Grenze International Journal of Engineering and Technology, Jan Issue

A Comprehensive Review of JavaScript Frameworks

Atal Dubey[1], Khushi Chaturkar[2], Rajat Pawar[3], Amisha Sarode[4] and D. D. Shirbhate[5]
[1-5]Government College of Engineering, Department of Computer Engineering, Yavatmal, India
Email: raghavajaydubey@gmail.com, {kuchaturkar11, yesimrajat, amishasarode2002, shirbhate.dhiraj }@gmail.com

***Abstract*— The web is getting recognition all over the globe as it allows access to resources and working on them over the internet, the development sector of the internet is adapting the HTML5 and CSS3 generation and JavaScript is becoming more powerful. Developers make use of brand-new technologies to choose the JavaScript Framework which will help to reduce the complexity of coding steps and provide a platform for the developers to address the tasks of programming. This comprehensive review, named "A Comprehensive Review of JavaScript Frameworks," gives an overview of JavaScript Frameworks. Also, this overview represents the evolution of JavaScript itself, examining the role of Frameworks in enhancing the maintenance of code, the productivity of developers, and user experience. This paper focuses on the adoption of these described frameworks in the industry, their applications, and their impact on the traits of web improvement. This review paper describes the JavaScript Frameworks: AngularJs, Nodejs, React, jQuery, and their influence on ongoing and upcoming technologies. It gives the guarantee of the evolution of JavaScript frameworks, thinking about emerging technologies and evolving principles to form the landscape of web development.**

***Index Terms*— Web Development, JavaScript, AngularJs, Nodejs, React, jQuery, etc.**

I. INTRODUCTION

The Internet is a huge pool of languages, protocols, and practices that are used in the development of websites applications, and Internet packages. One of them is JavaScript, commonly denoted as 'JS', a programming language that is the core technology for the internet along with HTML and CSS. As per the current report of 2023, 98.7% of website, applications use JavaScript on the client side for the working of the website by incorporating the in-built libraries. JavaScript libraries or code run on its dedicated environment. All the available web browsers have specialized JavaScript engines to run the code on the system of the customer. JavaScript is a high-level, widely used compiled language that follows the ECMAScript Standard. It uses prototype-based object orientation and first-class functions. JavaScript is an interpreted and script-based language to build web applications that allow programmers to interact with web pages. As an end result, some of the less sophisticated ports of computer-like packages have become on hand as net pages. One example was the mail client Gmail of Google.
Here, we refer to internet pages as web applications that have APIs for running the text, dates, data structures, regular expressions, and the Document Object Model (DOM) to manage the files. Social Networking is a live example of the use of an internet package. Facebook is a famous website that has been listed by Alexa on Range 2. A total of 25 websites are used for social networks, e.g., Facebook, Twitter, etc. Much research has been done that shows that social networking net programs are present that are made available by JavaScript libraries. JavaScript has grown as a famous programming language after getting recognized by net packages. The review

Grenze ID: 01.GIJET.10.1.182

paper is determined to present four popular JavaScript frameworks - AngularJs, Nodejs, React, and jQuery to show their fault, features, and benefits which will offer a side-by-side evaluation as per the analysis or testing [1]. We can also add some technology that will make the use of packages more independent and secure. The rest of the paper is organized as follows. Segment II provides the Beginning of JavaScript, and then phase III provides the Timeline of JavaScript. Phase IV benchmarks the popular framework of JavaScript that we use for development. Phase V gives trending JavaScript frameworks. In Phase VI, we conclude our analysis of JavaScript frameworks.

II. Beginning Of Javascript

A. JavaScript Invention

JavaScript was developed in May 1995 by Brendan Eich within 10 days. He worked at Netscape and applied the observed concept of JavaScript for his web browser, Netscape Navigator. Earlier most of the parts of client-side web pages had to be carried out in Java. JavaScript has become a linking language to connect and coordinate with Java. It helps different parts of a system or software that use Java to work together effectively. JavaScript also helps to make static web pages more interactive. As JavaScript holds the position of helping Java, JavaScript had to behave and appear like Java.

