IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 7.301

Volume 3, Issue 2, March 2023

Comparing the Performance of Flutter and ReactJS for Web Development

Krishnaprasad Awala
Student, Department of Information Technology
MIT Art, Design and Technology, Pune, Maharashtra, India

Abstract: *This research paper aims to assess the performance of Flutter, a cross-platform development framework, as compared to other widely-used web development frameworks like ReactJS for developing web applications. The study will analyse the advantages and limitations of each framework with respect to cross-platform compatibility, development time, and community support. The research aims to provide significant insights to web developers seeking to build efficient and user-friendly cross-platform web applications. By comparing the frameworks' performance, this research will offer valuable guidance for selecting the most suitable framework for developing high-performing cross-platform web applications.*

Keywords: SEO (Search Engine Optimization), Responsiveness, Cross platform development, Web development, ReactJS, Flutter, Programming Languages, Frameworks

I. INTRODUCTION

Web development is a crucial component of website building, encompassing the technical aspects that bring websites to life, maintain their functionality, and enhance the user experience. This involves using a variety of coding languages tailored to the task and platform at hand. With its high demand and competitive salaries, web development offers an accessible and promising career path that does not necessarily require a traditional university degree.

As cross-platform web application development gains popularity, various web development frameworks have emerged, including Flutter, ReactJS, Angular, and Vue.js. These frameworks provide web developers with powerful tools for creating high-quality, responsive web applications that can run on multiple platforms, including desktop and mobile devices.

Flutter, in particular, has gained popularity among mobile app developers due to its ease of use and ability to create native-quality apps. However, Flutter's web support feature is still in beta, and its effectiveness for web development remains a topic of debate. This raises questions about Flutter's performance in comparison to other popular web development frameworks, such as ReactJS.

This research paper aims to investigate the performance of Flutter and React JS. We will evaluate the performance of these frameworks in terms of load times, file sizes, and user experience on both desktop and mobile devices. Furthermore, we will explore the advantages and limitations of each framework in terms of cross-platform compatibility, development time, and community support.

This study's findings will provide valuable insights for web developers seeking to create cross-platform web applications that deliver optimal performance and user experience. Ultimately, this research aims to contribute to the ongoing discussion on the effectiveness of Flutter for web development and inform web developers' decision-making process when selecting a framework for their projects.

II. FLUTTER VS REACTJS

2.1 History

Web development has been a growing industry for over 30 years since the introduction of the internet. In its early days, websites were primarily built and managed by IT professionals. However, as technology has advanced, web development has become an industry in its own right. Today, businesses of all sizes rely on web development to establish their online presence and reach their target audience.

DOI: 10.48175/IJARSCT-8892

Over time, web development has evolved from static and simplistic web pages to more complex and dynamic websites. As new programming languages have been introduced and learned, developers have been able to create more sophisticated web applications. One such language that has been gaining popularity in recent years is Dart, introduced by Google.

Flutter, a UI software development kit built using the Dart language, has emerged as a powerful tool for cross-platform development. With Flutter, developers can create applications that run seamlessly on Android, iOS, Linux, MacOS, Windows, Google Fuchsia, and the web from a single codebase. As an open-source framework, Flutter offers a range of features that make it a compelling option for web development.

2.2 Advantages of Cross-Platform Development

1. Using the same application on different devices provides familiarity and a consistent user experience.
2. Cross-platform mobile app development allows faster time to market by building and publishing applications on multiple app stores simultaneously.
3. One source code simplifies the development process and allows for code reuse.
4. Cross-platform development offers easy implementation through the use of various technologies that convert code for different platforms.
5. Maintenance of cross-platform apps is easier as there is only one codebase to focus on, resulting in fewer tests and an easier way to deploy fixes.

From a different perspective, cross-platform development offers:

- Faster development with a single code and fewer errors from the developer's perspective
- Uniformity and satisfaction for customers
- Fewer expenses and a higher return on investment for businesses.

2.3 Programming Languages

A. React JS

- JavaScript: React JS is primarily based on JavaScript, which is a popular programming language used for web development.
- HTML: It is markup language used in React JS.
- CSS: It is styling language used in React JS.
- JSX: It is a syntax extension of JavaScript used in React JS to define and structure the user interface components.
- TypeScript: React JS also supports TypeScript, a superset of JavaScript that adds static typing to the language.

