IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 10, October 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

© 2024 IJIRCCE | Volume 12, Issue 10, October 2024| DOI: 10.15680/IJIRCCE.2024.1210051

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Research and Analysis of the Front-End Frameworks and Libraries in Web Development

Mohanapriya, Keerti Jadhav, Jarshini M

Assistant Professor, Department of Computer Science & Applications, The Oxford College of Science, Bangalore, India

MCA Students, Department of Computer Science & Applications, The Oxford College of Science, Bangalore, India

ABSTRACT: With the rapid growth of online technology in recent years, HTML5 is poised to become a global standard in web development, potentially dominating the front-end landscape of the internet. However, there are many front-end development frameworks available to choose from, including libraries like Angular and React. Selecting the right framework or library is crucial when launching an e-business and enhancing its user experience, making this a key decision in web development.

This paper starts by introducing a list of the most popular frameworks and libraries used in front-end development, along with an analysis of web performance services. It reviews research findings from multiple perspectives, discussing the advantages and disadvantages of each framework. The study concludes with a summary of the contributions made and offers insights into the future of front-end development.

I. INTRODUCTION

A. Preface

The past decade, the rapid advancement of internet technologies has made e-business a significant part of consumers' daily lives, whether for shopping, applying for mortgages, or filing taxes. One of the key drivers of this evolution has been the introduction of HTML5, which revolutionized the web development landscape. HTML5, a markup language, is used to create the structure and display content on websites globally. Compared to earlier versions, HTML5 introduces and improves several semantic elements, such as <footer>, <aside>, and <nav>, which help developers define the structure of a web page more clearly and adhere to stricter guidelines.

Additionally, HTML5 brings new features to application programming interfaces (APIs). For instance, the <canvas> element allows access to graphical sections on mobile devices, enabling developers to create websites with more advanced functionalities. While HTML5 offers numerous improvements, one downside is that its rendering efficiency can still be slow compared to older technologies like Flash. To address this, Google developed the Chrome V8 engine in 2008, which optimized the performance of JavaScript, placing it at the forefront of web development.

Before the Chrome V8 engine, JavaScript's primary role was to enhance user interfaces by interacting with Cascading Style Sheets (CSS) and handling basic script tasks like form validation. However, with the V8 engine being over 56 times faster than older versions of Internet Explorer (IE), it redefined the capabilities of JavaScript. Traditionally, web browsers generated JavaScript by parsing bytecode, compiling the entire project, and executing the code from a file system, making JavaScript execution slower compared to languages like Java or C++. The V8 engine introduced inline caching, improving JavaScript's speed without relying on traditional methods.

Following the release of the V8 engine, JavaScript began to perform on par with Java or C++ in terms of execution speed, allowing web applications to achieve similar performance levels as desktop applications. The V8 engine's success led to the development of various JavaScript platforms based on its architecture, marking a new chapter in the evolution of internet development.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

In 2009, Node.js was introduced, utilizing the V8 engine. Node.js expanded JavaScript's role from being limited to browser scripting to enabling the development of server-side applications. This allowed developers to build event-driven server applications with ease. Despite Node.js's release, numerous JavaScript frameworks have emerged in the past decade, shaping the future of the web. The following sections will explore the key front-end frameworks and libraries that have influenced modern web development.

B. Front-End Framework and Libraries

With the advent of the V8 engine, a variety of JavaScript-based front-end frameworks and libraries have been developed, each presenting distinct benefits. To identify the most widely adopted frameworks that adhere to industry standards, we analyze usage data from platforms such as Stack Overflow and GitHub. These platforms provide a clear picture of developer preferences worldwide. Frameworks like React.js, Angular, and Vue.js, along with others shown in the image, rank as the leading options for front-end development. By studying these trends, we can better understand the tools developers rely on to create responsive and effective web applications.

