IPTEK The Journal of Technology and Science, 33(1), 2022 (e/pISSN:2088-2033, 0853-4098)

DOI: 10.12962/j20882033.v33i1.13904 | Received 12 Jan, 2022; Revised 14 June, 2022; Accepted 27 July, 2022

REVIEW PAPER

A SYSTEMATIC LITERATURE REVIEW ON PROGRESSIVE WEB APPLICATION PRACTICE AND CHALLENGES

Reza Fauzan*[1] | Ice Krisnahati[2] | Bima Dinda Nurwibawa[2] | Della Aulia Wibowo[2]

[1]Electrical Engineering, Politeknik Negeri Banjarmasin, Banjarmasin, Indonesia

[2]Dept. of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Correspondence

*Reza Fauzan, Electrical Engineering, Politeknik Negeri Banjarmasin, Banjarmasin, Indonesia. Email: reza.fauzan@poliban.ac.id

Present Address

Gedung D Politeknik Negeri Banjarmasin,
Jl. Brig Jend. Hasan Basri, Banjarmasin
70124, Indonesia

Abstract

This research aims to establish a knowledge of Progressive Web Application (PWA) method practices based on published empirical investigation, the problems that the PWA method may face, and the PWA method's approach to experimental studies. We also looked for PWA practices that might address the issues with the prior method. We performed a broad, automated search to find SLRs and 43 papers published from 1 January 2015 to 6 November 2021 that discuss PWA. Specific inclusion and exclusion criteria were applied to determine which relevant studies we used for our research purposes. The review identified 31 practices of PWA, Six challenges of the previous approach of mobile application development that get resolved by PWA, and seven challenges posed by the practice of PWA. PWA was introduced in 2015, and since 2017 the studies on PWA topics have started increasing. The distribution of studies based on nations is spread evenly. 74% of selected studies are about the practice of the PWA. Although it cannot be generalized that the PWA approach handles the challenge of the previous method, some studies reported that PWA covers the problems of the previous methods. However, despite its benefits, PWA still has some challenges, mostly related to browser support.

KEYWORDS:

Progressive Web App, PWA, Survey, Systematic Literature Review, Web Application

1 | INTRODUCTION

According to Malavolta et al.[1], Progressive Web App (PWA) is a new mobile application development architecture with advantages over its predecessors. It can work with low or no internet connection, run in the background, and assist the push notification. It is unique due to its progressive steps[1]. Not all progressive web apps can be called progressive web apps, but PWA is always a progressive web app[2]. PWA was introduced in 2015[3, 4].

TABLE 1 Summary of selected literature on progressive web app reviews at large.

Reference	Goal	Research Questions
Lee et al.[9]	Review of security and privacy aspects unique to PWAs.	1. Conducted the first systematic literature review of PWA focused on security and privacy risk. 2. Analyzed push notifications to detect the risk of phishing. 3. conducted an in-depth security analysis of push services of PWAs. 4. Presented a novel cache-based side-channel attack. 5. Show a new form of exploitation that targets service workers. 6. Offer mitigations for the security and privacy issues that have been addressed.
Biørn et al.[2]	Review the concepts of technologies behind PWA and compare them with other approaches.	1. Introduce the principles and technology that underpin advanced web applications. 2. Provide a technical comparison.

It is a combination of web and app. Users can visit the browser in a tab and decide whether to install it or not. As the users build relationships with the app through repeat use, the app can send relevant push notifications. Literature reports those applications that adopt PWA exhibit more efficiency[5], perform better than the Android version[6], and are proven as a credible solution for software as a service in an IoT[7] increased performance[8].

Although the practices mentioned above claimed the benefit of the PWA method for developing apps, the PWA method might have some challenges faced by the developer and event to the end-users. The developers or users are the ones driving us to map out the published evidence about PWA techniques and issues. The purpose is to learn how the PWA approach solves previous approaches.

The remainder of this paper is structured as follows: Section 2 discusses the importance of creating literature reviews on PWAs. Section 3 presents our research question and the method followed for reviewing PWAs. The key findings of our research are outlined in Section 4. The results are discussed in Section 5. The essay is concluded in Section 6, which also outlines the study's shortcomings and offers recommendations for future research and industry professionals.

2 | RELATED WORKS

During this review, we found two papers[2, 9] that have reported reviews on different aspects of PWA. Table 1 shows the summary of selected literature on progressive web app reviews at large. Lee et al.[9] presented a systematic literature review to identify security and privacy aspects unique to PWAs. Their defensive proposals were to improve the safe usage of PWAs in practice. The results are the consequence of intrinsic PWA capabilities that deliver native app-like Web browsing experiences, putting the concerns addressed exclusively to PWAs.

Biørn et al.[2] conduct a systematic literature review to introduce the concepts of technologies (PWA) and compare them with other approaches. Andreas finds that PWA can be used as an approach to web-native development without using a cross-platform framework.

Although PWA is a new method, academic contributions regarding PWA have started increasing. However, no studies are still mapping out PWA's approach, practices, and challenges. The goal of doing a complete systematic literature review is to gather information on PWA and, as a result, close or at least decrease the research gap. The table below outlines some of the research primarily focused on PWA.

3 | RESEARCH METHOD

In our research, we basically followed Kitchenham and Charters[10] guidelines and other additional research[11–13]. We next went over the essential stages of our systematic review, which included planning, conducting, and reporting the results.

TABLE 2 Search sources

Electronic Databases	ACM Digital library, IEEE Xplore, SpringerLink, and ScienceDirect
Searched items	Search applied on Full text—to avoid missing any of the papers that do not include our search keywords in titles or abstracts but are relevant to the review object
Language	English
Publication period	From January 2015 to November 2021

3.1 | Research Question

We investigate the role of PWA approaches. This research aims to establish a knowledge of PWA method practices based on published empirical investigations, the problems that the PWA method may face, and the PWA method's approach to experimental studies. We also looked for PWA practices that might address the issues with the prior method. To satisfy these goals, we figured out the accompanying research questions:

RQ1. What are the adopted practices of Progressive Web Apps?

RQ2. What challenges of the previous approach to mobile app development get alleviated by Progressive Web Apps?

RQ3. What are the challenges of Progressive web apps?

3.2 | Search Process

Kitchenham and Charters[10, 11] work was utilized as a guideline for conducting the investigation. After outlining our research goals and queries, we began by formulating a systematic search strategy to analyze all accessible empirical materials related to the review's purpose. As illustrated in Table 2 , the plan included establishing the search space encompassing electronic databases and printed proceedings. The papers were first acquired from electronic databases. Then they were analyzed to find more relevant research using reference searches (snowballing). Then, as indicated in Section 3.3, the inclusion and exclusion criteria were applied to the recovered studies in two rounds, each involving a different number of researchers.