B. Flutter

- Dart: Flutter is based on the Dart programming language, which is an object-oriented, client-optimized language developed by Google. Dart is easy to learn, read, and write and has a syntax similar to Java and JavaScript.
- Widgets: Flutter uses its own set of widgets, which are pre-built elements that help in building the UI of the app. Widgets in Flutter are similar to components in React JS.

So, while React JS primarily uses JavaScript and JSX, Flutter is based on Dart, with its own set of widgets. Additionally, React JS also supports TypeScript for adding static typing to JavaScript code.

2.4 Styling and Responsiveness

ReactJS and Flutter have different approaches to styling and responsiveness. ReactJS uses CSS for styling and responsiveness. CSS is a well-established language for styling web pages and has a large number of libraries and frameworks available. ReactJS uses the same approach as HTML and CSS to define the layout of components. CSS allows developers to create responsive designs using media queries, which can change the layout of the page depending on the device's screen size. On the other hand, Flutter uses its own styling system, which is called Flutter widgets.

Flutter provides a rich set of widgets that developers can use to create the UI of the app. The widgets can be styled using the Flutter styling system, which uses a declarative syntax similar to CSS. The styling system allows developers to create responsive designs by defining constraints on the widgets, which adapt to different screen sizes. Flutter widgets are designed to automatically adjust themselves to fit the screen size and device orientation, providing a consistent user experience across different devices and screen sizes.Both ReactJS and Flutter provide mechanisms for creating responsive designs. However, ReactJS relies on CSS, which is primarily used for web development, while Flutter has its own styling system optimized for mobile app development. Developers familiar with web development may find ReactJS easier to use for styling, while those focused on mobile app development may prefer the Flutter approach.

2.5 Search Engine Optimization (SEO)

Search Engine Optimization (SEO) is a crucial aspect of website development as it determines the visibility and ranking of a website in search engine results pages (SERPs). The process involves optimizing various on-page and off-page elements to increase the quantity and quality of organic traffic to a website.

Search engines such as Google use a 3-step process to generate search results: crawling, indexing, ranking, and serving. Crawling involves discovering new and updated content by following links from one page to another. Indexing is the process of storing and organizing the processed data in a database, and ranking and serving involves selecting the most relevant information from the index in response to a user's query.

Flutter is a popular framework for building cross-platform mobile and web applications. However, it heavily relies on the HTML canvas tag, which makes it challenging to optimize for SEO. Google's crawlers have limited information about the page itself, making page-level SEO impossible. While relevant SEO meta tags can be added to improve the situation, they are applicable to the entire website since Flutter web is a single-page application.

However, SEO is a complex process, and there are various tactics that can be used to improve rankings. If SEO is not a priority, and the focus is on building a performant and visually stunning web app or website with support for iOS and Android, then Flutter may be an excellent choice due to its ability to provide full-fledged mobile apps at a fraction of the development and maintenance cost.

On the other hand, React JS is a popular frontend JavaScript library that allows for building dynamic user interfaces. It is generally considered to be more SEO-friendly than Flutter due to its ability to support server-side rendering (SSR) of content. SSR allows search engines to see the fully rendered page, which can improve the website's SEO performance by providing search engines with more information about the page.

In addition, React JS has a feature called "code splitting," which allows for splitting the application code into smaller chunks, improving page load times and, therefore, SEO performance. React JS also allows for adding relevant SEO meta tags to individual pages, which can further improve the website's visibility in search engine results.

Overall, while both Flutter and React JS have their strengths and weaknesses when it comes to SEO, React JS is generally considered to be more SEO-friendly due to its ability to support server-side rendering and code splitting. However, it is essential to evaluate all aspects of the project's requirements and priorities before choosing a technology for development.

2.6 Backend Connectivity

Backend connectivity is a critical aspect of web development that allows the front end of an application to communicate with the server and perform various functions. Both Flutter and React JS provide developers with tools for backend connectivity, but they differ in their approaches.

A. Flutter

Flutter provides developers with a set of libraries and APIs to connect their applications to a backend server. Developers can use HTTP requests to communicate with REST APIs and perform various functions such as fetching data, updating data, and deleting data. Flutter also provides packages such as Dio and Chopper to make HTTP requests easier and more efficient.

