A Study on Progressive Web Apps As A Unifier for Native Apps and the Web

Dhananjaysinh Jhala

4th year, Computer Engineering Department

Sardar Vallabhbhai Patel Institute of Technology (SVIT), Vasad

Vadodara, Gujarat, India, 390021

Abstract — **As of January 2021, 59.5 percent of the global population are active internet users worldwide out of which 92.6 percent accessed the internet via mobile devices. The Apple App Store has 1.96 million apps and the Google Play Store has 2.87 million apps available for download. However, there is a restraint to the number of mobile apps one can have in their phone. Progressive Web Apps do not need separate bundling or distribution. Progressive Web Apps are web apps built using common web technologies to bring a native app-like user experience to cross-platform web applications. The qualities of a PWA combine the best of the web and native apps. This study aims at looking at the features, working and future scope of progressive web apps as a unifier between native apps and the web for mobile development.**

Keywords — Web app, progressive web apps, service workers, cross-platform

I. WHAT ARE PROGRESSIVE WEB APPS?

Google chrome engineer Alex Russell published the landmark blog post defining progressive web application. It is intended to work on any platform that uses a standards-compliant browser, including both desktop and mobile devices. Developers can publish the web application online. Users can add the application to their home screen. Publishing the app to digital distribution systems like Apple App Store or Google Play is optional. The motive is to help developers build cross-platform apps with more ease than they would with native apps. Progressive Web Apps are web applications that have been designed so they are capable, reliable, and installable.

II. ESSENTIAL REQUISITES FOR MAKING PROGRESSIVE WEB APPS

A. *Application shell:* An application shell (or app shell) architecture is one way to build a Progressive Web App that reliably and instantly loads on your users' screens, similar to what you see in native applications. It is the minimal HTML, CSS and JavaScript required to power the user interface and on caching offline, it ensures quick, reliable and good performance to users visiting frequently. The app shell is similar to the bundle of code that you'd publish to an app store when building a native app. It is the skeleton of your UI and the core components necessary to get your app off the ground, but likely does not contain the data. The application shell is not loaded from the network every time the user visits. Only the essential content is needed from the network. For applications with a single-page using JavaScript-heavy architectures, an application shell is a proper approach.

B. *Service worker:* An application shell (or app shell) architecture is one way to build a Progressive Web App that reliably and instantly loads on your users' screens, similar to what you see in native applications. It is the minimal HTML, CSS and JavaScript required to power the user interface and on caching offline, it ensures quick, reliable and good performance to users visiting frequently. The application shell is not loaded from the network every time the user visits. Only the essential content is needed from the network. For applications with a single-page using JavaScript-heavy architectures, an application shell is a proper approach.

C. *Manifest:* The web app manifest is a JSON file via which the browser knows about your Progressive Web App and how it should behave when installed on the user's desktop or mobile device. Registering a PWA's web manifest requires a special link element in the HTML head element. A typical manifest file includes the app name, the icons the app should use, and the URL that should be opened when the app is launched. This metadata is crucial for an app to be added to a home screen or otherwise listed alongside native apps.

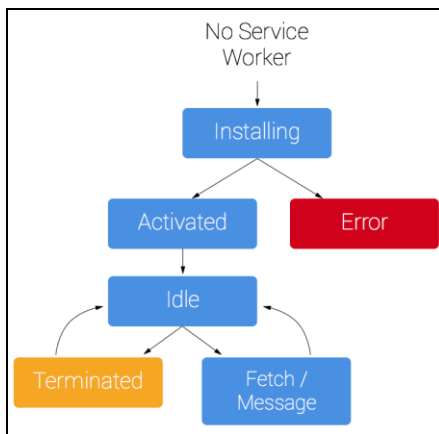Fig. 2.1 example of app shell for 2048 game project

Fig. 2.2 Service worker life cycle

Fig. 2.3 Example of manifest file code

III. CHARACTERISTICS

- Progressive — Works for all users no matter what browser they are using. For example. Google Chrome support PWAs for Windows, Linux, macOS and Android. Other browsers have PWA support in varying degrees as well.

- 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.

- Unconstrained from Connectivity — Service workers allow offline uses, or on low quality networks.

- Fast — After the initial loading has finished, the same content and page elements do not have to be re-downloaded each time.

- App-like — Feels like an app to the user with app-style interactions and navigation.

