ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Visit: www.ijirset.com*

Vol. 7, Issue 9, September 2018

# Impact of Progressive Web Apps on Web App Development

Sayali Sunil Tandel [1], Abhishek Jamadar [2]

Student, Department of Computer Engineering, Smt.Indira Gandhi College, Navi Mumbai(Ghansoli), Maharashtra, India[1,2]

**ABSTRACT**:Innovation in technology always makes an impact on how products and services are designed. The invention of Smartphone especially and after the launch of Android OS which is free there is a huge increase in the use of smartphones. Most of the users use native mobile applications for browsing the contents of the particular industry. The other way is to browse the contents is through a web browser. But both the ways have limitations. The first way which is the native app, user is required to download the app firstly and then they must use it as per their requirements. This has two major disadvantages, one is it takes space on local storage of smartphone device and the network connection must be strong enough to operate it smoothly. The areas where 2G or lesser bandwidth or 3G network is available, it becomes a slow process to access this native app. The second way which is through the web browser has disadvantages since the user experience is not as great as native app. In order to overcome the above limitations, Google has provided a solution of Progressive Web App (PWA) which combines the best of web and mobile apps giving us a rich experience just like the native apps. It is a website built using web technologies that acts like an app. PWA is a website built using web technologies that acts like an app and is not required to be installed like a native app.

**KEYWORDS**:Progressive, Offline, Service Worker, AppShell, App Manifest.

## I. INTRODUCTION

In 2015, Designer Frances Berriman and Google Chrome engineer Alex Russell coined the term "progressive web apps" to describe apps taking advantage of new features supported by modern browsers, including service workers and web app manifests, that let users upgrade web apps to progressive web applications in their native operating system (OS). Native mobile applications can send push notifications, work offline, load on the home screen. Mobile Web Apps accessed in a mobile browser, by comparison, historically haven't done those things. Progressive Web Apps fix that with new Web APIs, new design concepts, and new buzzwords. Progressive Web Apps bring features we expect from native apps to the mobile browser. Such that it uses standards-based technologies and run in a secure container which is accessible to anyone on the web.Even with the availability of fully developed mobile web application, it still struggles to provide an eye pleasing and satisfactory experience to the users mainly because of lack of strong network connection across various areas. Therefore, PWA (Progressive Web Apps) is a new technology designed and developed by Google to overcome the limitation of mobile browsing and native applications. PWA is launched by clicking on an icon on the home screen of the device just like how one goes with native apps. PWA's are instantly loaded on your screen regardless of the network connectivity is available. They support the splash screen through push notifications. In the background, the service worker (set of APIs) allows developer to programmatically cache and preloaded assets and manages the data through a concept called push notifications. Service Worker is a module that runs its own thread and it is responsible to provide generalized entry points by which PWA can process the background task. PWA are linkable with an URL which is fully responsive and secure. Progressive Web Apps start out as tabs in Chrome and become progressively more "app" like; the more people use them, to the point where they can be pinned on the home screen of a phone or in the app drawer and have access to app-like properties such as notifications and offline use.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 9, September 2018

## II. RELATED WORK

In 2014, the number of global users accessing the web on mobile devices surpassed those accessing it on a desktop. This shows that making your web applications mobile-friendly is more important now than ever. Companies often see the need to develop nativeapplications or hybrid applications to overcome the limitations that the web as a platform imposes on mobile devices. In many cases, they must develop their application for both the web, iOSand Android.

A native application is generally coded in a device specific programming language and integrated development environment (IDE).While native Android application is usually coded in Java with the IDE Android Studio. These applications are generally installed through app stores on mobile phones and have rich access to device hardware through platform specific APIs.

These applications are installable from the respective operating systems application store, and run inside a native environment, with all features available to a native application. Taken out of this native environment, these applications fail to deliver this experience due to browser constraints.Progressive Web Applications (PWA) could solve this problem.

A hybrid application denotes an application built with web-based technologies but that can with the help of hybrid development frameworks (e.g., Apaches Cordova), appear and act as a native application.

Therefore, when developing an application targeting mobiles, three common alternatives have traditionally been to build i.e. native, hybrid or mobile web applications; However, progressive web apps (PWAs), introduced by Google in 2015, can be considered a forth alternative to these.

A PWA is a web application that aims to deliver a native-like user experience on a mobile device, such as offline support and push notifications.