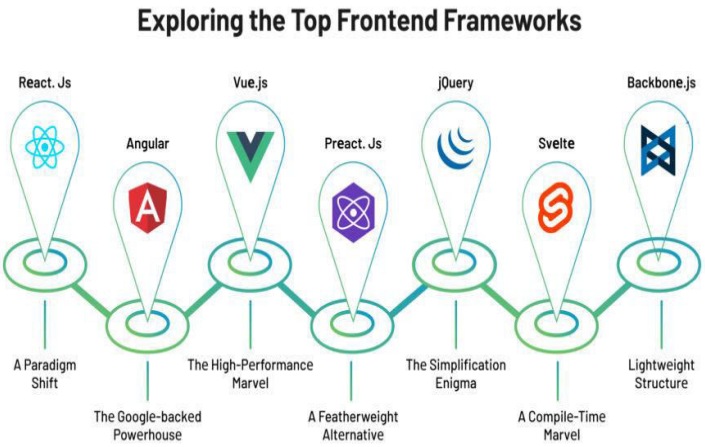

Figure 1 Exploring the Top Frontend Frameworks

1. Vue. js

Evan, you developed Vue.js, a popular JavaScript ES6-based open-source framework, in 2014. Vue's primary goal was to provide a simple API that allows for responsive data binding and the creation of user interface components. While initially designed for single-page applications, Vue.js has often been considered to have limited features, making it challenging to adopt for larger commercial applications. However, the open-source community has built a robust ecosystem of third-party libraries and packages, enabling Vue to handle complex single-page applications through tools like routing, state management, and build tooling.

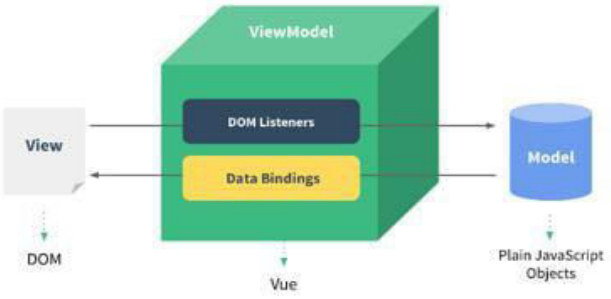

Figure 2 Data Driven Concept

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Vue.js operates with three key components to manage data, as illustrated in Fig. 2: View, View Model, and Model. The View component represents the website's content, displayed as the DOM. The View Model handles DOM listeners and data bindings, serving as an intermediary between the View and Model. When users interact with the View, Vue detects these actions through DOM listeners and updates the Model accordingly. Vue updates the website's content by adjusting the DOM using data bindings whenever there is a change in the Model's data. In essence, Vue.js achieves two-way data binding by combining one-way binding with DOM listeners.

2. Angular 1 & 2

Angular is a widely recognized open-source front-end web application framework based on JavaScript ES5, developed by Google in 2010. Its primary objective was to simplify the process of building persistent web forms for web designers. As front-end development has evolved, Angular's role has also expanded to meet more complex development requirements, enabling developers to create more sophisticated applications.

However, its basic design model caused Angular to lag behind other front-end frameworks in recent years. To address this, Google released Angular 2 in 2016, a completely overhauled version of the original framework.