The search was performed on four digital libraries in the IEEE Computer Society Digital Library, ACM, SpringerLink, and Science Direct. All searches are based on metadata. The survey was conducted between January 2015 and November 2021. researchers used a series of simple search strings to aggregate the results of each search for each source.

We conduct a query from each database by quoting to search the phrase and add a wildcard (*). However, some databases did not support wildcards nested in the quotations. Finally, we used the keyword "*progressive web application*" OR "*progressive web applications*" OR "*Progressive Web App*" OR "*progressive web apps*" for all databases. As indicated in Table 2 , the plan included establishing the search space, which comprised electronic databases. Then we conducted a snowball search method for each retrieved paper. We created the search string manually based on each database's search functionality. Each database search was treated as a learning and testing opportunity.

3.3 | Study Selection

In deciding a study that should be included, the following inclusion and exclusion criteria were used:

Inclusion criteria: (I1) it is a peer-reviewed publication; (I2) it is written in English; (I3) it is relevant to the search terms defined in Section 3.2; (I4) it is an empirical research paper, an experience report, or a workshop paper; and (I5) it was published between January 2015 and November 2021.

Exclusion criteria: (E1) studies that do not explicitly focus on PWA methods but mention them as a side topic (e.g., studies that use PWA as an adjective); (E2) studies that do not meet inclusion criteria; and (E3) opinion, viewpoint, keynote, discussions, editorials, comments, tutorials, prefaces, and anecdote papers and presentations in slide formats without any associated papers.

The specified electronic databases were searched, and the studies were retrieved using the search technique (described in Section 3.2). We found 27 results from IEEE Xplore, 34 results from Science Direct, 72 from ACM, 222 from Springer Link, and 355

TABLE 3 Several studies were discovered during the several iterations of our comprehensive search.

Database	Retrieve	Round-1 Include	Round-1 Exclude	Round-2 Include	Round-2 Exclude
IEEExplore	27	26	1	12	14
ACM	72	23	49	6	18
Springer Link	222	55	167	22	33
Science Direct	34	17	17	3	14
Total	355	122	232	43	79

TABLE 4 Quality criteria for study selection.

Criteria	Respond grading
(C1) Is the study's goal/objective well-defined?	1, 0.5,0(Yes, nominally, No)
(C2) Is the research context adequately addressed?	1, 0.5,0(Yes, nominally, No)
(C3) Are the findings conveyed clearly?	1, 0.5,0(Yes, nominally, No)
(C4) How beneficial is the research based on the findings?	>80% = 1, <20% = 0, in-between = 0.5

studies in this original search, as shown in Table 3 . We should note that we just chose data sets that distribute peer-evaluated papers (I1). By applying the consideration measures, a broad investigation of the examinations' titles and edited compositions was made by the scientists haphazardly (Round 1). A large portion of the recovered examinations fell inside the inclusion criteria I3. Although we set English as the language criteria in the databases (I2), we still found 1 study in another language except for the title and abstract. The search engines' restrictions prevented them from applying the search string to the whole body of the paper's text. Thus a sizable portion of the results they had found had to be dropped. After this initial categorization, we were left with 122 potential studies. Additionally, we ensured that the articles we got did not contain debates, editorial observations, tutorials, prefaces, or presentations (I4). The studies were published between January 2015 and November 2021.

The researchers then examined the selected papers at random in Round 2 in order to apply the exclusion criteria (E1, E2, E3, and E4). We had a virtual consensus conference to evaluate the agreements and disputes expressed by the researchers in their assessments. The researchers reviewed the whole manuscript and then rejected the studies based on the established exclusion criteria for the publications where consensus could not be reached after applying the inclusion criteria to the 122 papers that were pre-selected. Seventy-nine were eliminated since they did not explore any issue directly connected to the scope of our inquiry (E1 to E4). As a result, we have narrowed our final list to 43 studies (see the two rightmost columns in Table 3).

3.4 | Quality Assessment

The methodological quality of the main papers selected for review was assessed using the quality criteria developed by Guyatt et al.[14]; the criteria were also utilized by Dybå and Dingsøyr[15] to assess the quality of empirical research on PWA approaches. These quality criteria (shown in Table 4) consist of inquiries determining how satisfying research is and how it will add to the investigation's scope. The criteria involve the research' completeness, reliability, and importance. We picked these criteria because (i) the quality indicators related to these criteria have been utilized in various recent systematic reviews, and (ii) they may be used to assess the value of synthesis findings and interpretation[10, 15, 16].

Each study was assessed using the quality evaluation criteria listed in Table 4 . Instead of a binary scale, we used an ordinal scale based on our quality evaluation criteria (Table 4) to properly categorize and rate the research. The first criteria (C1) entailed evaluating each study's goal. In all of the research, this question was answered affirmatively. The second criteria (C2) examined how well the research setting was handled and presented. In 94 percent of the trials, this question was responded to affirmatively. We sought a clear description of findings in each research in the third criteria (C3). In 71% of the investigations, this question was responded to affirmatively. We stated above developed the heuristic ratings for the quality measures (C4).

The Kappa coefficient, a statistical metric, and Intraclass Correlation Coefficients (ICC) were employed to determine inter-rater agreement among the researchers. The ICC value was determined to be 0.659, indicating strong or considerable agreement[17, 18]. Independent quality assessments were undertaken for the studies, and disputes were addressed through discussion.

TABLE 5 Distribution of research based on the mode of publishing.

Publisher	Type	Number
ACM	Conference	6
Science Direct	Conference	1
	Workshop	2
IEEE	Conference	11
	Journal	1
Springer	Conference	19
	Journal	3

3.5 | Data Extraction

We created a data extraction process to gather important data from the 43 primary publications included in the study that answered our research concerns, following the criteria offered by Kitchenham and Charters[10, 11]. Our data extraction procedure includes the following steps: We started by making a form to keep track of each study's ideas, views, contributions, and findings. Using this format assures that higher-order interpretation may be made afterward. Each publication yielded the following information: (i) publication source; (ii) title; (iii) authors; (iv) reference; (v) database; (vi) connection to the subject, i.e. PWA issues, problems, practices, models, methods, techniques; (vii) methodology (interview, case study, report, survey); (viii) methodology (interview, case study, report, survey); (viii) citation; (x) future work; (xi) limitations; (xii) analysis country/location; and (xiIi) publication year.

4 | FINDINGS OF OUR REVIEW

4.1 | Overview of Studies

We found 43 studies, as previously mentioned. The distribution of study publishing sources is shown in Table 5 . About 14 percent (6 research) were presented at conferences published in ACM, 2 percent (1 study) in Science Direct, 26 percent (11 studies) in IEEE, and 44 percent (19 studies) in Springer. The studies presented in Journal were 2% (1 study) in IEEE and 7% (3 studies) in Springer. The studies presented at the workshop were only published in Science Direct, about 5% (2 studies).

