VIVA-TECH IJRI
ANNUAL RESEARCH JOURNAL
VIVA-TECH INTERNATIONAL JOURNAL FOR RESEARCH AND INNOVATION
ANNUAL RESEARCH JOURNAL
ISSN(ONLINE): 2581-7280

Progressive Web Applications (PWA's): Detailed Comparison with existing Mobile Application Development Architectures.

Shloka Patil[1], Chirag Shah[2]

[1](Department of Master of Computer Applications, Viva Institute of Technology, INDIA)
[2](Department of Master of Computer Applications, Viva Institute of Technology, INDIA)

Abstract : *Progressive Web Apps (PWAs) are web applications built using common web technologies including HTML, CSS, JavaScript, and Web Assembly. PWA's which combine the features of native as well as web development turned out as a better alternative amongst other development approaches due to additional benefits such as offline capability, background synchronization, etc. Service Workers allows the app to work offline by intercepting network requests. This research highlights at carrying out a detailed comparative study on the mobile application development architectures using the Systematic Literature Review technique, performing feature comparison on the Native Application, Mobile Web Application, Hybrid Application and Progressive Web Application (PWA) architecture and finally contend for PWA development architecture based on the comparisons. This comparison will assist research scholars and development firms in understanding the concept of PWA, thus encouraging them to adopt this strategy for further development.*

Keywords - *Cross-platform, Hybrid Applications, Mobile Application Development, Native Applications, Progressive Web Applications, Service Workers.*

I. Introduction

Over the previous couple of years, there has been a continuing rise within the number of mobile devices and its users. Mobile technology has evolved rapidly over the last decades which has made quite five (5) billion people possess a mobile device during which 57 percent of mobile devices are smartphones [1,2]. These statistics show a constant rise within the affinity of individuals towards mobile devices, especially smartphones. Therefore, it is imperative to satisfy the requirements of the increasing number of smartphones users by constantly developing applications (apps) that span through different sectors of life starting from education to health to entertainment and so on. This has given mobile applications a distinct nomenclature like m-education, m-health, m-government, m-entertainment, then on so as to completely differentiate it from other varieties of applications. Different smartphones vendors adopt a specific mobile platform like android, windows, iOS, blackberry, Symbian, then on [3] upon which mobile applications are built. Mainly, mobile application architecture is classified into the native app which is totally smitten by a mobile device platform, mobile web apps which makes use of web technologies like HTML, CSS, and JavaScript providing more flexibility for mobile application development across various platforms and also the hybrid architecture which harnesses the pros of both the native and mobile architecture [4,5]. Progressive web application is an emerging technology that has been adopted by some mobile developers within the industry, however, because of existing applications that are developed over the period of time using the native, mobile web, and hybrid architecture, doubts are raised about the requirement, success, and acceptability of progressive web application (PWA).

II. LITERATURE REVIEW

PWA is a leading-edge technology that is gradually gaining academic involvement in terms of research. This is obvious from the handful of research articles as regards PWA across various academic search engines which will be duly reviewed. Mobile Application Development team can adopt one or more of the existing development strategies from native apps to mobile web apps, hybrid apps and now the PWA. Native app development strategy happened to be the first that existed, it consists of binary executable files that are directly downloaded and kept stored in to a user's mobile device [6]. Apps developed using this architecture is platform dependent and is simply distributed via a dedicated app store (Google Play Store, Apple App Store) depending on the platform adapted by the mobile device vendors. [7] Identified high development time, high testing and maintenance cost as a major challenge of the native app, [6] called this a challenge of mobile device fragmentation which indicates that a code written for one mobile platform (for e.g., java code for android app) cannot be used for another platform such as Apple iOS app which is in Objective C. In an attempt to overwhelm the challenges of the native app where each platform has its own Software Development Kit (SDK), several cross-platform architectures were developed which allows deployment of mobile solutions using a single SDK. A survey of several cross-platform approaches was carried out by [8] while [3] discussed the taxonomy of these cross-platform approaches. These approaches recognized are the web approach which are used in developing mobile applications using web technologies (Hypertext Markup Language (HTML), Cascading Style Sheet (CSS) and JavaScript (JS)) hosted on a remote server thereby making it platform independent because the mobile-optimized website/app are accessed via a browser app such as Google Chrome, Mozilla Firefox or Safari which must be pre-installed on user's mobile devices [7,9]. A major challenge of this approach is that apps are only accessed via a Uniform Resource Locator (URL) using a reliable and persistent internet connections which implies that apps cannot be downloaded via various app stores. The hybrid approach according to [3,8] tried exploiting the advantages of the native and web architecture. In the hybrid approach, mobile applications are developed using the common web technologies but displayed inside the native apps and are distributed via various app stores. Other approaches which uses a common programing language such as JavaScript to write a code which successively generates the equivalence for the native component for each platform, the cross-compile approach which enables developers to write code using any common programming language which are then transformed by cross compilers to a specific native code. To overcome these challenges proposed by the various mobile development approaches (architectures) as identified, another development approach known as **Progressive Web App (PWA)** as coined by [10] was developed. Reference [11] provides a general introduction to the concept and technologies behind PWA by exhibiting some major features and providing technical comparison alongside existing mobile development architectures. Biørn-Hansen and his group performed a measurement-comparison of the size of installation, launch time and the time from application-icon tap tool bar render among the hybrid, interpreted and PWA mobile development approach. The result indicated that PWA had the least size of installation as well as the smallest launch time but has the highest time from app-icon tap tool bar render. To further elaborate the general concept and technology of PWA, [12] discussed the architectural designs on which the PWA is based that is responsible for the improved performance, loading time of mobile apps.

