

Progressive Web Apps: An optimistic approach to traditional application development

Jasmine Muman

Student

Department of Computer Science and Engineering,

Guru Tegh Bahadur Institute of Technology affiliated to Guru Gobind Singh Indraprastha University, New Delhi, India

Abstract: Technical innovations have made a greater impact on how products and services are designed and developed. After the launch of Android OS, which is free of cost, there has been a huge increment in the use of smartphones and the development of mobile apps and mobile-friendly web apps. Nowadays, for browsing and scrolling contents of a particular field, most of the users use native mobile apps and some use web browsers. But both of these mediums have their own limitations. For native mobile apps, the user first has to download the apps, then install them and then use them as per their respective requirements giving rise to two major limitations. Firstly, these mobile apps use variable space according to their usage and development in local mobile storages thereby limiting the free space for users' other resources. And, the other main issue is network connectivity, the decrement of which leads to slower processing and accessing of the native mobile app. And, browsing through web browsers is disadvantageous over native mobile apps in terms of users' experience. This paper discusses in detail about overcoming the mentioned limitations using the google provided solution i.e., PWA (Progressive Web Apps). Using PWA to decrease the loading time of the app will also be discussed.

Index Terms: service worker, offline, progressive, user-experience, platform-independent.

I. INTRODUCTION

Mobile and web-based application development can take several paths. With time, the bottom necessities of developing web and mobile apps can be said to have become both more and less complex at the same time. The development methods have also been matured with time. Mobile apps can be developed by building native apps that are platform-dependent and run on specific operating systems such as Android and iOS. Native applications are native to the device's Operating System and therefore the application needs to be coded in different programming languages, as native apps are coded in different programming languages for different operating systems such as Java and Swift. Java programming language was used to code native Android apps whereas Swift was used to code Native iOS apps. These native apps altogether provide a better user experience as they are built for a particular device. However, the cost involved in developing a particular app for a particular operating system could be a major disadvantage.

Another approach to developing mobile applications could be building mobile-based web apps. The technology stack for web apps remains the same as that of a traditional website. Web apps, however, are independent of platform or OS as experienced in the case of Native Apps. Web apps are served via URL running on remote servers. They are also comparatively faster but they fail to provide an app-like experience. Also, the native mobile applications can send push notifications, work offline, load on the home screen, and perform more such activities which may make the app experience more user-friendly. Mobile Web Apps accessed in a mobile browser, by comparison, historically haven't done those things yet. And also, device fragmentation remains another problem.

To beat the mentioned problems, the term "progressive web apps" came into existence by a Designer, Frances Berriman, and a Google Chrome engineer, Alex Russell in 2015 to describe new methods with which the apps could update themselves with new features which were supported by latest browsers, including service workers and web app manifests, that let users upgrade their web applications to progressive web applications in their native operating system. Progressive Web App emerged as an optimal solution to provide the users with a native app-like experience and offer the advantage of web applications. Progressive Web Apps bring features we expected from native mobile apps to mobile browsers such that they use standards-based technologies and run in a secure container that is made accessible to anyone on the web. PWAs have become a Nobel way to develop apps as they promise to combine ease of development and the portability of web technologies with the versatility of native apps. This is achieved without any profound performance penalty but with a gradual decrease in app size.

PWAs are instantly and easily loaded on the laptop's or mobile phone's screen regardless of the good or poor network connectivity. In the background, the service worker i.e., the set of APIs allows developers to programmatically create cache and preloaded assets and manage the data through a concept called push notifications. PWAs are linkable with URLs that are fully responsive and secure. Progressive Web Apps do start as tabs in browsers and become progressively more "app" like by having access to ultimate app-like properties such as notifications and offline use.

This paper makes several contributions. Firstly, we thoroughly assessed competing concepts for developing multi-platform apps. Secondly, this paper studies and proves PWAs as a possible approach or solution to overcome the shortcomings of existing cross-platform application development approaches.

II. RELATED WORK

The technology perspective is changing with every passing year. Over 58.8 percent of the world's population is getting access to the internet and how they are accessing the web all over the world is changing, too. However, this is not the only way the digital world is changing. Thanks to the new devices, desktop computers, laptops, tabs which are not the only primary way people access the internet, instead, they are shifting to mobile phones and the data seems to show that the trend will only increase as more and more people around the globe primarily use mobile devices to get online. Mobile users' behavior and demands change constantly, and there is always a need for optimizing and improving their experiences.

It has been seen that there is a rapid increase among people, globally, using the web on mobile phones is transcending from those using it on desktop. Companies are constantly seen in the demands of native or hybrid applications to overpower the limitations imposed on mobile devices by the web platform. A native application is coded in a platform-specific programming language on a device-specific IDE. These applications are made available to the people on mobile phones via OS-based app stores and these applications may also have vast access to the device hardware through platform-specific APIs.

The native applications can be installed from the respective operating systems-based application stores such as Google play stores and iOS app stores, and run inside a native environment, having all the functionalities and features completely available to the native app. When these apps are taken out of this native environment, they fail to project such an experience due to browser limitations. The best alternative we could think of is using mobile-based web applications by making them more mobile-friendly than ever and the one way we can do that is by using Progressive Web Apps, as they are always in trend and are known for improving the quality of user's experience.