Our findings in Table 5 indicate that the research primarily studies PWA topics published in Springer at conferences about 44%, almost half of the studies we have found. It means there are slight preferences that PWA authors prefer at the Conference in Springer.

We found no relevant studies connected to our study subject before 2016 regarding publication years. There were few studies we found until 2017, which means three years after PWA was published in 2015. In 2017, we can see that the number of studies started increasing. The distribution of peer-reviewed articles published between 2016 and 2021 is shown, along with the investigation's central themes of interest (see Fig. 1 and 2 , respectively).

According to Fig. 2 , 65% of the 43 studies were focused on PWA practices. In comparison, 23 percent were focused on newly presented concepts in the form of techniques. Only 7% of the methodologies and models for PWA are based mainly on comparing previous approaches with PWA. The other 5% of the research explores PWA in general.

Our sample of 43 papers was from the same nations within the same research. We can see that most of them are from South America, Asia, and Europe (refer to Fig. 3). It is also clear that there is little research on PWA from North America and none from Australia and Africa.

Fig. 4 shows the research methodologies used in 43 selected studies. We discovered that the vast majority of research is exploratory. We found that 15 studies are "empirically evaluated" or based only on the assessment of methodologies, without any experimental research. Otherwise, we found 28 studies "empirically based." We discovered that 27 empirically-based studies are case studies and only one survey. We also found that in empirically evaluated studies, 12 are case studies, 1 is a survey, and 2 are reports.

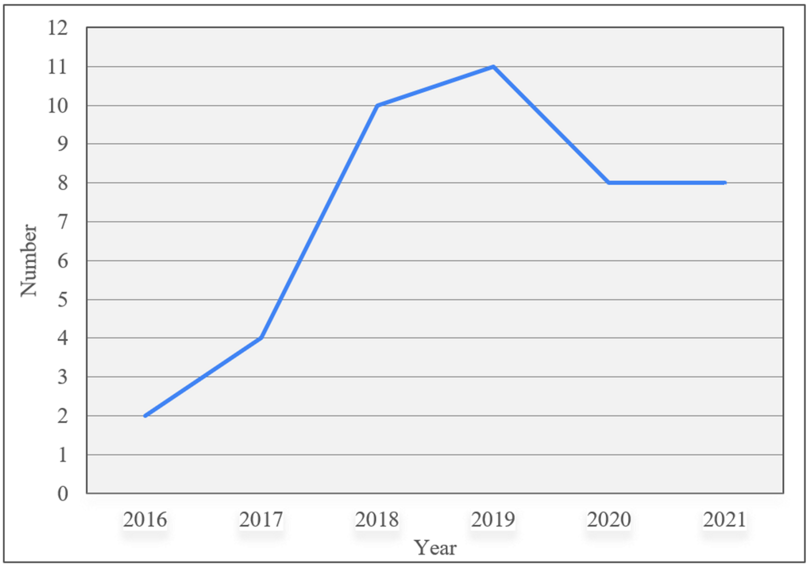

FIGURE 1 Distribution of selected research by year.

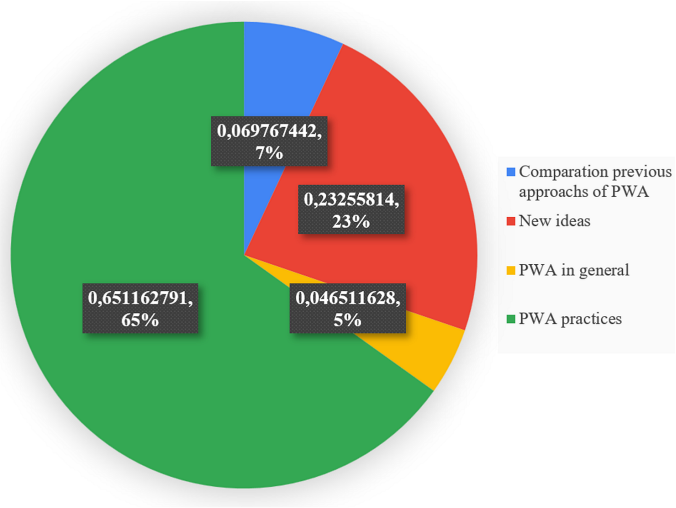

FIGURE 2 Categorization of study basis.

4.2 | (RQ1) What are The Adopted Practices of Progressive Web Apps?

Below, we describe the six practice categories we found from 43 studies to be adopted in PWA development. We identify each practice for potential challenges. Table 6 shows the surveys that report the frequency of occurrence and each practice.

Architecture - Mena et al.[19] use microservice architecture. PWA is a hybrid solution that allows it to be accessed on various platforms. The miniature frontend procedure makes it conceivable to make the U.I. powerfully and autonomously develop visual parts, thinking of new strategies to show user information. Santos[20] utilize a range of ordinary mobile phones, respectively iOS and Android systems. With PWA, everything is discoverable, shareable, linkable, and rankable. It means PWA makes it easy for users to share links and easy to be installed[21]. De Andrade[22] propose the PWA-EU strategy, a development of the PWA architecture that incorporates customer preferences into run-time interface modification.

Framework requirement - It were declared that service workers do not significantly affect the energy efficiency of PWA on low-mid and high-end mobile devices also on empty or populated caches. Malavolta et al.[1, 28] performed an application from

