaScript frameworks. While the terms "library" and "framework" are

sometimes used interchangeably, they bear significant distinctions. A library comprises a collection of classes and functions that users can freely invoke and control. In contrast, frameworks resemble templates with predefined structures and limited options, providing users with a predetermined blueprint, which means they have less flexibility and control.

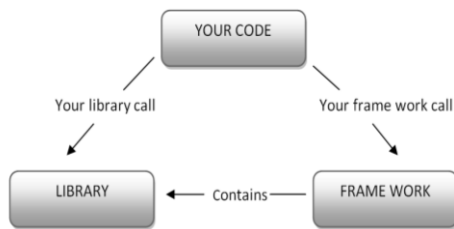


Fig.2.1. Functioning of Library and Framework

The fundamental distinction between a library and a framework hinge on a concept known as "Inversion of Control," often abbreviated as IoC. When we utilize methods from a library, we retain control and make the calls ourselves. Conversely, in the case of a framework, the control is inverted, meaning the framework initiates the calls. The program establishes a framework or structure and outlines its features, which are then filled in accordingly. JavaScript, being an interpreted language that executes code line by line, imparts high efficiency to these tools. This efficiency is achieved by sidestepping the need for intermediary code transformation and a runtime environment, which would introduce delays. React finds applications in the development of both web and mobile applications, whereas Angular is typically employed for building single-page applications (SPAs). Additionally, the stack is suitable for creating hybrid applications due to its reliance on fundamental web technologies like HTML, CSS, JavaScript, and jQuery. Vue is another framework used for crafting SPAs and designing user interfaces. SPAs represent a contemporary approach to web application development, particularly well-suited for applications with relatively limited content. In a single-page application, only the necessary information is reloaded for the user, as opposed to a multi-page application where every requested page and the entire web page content are updated. When a user interacts with a SPA, the server loads the complete page initially, and subsequently, only essential data is transmitted in the form of JSON files in response to specific requests.[2]

III. POSSIBLE COMPARISON DIMENSION

3.1. Current Trends and Popularity:

When it comes to programming languages, JavaScript stands out as a versatile choice with a vast array of functionalities and multiple development environments. Opting for widely-adopted programming frameworks and libraries holds several compelling advantages:

1. **Sustainability:** Renowned frameworks are less prone to disappearing, as their core teams are more likely to ensure ongoing maintenance.
2. **Community Support:** Popular frameworks boast thriving communities that continually develop complementary libraries and tools, enhancing the overall ecosystem.
3. **Abundant Resources:** Accessing resources like tutorials, forum discussions, and documentation is considerably easier when working with well-liked libraries and frameworks.

In terms of popularity, we can turn to the Stack Overflow survey report for insights. Among the top three frameworks—React, Angular, and Vue—React commands the largest market share, with 35.9% of developers favoring it. Angular, on the other hand, is currently surpassing Vue.js in terms of popularity, being the choice of 25.1% of developers, while Vue.js is utilized by 17.3% of developers.

Examine the demand trends depicted in Google Trends over the past five years. The lines colored in blue, red, and yellow correspond to React, Angular, and Vue.js, respectively.

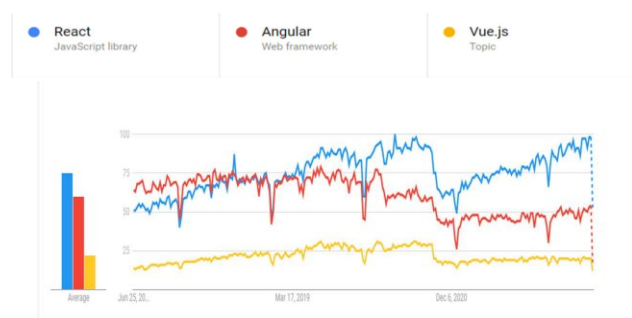


Fig.3.1. GOOGLE TRENDS

The chart illustrates a subtle shift in the volume of inquiries related to React and Angular between 2015 and 2016. Afterward, there was a noticeable increase in the disparity between the two, but by mid-2017, the demand had leveled out. At this juncture, React embarked on a growth trajectory, gradually closing in on Angular's popularity. Meanwhile, Vue.js, although initially less recognized, steadily carved out its niche among frameworks, indicating potential for future expansion. In recent years, React has not only overtaken Angular but has established itself as the frontrunner

among developers, solidifying its position as the most widely used front-end framework. Angular follows closely behind in terms of popularity. Overall, React and Angular have exhibited organic development patterns with fairly similar dynamics. When we attempt to forecast the future popularity of these front-end frameworks, React shows a promising trend, while Angular appears to experience a modest decline. The demand for Vue.js development services remains uncertain, but its well-thought-out structure suggests growth potential, albeit possibly at a slightly slower pace compared to the leading frameworks. According to NPM trends, React, Angular, and Vue.js are the most frequently downloaded JavaScript frameworks, and this competition is expected to persist in 2022. If we assess the download statistics of these web development frameworks over the past two years, React clearly takes the lead, while Vue.js demonstrates higher download figures than Angular.