III. MOBILE APPLICATION DEVELOPMENT ARCHITECTURES

Mobile applications commonly referred to as an apps are application programs developed and optimize for mobile devices such as smart phones and tablets [16]. Mobile apps are like the traditional application but have some distinct features that distinguishes it from regular apps. [17] identified requirements that clearly distinguishes a mobile applications from traditional apps some of which are:

1. **Potential Interactions with Other Applications:** This means that mobile devices might have numerous applications from different sources are likely to interact with other applications residing in the device.

2. **Sensor Handling:** Mobile applications can access several sensors such as accelerometer, GPS, microphone, cameras and so on.

3. **Families of Hardware and Software Platforms:** There are dissimilar mobile platforms which might require developers to build several apps for different platforms.

4. **Security:** Mobile platforms are vulnerable due to different attacks because they are "open" which can allow the installation of new malware applications that can affect the overall operations of the device.

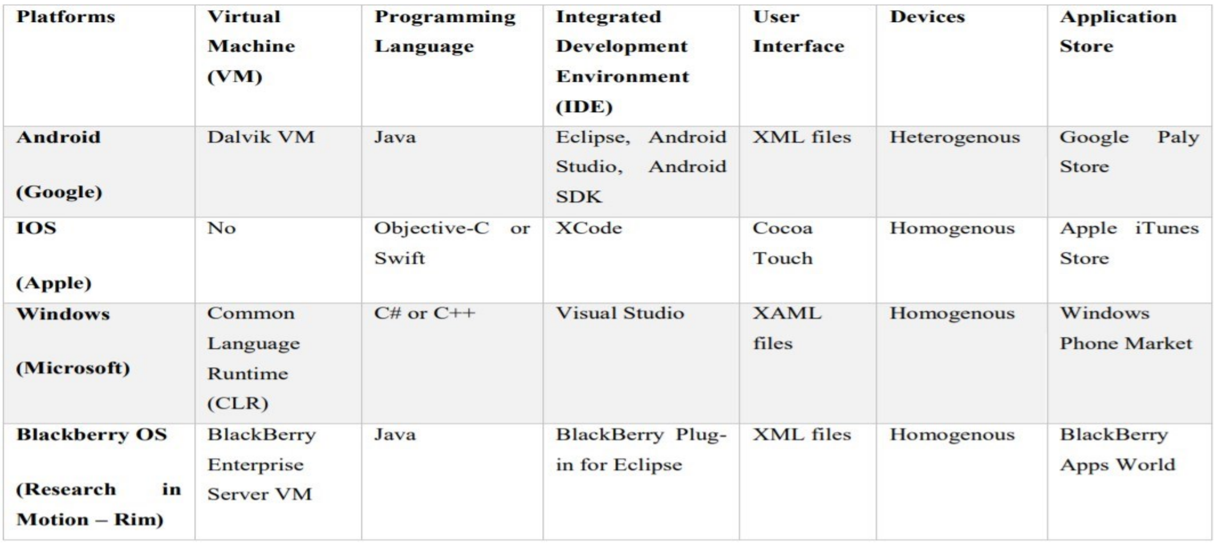
5. **User Interface:** Mobile applications cannot be designed in a singular manner due to the fact that mobile devices come in different sizes and shapes.