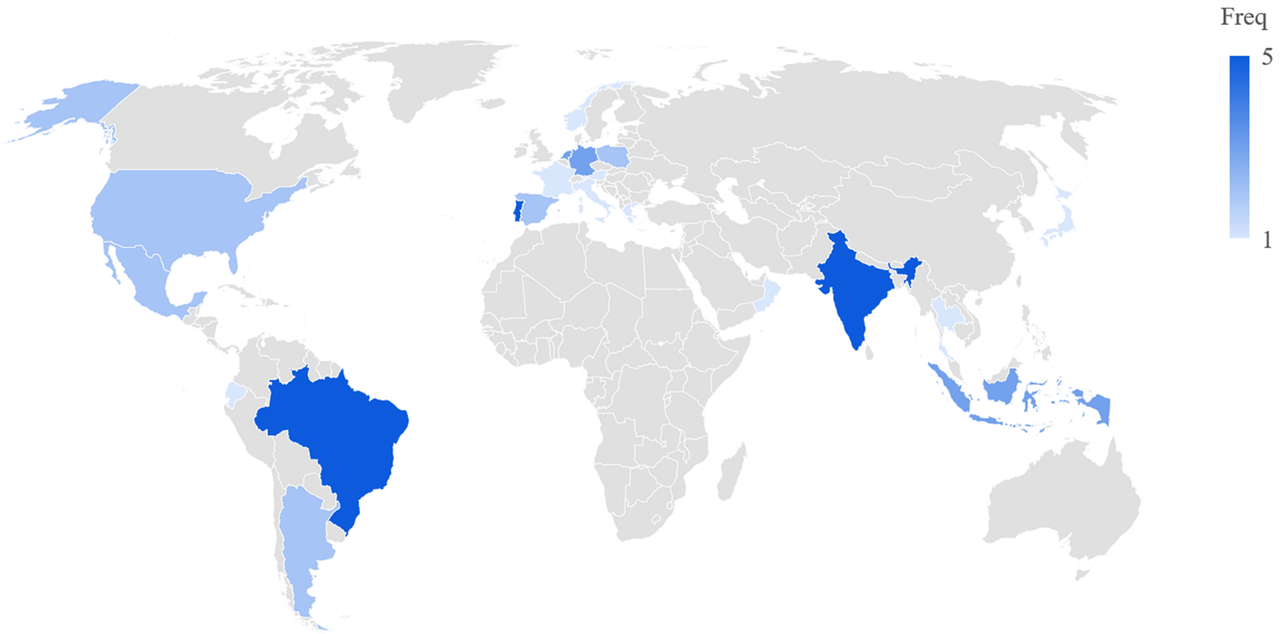

FIGURE 3 Geographical distribution of the authors.

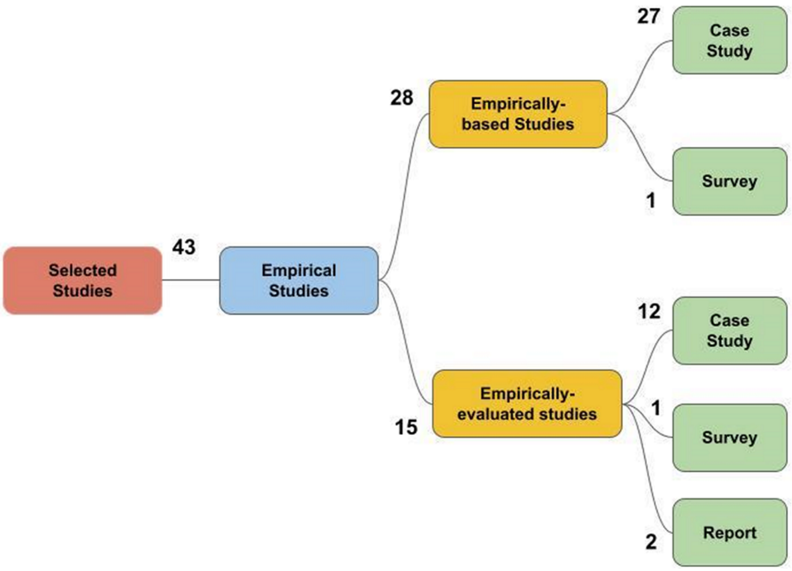

FIGURE 4 Distributions of research studies depending on research methodology include.

a proxy that controls the service worker is on/off state. In other studies, service workers make applications run faster. Pre-caching work done by the backend service employee is the cause of this[6, 27]. Pande et al.[26] proposes harnessing service worker infusion framework-equipped proxy servers for improved user experience, quicker page load times, and practical resource caching[24, 25]. PWA makes it possible to work offline with cache management using service workers. Huber et al.[23] have compared PWAs, and other mobile development approaches. The outcomes demonstrated that PWAs are a practical substitute for mobile cross-platform development.

Network connection - In this context, PWA means offline availability. Due to a pre-cache management strategy used by PWA with the assistance of service personnel, offline accessibility is possible[29]. When the mobile device has an internet connection, the PWA user can fill out and submit a form offline. The data will be synchronized to the server. The service worker, the cache

TABLE 6 Summary of procedures and the corresponding research that looked at them.

Practice	Freq.
Architecture [19–22]	4
Framework requirement [1, 6, 23–28]	8
Network Connection [5, 29]	2
Security [30]	1
Tools requirement [2, 7, 25, 27, 31–38]	12
User experience [39–42]	4

API, and the BackgroundSync API are three essential components that allow the application to operate independently of network circumstances [5].

Security - Security was assessed in PWA; push web service is considered a security. After checking new features for security flaws, a thorough examination of the Web push features was performed [30]. Channels on the Subresource Integrity Mechanism (SIM) were examined to use the discovered flaws. The man-in-the-middle assault is only a model for an attack that may be carried out on several communication levels. Man-in-the-Middle attacks can be prevented by avoiding suspicious public networks, and the security of the transport must be assured.

Tools requirements - PWA integrated with API, which can support the use of technology as monitoring information [31], Mantellos [32] also takes advantage of the API which is integrated with PWA as a front end is written in HTML5 and is using the Bootstrap frontend framework. JavaScript-based and open source technologies were employed in creating PWA: Front-end use of React.js [2], MongoDB serves as the database, with Node.js serving as the back end. Service workers are used to handling assets like CSS or JSP files and all URLs of web applications [27]. PWA-EU, an expansion of the conventional PWA technique linked to EUD principles, was presented. It would enable end users to choose design preferences, including them in the app's creation [33]. Npm is used in package management on the backend [34]. Form.io allows users to decorate component groups and field input [35]. To store variables that would permit sending active alerts, Firebase has to be set up [36]. Rivera et al. [37] suggests that PWAs built using the Ionic framework will function with Angular, enabling access to REST services via HTTPS requests to retrieve data from databases. PWA utilizing Angular or React now get support from Ionic. It was essential to build two files during the development transformation into a PWA: one for the manifest and the other for the service worker configuration. Installing the "@angular/pwa" Angular package will automate the process. While the manifest gives the web browser information, such as the program name and icon, the service worker file configures the application cache strategy for offline execution [25]. Wijaya et al. [7] Used Google Maps Javascript API to get real-time data from Firebase Firestore. A web application may fully use cloud computing by implementing the Twelve-Factor App approach driven by docker. In the meantime, a PWA also offers advantages like unique contexts since it only utilizes HTTPS, the ability to leverage web browser APIs to store necessary data on the client side, and the ability to create a native-like user experience with self-update without requiring manual updating [38].

User experience - Alvarado-Uribe et al. [39] propose that PWA improves tourist experience with an intelligent Point of Interest (POI). PWA is easy to use for all kinds of users [40]. Mukhopadhyay et al. [41] in financial transactions, PWA offers security and customers comfort doing transactions. Whittemore et al. [42] PWA provides users a seamless experience regardless of Internet connection type and allows us to utilize push notifications.

4.3 | (RQ2) What Challenges of The Previous Approach to Mobile App Development Get Alleviated by Progressive Web Apps?

In traditional mobile application development, the ability of the web platform to compete with native or cross-platform apps fell behind mobile-centered innovation. PWA approach has been introduced with features such as offline support, background synchronization, and home-screen installation to the web to counter many challenges in traditional mobile application development. According to Andreas et al. [22], the absence of academic study on the PWA topic was noted as part of the theory search, despite the industry's investment and interest. Summary of challenges of the previous mobile app development approach resolved by PWA practices is shown in Table 7 .