B. Standardisation of JavaScript

JavaScript follows certain standards: The number 1 well-known is ECMA_262, which is hosted through ECMA-International. The secondary widespread is ISO/IEC 16262, hosted by the Worldwide Company for Standardization (ISO) and the Worldwide Electro-technical Fee (IEC). Using above mentioned standard some languages were defined that are referred to as ECMAScript, instead of JavaScript. The new name was chosen by Sun Microsystem which is now known as the Oracle as the trademark for the former name. The word "ECMA" in "ECMAScript" comes from the first standard which was hosted by the name of an organization. Earlier, the organization was named ECMA, which was represented as the European Computer Manufacture Association. Afterward, the name was changed to Ecma International, Ecma being an actual name and no longer an acronym. This was transformed because the Organization expanded its operation which was beyond Europe. The preliminary use of all capital letters in the acronym clarifies the word ECMAScript.

C. Evolution of JavaScript

Over the years JavaScript has taken control of the progress of web development and has evolved totally. In the initial days of JavaScript, it was only being used to add simple interactivity within web pages. But with the continuous updation and evolution in JavaScript, new features have been added that make it more powerful and easier to use. For example, ECMAScript 2019, the latest version of JavaScript, introduces new features and improvements to JavaScript such as asynchronous functions, classes, and modules. These features have made a huge impact on JavaScript development by enhancing code readability, maintainability, and performance. The evolution of JavaScript has made it a powerful and versatile programming language. Now it is used to create a wide variety of web applications. As there is continuous evolution and updation in the language, it will become even more powerful and popular.

III. Timeline Of Javascript

Over the years, JavaScript has turned into a multiprogramming, versatile, and powerful programming language from a simple scripting language that can be used for web development, server-side programming, and many more. Here's a timeline of the major milestones and developments in the history of JavaScript:

A. Birth of JavaScript (1995)

Brendan Eich created JavaScript in only ten days while running at Netscape Communications and is first known as "Mocha" and later known to be "LiveScript," it is ultimately named "JavaScript".

B. JavaScript Standardization (1996)

JavaScript is submitted to ECMA global (European Computer Manufacturers Association) for standardization, resulting in the first version of the ECMAScript general.

C. ECMAScript-2 (1997)

The second edition of ECMAScript is launched with minor updates to the language.

D. ECMAScript-3 (1999)

ECMAScript 3 is delivered, including numerous new capabilities and enhancing the language's balance. This model will become widely supported in web browsers and remain well-known for decades.

E. Browser wars and DOM Manipulation (2000s)

JavaScript will become critical to net development, with the introduction of the document object version (DOM) for dynamic internet page manipulation. The "browser wars" era ended in compatibility troubles as distinct browsers put in force JavaScript differently.

F. ECMAScript-5 (2009)

JavaScript will become critical to net development, with the introduction of the document object version (DOM) for dynamic internet page manipulation. The "browser wars" era ended in compatibility troubles as distinct browsers put in force JavaScript differently.

G. Node.js (2009)

ECMAScript 5 is launched, introducing strict mode, JSON help, and numerous new array strategies.

H. ECMAScript-6 (ES6) (2015)

ECMAScript 6, additionally called ES6 or ECMAScript 2015, is launched, bringing foremost language improvements like arrow functions, training, modules, and template literals.

I. WebAssembly (Wasm) (2017)

Web Assembly (Wasm) is introduced, making allow to the near-native for the speed execution of the low-level code in internet browsers.

J. ECMAScript 2019 (ES10) (2019)

ECMAScript 2019 introduces capabilities like Array.prototype.flat(), Array.prototype.flatMap(), and non-compulsory capture binding.

K. ECMAScript 2020 (ES11) (2020)

ECMAScript 2020 consists of functions like Big int for the arbitrary precision integers.

L. ECMAScript 2021 (ES12) (2021)

ECMAScript 2021 introduces new capabilities along with the "String.prototype.replaceAll()" technique, logical task operators, and Promise.any().

M. Future ECMAScript Versions (Ongoing)

The TC39 committee continues to paint on proposals for the upcoming ECMAScript versions, ensuring that JavaScript stays a dynamic and evolving language.