6. Complexity of Testing: This is a difficult task as a simple application need to be tested on several devices as well as under different network conditions this is so because the development platform is not the same as where the application will be used.

7. Power Consumption: Software must be optimized to maximize battery life.

Table 1 depicts a detailed difference between mobile development platforms. Despite the differences across platforms, the uniqueness of each platform – specific API, tools and technologies enable developers to create applications with a good user experience and increased performance [7,9].

Table 1: Difference between four mobile platforms



Platforms	Virtual Machine (VM)	Programming Language	Integrated Development Environment (IDE)	User Interface	Devices	Application Store
Android **(Google)**	Dalvik VM	Java	Eclipse, Android Studio, Android SDK	XML files	Heterogenous	Google Paly Store
IOS **(Apple)**	No	Objective-C or Swift	XCode	Cocoa Touch	Homogenous	Apple iTunes Store
Windows **(Microsoft)**	Common Language Runtime (CLR)	C# or C++	Visual Studio	XAML files	Homogenous	Windows Phone Market
Blackberry OS **(Research in Motion – Rim)**	BlackBerry Enterprise Server VM	Java	BlackBerry Plug-in for Eclipse	XML files	Homogenous	BlackBerry Apps World

There are **four (4)** ways in which mobile applications can be developed. These are **native app**, **mobile web app**, **hybrid app** and the emerging Progressive Web Application **PWA**. A relative study on the various development approaches will be carried out based on a SLR.

3.1. Native Applications

These are apps developed using tools and programming languages dedicated for a certain mobile platform [3]. Native applications are platform dependent, hence programmers must confirm to the specific programming languages and tools needed to successfully develop the app. A major disadvantage of this development approach is mobile platform fragmentation as identified by [13] – meaning that for a development firm to gain more audience across varying platforms, there must be the re-development of the same app across different technologies and tools specific to each desired platform. This leads to rise in development time, development cost, efforts, maintenance cost and low portability.

Figure 1 diagrammatically shows the approach of native mobile development.

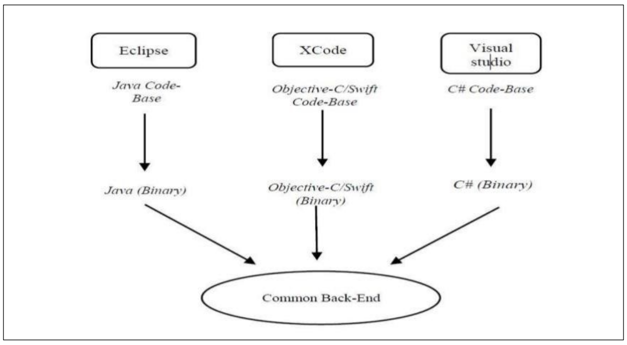


Fig. 1: Mobile Native Development Approach [8]

3.2. Mobile Web Applications

These are mobile optimized web applications developed based on common web technologies such as HTML, CSS and JavaScript. They are hosted on remote servers and are accessed using specific URL via web browsers installed on a user mobile device [3,7,9]. This makes the mobile web platform independent because the web browser serves as its runtime environment. This approach enforces optimization of web application such as taking into consideration the screen sizes of various devices as well as their usage philosophy.

Figure 2 diagrammatically shows the web approach of mobile development as amended from [8]. Mobile web apps adopt the client-server model where a service requester (client) makes certain calls or request to a service provider (server) which in turn respond to the request of the client. The back-and-forth communication is handled by an application-level protocol (HTTP).