TABLE 7 Summary of challenges of the previous mobile app development approach resolved by PWA practices.

No	Challenge	Practice	Description
1	Cost [40, 43]	Tools requirement	PWA's approach is used in order to avoid the cost of a native implementation.
		User experience	Similar user experiences may be delivered by PWA and native applications. In contrast, native apps for iOS and Android must be created, which results in high development expenses.
2	Development issue [8, 20, 40]	Tools requirement	PWA is more efficient than simple web apps with less complexion of the development and installation, unlike native apps. Provide support for various popular smartphones, both in iOS and Android systems.
3	Energy consumption [23]	Framework requirement	When running in Chrome, PWA consumes much less energy than React Native.
4	Low internet connectivity [5–7, 26, 29, 44]	Framework requirement	PWA size in the home feed load is 150 KB (Analysis of Cache in Service Worker and Performance Scoring of Progressive Web Application). The Tinder PWA application is 15 times smaller to install than the original Tinder application. The PWA benefits offline accessibility thanks to the service worker's pre-cache technique.
5	Storage issue [44]	Framework requirement	PWA may be updated on the server-side and needs less storage space.
6	User experience [1, 7, 21, 22, 42, 45, 46]	Architecture	PWA expands the possibilities for user experience regardless of the user's environment.
		Tools requirement	Increased client engagement rates are possible because of the PWA's ability to incorporate native app features like alerts.

The problems we discovered in our review, including the approaches to solve them, are outlined below. These problems are summarized in Tables 6 and 7 , together with the studies that discovered them. We divide the category of the issue into the developer side and end-user side. Some issues in the previous approach by the developer side are follows.

Cost is one of the considerations in developing applications [43]. The need to create native apps for iOS and Android results in high development expenses [40]. The user interface of a PWA might be identical to that of a native application. It can support a wide range of popular devices running on both iOS and Android. Therefore, it costs less than developing both the iOS and Android platforms for the same application [40].

Development issues occur in previous app development approaches. According to Luntovskyy [8], using the PWA approach, which has similarities with native apps in the user experience case, can be a solution for the complexity of developing in different O.S. Instead of the time-consuming development and installation of native applications, a web browser can be utilized in conjunction with optimization software.

After summarising the challenge of app development's previous approach, we present some of the challenges from the previous app development approach based on the end user's side. Some of the challenges based on end-users of previous app development approaches are follows.

Energy consumption in react native is reported to have higher energy consumption than PWA. However, the study also reported that PWA exhibited a higher energy footprint than native development [23].

A low internet connection can be a problem for web apps. The PWA approach comes with a solution. Like native apps, the PWA can work with a low internet connection, even in a no-connection environment. Offline caching also helps speed up page loading by avoiding asking the server for data that has already been placed into the cache [7, 26].

The storage issue is a common problem in mobile apps. Because of bandwidth and storage constraints, Robert converted his mobile app to a progressive online app. PWA takes up less storage space and may be updated on the server-side [44].

User experience related to the other issues such as low internet connection, energy consumption, and storage issues. Progressive Web Apps provide a consistent user experience across various devices [21].

4.4 | (RQ3) What are The Challenges of Progressive web apps?

Web pages are now backed by developments in web application development with capabilities close to natively developed applications. Users can use the progressive web app (PWA) approach as a native application with several features. The PWA

TABLE 8 Summary of challenges of PWA.

Challenge	Description	Impact	Solution
Browser compatibility [2, 5, 6, 25, 28, 29, 41, 47]	The PWA architecture still needs work regarding browser compatibility and the necessity for a storage space akin to play stores. For instance, not all of the APIs needed to operate PWAs are supported by Apple's Safari web browser. PWAs can only be used with web browsers that support them since they must be installed through them.	Functional issue	-
End-user development [33, 48]	The effects of connecting EUD and PWAs have not been fully explored because PWAs are a relatively new technology. The typical PWA method has limitations when it comes to involving users as co-designers.	User satisfaction	PWA-EU approach [33]
Framework issue [27, 31, 49]	Browser compatibility is one of the elements that might make the adoption of this app challenges. Because Google is the primary force behind PWAs, this is not a problem if the user is using Google Chrome.	Functional issue	Intelligent framework [49]
Network issue [50]	Although service workers in advanced online applications are successfully solving this issue, web programs are inherently cross-platform and heavily rely on connection.	Functional issue	-
Novel device compatibility [51, 52]	The majority of innovative device types, such as the popular wristwatch operating systems watchOS and Wear O.S. from Google, do not, however, include WebView components or browser engines that permit the execution of JavaScript code. As a result, this method cannot be applied to multi-platform development that targets a broader variety of devices.	Functional issue	Munster App Modelling Language (MAML) [51, 52]
User Interface [53, 54]	The PWA received a "Performance" score of 86, meaning that it loads quickly regardless of the state of the network. PWA received an "Accessibility" score of 78 because particular objects lacked labels or the contrast between the background and foreground colors was insufficient.	User Satisfaction	-
Security [41]	Progressive web applications (PWAs) strive to strike a balance. However, they have significant drawbacks, such as a lack of stricter security measures, which are crucial for financial operations.	Trust issue	They are creating a banking transaction-specific browser [41].

development method makes it simple for both users to access content [31]. Web apps will be able to enable native applications thanks to this advancement, known as the Progressive Web App (PWA), notably in the creation of multi-platform applications that can be accessed by multiple devices and run on various operating systems [7]. Summary of challenges of PWA are shown in Table 8 .

The challenges of Progressive web, description, impact and solution approaches identified in our review are discussed below. Our review discusses the progressive web approach challenges, descriptions, impacts, and solutions identified below. Seven summaries discuss.

First, Progressive Web Apps (PWAs) allow web apps to be designed using an offline-first strategy due to their browser compatibility. Traditional temporary Web apps struggle to function without an internet connection. PWAs enable offline download, installation, and use [2]. A preliminary empirical study on PWAs. Our findings demonstrate the potential for energy efficiency of PWA and service employees' technologies [1, 28].

Second, Google introduced the Progressive Web App (PWA), an innovative method for creating mobile applications (apps), in 2015. The effects of connecting EUD and PWAs have not been fully explored because PWAs are a relatively new technology. We propose the PWA-EU approach, an extension of the standard PWA architecture that incorporates EUD notions. The existing PWA method is constrained regarding users participating as co-designers [33].

Third, framework issue is used for lightweight web applications and consumes less memory. Web applications use the browser engine and take up much time to launch compared to the original application, especially after booting the device or if the browser [49]. Recent technology called a "Progressive Web App" (PWA) offers offline support, giving the same app a native feel on mobile devices, and making it possible for the app to run on most devices [27]. Utilizing technology, such as the Internet of Things, is one enhancing (IoT) approach. To have the actual and current circumstances [31].