III. CHARACTERISTICS

1. Progressive- PWAs are independent of platforms and operating systems they are accessed on. They do not even depend upon the locations from where they are accessed. You can run these apps on any of the browsers, be it Google Chrome or be it Mozilla Firefox, these apps can be used by anyone on the world wide web because they are built with progressive enhancement as a core feature.
2. Responsive- PWAs are responsive and can align or fit any type of device such as mobile phones, laptops, desktops, etc.
3. Installation- Users can access most frequently used applications with greater ease and also set them up on their home screens or in favorites without any interference from the mobile app stores.
4. User Experience- Progressive Web Apps give an app like user-friendly experience to their users by providing app-style interactions and navigations.
5. Discoverable- The service workers and w3c manifests help to make PWAs highly discoverable by allowing the search engines to find them easily.
6. Independence of Network Connectivity- Progressive Web Apps allow the users to access them in either case of low or poor network connectivity or when they are offline. This feature is possible because of service workers.
7. Up-to-Date- Because of service workers PWAs are always up to date with modern trends and updates.
8. Secure- Progressive Web Apps provide more security to their users as they are served via HTTPS thereby, disallowing anyone to tamper with them.

IV. PREPARE YOUR PAPER BEFORE STYLING

According to the statistics studied, the reach and impact of mobile based web apps is highly greater than that of native mobile apps. Mobile based web apps experience around 11.4 million unique visitors per month whereas, native mobile apps experience about 4 million unique visitors per month.

Below is the comparison between Native Apps, Standard web apps and PWAs based on some of the considerable parameters as mentioned: -

	Native Apps	Standard Web Apps	PWAs
Installation	To be downloaded by hitting the download button in platform or OS based app stores such as Google Play and Apple App Store.	Installation not required.	Can be added to mobile phone's home screens by just a single click.
Update	Updates need to be added by the developers and updated applications need to be submitted to the app stores from where the users can download them.	Instantly updated.	Instantly updated.
Size	Heavy in size and can take time to be downloaded on users' devices based on their respective network connectivity.	Small and fast.	Small and fast.
User Experience	Excellent only when applications are well designed and developed.	Confusing because of the double menus namely app menus and browser menus.	Confusing because of the double menus namely app menus and browser menus.
Push Notifications	Yes.	Yes (But only possible with third party services).	Yes (Android Only).
Discoverability	Not good.	Not required.	Good.

V. TECHNOLOGICAL PROSPECTS AND LIMITATIONS

In Google IO 2017, a technical baseline was presented for testing Javascript frameworks against PWA. HNPWA was the baseline which was launched to help developers a javascript framework for developing PWAs.

When compared to native or cross-platform application development, PWAs projected a major limitation that is the constrained access to device and platform APIs with respect to the APIs that are supported by users' browsers. While in case of native or cross-platform development, all the open device APIs were made available to the developers through the native SDKs. But, the Google Chrome team has been launching several new Apis to bridge the gap between PWAs and native mobile apps.

With time, the support of multiple programming languages might introduce more new developers to the web apps and PWAs based platforms thereby leading to further adoption proliferation of those apps.

VI. CONCLUSION

In this paper, modern methods of mobile and web computing have been presented. The possibilities of PWAs being the unified technology for application development have been discussed. This paper concludes that PWAs can fulfill many requirements for unified multi-platform development and also can contribute to a richer and vast development experience and eventually lead to the development of better apps. This paper suggests a balanced approach of qualitative and experimental work.

The browsers download and parse any stylesheets which are required to layout a webpage before rendering that web page. Most of the websites reuse the same external CSS file for all their web pages even if many of the functions defined in the file aren't used on the current page. The potential savings of latency caused by rendering and loading time style sheets of PWA enabled webpages is smaller (187KB) than that of the webpages without PWA (283KB).

REFERENCES

- [1]. Pavel Břoušek, "Evaluation and usage of Google Progressive Web Apps technology", bachelor's thesis, Masaryk University Faculty of Informatics, 2017.
- [2]. Bob Frankston, "Progressive Web Apps", IEEE Consumer Electronics Magazine, vol.7, no 2. pp.106, 2018.
- [3]. Rebecca Fransson and Alexandre Driaguine, "Comparing Progressive Web Applications with Native Android Applications", bachelor's thesis, Linnaeus University, 2017.
- [4]. Ivano Malavolta, Giuseppe Procaccianti, Paul Noorland and Peter 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.
- [5]. M. Macedonia, "iPhones Target the Tech Elite," Computer, vol. 40, pp. 94-95, 2007.

- [6]. M. Usman, M. Z. Iqbal, and M. U. Khan, "A product-line model-driven engineering approach for generating feature based mobile applications," *JSS*, vol. 123, pp. 1 – 32, 2017.
- [7]. T. A. Majchrzak, J. C. Dageforde, J. Ernsting, C. Rieger, and " T. Reischmann, "How Cross-Platform Technology Can Facilitate Easier Creation of Business Apps," in *Apps Management and E-Commerce Transactions in Real-Time*, S. Rezaei, Ed. IGI Global, 2016.
- [8]. M. Ciman, O. Gaggi, and N. Gonzo, "Cross-platform mobile development: A study on apps with animations," in *Proc. 29th Annual ACM SAC*. ACM, 2014, pp. 757–759.
- [9]. Parbat Thakur, "Evaluation and Implementation of Progressive Web Application", bachelor's thesis, Helsinki Metropolia University of Applied Sciences, 2018.