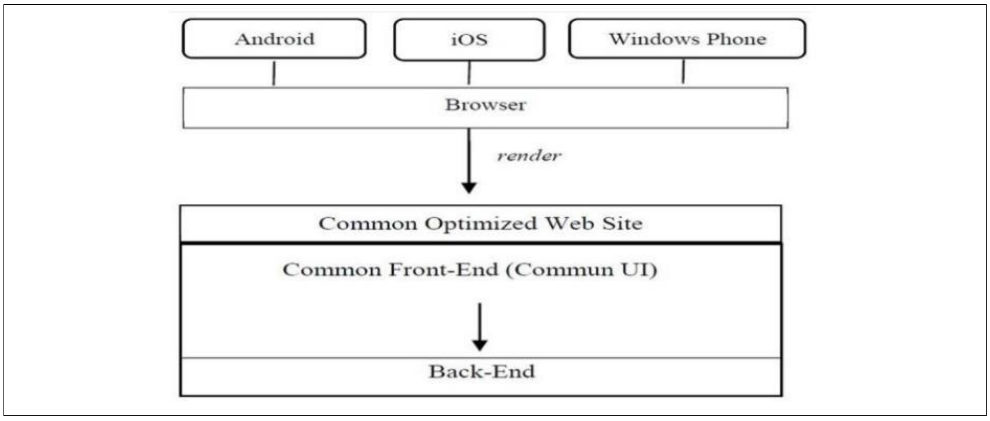


Fig. 2: Mobile Web Development Approach [8]

3.3. Hybrid Approach

The approach tries to herness the benefits of both the native and web approach thereby overcoming some limitations posed by both the approaches. Applications developed using the hybrid approach utilizes the browser engine in the mobile device and embeds the HTML content in the native web container (for example, WebView for android, UI WebView for iOS) [8]. The provision of certain mobile hybrid development frameworks such as Cordova, Ionic, PhoneGap, Mo Sync provides a native wrapper that contains the web-based codes and also a generic JavaScript API that serves as a bridge of the service request from the web-based code to corresponding platform's API [7].

Figure 3 shows a diagrammatic view of the Hybrid approach.

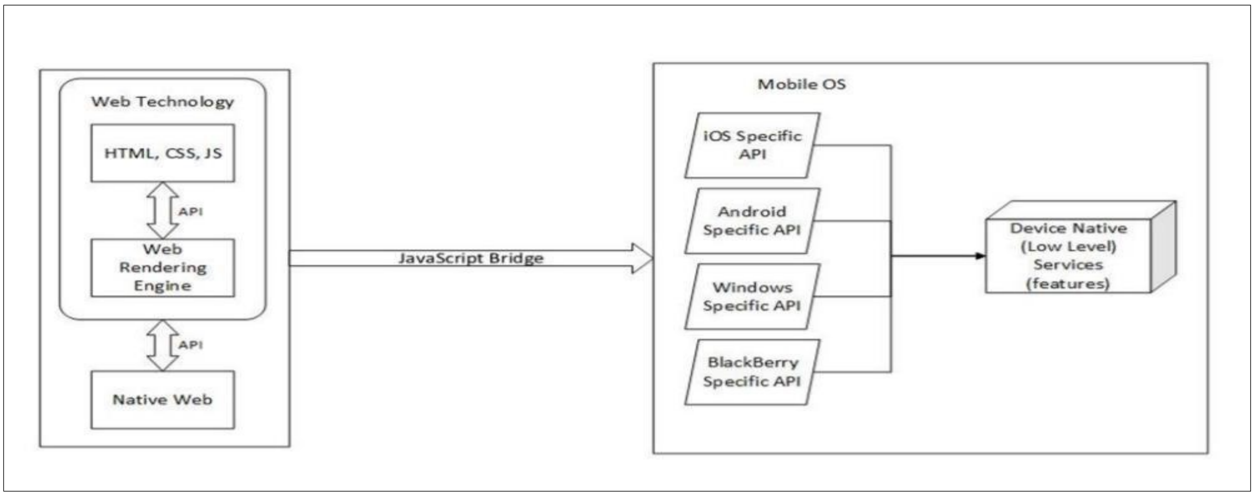


Fig. 3: Diagrammatic view of the Hybrid Mobile Development Approach [8]

The fourth approach for developing mobile app is the PWA. This will be discussed next as it is the main focus of this research work.

3.4. Progressive Web Application (PWA)

PWA is a mobile application development approach that seek to overcome the challenges or weaknesses of earlier approaches. Embracing this approach produces special kind of web apps which requires no installation before using and is served from a remote server via a secured Hypertext Transfer Protocol (HTTPS) unlike regular mobile web applications which might be served using the HTTP [7,12,22]. User of PWA is provided with a native app like experience by promoting the PWA to a top-level mobile application with a full screen support (no browser) after deciding to install the PWA on the user's device [7]. The PWA is based on the concepts of a single application for all platforms [5] just like the hybrid approach. However, it possesses distinct capabilities such as instant loading, push notification even in the offline state.
Figure 5 diagrammatically shows the PWA development approach.