Fourth, network problem Applications for WIVR are distributed as native apps. Web-based and using the VRaaS (Virtual Reality as a Service) paradigm, XOOM apps. A few years ago, native desktop programs were replaced by cloud-based services (like SaaS), and we now see a similar trend on mobile platforms. End users of WIVR can gain from a transfer of this kind of app to the web. They do not bother downloading programs that they might use once. The Internet allows quick access to materials by avoiding the time-consuming installation process (download, install, obtain rights, open) instead of just clicking a link to launch a virtual reality environment in a browser. Because they just require a browser and are independent of the underlying operating system, web applications are inherently cross-platform. Consequently, they are simpler to distribute and manage[50].

Fifth, we suggest the Munster App Modeling Language (MAML) express business apps using graphical building pieces with a high degree of abstraction for innovative device compatibility. Unlike current process modeling notations, these models can be instantly converted using code generators into apps for many platforms and device classes without the need for manual programming[51, 52].

Sixth, the user interface (U.I.) is a collection of graphical components users use to interact with applications. The interface components could consist of. As a result, the responsive web design (RWD) guidelines must be followed. It implies that the page layout must change to automatically fit the application window's size. For web applications, this objective is accomplished using a specially modified HTML structure and specific cascading style sheet rules (CSS). For instance, the application's buttons, multimedia components, and various colors[53].

Seventh, Progressive Web Applications (PWA) attempt to keep security in a middle ground. However, there are significant drawbacks, such as a lack of stricter security controls, which are crucial for financial operations[41].

5 | DISCUSSION OF THE FINDINGS

The geographic locations of authors were a critical factor in our study of the 43 papers we chose. We found that the highest number of selected studies in nations is less than six studies. In most studies, about 46% were distributed in European nations. It was observed that 25% of it was from Portugal (5 studies). The rest were distributed in European countries such as Austria, French, Germany, Greece, Italy, Netherlands, Norway, Poland, and Spain. The second place is Asia with about 26% (11 studies). The most contributed country in Asia was India, with about 46% of total studies in Asia.

Moreover, she was followed by Indonesia at 27%, Japan at 9%, Oman at 9%, and Thailand at 9%. The third place is South America with about 19% (8 studies). Brazil had the highest number of studies from South America, with about 63% of total studies in South America. They were followed by Argentina at 25% and Ecuador at 12%. The last place in North America, about 9% (4 studies), contributed by the United States and Mexico. However, considering the deviation between countries, there is no significant gap in the number of studies. From the above discussion, it can be said that the distribution of studies was spread evenly between Europe, North America, South America, and Asia. As for Australia and America, we could not find any studies from selected studies. However, it cannot be concluded that there are no studies in both regions. It might be because the studies did not match our criteria, as explained in Section 3.3.

Important aspects were highlighted in the analysis of our 43 selected studies. Empirical Studies are divided into two parts, Empirically based Studies and Empirically- evaluated studies. Empirically-based Studies has 28 papers from that section, and there are two aspects. In the case study and survey aspects, 27 papers are part of the case study, and one paper is part of the survey. The second empirical study, Empirically-evaluated studies, contains 15 papers in this section. Twelve papers are included in the case study section, one in the survey, and two papers are included in the report.

From 43 studies adopted in PWA development, we identified the six current practices in response to RQ1. We can see that every year there is growth in practice at PWA. In category architecture, research on PWAs in 2021 is collaborating with the E.U., where user preferences are considered in interface adaptation at run-time. The framework requirements category highlights that the service worker plays an essential role in PWA. The service worker manages pre-caching in the background, which allows PWA to load pages faster and makes it possible to work offline for a better user experience. If we look at the tools requirements category of PWA, the subsequent development trend is more towards cloud computing. Security needs to be considered in the development of PWAs. There is little research that discusses security in PWAs. However, the research we reviewed explains that there are vulnerabilities in PWAs. Hence, we feel that more research is needed to define specific practices for security issues from PWA.

In response to RQ2, we highlighted previous approach difficulties that can be addressed by the PWA approach. As previously mentioned, we divided it into two sides: the developer and the end-user. We found cost and development issues from the developer side. For example, native apps must be developed both in Android and iOS, leading to some complexity, time-consuming, and impact on app development costs. React native is one of the solutions to develop mobile apps for two operating systems (Android and iOS) at once. However, it would raise another challenge for the end users. We found some challenges in the previous approaches from the end user's side. Generally, it can be grouped as user experience. However, we can break it down into energy consumption, low internet connection, and storage issues. As we can see, the challenges arise from various practices. Common challenges of traditional web apps were common connectivity issues, which would not be a problem for native apps. As previously mentioned, react native or cross-platform can be a solution for developers. However, it raises challenges for end users, such as energy consumption. The studies that answered RQ1 proved that the PWA approach was used to handle the challenge of the previous app development approach. PWA could work as well as native in low or no network connectivity. It also has lower energy consumption than react native but can handle complexity and cost problems in native apps. However, not all selected studies focused on previous traditional approaches lead to incomplete information about other challenges the previous approach might have. Therefore, more research should be conducted to cover the gap.

We have identified 16 from 43 studies that mentioned PWA challenges. Browser compatibility was the most reported by studies (about 44%) as a PWA challenge. There is no clear solution identified. Since browser compatibility refers to Safari, it might be a solution if Apple considers supporting Service Workers API. Other challenges can be categorized based on the impact. The challenges that impact user satisfaction, such as end-user development and user interface reported in the study, are related to displaying and end-user development problems. The next is the trust issue. It is quite an important topic related to security. However, we found only a few talks about this topic. A solution was provided, but it is limited only to Banking applications. Besides browser compatibility, we found other challenges such as framework issues, network issues, and novel device compatibility, e.g., smartwatch, which lead to functional issues of the apps.

6 | CONCLUSIONS

This paper presents a systematic literature review on the practices and challenges of PWA. This study was conducted by searching and categorizing all current and available literature on PWA using published principles for performing systematic literature reviews. A multistage screening approach with independent validation at each phase was used to retrieve 43 relevant papers from 355 initial papers found in reputed electronic research databases.

These papers were then scored through quality assessment between the researchers. The extraction of features was done and categorized based on research questions: (i) practice of the PWA approach; (ii) challenge of the previous approach that PWA resolved; (iii) practical challenge of PWA. The outcomes of our study give business and research practitioners new perspectives on how to work on the PWA in the future.

As we noticed, the practice of the PWA approach is increasing not only in industrial areas but also in studies yearly since its first introduction. Therefore, more research is required on PWA and its adjustment to narrow the challenges of the PWA approach based on security. We also noticed that the study distribution was spread evenly through regional countries. Despite its benefits, PWA still has challenges. In this study, it was revealed that the most challenging reported are browser compatibility issues. Furthermore, the studies show that the topic is mostly about tool requirements and framework requirements of PWA, which are also primarily identified for handling challenges of the previous app development approach.