DOI: 10.48175/IJARSCT-8892

Flutter also supports WebSockets, which allows for real-time communication between the client and server. WebSockets are useful in applications that require real-time updates, such as chat applications, games, and stock market applications.

Flutter also has built-in support for Firebase, a backend as a service (BaaS) platform by Google. Firebase provides developers with a range of services, including authentication, real-time databases, cloud storage, cloud messaging, and more. Developers can use Firebase to build powerful web applications without worrying about server maintenance or scalability.

B. ReactJS

React JS also provides developers with tools for backend connectivity, but its approach is different from Flutter. React JS relies on third-party libraries and frameworks to connect with backend servers. Popular libraries and frameworks for backend connectivity in React JS include Axios, Fetch, and Superagent.

React JS also supports WebSockets, which allows for real-time communication between the client and server. Developers can use libraries such as SocketIO and SockJS to implement WebSockets in their React JS applications.

React JS also has built-in support for Firebase, similar to Flutter. Developers can use Firebase to build web applications without worrying about server maintenance or scalability.

2.7 Conclusion

Both Flutter and React JS provide developers with tools for backend connectivity, but they differ in their approaches. Flutter provides a built-in set of libraries and APIs for backend connectivity, while React JS relies on third-party libraries and frameworks. However, both frameworks support WebSockets and Firebase for real-time communication and scalable backend services. Ultimately, the choice between Flutter and React JS for backend connectivity will depend on the developer's familiarity with the frameworks, project requirements, and personal preferences.

A. Cross Platform Compatibility

ReactJS is primarily a JavaScript library that is used for building user interfaces. It can be used to build web applications and mobile applications using React Native. However, ReactJS is not a cross-platform development framework for web applications, as it is specifically designed for building user interfaces for web applications.

On the other hand, Flutter is a cross-platform development framework for building mobile, web, and desktop applications. It uses a single codebase for developing applications for multiple platforms, which makes it a great choice for developers who want to build cross-platform applications.

In terms of cross-platform compatibility for web applications, Flutter has an advantage over ReactJS because it is specifically designed for building cross-platform applications. With Flutter, developers can build web applications that are compatible with different browsers and operating systems.

In summary, while ReactJS can be used to build web applications and mobile applications with React Native, it is not specifically designed for cross-platform web application development. Flutter, on the other hand, is a cross-platform development framework that can be used to build web applications and mobile applications with a single codebase, making it a better choice for cross-platform application development.

B. Community Support

Both Flutter and ReactJS have large and active developer communities that provide support and resources to help developers create web applications.

ReactJS has been around longer than Flutter and has a more extensive community with a larger number of libraries and resources. There is a vast amount of documentation available online, and many developers are familiar with ReactJS, which means that it's often easier to find help and resources when facing issues.

Flutter is relatively new compared to ReactJS, but it has gained popularity quickly, and its community is growing rapidly. Google provides extensive documentation, tutorials, and support for Flutter, and there is also a large and active community of developers who contribute to open-source projects, share knowledge, and provide support on forums and social media platforms.

DOI: 10.48175/IJARSCT-8892

In terms of community support, both frameworks have a lot to offer, but ReactJS has been around longer and has a larger community, which means that it may be easier to find help and resources. However, Flutter's community is growing quickly and is also very active, so it is also a great option for developers looking for community support.

C. Resource Consumption

In terms of resource consumption, both ReactJS and Flutter for Web are known for being efficient, but there are some differences in the way they handle resources.