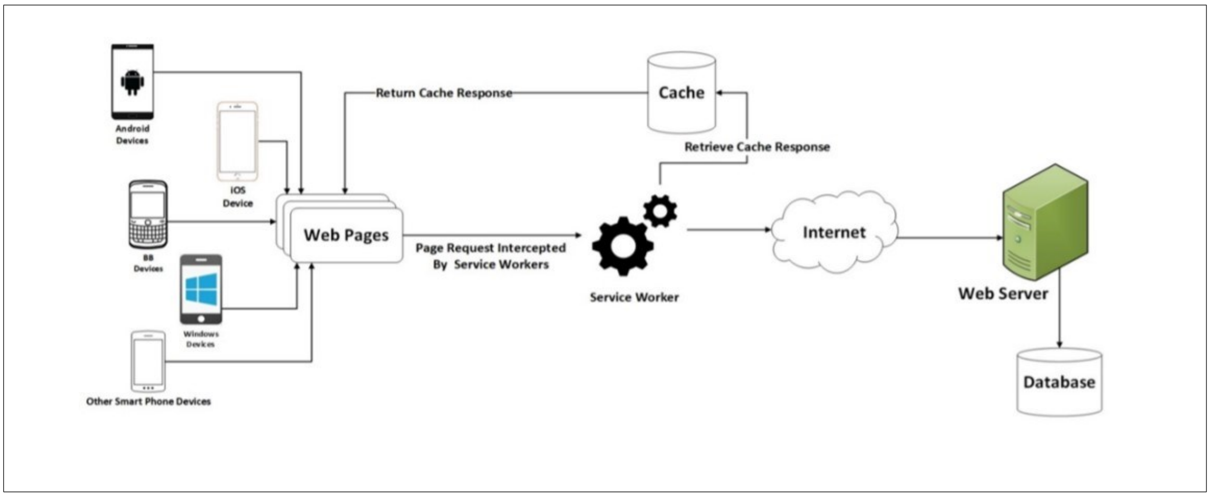


Fig. 5: PWA Development Approach Architecture (Researcher's Diagram)

Features of PWA

The PWA has compiled list of features that are the standard requirements for a PWA as identified below.

1. **Offline Capabilities:** PWAs have the ability to work to a great extent even if the device is offline (airplane mode or out of network coverage).

2. **Push Notification:** PWAs have the ability to display re-engaging notifications as defined in the push API.

4. **Background Synchronization:** Ability to synchronize data in background.

5. **Storage Estimation:** Ability to estimate the available storage that an application uses and also to know the amount of storage left.

6. **Web Share:** Ability to make use of the native sharing widget belonging to the Operating System (OS) as specified by the web share API.

7. **Cross-Browser Consumption:** Ability to work on major browsers.

8. **Page Unique Identity:** Every page has a unique URL which makes it linked with other pages.

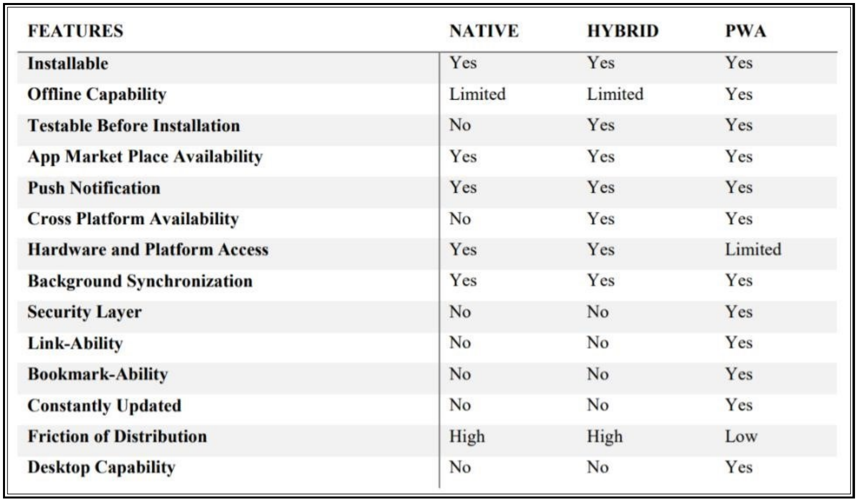
9. **Payment Request:** Ability to use the web payment request API to act as an intermediary among merchants and users.