6.1 | Implication of The Study

This study might have several implications for both practitioners and researchers. For researchers, it might help to provide insight into the PWA topic in the world, how PWA approaches are implemented, and research trends for related topics. As we learn that the practice of PWA has increased, challenges will be found. Therefore, the studies of new ideas might help to handle new challenges that may arise. As for the practitioners, this study might help in considering the right approach for their app development that is suitable for their needs. This study might give more views on PWA practice and trends for related topics of PWA.

6.2 | Limitation of Our Study

This study might have limitations, especially bias in study selection. Therefore, we define a search string key to ensure all related studies were included. Next, we selected the studies based on inclusion and exclusion criteria. Then the ICC calculation was conducted to see an inter-rater agreement score. We got 0.659 for an ICC score that indicates good agreement.

Furthermore, we discovered that several of the publications we considered for inclusion in our analysis lacked adequate information. More precisely, we discovered that the degree of information used to describe the study process in the 43 publications differed significantly. Some research, for example, assessed validity risks in greater detail than others. Because of these differences in research, it is possible that mistakes in the data extraction method occurred.

We know that the underlying reasons for the identified issues may not have been examined in detail in any studies documenting challenges. We further believe that the writers of some research selected the challenges for specific reasons not expressed directly in the articles. However, because the authors were focused on their specific study aims and topics, it is possible that the fundamental causes for the problem were neglected.

CREDIT

Reza Fauzan: Conceptualization, Methodology, Writing - review and editing, Validation, Formal analysis, and Supervision. **Ice Krisnahati:** Writing - original draft preparation, Data Curation, Formal analysis and investigation, Resources, and Visualization. **Bima Dinda Nurwibawa:** Writing - original draft preparation, Data Curation, Formal analysis and investigation, Resources, and Visualization. **Della Aulia Wibowo:** Writing - original draft preparation, Data Curation, Formal analysis and investigation, Resources, and Visualization.

References

1. Malavolta I, Chinnappan K, Jasmontas L, Gupta S, Soltany KAK. Evaluating the impact of caching on the energy consumption and performance of progressive web apps. In: Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems; 2020. p. 109–119.

2. Biørn-Hansen A, Majchrzak TA, Grønli TM. Progressive web apps for the unified development of mobile applications. In: International Conference on Web Information Systems and Technologies Springer; 2017. p. 64–86.

3. Hume D. Progressive web apps. Simon and Schuster; 2017.

4. Fortunato D, Bernardino J. Progressive web apps: An alternative to the native mobile Apps. In: 2018 13th Iberian Conference on Information Systems and Technologies (CISTI) IEEE; 2018. p. 1–6.

5. Rêgo F, Portela F, Santos MF. Towards PWA in healthcare. Procedia Computer Science 2019;160:678–683.

6. Gambhir A, Raj G. Analysis of cache in service worker and performance scoring of progressive web application. In: 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE) IEEE; 2018. p. 294–299.

7. Wijaya PR, Crisgar PV, Pakpahan MD, Syamsuddin EY, Hasanuddin MO. Implementation of Motor Vehicle Tracking Software-as-a-Service (SaaS) Application Based on Progressive Web App. In: 2021 International Symposium on Electronics and Smart Devices (ISESD) IEEE; 2021. p. 1–6.

8. Luntovskyy A. SLMA and novel software technologies for Industry 4.0. In: International Multi-Conference on Advanced Computer Systems Springer; 2018. p. 170–184.

9. Lee J, Kim H, Park J, Shin I, Son S. Pride and prejudice in progressive web apps: Abusing native app-like features in web applications. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security; 2018. p. 1731–1746.

10. Kitchenham B, Charters S. Guidelines for performing systematic literature reviews in software engineering. Keele University; 2007.

11. Kitchenham B, Pretorius R, Budgen D, Brereton OP, Turner M, Niazi M, et al. Systematic literature reviews in software engineering–a tertiary study. Information and software technology 2010;52(8):792–805.

12. Inayat I, Salim SS, Marczak S, Daneva M, Shamshirband S. A systematic literature review on agile requirements engineering practices and challenges. Computers in human behavior 2015;51:915–929.

13. Triandini E, Fauzan R, Siahaan DO, Rochimah S, Suardika IG, Karolita D. Software similarity measurements using UML diagrams: A systematic literature review. Register: Jurnal Ilmiah Teknologi Sistem Informasi 2021;8(1):10–23.

14. Guyatt GH, Oxman AD, Kunz R, Woodcock J, Brozek J, Helfand M, et al. GRADE guidelines: 7. Rating the quality of evidence—inconsistency. Journal of clinical epidemiology 2011;64(12):1294–1302.

15. Dybå T, Dingsøyr T. Empirical studies of agile software development: A systematic review. Information and software technology 2008;50(9-10):833–859.

16. Pacheco C, Garcia I. A systematic literature review of stakeholder identification methods in requirements elicitation. Journal of Systems and Software 2012;85(9):2171–2181.

17. Landis JR, Koch GG. An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. Biometrics 1977;p. 363–374.

18. Fauzan R, Siahaan DO, Rochimah S, Triandini E. A novel approach to automated behavioral diagram assessment using label similarity and subgraph edit distance. Computer Science 2021;22(2).

19. Mena M, Corral A, Iribarne L, Criado J. A progressive web application based on microservices combining geospatial data and the internet of things. IEEE access 2019;7:104577–104590.

20. Santos-Gago JM, Álvarez-Sabucedo L, González-Maciel R, Alonso-Rorís VM, García-Soidán JL, Wanden-Berghe C, et al. Towards a personalised recommender platform for sportswomen. In: World Conference on Information Systems and Technologies Springer; 2019. p. 504–514.

21. Fernandes G, Portela F, Santos MF. PWA and pervasive information system–a new era. In: World Conference on Information Systems and Technologies Springer; 2020. p. 334–343.

22. de Andrade Cardieri G, Zaina LM. Analyzing user experience in mobile web, native and progressive web applications: A User and HCI specialist perspectives. In: Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems; 2018. p. 1–11.

23. Huber S, Demetz L, Felderer M. Pwa vs the others: A comparative study on the ui energy-efficiency of progressive web apps. In: International Conference on Web Engineering Springer; 2021. p. 464–479.

24. Cândido PH, da Silva-Santos CH, Votto LF, Ambrosio LA. Semantic Web-based System for Light Scattering Using the Generalized Lorenz-Mie Theory. In: 2019 PhotonIcs & Electromagnetics Research Symposium-Spring (PIERS-Spring) IEEE; 2019. p. 3217–3224.

25. Aguirre V, Delía L, Thomas P, Corbalán L, Cáseres G, Sosa JF. PWA and TWA: Recent Development Trends. In: Argentine Congress of Computer Science Springer; 2019. p. 205–214.