- Fresh — Always up-to-date due to the service worker update process.

- Safe — Served via HTTPS to prevent snooping and ensure content hasn't been tampered with.

- Engageable — Feature like push notifications makes Progressive Web Apps more engaging. Users will be able to add the application to their home screen.

- Discoverable — Identifiable as an "application" by manifest.json and service worker registration, and discoverable by search engines.

IV. FEATURE COMPARISON

There are many different ways to deliver a mobile experience to the users. Building mobile apps can be done using various methods such as:

- Native apps are applications that were made with one specific technology dedicated to a concrete platform (Android or iOS).

- Hybrid mobile app development is characterized by combining features of a native app and a web app.

- Cross-platform mobile applications are using native elements to give the user great experience despite the device they use.

- In Progressive web apps, the main attribute is running in a web browser, even though it gives the user the native feeling.

TABLE I. FEATURE COMPARISON

Feature	Interpreted	PWA	Hybrid	Native	Standard web app
Installable	Yes	Can add to home screen	Yes	Yes	Not needed
Offline access	Yes	Need to use the app once online, then you can access offline	Yes	Yes	Not needed
Testable before installation	No	Yes	No	No	Yes
Push notification	Yes	Yes (varying as per browsers)	Yes	Yes	Yes (Only possible with third party services)
Discoverability	Comparatively limited	Good – to make appear in search results, need to be optimized for SEO	Comparatively limited	Comparatively limited	Not needed

V. ADVANTAGES OF PROGRESSIVE WEB APPS AND REAL-LIFE EXAMPLES

- Not only is the web at par with native applications, websites have always been easily discoverable. Reach is and has always been the webs greatest superpower. Progressive web apps aren't packaged and deployed through stores, they're just websites.

- In 2017, Twitter released Twitter Lite, a PWA alternative to the official native Android and iOS apps. According to Twitter, Twitter Lite consumed only 1-3% of the size of the native apps. In July 2019 Twitter started serving all website users to Twitter Lite by default. On June 1, 2020, Twitter deactivated the legacy website layout, leaving the progressive web app version as the only option.

- Midway into 2021, in the latest effort to improve the web apps experience, Google appears to be working on a new API that will allow Chrome progressive web apps (PWAs) to 'handle' (read) files in the operating system's file system. Google is working on a new permission prompt that will ask users to confirm whether a PWA can register itself as a file handler for certain file extensions.

- In 2021, Microsoft is working on a way to let people open progressive web apps (PWAs) with the "Open With" dialog box on Windows 10. The feature makes PWAs feel more native since you no longer have to jump around between different ways of accessing the same service. PWAs will appear alongside other apps on Windows 10 once the feature rolls out.

- Activities by Apple in 2021 provide signals businesses should consider using Progressive Web Applications instead of native apps for iOS 13. These actions include purging existing apps from the store and blocking access to other, corporate applications. At the same time, they have begun adding support for service workers and the web manifest specification to iOS Safari.

- Tinder Online started with the goal of getting adoption in new markets, striving to hit feature parity with V1 of Tinder's experience on other platforms. The MVP for the PWA took 3 months

to implement using React as their UI library and Redux for state management. The result of their efforts is a PWA that delivers the core Tinder experience in 10% of the data-investment costs for someone in a data-costly or data-scarce market. Tinder are still iterating on their Progressive Web App but have already started to see positive results.

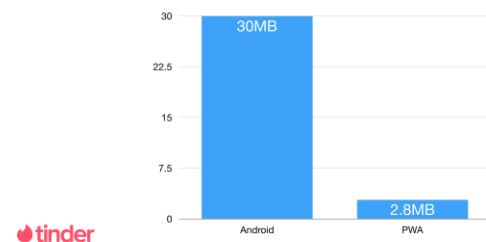Fig. 5.1 Capabilities v/s reach comparison

Fig. 5.2 Size comparison by Tinder

VI. CONCLUSION, FUTURE SCOPE AND SUGGESTIONS

Progressive Web Apps (PWAs) take advantage of major advances in modern web browsers, web APIs, and front-end frameworks to deliver great app experiences to mobile and desktop users. PWA can contribute to a richer development experience, and – eventually – better apps. We see that there is much flair for PWAs to become a unifier for web-native development. This is an area where thoughtful application design and construction will give early movers a major advantage. This study attempts to give an insight into the future benefits PWA could give based on feature comparison and analysis. Few suggestions for further research or how we could deeper into PWA's and gain more from them are provided below:

- Mozilla released Firefox 85 in 2021, adding protections against so-called super cookies but it's also taking a major step back from Progressive Web Apps (PWAs) owing to privacy concerns. While PWAs allow businesses to offer a better scale of performance regardless of connectivity, privacy and security concerns should be addressed as well and future improvements should be made in this matter.