Analysis of Feature Comparison:

1. **Installable:** This is the ability of mobile applications to be installed on the user's mobile device. This feature is possible in the three mobile development approach.

2. **Offline Capability:** This is the ability for a mobile app to work without an internet connection (in airplane mode or out of network coverage). Every app developed using the PWA approach has the ability to run to a great extent without an internet connection due to the presence of service workers.

Table 2: Feature comparison among the Native, Hybrid and PWA Mobile Development Approach



FEATURES	NATIVE	HYBRID	PWA
Installable	Yes	Yes	Yes
Offline Capability	Limited	Limited	Yes
Testable Before Installation	No	Yes	Yes
App Market Place Availability	Yes	Yes	Yes
Push Notification	Yes	Yes	Yes
Cross Platform Availability	No	Yes	Yes
Hardware and Platform Access	Yes	Yes	Limited
Background Synchronization	Yes	Yes	Yes
Security Layer	No	No	Yes
Link-Ability	No	No	Yes
Bookmark-Ability	No	No	Yes
Constantly Updated	No	No	Yes
Friction of Distribution	High	High	Low
Desktop Capability	No	No	Yes

3. **Testable Before Installation:** This implies that an app can be tried to see how it functions or operates before installation on the user's device. This feature is negative for the Native development approach and positive for the Hybrid and PWA approach.
4. **App Market Place Availability:** PWA apps are only accessed via a dedicated and unique URL, however from Google Chrome version 72 (android platform), the Trusted Web Activity (TWA) feature has been embedded which allows PWAs to be distributed via the Google Play Store.
5. **Push Notification:** This is the ability to display continuous engaged information to users. This feature is available for the three-development approach being compared.
6. **Cross Platform Availability:** This is the ability for a mobile app to be distributed or made available on all mobile platform such as the android, iOS, Windows, BlackBerry and so on. The PWA development approach is the only approach that makes a mobile application available to all mobile platforms without re-development for each platform and also with no overhead incurred.
7. **Hardware and Platform Access:** The amount of hardware features and sensors that can be accessed by PWA depends on the type of smartphone use. PWAs have greater possibilities of accessing more device features on Android smart phones compared to the iOS. This can easily be confirmed by visiting What Web Can Do Today on the smart phone that interest the developer.
8. **Background Synchronization:** All the mobile development approach as compared in Table 2 have the ability to synchronize data with the server in the background.
9. **Security Layer:** Mobile applications developed using the Native and the Hybrid development approach are not deployed on a secured layer which can lead to a compromise in the integrity of the application. On the flip side, PWAs can only be accessed via a secured layer – Hypertext Transfer Protocol Secured (HTTPS) which provides a high level of security for the app.
10. **Constantly Updated:** Applications developed using the Native and the Hybrid development approach are usually downloaded to the user's mobile devices and can only be updated whenever an update is provoked and accepted by the owner of the mobile device. This is not the case for apps developed using the PWA approach due to the fact that the apps are loaded from the web server, once an update is made by the developer, the apps are automatically updated and integrated on all mobile devices where the app resides which also facilitates the same view for all users.
11. **Desktop Compatibility:** Applications developed using the PWA development approach are desktop compatible, that is, they can be viewed and used on laptops and desktop computers without any distortion or hindrance. However, this is not the case for the Native and Hybrid approach where apps can only be accessed on mobile devices with the required and specific platform.

IV. CONCLUSION

The promises offered by PWA's can neither be underestimated nor compared to existing (traditional) mobile development strategies. Development firms strive to reduce development time, testing time and cost as well as general maintenance cost – which is relatively impossible while adopting the native and hybrid development architecture. The mobile web development approach has completely eradicated the challenge of mobile fragmentation which implies that a mobile app can now run on any mobile platform with the help of a browser and does not need to be re-developed. PWA has completely brought in a new dimension with the help of the service worker, app shell and other components which has facilitated the offline loading, background synchronization, push notification of mobile applications thereby making web apps look, feel and act similar to native and hybrid apps. This research makes a recommendation of the PWA to mobile app developers based on feature comparison and analysis. However, further experiments on the mobile development approach can be carried out in terms of memory management and efficiency on smartphones to further validate the claims of this work.