26. Pande N, Somani A, Samal SP, Kakkirala V. Enhanced web application and browsing performance through service-worker infusion framework. In: 2018 IEEE International Conference on Web Services (ICWS) IEEE; 2018. p. 195–202.

27. Loreto P, Braga J, Peixoto H, Machado J, Abelha A. Step towards progressive web development in obstetrics. Procedia Computer Science 2018;141:525–530.

28. Malavolta I, Procaccianti G, Noorland P, Vukmirovic P. Assessing the impact of service workers on the energy efficiency of progressive web apps. In: 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft) IEEE; 2017. p. 35–45.

29. Khan AI, Al-Badi A, Al-Kindi M. Progressive web application assessment using AHP. Procedia Computer Science 2019;155:289–294.

30. Wróbel T, Kędziora M, Szczepanik M, Jóźwiak PP, Jóźwiak AM, Mizera-Pietraszko J. Progressive Mobile Web Application Subresource Tampering During Penetration Testing. In: International Conference on Advanced Information Networking and Applications Springer; 2021. p. 297–306.

31. Nugroho LE, Pratama AGH, Mustika IW, Ferdiana R. Development of monitoring system for smart farming using Progressive Web App. In: 2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE) IEEE; 2017. p. 1–5.

32. Mantellos G. WOL Ecosystem: Secure Remote Power–State Control of Computer (s) Over the Internet for Telemedicine Purposes and Dementia Patients. In: GeNeDis 2018 Springer; 2020.p. 67–71.

33. de Andrade Cardieri G, Zaina LA. PWA-EU: Extending PWA approach for promoting customization based on user preferences. Proceedings of the ACM on Human-Computer Interaction 2019;3(EICS):1–28.

34. Inkane NS, Kotak SA, Manekar AS. Splay: A Lightweight Video Streaming Application. In: 2019 International Conference on Innovative Trends and Advances in Engineering and Technology (ICITAET) IEEE; 2019. p. 139–143.

35. Namee K, Phoarun R, Albadrani GM, Polpinij J, Tanessakulwattana S, Sphanphong P. A Form and API Data Management Platform for Progressive Web Application and Serverless Application Architecture. In: Proceedings of the 2019 2nd International Conference on Computational Intelligence and Intelligent Systems; 2019. p. 144–149.

36. Nayak B, Mugali PS, Rao B, Sindhava S, Disha D, Swarnalatha K. Geofencing-based accident avoidance notification for road safety. In: Emerging Research in Computing, Information, Communication and Applications Springer; 2019.p. 379–386.

37. Rivera J, Tapia F, Terán D, Aules H, Moncayo S. Use of e-Health as an Accessibility and Management Strategy Within Health Centers in Ecuador Through the Implementation of a Progressive Web Application as a Tool for Technological Development and Innovation. In: International Conference on Software Process Improvement Springer; 2020. p. 199–212.

38. Adhiguna KA, Rusli FM, Irawan H. Building an ID Card Repository with Progressive Web Application to Mitigate Fraud based on the Twelve-Factor App methodology. In: 2021 9th International Conference on Information and Communication Technology (ICoICT) IEEE; 2021. p. 544–549.

39. Alvarado-Uribe J, Gómez-Oliva A, Molina G, Gonzalez-Mendoza M, Parra-Meroño MC, Jara AJ. Towards the development of a smart tourism application based on smart POI and recommendation algorithms: Ceutí as a study case. In: International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing Springer; 2017. p. 904–916.

40. Kinoshita K, Ni B. Development of Learning Support Platform as PWA and Its Usability Evaluation. In: 2020 6th International Conference on Engineering, Applied Sciences and Technology (ICEAST) IEEE; 2020. p. 1–3.

41. Mukhopadhyay I, Ghosh A. Banking Transaction Considerations on Standardized Browser Architecture. In: Proceedings of International Conference on Computational Intelligence, Data Science and Cloud Computing Springer; 2021. p. 635–644.

42. Whittemore M, Toubeau S, Griffin Z, Deligiannidis L. ActiviX: Noninvasive Solution to Mental Health. In: Advances in Computer Vision and Computational Biology Springer; 2021.p. 339–347.

43. Mendes FM, Poppi R, Parra H, Moreira B. EJ: A Free Software Platform for Social Participation. In: IFIP International Conference on Open Source Systems Springer; 2019. p. 27–37.

44. Bwana RM, Baart A, de Boer V, Lenfant F, Morisho N, Westermann-Behaylo M, et al. Developing a crowdsourcing application for responsible production in Africa. In: 12th ACM Conference on Web Science Companion; 2020. p. 48–53.

45. Peraza J, Quiñonez Y, Lizarraga C, Ortega J, Olivarría M. Identification of visually impaired users for customizing web pages on the Internet. In: International Conference on Software Process Improvement Springer; 2016. p. 285–292.

46. Andrade Cardieri Gd, Zaina LA. How Communication Breakdowns Affect Emotional Responses of End-Users on Progressive Web Apps. In: International Conference on Enterprise Information Systems Springer; 2020. p. 660–682.

47. Steiner T. What is in a web view: An analysis of progressive web app features when the means of web access is not a web browser. In: Companion Proceedings of the The Web Conference 2018; 2018. p. 789–796.

48. Guarda T, Haz L, Augusto MF, Vitor JA. Pervasive Smart Destinations. In: World Conference on Information Systems and Technologies Springer; 2018. p. 375–382.

49. Gudla SK, Sahoo JK, Singh A, Bose J, Ahamed N. Framework to improve the web application launch time. In: 2016 IEEE International Conference on Mobile Services (MS) IEEE; 2016. p. 73–78.

50. Garzotto F, Gelsomini M, Matarazzo V, Messina N, Occhiuto D. XOOM: An end-user development tool for web-based wearable immersive virtual tours. In: International Conference on Web Engineering Springer; 2017. p. 507–519.

51. Rieger C, Majchrzak TA. A taxonomy for app-enabled devices: mastering the mobile device jungle. In: International Conference on Web Information Systems and Technologies Springer; 2017. p. 202–220.

52. Rieger C, Majchrzak TA. Towards the definitive evaluation framework for cross-platform app development approaches. Journal of Systems and Software 2019;153:175–199.

53. Olczak A, Kaczmarek JM. Modern Devices and Software Solutions as a Tool for Education on Local Biodiversity: A Case Study. In: International Scientific Conference on Brain-Computer Interfaces BCI Opole Springer; 2021. p. 307–321.

54. Pombo N, Martins C. Test Driven Development in Action: Case Study of a Cross-Platform Web Application. In: IEEE EUROCON 2021-19th International Conference on Smart Technologies IEEE; 2021. p. 352–356.

How to cite this article: Fauzan R., Krisnahati I., Nurwibawa B. D., Wibowo D. A., (20XX), A systematic Literature Review on Progressive Web Application Practice and Challenges, *IPTEK The Journal of Technology and Science*, *33(1):43–58.*