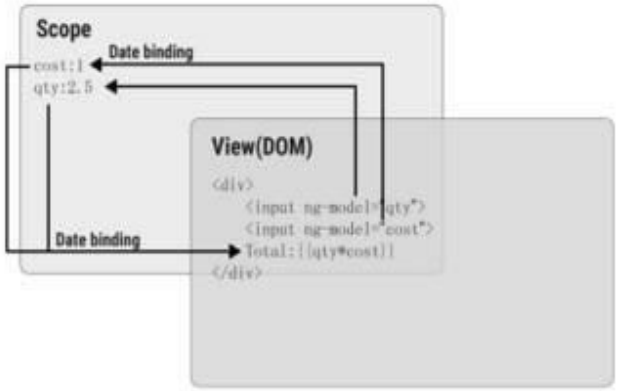

Figure 3 Angular 1 two-way data binding

Angular 1's key feature is two-way data binding within web browsers, which helps reduce the burden on web servers by handling more data processing on the client side. The process of how Angular 1 manages data binding is illustrated in Figure 2. Custom tag attributes embedded in JSON are used as directives in Angular 1 to link input or output elements on a web page to a model represented by Scope. When users interact with the web page, certain JavaScript variables are updated through dynamic JSON resources, and the data is sent to the server. Angular's two-way data binding allows interactions to occur within the browser, eliminating the need to wait for server-side processing and enabling the immediate rendering of updated data in the front-end via HTML. This technology allows for faster HTML rendering since it bypasses delays caused by waiting for server responses.

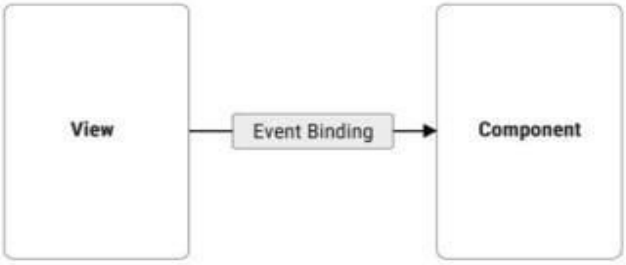

Figure 4 Angular 2 one-way data Binding

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Angular 2 brought significant changes and optimizations to the data-binding process that was present in Angular 1. The most notable change was the removal of the template directive and controller, which were combined into a new module called 'Component.' Additionally, Angular 2 introduced event binding, which provides one-way data binding from the view to the component—something that was absent in Angular 1. Another major change was the switch from JavaScript to TypeScript, a syntactical superset of JavaScript developed by Microsoft. Moreover, Angular 2 replaced Scope with zone.js for monitoring user interactions. Lastly, while Angular 1 was designed primarily for desktop web applications with limited mobile support, Angular 2 was optimized for mobile platforms. By integrating these new features, Angular 2 offers a smaller framework size and faster performance, which are crucial for mobile development.

3. React js and React Native

To enhance user experience on its platforms, Facebook developed the React JavaScript library. Launched as an open-source project in 2013, React is built on JavaScript ES6 and offers powerful features for developers and companies worldwide. In 2015, Facebook expanded its offerings with React Native, allowing developers to create mobile applications for major platforms like iOS and Android using React