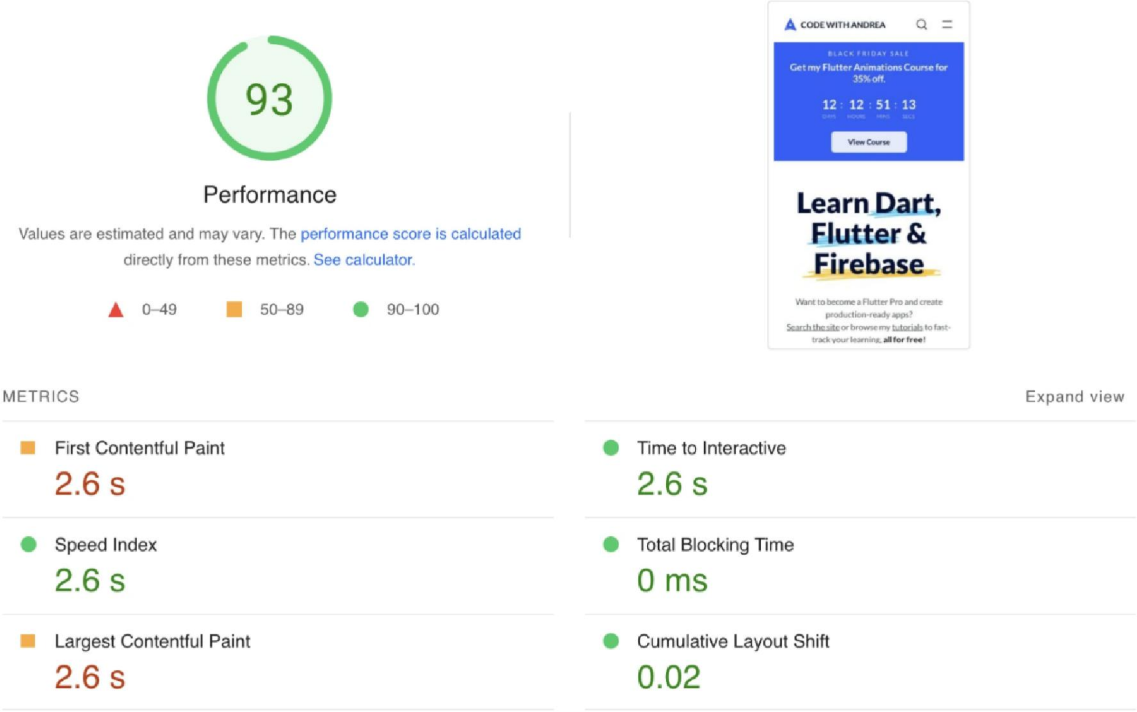
ReactJS uses a virtual DOM, which allows it to update only the necessary parts of the UI without having to reload the entire page. This approach can lead to lower memory and CPU usage since only the changes are being rendered instead of the entire UI.

Flutter for Web, on the other hand, uses its own rendering engine, which is optimized for performance and fast rendering of complex UIs. It also has a feature called Hot Reload, which allows developers to see changes to the UI in real-time without having to rebuild the entire application. This can save time and resources during the development process.

However, it's worth noting that Flutter for Web may require more resources upfront since it has a larger footprint than ReactJS due to the Dart runtime it uses. This means that it may take longer to load and initialize the application, especially on slower devices or networks.

Overall, both ReactJS and Flutter for Web are known for being efficient in their resource consumption, but the specific resource requirements may vary depending on the complexity of the application and the specific implementation choices made by the developer. It's important to consider these factors when choosing between the two frameworks for your web development needs.

D. Performance Comparison



Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

Get my Flutter Animations Course for 35% off.

Want to become a Flutter Pro and create production-ready apps? Search the site or browse my tutorials to fast-track your learning, all for free!