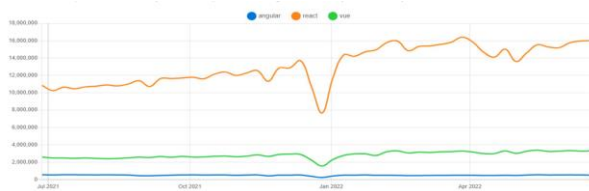


Fig.3.2NPM trends

3.2.Characteristics

3.2.1. ANGULAR

Among the three options, Angular is the most extensive, and it is often referred to as a "platform" rather than just a framework. This distinction arises from Angular's comprehensive support for a wide range of functionalities, including:

- Managing the User Interface (UI)
- Handling User Input Responsiveness
- Validating User Input in Forms
- Implementing Routing and State Management
- Facilitating Ajax HTTP Requests
- Enabling Offline Support and Progressive Web App (PWA) Capabilities
- Ensuring Comprehensive End-to-End Testing
- Streamlining Application Building
- Managing and Connecting Multiple Applications

Angular's robust capabilities make it a versatile and powerful tool for developing complex web applications. One of Angular's standout qualities is its consistent and frequent updates. Angular releases update every six months, with each new version building upon the foundations of its predecessors.

For instance, Angular 11 was a notable update that addressed and resolved issues from prior versions. It is important to keep an eye on these updates because significant changes can impact your existing codebase. However, Google adopts a considerate approach by maintaining the previous tools for a full year after introducing a new update, allowing developers ample time to adapt their code if necessary.

While all frameworks share the goal of simplifying the development of responsive and intricate user interfaces, Angular stands out by offering a comprehensive suite of tools. It not only supports DOM manipulation but also provides an official Command Line Interface (CLI). This CLI assists in creating and managing Angular projects, keeping them up-to-date, adding dependencies, and even handling deployment tasks. At its core, Angular is all about constructing reusable user interface components. These components can be efficiently managed with Angular and seamlessly combined with other elements to craft complete user interfaces, leveraging the power of Angular-controlled components.[4]

3.2.2. React

This popular open-source JavaScript library has gained significant popularity for its role in developing interactive web and mobile applications. React offers developers a range of features, including:

1. **Code Reusability:** React promotes the reuse of code, enhancing development efficiency.
2. **Ease-of-use:** It provides a user-friendly and intuitive development experience.
3. **Customizability:** React shines in terms of customization. Unlike Angular, which has limitations in this regard, react allows users to have more control and integrate specific library components as needed.
4. **Virtual DOM:** React leverages a virtual Document Object Model (DOM), rendering nodes when necessary to optimize performance.
5. **Server-side Rendering:** It offers server-side rendering capabilities for certain implementations, particularly content-focused applications.
6. **Tree Shaking and Bundling:** React supports tree shaking and bundling to reduce the resource load on users.
7. **SEO-Friendly:** React's clean and straightforward code contributes to minimal website page load times. This is crucial, as a load time exceeding 5 seconds can lead to a significant 38% increase in bounce rates.
8. **Efficient Content Control:** React efficiently renders content to the DOM and revolves around component-based development. It equips developers with the necessary tools to specify what should be rendered, how it

should appear, and under what conditions it should be displayed.

In essence, React offers developers a flexible and efficient toolkit for creating user interfaces and web applications while emphasizing code reusability, customization, and optimal performance. However, React does not include native functionality for form validation, nor does it provide an integrated HTTP client or a built-in router for handling component rendering based on URL changes. While it offers some built-in state management support, it does not introduce any distinctive, standalone features and is undeniably leaner compared to Angular in terms of out-of-the-box features. To address these needs, React developers often depend on the robust and vibrant React community for the addition of these functionalities.[5]

3.2.3. VUE

Vue is a structured framework that occupies a middle ground between React and Angular. It doesn't carry the same level of complexity as Angular, but it offers more features compared to React.