## III.CHARACTERISTICS

- Progressive – Regardless of the browser that you are using or even where you are situated in the world,Progressive Web Apps work for every user. So, if a user uses Chrome or Opera or whether the user is living in India, UK or even North Korea, it doesn't matter! Progressive Web Apps will work just as well because they're built with progressive enhancement as a core tenet.
- Responsive – Progressive Web Apps can fit any type of device. It can fit in any device like desktop, mobile, tablet, or devices yet to emerge.
- Application feel— Due to the app-style interactions and navigation inProgressive Web Apps, they give a feel of an application to users.
- Independent of connectivity—Service workers help Progressive Web Apps in such a way that they work offline, or even on networks with low quality.
- Installation—Users can keep the widely used PWAs on their home screen of their device without the interference of app stores.
- Discoverable - Progressive Web Apps are recognizable as applications. The service worker and W3C manifests registration scope allows search engines to find them easily.
- Engageable—Feature like push notifications makes Progressive Web Apps more engaging. These apps are installable and live on the user's home screen, without the need for an app store. User can specify icons for the home screen and splash screen when the app is loading. Page to be loaded when the app is launched and screen orientation.
- Safe—Progressive Web Apps are served via HTTPS, which ensures that without authentication it cannot be tampered by anyone.
- Fresh- Progressive Web Apps are always up-to-date all thanks to the service worker update process.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 9, September 2018

## IV. NATIVE APP Vs PWA Vs STANDARD WEB APP

According to the study, mobile web reach is way higher than native app reach. It was 11.4 million unique visitors per month compared to 4 million visitors.

Whereas the stats of user engagement with services showed that users tend to spend more time on native mobile apps compared to the standard web app. It was an average of 188.6 minutes on app against 9.3 minutes on the web.
So, the idea was clear. They wanted to provide a native app like engaging experience to users on the mobile web. In this way, Progressive Web Apps were developed to deliver amazing user experience on the web.

Below is the comparison between Native App, PWA and Standard web app on various important parameters:

| | **Native App** | **PWA** | **Standard web App** |
|---|---|---|---|
| **Installation** | Need to go to the App store or Play Store, click download | Just click a button to add them to their phone home screen (only on Android) | Installation not required |
| **Updates** | Need to be submitted to the store, then downloaded by the user | Updates are instant | Updates are instant |
| **Size** | Mostly heavy in size. They can take time for downloading on a users' device | Small and fast | Small and fast |
| **Offline access** | Available | Need to use the app once online, then should be able to access the cached content offline | Not required |
| **User experience** | Excellent when the application is well designed | Confusing because of the double menus (app menu and browser menu) | Same as progressive web app |
| **Push notification** | Yes | Yes (Android Only) | Yes (Only possible with third party services) |
| **Discoverability** | Not good–need to work hard on app store optimization | Good – to make appear in search results, need to be optimized for SEO | Not required |