■ First Contentful Paint
2.6 s

● Time to Interactive
2.6 s

● Speed Index
2.6 s

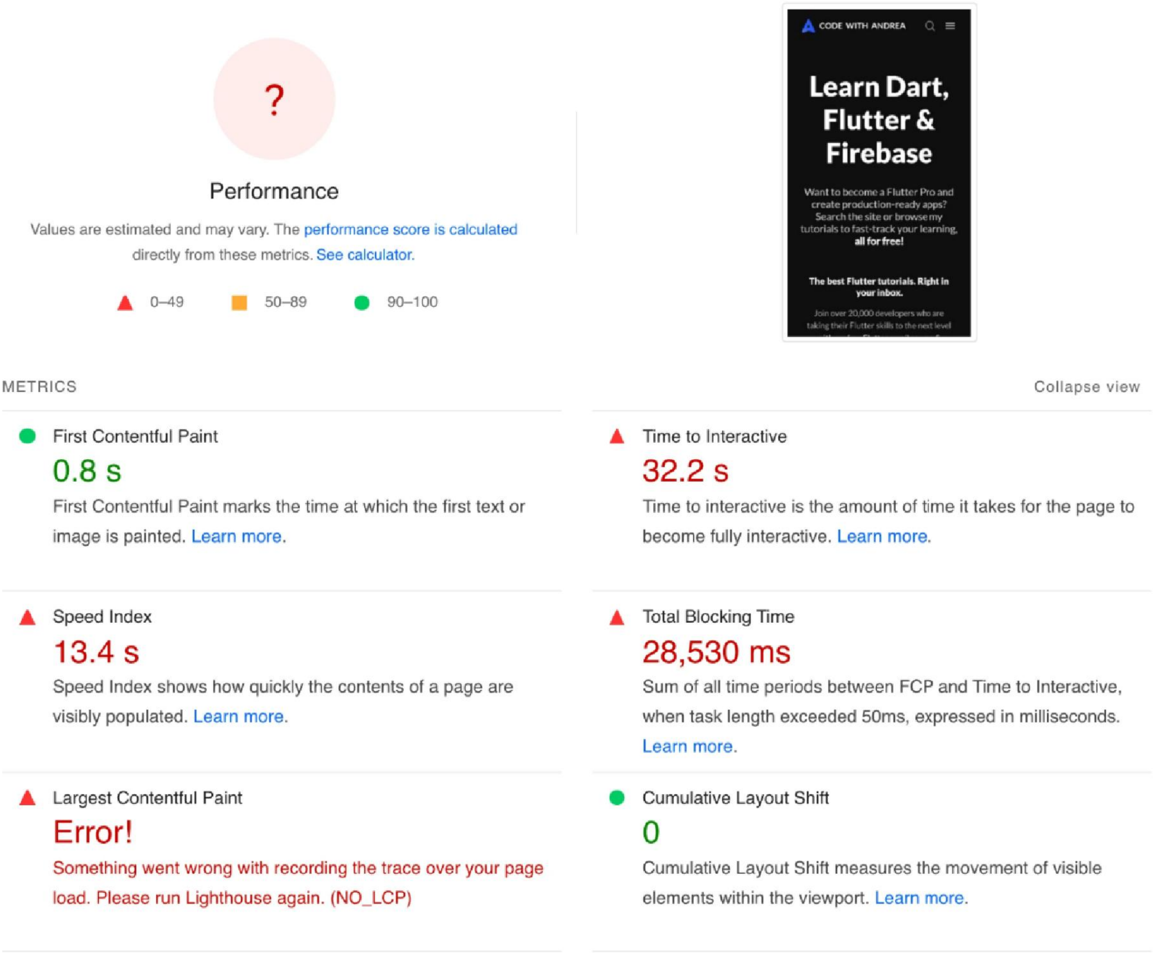
● Total Blocking Time
0 ms

■ Largest Contentful Paint
2.6 s

● Cumulative Layout Shift
0.02

Fig. 1. HTML, CSS, JavaScript Webpage

DOI: 10.48175/IJARSCT-8892



?

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

CODE WITH ANDREA

Learn Dart, Flutter & Firebase

Want to become a Flutter Pro and create production-ready apps? Search the site or browse my tutorials to fast-track your learning, **all for free!**

The best Flutter tutorials. Right in your inbox.

Join over 20,000 developers who are taking their Flutter skills to the next level

METRICS Collapse view

● First Contentful Paint
0.8 s
First Contentful Paint marks the time at which the first text or image is painted. Learn more.

▲ Time to Interactive
32.2 s
Time to interactive is the amount of time it takes for the page to become fully interactive. Learn more.