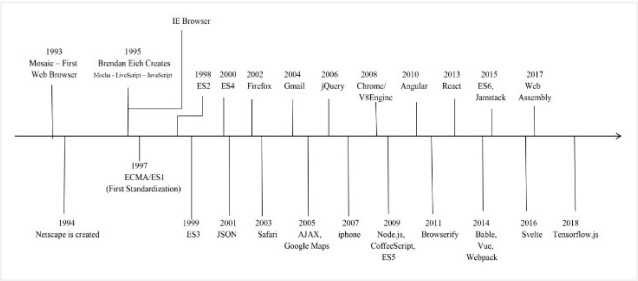

Fig 1. Evolution of JavaScript

IV. POPULAR JAVASCRIPT FRAMEWORKS

A web framework that is abbreviated as 'WF' or the web application framework (i.e. WAF) is a software interface that supports the web application or the development of frameworks that include the resources and services required for the web and web API's. A JavaScript Framework is a comprehensive set of in-built JavaScript Snippet's code libraries that gives pre-written code for regular programming tasks to a web developer or

programmer to build a web application more efficiently. JavaScript works well for both front-end as well as back-end development with its frameworks.

TABLE I. JAVASCRIPT FRAMEWORKS

Front-end Frameworks		Back-end Frameworks	
1.	React	1.	jQuery
2.	Angular	2.	Node.js
3.	Vue.js	3.	Express.js
4.	Ember.js	4.	NetJS
5.	Backbone.js	5.	Koa.js

A. Why do we have a number of JavaScript frameworks?

JavaScript Frameworks are created to provide pre-written code so that it can fulfill the changing needs or emerging advancements in web development. These Frameworks are designed to take advantage of new features and capabilities evolved. These Frameworks made it possible for the developers to create modern web applications.
One of the factors on which the development of Frameworks depends is Developer preferences and philosophies. Some developers focus on user requirements, structure, and opinions which were observed in Angular and Ember. Others prefer the use of lightweight libraries like React or Vue. These differences in the opinions and preferences of developers were the main reason for the creation of so many frameworks, where every framework carries out a different task.

B. Analysis of frameworks

Some of the frameworks that our review paper is basically based on:

1. Angular

AngularJS is a front-end-based web application. The first version of this framework was created in 2010 by Google, and the recent version that is actively in use is version 1.7 which was developed in 2018. It can be used in the HTML page by adding a <script> tag. AngularJS focuses on extending the attribute of HTML by making use of pre-processor directives and binds data by using the expressions to HTML.
AngularJS follows the MVC design pattern that divides the program into three major parts: the model, the view, and the controller. This division can help to represent how data is stored, its presentation to the user, and management by the program. Angular has transformed the traditional server-side rendering service into client-side web programs with the help of "dependency injection". This will also help to reduce the burden on the server. Out of the three parts: the model shows how the data is stored, the view displays the data and the controller focuses on managing the input and making the updation in the model whenever needed.
AngularJS is a well-known framework for creating single-page applications. In a single-page web application, all the content is loaded on a single page which has become more responsive and faster than traditional web applications after using the AngularJS features. AngularJS is a powerful framework that is used to build a number of web applications. It is mainly used by developers who want to create dynamic and interactive web applications.

2. React

React was developed by Jordan Walke, who was a software engineer at Meta, who released an early rough model of React called "FaxJS". Its first deployment was in 2011 on Facebook's News Feed and later it was deployed on Instagram in the year 2012. It was open-sourced at JSConf US which was held in May 2013. React Native supports iOS, and native Android and also allows the creation of a Universal Windows Platform (UWP) app using React. The official announcement about this was made in February 2015 at Facebook's React Conf. Afterward, in March 2015; it was made open-source.
React or React.js or ReactJS is a frontend JavaScript library that is free and is being used to build application-based interface components for users as it is open-sourced. Now, Meta which was earlier known as Facebook, and Individual developers groups, and companies maintain React. It is combined with the Next.js framework to develop mobile, single-page, or server-side rendering web applications. React mostly focuses on rendering components and the user interface to the DOM, React provides routing and client-side functionality, which relies on packages.
In React, the code works in components that are made up of elements. These components are used to develop a software application in terms of modules. Programs built in React mainly consist of multiple layers of components. With the help of the DOM library provided by React, we can display these components on the web page. To display

components and share information between these components values are used through properties (mainly called props). Internally, information about a component is stored in something called 'state'. Components can be declared primarily in React, i.e., through class and function components.