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 9, September 2018

**V.CORE TENETS OF PWA**

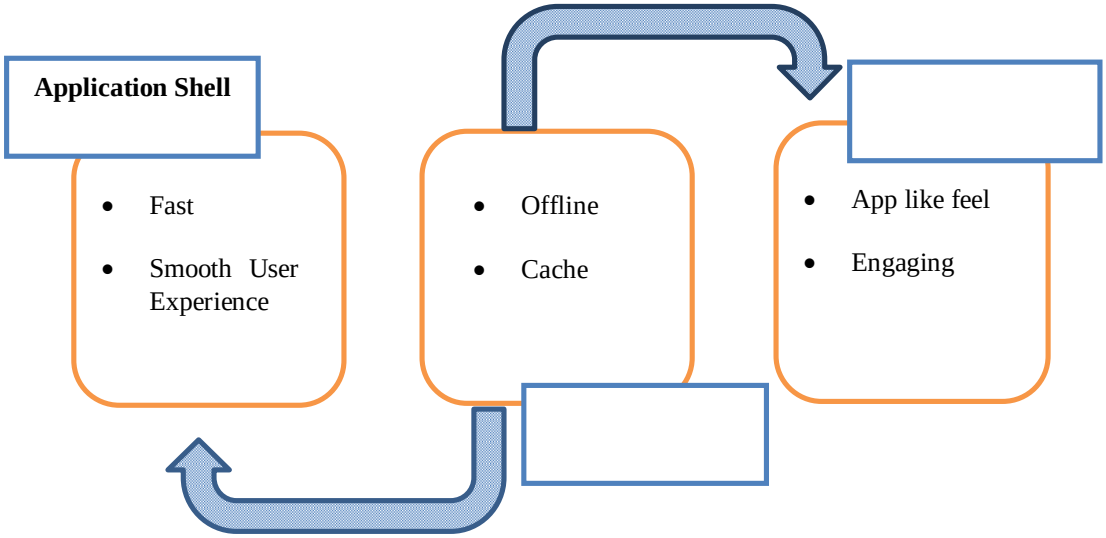


FIG 1: CORE TENETS OF PWA

**SERVICE WORKERS:**

Service Workers, an incredibly powerful tool behind Progressive Web App. The features provided by service workers are:

- Offline Access.
- Push Notifications
- Background content updating
- Content caching

Service workers perform the following functionalities:

1. Caches the App Shell.
2. Updates the Content in background.
3. Gets the push notification id from the user to send the notification.
4. Invalidates the cache when needed

**APP SHELL:**

Application Shell Architecture is served up by the Service Worker and then the content is delivered. These are often cached by the service worker from its source through API requests. The sites that people visit more often will be able to hold the last content the person visited while waiting for the network to dynamically load the latest refresh. With

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 9, September 2018

the App Shell model, the focus is on keeping the shell of app UI and the content inside of it separate, and they arecached separately. Ideally, App Shell is cached such that it loads as quickly as possible when a user visits and returns
later. Having the shell and the content load separately theoretically improves the user's perception of the performance and usability of the app.

**WEB APP MANIFEST:**

The web app manifest has the role to provide information about an application (such as its name, author, icon, and description) in a JSON text file.The manifest must inform details for websites installed on the home screen of a device, providing users with quicker access and a richer experience. The collection of web technologies called progressive web apps includes Web app manifests as its part through which websites that can be installed to a device's home screen without an app store, along with other capabilities like working offline and receiving push notifications.

## V. RESULTS

Google Lighthouse can be run as a Chrome Extension, from the command line, or used programmatically as a Node module. Lighthouse focuses on performance metrics and the quality of PWAs. The standard Lighthouse report provides information on a wide array of factors, in an easily digestible format. Some notable inclusions for website performance are:

1. HTTP/2 – Provides a list of every internal resource that was not served over HTTP/2.
2. First Meaningful Paint – The time required to begin rendering content within the browser.
3. Time to Interactive – A measure of the point at which a page has loaded enough for a user to interact with it.

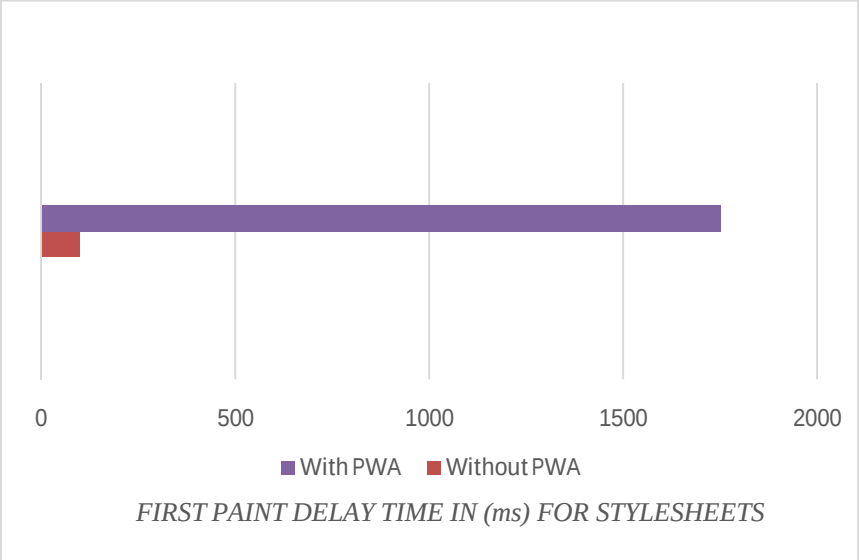


FIG 2: FIRST PAINT DELAY TIME

First page is the first point where the browser has all the information that it needs to render the page. This is with reference to the first time the browser will paint an image of the rendered page on the screen.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 9, September 2018

Style sheets are a type of template file consisting of font and layout settings to give a standardized look to documents. The time taken for the first paint from the style sheet of PWA enabled webpage is smaller (85 ms) when compared to webpage without PWA (1769 ms).