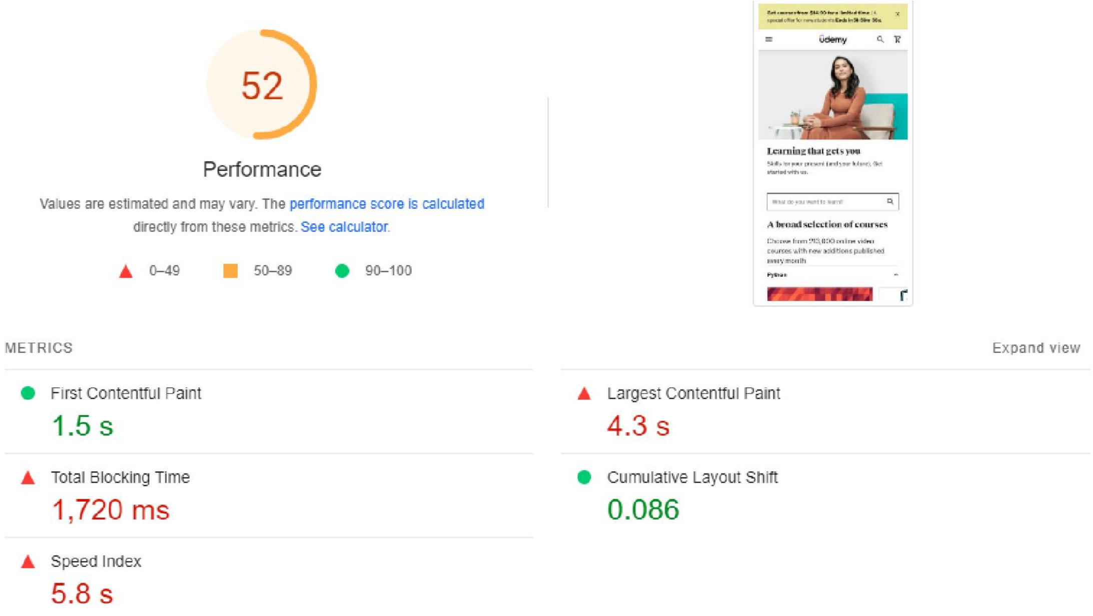
▲ Speed Index
13.4 s
Speed Index shows how quickly the contents of a page are visibly populated. Learn more.

▲ Total Blocking Time
28,530 ms
Sum of all time periods between FCP and Time to Interactive, when task length exceeded 50ms, expressed in milliseconds. Learn more.

▲ Largest Contentful Paint
Error!
Something went wrong with recording the trace over your page load. Please run Lighthouse again. (NO_LCP)

● Cumulative Layout Shift
0
Cumulative Layout Shift measures the movement of visible elements within the viewport. Learn more.

Fig. 2. Flutter web – Canvas Kit renderer



52

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

Udemy

Learning that gets you

A broad selection of courses

Python

METRICS Expand view

● First Contentful Paint
1.5 s

▲ Largest Contentful Paint
4.3 s

▲ Total Blocking Time
1,720 ms

● Cumulative Layout Shift
0.086

▲ Speed Index
5.8 s

Fig. 3. React JS

DOI: 10.48175/IJARSCT-8892

III. CONCLUSION

Both React JS and Flutter web have well-established communities and a rich package ecosystem, which means that developers can easily find the tools and resources they need to build their applications. Consequently, the development experience is similar between the two. In terms of SEO and accessibility, however, there are notable differences between React JS and Flutter web. While React JS is a UI library that can create standard single-page applications, it offers various options for managing SEO, such as server-side rendering, pretender, or managing SEO through the frontend server. In contrast, Flutter web is not suitable for text-heavy or SEO-focused applications, and the Flutter team has acknowledged that it may only be suitable for a limited range of web use cases. Therefore, if SEO, Performance and accessibility are a priority, React JS may be a more appropriate choice.

REFERENCES

[1]. Web Development - https://blog.openclassrooms.com/en/2018/03/28/web-development-definition/

[2]. Coded Websites vs WordPress Websites (Research Paper)

[3]. Flutter - https://en.wikipedia.org/wiki/Flutter_(software)

[4]. Cross-platform Development Advantages - https://www.clariontech.com/blog/cross-platform-application-development-benefits-and-technology

[5]. Programming languages- https://www.browserstack.com/guide/flutter-vs-react-native

[6]. Search Engine Optimization - https://cinnamon.agency/blog/post/flutter_and_seo

[7]. Backend Connectivity - https://betterprogramming.pub/how-to-connect-flutter-to-node-js-3e13b356c08d

[8]. Cross-Platform App Development With Flutter - https://medium.com/rpdstartup/cross-platform-app-development-with-flutter-2b97a7eacfa0

[9]. React Native vs Flutter: Which One is Better for 2023? - https://fireart.studio/blog/flutter-vs-react-native-what-app-developers-should-know-about-cross-platform-mobile-development/

[10]. Performance Comparison - https://codewithandrea.com/videos/flutter-web-html-css-js-performance-comparison/

[11]. https://www.reddit.com/r/FlutterDev/comments/oueci0/how_would_you_compare_react_and_flutter_web/

DOI: 10.48175/IJARSCT-8892