3. Jquery

Originally jQuery was created at BarCamp by John Resig in January 2006. He was influenced by the earlier CSS Query library that was created by Dean Edward. Currently, a group of developers, under the leadership of Timmy Willison, is maintaining this framework with a selector engine of jQuery, with its component, Sizzle, led by Richard Gibson. This design was developed to make it easier for the developer to traverse the tree structure of HTML DOM and handle the changes in HTML, manage events, apply CSS animations, and work with Ajax. It is open-sourced, free-for-all software under the agreeable MIT License.

The syntax of jQuery is designed to simplify document navigation, make the DOM element selection easier, manage events, and develop Ajax applications more efficiently. jQuery is highly valuable for developers as it offers the ability to develop plugins above the libraries of JavaScript. jQuery operates a Domain Object Model (DOM) based library that provides powerful manipulation capabilities. The DOM serves as a tree-like structure representing all the components within a Web application, and jQuery simplifies the syntax to find, select, and manipulate these DOM components, enhancing the development process.

jQuery offers compatibility with the "current-1 versions" of popular web browsers, including Firefox, Safari, Edge, Chrome, and Internet Explorer 9, which is the current stable version of the browser. For mobile devices, jQuery supports iOS 7 and Android 4.0, along with later versions of these operating systems.

4. Node.Js

The very first server-side rendering environment for JavaScript, 'Netscape's LiveWire Pro Web' was developed in 1996 and after about thirteen years Node.js was introduced in 2009. The available versions initially supported operating systems like Linux and Mac OS X only. Initially, Node.js was maintained by Dahl and afterwards, it was owned by Joyent. Node.js can run on various operating platforms such as Unix, Linux, Windows, and Mac OS because it is versatile and has an open-source server environment. Node.js is a backend framework that provides an environment for running the program on the V8 engine of JavaScript and it executes the code external to a web browser. Node.js can allow the programmer to write command-line arguments using JavaScript tools and can also be used for scripting server-side.

Node.js follows a "JavaScript everywhere principle" which represents that a single programming language is enough to develop a web application instead of using multiple programming languages for server-side programming versus client-side programming. Node.js supports event-driven architecture which is used to improve scalability, performance, and flexibility in a software design.

In Node.js event-driven architecture is implemented using the modules of the event loop and event emitter that provide a powerful group of tools for managing different object events and making it able to asynchronous with I/O. The motive for the use of this design is to improve the output in web applications for real-time communication programs and various web applications and to improve scalability with too many operations of input/output.

C. Comparison of frameworks

We have compared the above-discussed framework based on various metrics. One can easily find the best-suited framework for their project by reviewing this comparison.

TABLE II. COMPARISON OF FRAMEWORKS

Parameters	Angular	React	jQuery	NodeJs
Type	Front-end framework	JavaScript library	Server-side runtime	Front-end library
Component-based	Yes	No	No	Yes
Templates	HTML-based templates	HTML/ JavaScript	No	JSX (JavaScript)
Routing	Built-in(Angular Router)	Manual/Third-party	Express.js	React Router
Community & Ecosystem	Large, but divided	Large	Large	Large
Popularity	Declining	Declining	Growing	Very popular
Real-time applications	Not typically used	Possible, but manual	Ideal for real-time	Used with web socket

Let us understand the emerging frameworks which are useful in designing the architecture of web applications in day-to-day life.