Acknowledgements

We are thankful to our institute for giving us this opportunity to make this research a success. We give our special thanks and sincere gratitude towards Prof. Neha Lodhe for encouraging us to complete this research paper, guiding us and helping us through all the obstacles in the research. Without her assistance, our research paper would have been impossible and also, we present our obligation towards all our past years teachers who have bestowed deep understanding and knowledge in us, over the past years. We are obliged to our parents and family members who always supported me greatly and encouraged us in each and every step.

REFERENCES Journal Papers:

[1] Mayuran Sivakumaran and P. Iacopino, “The Mobile Economy 2018,” GSMA Intelligence, pp. 5–11,2019.

[2] B. Y. K. Taylor and L. Silver, “Smartphone ownership is growing rapidly around the world, but not always equally,” Pew Res. Cent., no. February, 2019.

[3] W. S. El-Kassas, B. A. Abdullah, A. H. Yousef, and A. M. Wahba, “Taxonomy of Cross-Platform Mobile Applications Development Approaches,” *Ain Shams Eng. J., vol. 8, no. 2*, pp. 163–190, 2017.

[4]. N. Pande, A. Somani, S. Prasad Samal, and V. Kakkirala, “Enhanced Web Application and Browsing Performance through ServiceWorker Infusion Framework,” in Proceedings - *2018 IEEE International Conference on Web Services, ICWS 2018 - Part of the 2018 IEEE World Congress on Services*, 2018, pp. 195–202.

[5]. A. I. Khan, A. Al-Badi, and M. Al-Kindi, “Progressive Web Application Assessment Using AHP”, *Procedia Comput. Sci., vol. 155*, pp. 289–294, 2019.

[6]. IBM, “HTML5 , Hybrid or Native Mobile App Development,” White Paper, IBM Corporation, p. Document Number: WSW14182USEN, 2012.

[7]. I. Malavolta, “Beyond Native Apps: Web Technologies to the Rescue! (Keynote),” in Mobile! 2016 - Proceedings of the 1st International Workshop on Mobile Development, co-located with SPLASH 2016, 2016, pp. 1–2.

[8]. M. Latif, Y. Lakhrissi, E. H. Nfaoui, and N. Es-Sbai, “Cross platform approach for mobile application development: A survey,” 2016 Int. Conf. Inf. Technol. Organ. Dev. IT4OD 2016, pp. 1–5, 2016.

[9]. F. Johannsen, “Progressive Web Applications and Code Complexity-An analysis of the added complexity of making a web application progressive,” Linköping University, 2018. [10]. A. Russell, “Progressive Web Apps: Escaping Tabs Without Losing Our Soul,” Infrequently Noted, 2015. [Online]. Available: https://infrequently.org/2015/06/progressive-apps-escaping-tabs-withoutlosing-our-soul/. [Accessed: 05-Feb-2020].

[11]. A. Biørn-Hansen, T. A. Majchrzak, and T. M. Grønli, “Progressive web apps: The possibleweb-native unifier for mobile development,” in WEBIST 2017 - Proceedings of the 13th International Conference on Web Information Systems and Technologies, 2017, no. Webist, pp. 344–351. [12]. K. Behl and G. Raj, “Architectural Pattern of Progressive Web and Background Synchronization,” Proc. 2018 Int. Conf. Adv. Comput. Commun. Eng. ICACCE 2018, no. June, pp. 366–371, 2018.

[13]. I. Malavolta, G. Procaccianti, P. Noorland, and P. Vukmirovic, “Assessing the Impact of Service Workers on the Energy Efficiency of Progressive Web Apps,” in Proceedings - 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2017, 2017, pp. 35–45.

[14]. A. Gambhir and G. Raj, “Analysis of Cache in Service Worker and Performance Scoring of Progressive Web Application,” Proc. 2018 Int. Conf. Adv. Comput. Commun. Eng. ICACCE 2018, no. June, pp. 294–299, 2018.

[15]. L. E. Nugroho, A. G. H. Pratama, I. W. Mustika, and R. Ferdiana, “Development of monitoring system for smart farming using Progressive Web App,” 2017 9th Int. Conf. Inf. Technol. Electr. Eng. ICITEE 2017, vol. 2018-Janua, pp. 1–5, 2018.

[16]. [V. Sharma, R. Verma, V. Pathak, M. Paliwal, and P. Jain, “Progressive Web App (PWA) - One Stop Solution for All Application Development Across All Platforms,” Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 5, no. 2, pp. 1120–1122, 2019.