- Providing Progressive Web Apps for IoT (Internet of Things) devices could be a huge step forward for real-world applications. Another question that may arise is whether they make sense for Internet-of-Things (IoT) devices, which might come without a graphical user interface. Future research could explore this direction. It would be important to learn about the gaps that PWAs pose regarding IoT.

REFERENCES

[1] Tandel, Sayali & Jamadar, Abhishek. (2018). Impact of Progressive Web Apps on Web App Development. 10.15680/IJIRSET.2018.0709021.

[2] Progressive Web Apps a New Way to Experience Mobile, Bharath Kumar NJ, International Journal of Engineering Research & Technology (IJERT) ,ISSN: 2278-0181

[3] Biørn-Hansen, A., Majchrzak, T. and Grønli, T-M. Progressive Web Apps: The Possible Web-native Unifier for Mobile Development. DOI: 10.5220/0006353703440351 ISBN: 978-989-758-246-2 Copyright © 2017 by SCITEPRESS – Science and Technology Publications , Lda.

[4] Progressive Web Apps: the Definite Approach to Cross-Platform Development? Majchrzak, Tim A, Biørn-Hansen, Andreas, Grønli, Tor-Morten ISBN: 978-0-9981331-1-9, DOI: 10.24251/HICSS.2018.718M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[5] Progressive Web App for Educational System ,Amit Mhaske, Aditya Bhattad, Priyanka Khamkar, Radhika More ,International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 ,p-ISSN: 2395-0072

[6] Dawning of Progressive Web Applications (PWA): Edging Out the Pitfalls of Traditional Mobile Development, Oluwatofunmi Adetunjia, Chigozirim Ajaegbub, Nzechukwu Otunemec, Olawale J. Omotoshod, American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS) ,ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

[7] Progressive Web App (PWA) for Organization System, Aishwarya Bhilare, Yogita Gaikwad, Varsha Kokare, Satish Kumbhar, Jalindar Ekatpure, International Journal for Research in Applied Science & Engineering Technology (IJRASET) ,ISSN: 2321-9653

[8] PERFORMANCE ENHANCEMENT OF WEBPAGE USING PROGRESSIVE WEB APP FEATURES, Dr. V. Karpagam, Padmavathe. R, ,Lakshana. R, Priyadharshini.S, Number: IJIRAE/RS/Vol.04/Issue03/MRAE10102

[9] Progressive Web Application Development by Example: Develop fast, reliable, and engaging user experiences for the web, Book by Chris Love, Packt Publishing

[10] https://www.statista.com/statistics/617136/digital-population-worldwide/#:~:text=How%20many%20people%20use%20the,the%20internet%20via%20mobile%20devices.

[11] https://buildfire.com/app-statistics/#:~:text=Key%20Mobile%20App%20Statistics&text=The%20Apple%20App%20Store%20has,app%2011%2B%20times%20each%20day.

[12] https://docs.microsoft.com/en-us/microsoft-edge/progressive-web-apps-chromium/

[13] https://medium.com/@slightlylate/progressive-apps-escaping-tabs-without-losing-our-soul-3b93a8561955

[14] https://developers.google.com/web/fundamentals/architecture/app-shell

[15] https://developers.google.com/web/fundamentals/primers/service-workers

[16] https://itcraftapps.com/blog/cross-platform-vs-native-vs-hybrid-vs-pwa-development/

[17] https://www.thurrott.com/cloud/web-browsers/mozilla-firefox/246716/firefox-85-is-here-but-mozilla-is-killing-pwa-features

[18] https://www.bleepingcomputer.com/news/google/google-chrome-is-getting-a-new-progressive-web-app-feature/

[19] https://www.windowscentral.com/pwa-windows-10-open

[20] https://medium.com/@addyosmani/a-tinder-progressive-web-app-performance-case-study-78919d98ece0