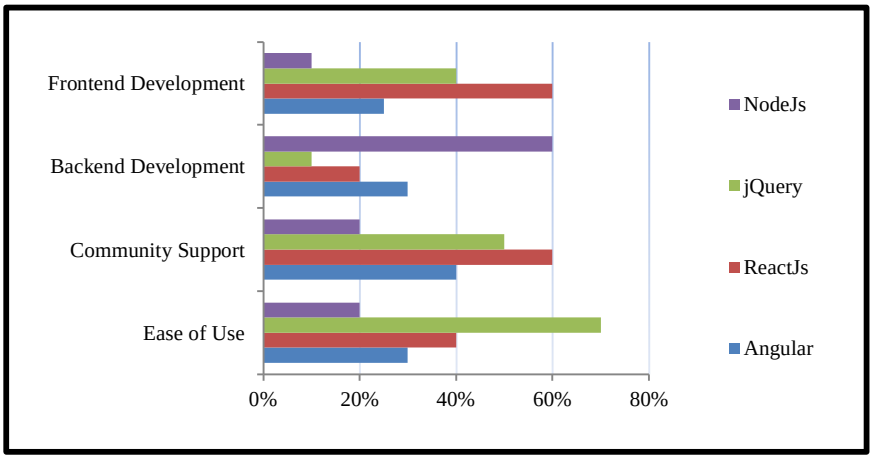

Fig 1. Graphical Representation of Trending Frameworks

D. Architectural view of frameworks

JavaScript frameworks such as Angular, React.js, and Vue.js work on a component-based architectural pattern, focusing on the creation of UI components that are reusable and summarize both the logical and the presentation layers(JSX or template). These frameworks allow structured rendering by utilizing a virtual DOM(for React) and imparting working for statistics binding, where it ensures the view should display changes in the underlying records. Angular supports a dependent technique while Vue and React provide greater flexibility. In review, Node.js is not a frontend framework anymore still it uses an event-pushed and non-blocking I/O structure to construct server-aspect applications and APIs, and routinely interacts with frontend frameworks and libraries with the help of restful APIs [1].

TABLE III. ARCHITECTURE AND TECHNICAL POSSIBILITIES

Runtime/ Framework	Angular	React.js	jQuery	Nodejs
Architecture	Component-based	Component-based	Not component-based	Event Driven, Non-Blocking I/O
Templates	Angular Templates	JSX (JavaScripts XML)	HTML/ JavaScripts	Not allowed templates
Built-in Validation	Yes	Limited(with third-party libraries)	No	Not allowed built-in validation
Building of Models & Views	Components & Services	Components & State	DOM Manipulation & Event Handling	Server-side Development
Compatibility with Other Libraries	Compatible with RxJS for observables and various libraries	Compatible with many libraries due to its component-based architecture	Compatible with various plugins and libraries, but it's not a framework	Compatible with a wide range of npm packages and libraries

E. Analysis of frameworks

Different types of testing have been supported by all four frameworks. Some support functional or black box testing and others support structural or white box testing, Karma and Jasmine are the famous JavaScript testing modules supported by AngularJS, Jest, mocha by React, Qunit by jQuery, and many others.

1. Performance

Different Characteristics have been revealed after performing the performance testing of Angular, jQuery, Node.js, and React. AngularJS is a robust technology but due to its framework size, it can exhibit longer initial load times, it recounts with fast runtime performance, especially with Ahead-of-Time(AOT) compilation.jQuery has a lightweight nature so it dominates the initial load time, but for complex applications, jQuery's runtime performance may lag behind modern frameworks.Node.js is proficient in server performance as it supports a non-blocking I/O feature that makes it ideal for real-time applications. React offers rational load times, mainly with server-side rendering, and also maintains an outstanding runtime performance because of its Virtual DOM and component-level updates. Each technology has its own choices, weaknesses, and strengths that should align with specific project requirements and priorities [1].

2. Security

All the Frameworks Angular, jQuery, Node.js, and React have evolved in the various considerations for security purposes. Angular provides protection against cross-website Scripting (XSS) attacks and strong safety features. The right setup is needed to prevent move-web site Request Forgery (CSRF) attacks as Angular's exceptional practices are analytical for protection. jQuery which consists of a DOM manipulation library doesn't provide security features instead it provides the environment for the user to be comfortable with coding practices, server-side security measures, and for entering validations. Node.js provides security when configured properly but due to bad coding practices, it can have powerlessness, authentication, remote server safety measures like Helmet, and making input validations. React's security measure depends on the right data management and coding practices that include input sanitization to prevent XSS attacks. We can implement server aspect rendering (SSR) to beautify protection. To retain software safety efficiently, we have to follow a configured approach and stay knowledgeable about best safety practices. The selection of best practices needs to be aligned with unique safety necessities and development practices [1].

V. JAVASCRIPT CUTTING-EDGE

We are entering into the third generation of net services. Web services and technologies have been evolving constantly and JavaScript continues to be an important part of this evolution. JavaScript has been transformed over the years and it is a core technology for web development. All the new Frameworks that have been developed are being advanced all the time. Therefore, we have to stay up-to-date in JavaScript development.