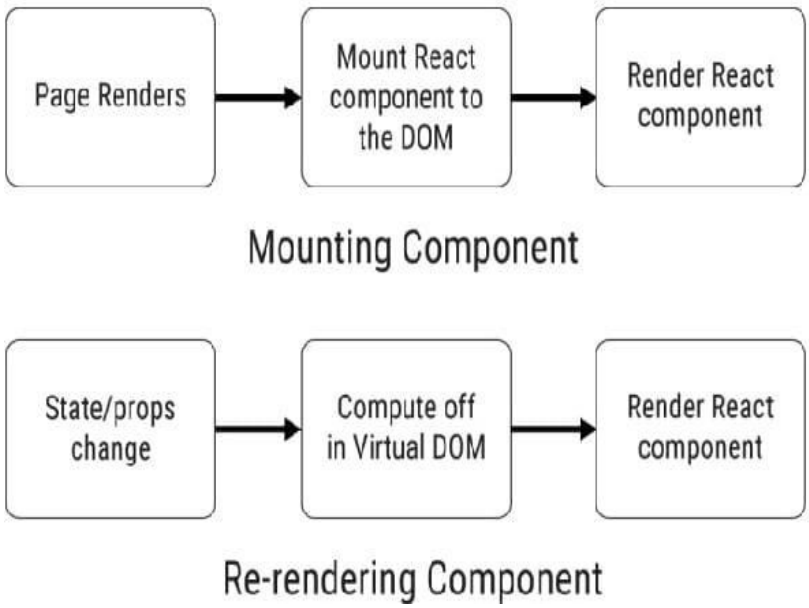

Figure 5 Mounting component in React and Re-rendering Component

Figure 5 illustrates how React technology facilitates page rendering. React utilizes the Document Object Model (DOM) to manage website content as individual components. The rendering process occurs in the browser via JavaScript, which operates more swiftly than traditional dynamic web pages, thanks to Chrome's V8 engine. A key innovation of React is its implementation of a virtual DOM. In a standard HTML website, when users refresh data or navigate to different subpages, the entire page is re-rendered, consuming more resources and leading to slower performance. In contrast, React adopts a more efficient strategy by first creating an updated virtual DOM and then comparing it with the actual DOM. This process allows React to identify changes and only re-render the components that have been altered, avoiding unnecessary re-renders for unchanged parts.

The most significant advantage of the virtual DOM is the enhanced speed of websites built with the React framework. Additionally, React promotes a modular approach to UI design. For instance, consider two websites with similar text forms serving the same purpose but featuring different designs. If these text forms are developed according to a standard module, clicking a link to navigate to another page won't cause the form to be re-rendered, as React recognizes them as identical. Ultimately, React not only transforms front-end programming but also sets a new benchmark for user interface design and development.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. FRONT-END SOLUTION ANALYSIS

In contemporary web development, React, Angular, and Vue are undoubtedly leading choices for front-end development. Each framework and library possess unique features and principles, which implies that various business requirements should guide the selection of the most suitable option to maximize application performance.

A. Data Handling

Data handling plays a crucial role in front-end development, as the effectiveness and quality of data management significantly impact the user experience when interacting with the application.

	Angular 1	Angular 2	React	Vue
Data Binding	Two-way	Two-way and one-way	One-way	One-way /Two-way

Table 1: Data Binding in Front-end Frameworks

Table 1 illustrates how each framework and library manages data binding. Vue supports two-way binding through DOM listeners, while it can also implement one-way binding without them. The primary distinction between React and Angular 1 lies in their approach to data binding; React utilizes a more advanced data flow mechanism to identify changes between the virtual DOM and the actual DOM.

However, employing a two-way binding approach between a component and a view can lead to unexpected states in the component due to conflicting data from various sources. In contrast, a one-way binding approach helps to mitigate conflicts arising from multiple data sources, especially in event-driven scenarios. Consequently, the Angular 2 team has enhanced this concept by allowing developers to utilize both one-way and two-way data binding.

B. Volume and performance

A larger size typically signifies that a framework or library offers more features and functionalities, but it can also lead to longer loading times. Among the frameworks, Angular 2 is the most substantial, with a size of 143 kilobytes (KB), followed by React at 43 KB and Vue at 23 KB. Angular 2's larger size comes with advanced and comprehensive functions; however, its complex architecture may result in slower performance compared to React or Vue, particularly regarding memory management.

Activity	Angular 2	React	Vue
Ready-memory	4.8 ± 0.0 (1.4)	4.5 ± 0.1 (1.3)	3.8 ± 0.0 (1.1)
Run-memory	10.9 ± 0.1 (2.7)	9.7 ± 0.1 (2.6)	7.5 ± 0.1 (1.9)

Table 2 Front-end Frameworks and Library Memory Allocation Performance

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Table 2 shows that Angular 2 requires more time to initialize and utilize memory, while Vue demonstrates faster performance due to its lightweight and efficient design.