- Vue includes built-in state management and comes with a router. However, it does not provide HTTP client functionality or built-in form validation.
- Third-party tests have shown that Vue outperforms both Angular and React when it comes to manipulating rows and columns in tables. This advantage can be attributed to Vue's use of a lighter virtual DOM compared to React and Angular.
- Debugging in Vue happens concurrently with the coding process, allowing developers to visualize the user interface as they write code, which simplifies the debugging process.
- Vue is a lightweight framework, with a small footprint of only 21KB. This means that downloading and starting up Vue applications is incredibly fast. Additionally, Vue code is concise and straightforward, facilitating efficient development of applications and templates.

At its core, Vue focuses on creating user interfaces through the integration of reusable components, akin to Angular and React. However, it offers a bit more in terms of features compared to React and is somewhat less complex than Angular.[6]

3.3. PERFORMANCE

Performance is a critical aspect of web development. When we create an HTML view and display it in the browser, the user sees it as a collection of HTML tags. However, internally, the code is transformed into a balanced tree structure known as the Document Object Model or DOM. The DOM API, designed by the W3C, defines how a document can be viewed and modified. It essentially converts HTML code into this structured tree format. Within the DOM, there exists a node for each UI element present in the web document. This structure proves advantageous as it empowers web developers to manipulate content through JavaScript. Furthermore, its well-organized nature simplifies the selection of specific elements, making it more efficient to work with the entire codebase. The DOM undergoes updates each time the application's state changes, ensuring that these changes are reflected in the user interface. The B-Tree algorithm, known for its high-speed capabilities, plays a pivotal role here. When users make alterations to the UI, these changes are immediately reflected. This rapid updating is facilitated by the B-Tree algorithm, which ensures that updating the tree structure is not a resource-intensive operation. Numerous tree algorithms are available to expedite updates. However, what tends to be resource-intensive is the need to re-render the changed element and its child components to refresh the webpage's user interface each time the DOM is updated. Consequently, both the DOM and UI components must be updated whenever a part of the page undergoes modification.[7]

When we write code in a JavaScript file or the console, several actions take place:

1. The browser parses the HTML to locate the node corresponding to the provided ID. It then removes any child elements of that specific element.
2. The browser adds the "updated value" to the element within the Document Object Model (DOM).
3. CSS values for both the parent and child nodes are updated as needed.
4. The browser refreshes the visual design to reflect the changes.
5. Lastly, it traverses the DOM tree and renders it on the browser's screen for the user to see.

3.4. BINDING DATA

Data binding connects an app's UI and models, enabling user interaction via web browsers. It updates the DOM when variables change. Angular uses it to link components and the DOM, ensuring real-time synchronization. Angular has featured data binding since AngularJS to Angular 9, including two-way data binding for

bidirectional data exchange between components and templates, ensuring continuous synchronization by setting element properties and listening to change events. AngularJS templates compile in the browser, creating a dynamic view reflecting the template. Changes in the view instantly update the model, and vice versa, creating a real-time connection between the model and the view. In AngularJS, the model is the sole source of truth, simplifying development with a direct link between the view and model via the controller. However, immediate view-model synchronization can impact performance as view changes are instantly reflected in the model. In ReactJS, components render UI content, and their logic defines the displayed data, referred to as data binding.

ReactJS predominantly employs one-way data binding. This means that data can flow in one of two directions:

1. Component to View: When there is a modification to the component's data, it is reflected in the view.
2. View to Component: Any changes made in the view are mirrored back in the component's data.

Data can flow bidirectionally between the view and the model, enabling data to be retrieved from the view and pushed to the model. As a result, when there are updates to the model's state, the view adapts accordingly. This characteristic contributes to React's modularity, speed, and enhanced performance. On the other hand, Vue employs a reactive two-way data binding approach. It intuitively selects the appropriate update mechanism for each element based on its input type.

IV. CONCLUSION

This paper undertook an assessment of the suitability of various front-end technologies for building web applications, with a particular focus on the front-end frameworks Angular, Vue.js, and React. A comparative analysis was conducted across multiple dimensions, including data binding, DOM manipulation, features, popularity, code syntax, and performance, among others, for React, Angular 2, and Vue. The paper identified the criteria guiding the selection of a specific framework.

It's worth noting that each of these three frameworks offers its own advantages: Angular boasts robustness and a proven track record, React is celebrated for its flexibility and speed, while Vue stands out for its simplicity and high performance. Furthermore, all three frameworks are actively developed, with continuous updates and releases. It's important to highlight that Vue, despite its more recent

emergence, exhibits significant growth, whereas Angular's expansion rate has slowed compared to its earlier trajectory.

Ultimately, the choice among these three frameworks depends on the suitability and objectives of the specific web development project at hand.

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] Difference between One-way Binding and Two-way Binding - GeeksforGeeks