Vue.js is making a great impact on today's web era with its flexibility, versatility, and overall performance. Vue.js can evolve seamlessly due to its modular structure to a variety of task requirements to manage small interactive widgets as well as complicated single-page packages. Vue.js offers strong aid for server-side rendering technology, growing internet apps, accessibility, and internationalization as it understands modern-day net technology.

Vue.js is poised to have a profound impact on the present-day web era with the aid of virtue of its simplicity, flexibility, and overall performance. Vue's modular structure lets evolve it seamlessly to an extensive variety of task requirements, from small interactive widgets to complicated single-page packages. Vue.js aligns with modern-day net technology trends by offering strong aid for server-aspect rendering, progressive internet apps, internationalization, and accessibility. They may be involved in the development of net applications and are determined to fulfill the changing desires of businesses and users in the present-day digital aspect.

VI. CONCLUSION

In this comprehensive review of JavaScript Frameworks, it focuses mainly on the evolving technologies of the web development. JavaScript is a necessary programming language, which has given uplift to a large number of libraries and Frameworks that help developers create interactive web pages and enhance the overall performance of web packages. Our study of distinguished frameworks like React, Angular, jQuery, and Node.js and emerging frameworks like Vue and Svelte, emphasizes their important role in making the modern web ecosystem. The important intention of this comprehensive review of JavaScript frameworks is to provide developers and selection-makers a piece of precise information and aid in choosing the most appropriate framework needed and also comparing the frameworks with others to choose the framework that optimizes web development initiatives and forward better knowledgeable technology choices.

Besides, this review paper stresses the importance of choosing the right framework that specifies the specific project's needs, considering factors such as security, server-side rendering, overall performance, developer experience, and community support. JavaScript is continuously evolving so the landscape of frameworks supplies

each and every possible demanding situation to the developers. The insights gained from this review serve as an important resource for directing the changing world of JavaScript development, predicting innovations, and assurance of the creation of web studies for worldwide users.

REFERENCES

[1] S. Delcev and D. Draskovic, "Modern JavaScript frameworks: A Survey Study," 2018 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2018, pp. 106-109, doi: 10.1109/ZINC.2018.8448444.

[2] P. Ratanaworabhan, B. Livshits, and B. Zorn, "JSMeter: Comparing the behavior of JavaScript benchmarks with real web applications," in Proceedings of the 2010 USENIX conference on Web application development. Berkeley, CA, USA: USENIX Association, 2010.

[3] Gude, S., Hafiz, M., & Wirfs-Brock, A. (2014). JavaScript: The Used Parts. 2014 IEEE 38th Annual Computer Software and Applications Conference. doi:10.1109/compsac.2014.69 Year: 2014 | Research Paper | Publisher: IEEE.

[4] Wahslen, J., & Lindh, T. (2018). A Javascript Web Framework for Rapid Development of Applications in IoT Systems for eHealth. 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom), Publisher: IEEE

[5] P. Thung, C. Ng; S. Thung, S. Sulaiman, "Improving a Web Application. Using Design Patterns: A Case Study," International Symposium in Information Technology 2010, IEEE, Kuala Lumpur, Malaysia, Sept.2010.

[6] Saxena, P., Akhawe, D., Hanna, S., Mao, F., McCamant, S., & Song, D. (2010). A Symbolic Execution Framework for JavaScript. 2010 IEEE Symposium on Security and Privacy. doi:10.1109/sp.2010.38, Year: 2010 | Research Paper | Publisher: IEEE

[7] S. Guarnieri and B. Livshits. Gatekeeper: mostly static enforcement of security and reliability policies for JavaScript code. In Usenix Security, 2009.

[8] C Ritwik and Anitha Sandeep, "REACT.JS and Front End Development", Bangalore in International Research Journal of Engineering and Technology (IJRET), vol. 7, no. 4, April 2020, ISSN 2395-0056.

[9] Sohan, M. F., & Basalamah, A. (2020). A Systematic Literature Review and Quality Analysis of Javascript Malware Detection. IEEE Access, 8, 190539–190552. Year: 2010 | Research Paper | Publisher: IEEE

[10] P. Likarish, E. Jung, and I. Jo, ''Obfuscated malicious Javascript detection using classification techniques,'' in Proc. 4th Int. Conf. Malicious Unwanted Softw. (MALWARE), Oct. 2009, pp. 47–54.