C. Language Based

Language-based factors are crucial, as different programming languages present unique challenges in project development, including learning curves and efficiency. Table 3 outlines the current landscape of language-based frameworks and libraries. React and Vue are developed using JavaScript ES6, which has been the industry standard since 2015. In contrast, Angular 1 utilizes the older JavaScript version, ES5. Angular 2 adopts TypeScript to enhance type inference and minimize type-related issues in web applications. Additionally, TypeScript aids developers by modernizing the language structure, moving away from the older JavaScript syntax. However, TypeScript has a relatively small user base, and there is a possibility that it may become less relevant if a new syntactical superset of JavaScript is introduced. Meanwhile, JavaScript ES6 remains a foundational element in JavaScript development, allowing for modifications but not replacement.

	Angular 1	Angular 2	React	Vue
Language-based	JavaScript ES5	TypeScript	JavaScript ES6	JavaScript ES6

Table 3: Language-based in front-end frameworks and library

D. Technical Support

Technical support is essential, as strong community backing can significantly enhance the reputation and adoption of a framework. React offers exceptional technical support and a highly reliable API. The official scripts provided make the process of updating and migrating relatively simple, ensuring long-term assistance for developers.

Angular also provides similar functionalities; however, its API does not match the robustness of React's. Additionally, certain APIs from earlier versions have been deprecated. While Vue offers a straightforward migration and update system across its versions, the official team has indicated that improvements may not be forthcoming due to budget limitations.

E. Online Business Solutions

From the analysis of various features of front-end frameworks and libraries, it is evident that each option has its unique strengths and weaknesses. Angular 2 stands out for providing an effective solution for both one-way and two-way data binding. Additionally, the technical support from the Google development team is reliable and consistent. However, Angular 2's extensive functionalities result in a larger framework size, which may impact performance, and its underlying language has a relatively small user community. Consequently, Angular 2 is well-suited for large-scale e-business solutions that require advanced data processing capabilities and complex features.

React is notable for its efficient rendering of the updated DOM and robust technical support, including a long-lasting API, which alleviates concerns related to updates and migration. Moreover, once developers become proficient in React, they can easily transition to building mobile applications with React Native. However, as a JavaScript library with a smaller footprint compared to Angular 2, React may not offer comprehensive functionalities, requiring developers to implement additional features independently. This demand for customizable features and fast rendering is particularly prevalent in social media and communication applications, making them potential clients for React.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Vue provides both one-way and two-way binding options in data processing. Its lightweight design enables efficient rendering and processing, outperforming both Angular 2 and React in this aspect. Although Vue excels in flexibility for front-end development, its technical support is often inconsistent due to the limited scale of its development team and unexpected changes in official plans. Additionally, its modest size means it only includes basic functionalities. Therefore, Vue is an excellent choice for small to medium-sized web projects that prioritize simplicity and flexibility in development, alongside rapid data processing capabilities.

III. CONCLUSION

This paper explores three front-end development frameworks and libraries—React, Angular 2, and Vue—for building online applications and potential web development solutions. By comparing these frameworks in areas such as data binding, programming language, technical support, size, and performance, we can draw several conclusions. Angular 2 offers the most comprehensive features, making it well-suited for large-scale commercial projects, especially in the e-business sector. On the other hand, React and Vue are better options for live streaming, communication, blogging, and small to medium-sized applications.

When creating a complete front-end component, using a UI framework is essential to ensure high-quality UI design. In the future, we aim to expand our research to include more front-end development methods and analyze their operational principles for web application development.

REFERENCES

1. Angular Official Blog. (2018, July). AngularJS to Angular concepts: Quick reference. Retrieved from https://angular.io/guide/ajs-quick-reference
2. Vue Official Blog. (2018, July). Reactive data binding research. In Science and Information Conference. Retrieved from https://v1.vuejs.org/guide/overview.html
3. MDN Web Docs. (2024). A re-introduction to JavaScript. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_reintroduction_to_JavaScript
4. React. (2024). Virtual DOM and internals. Retrieved from https://reactjs.org/docs/faq-internals.html
5. Web Technology Surveys. (2024). Historical trends in the usage statistics of client-side programming languages for websites. Retrieved from https://w3techs.com/technologies/history_overview/client_side_language/all

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details