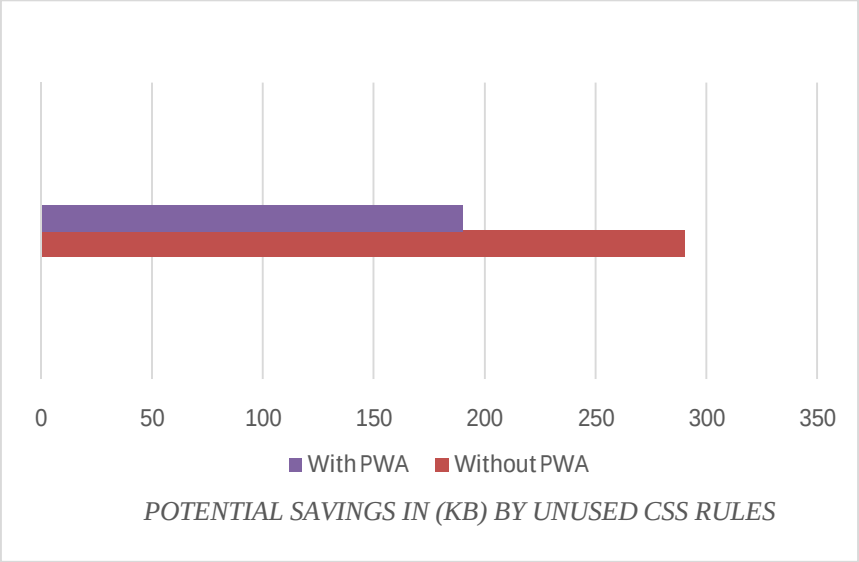


FIG 3:POTENTIAL SAVINGS BY UNUSED CSS RULES

Before a browser can begins to render a web page, it must download and parse any style sheets that are required to lay out the page. Mostly many web sites reuse the same external CSS file for all their pages, even if many of the rules defined in it don't apply to the current page. The best way is to minimize the latency caused by style sheet loading and rendering time is to cut down on the CSS footprint. The potential savings of PWA enabled webpage is smaller (187 KB) when compared to webpage without PWA (283 KB).

## REFERENCES

[1]. Parbat Thakur, "Evaluation and Implementation of Progressive Web Application", bachelor's thesis, Helsinki Metropolia University of Applied Sciences, 2018.
[2]. Mikael Wahlström, "Exploring Progressive Web Applications for Health Care"master'sthesis, Umeå University, 2017.
[3]. Thomas Steiner, "An Analysis of Progressive Web App Features When the Means of Web Access is not a Web Browser",2018.
[4]. Pavel Břoušek," Evaluation and usage of Google Progressive Web Apps technology", bachelor'sthesis, Masaryk University Faculty of Informatics, 2017.
[5]. Bob Frankston," Progressive Web Apps", IEEE Consumer Electronics Magazine, vol.7, no 2. pp.106, 2018.
[6]. Rebecca Fransson and Alexandre Driaguine," Comparing Progressive Web Applications with Native Android Applications", bachelor's thesis, Linnaeus University,2017.
[7]. Ivano Malavolta, Giuseppe Procaccianti, Paul Noorland andPeter Vukmirovic," Assessing the Impact of Service Workers on the Energy Efficiency of Progressive Web Apps", IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILE Soft), 2017.
[8]. Andreas Biørn-Hansen,Tim A. Majchrzak and Tor-Morten Grønli, "Progressive Web Apps: The Possible Web-native Unifier for Mobile Development", SCITEPRESS, pp.345,2017.
[9]. Farrugia Kevin, "A Beginner's Guide to Progressive Web Apps," Smash Magazine,2016.
[10]. Teplý Jan Bc., "Hybrid mobile application for project", Czech Technical University, Prague, 2016.
[11]. P. Kinlan, "Installable web Apps with the web App manifest in chrome for Android", Google Developers, 2017.
[12]. J. Parsons, "What are progressive web Apps? ", The Official Ionic Blog,2016
[13]. David Fortunato and Jorge Bernardino,"Progressive web apps: An alternative to the native mobile Apps", IEEE/13th Iberian Conference on Information Systems and Technologies (CISTI), 2018.