

Progressive Web App (PWA) for Organization System

Aishwarya Bhilare¹, Yogita Gaikwad², Varsha Kokare³, Satish Kumbhar⁴, Jalindar Ekatpure⁵

^{1, 2, 3, 4, 5}Department of Computer Engineering, S. B. Patil College of Engineering, Indapur-413106

Abstract: Now a days use our mobile devices a lot and a large amount of that time is spent using apps. However, the remainder of that time is spent browsing and searching the web from the mobile device. PWAs aim to capitalize on this time by marrying the experience of an app with the immediate accessibility of visiting a website. As smart phones get used in large amount, the number of apps are also increased. But the problem with these native web apps is it cannot work properly when network is poor or unreliable. Also these apps require large memory space and they cannot work on offline mode. So for overcoming all these problems, the concept of Progressive Web Apps comes in frame which can work on any platform. The application here is online tier shopping system for user and owner.

Keywords: JavaScript, native web app, network, Progressive web app, Service Workers

I. INTRODUCTION

The natural choice for developing apps is the native software development kit(SDK). It usually is provided by the vendor of the mobile platform. The SDK offers a development experience tailored to the platform. There are normally few programming languages available. In the case of ios, Objective-C and Swift can be leveraged.

On Android, the recent addition of the Javasuperset language Kotlin was welcomed by the otherwise Java-heavy developer community. If developing for the Windows platform, a so-called Universal Windows Platform app, multiple languages can be used, including C++, C#, Visual Basic and JavaScript.

Device features can be accessed through the platform's own application programming interfaces (APIs) .Progressive web apps are an enhancement of existing web technology. PWAs work in any browser, but "app-like" features such as being independent of connectivity, install to home screen & push messaging depend on browser support. This new technology is nothing but Progressive Web Apps (PWA).

Progressive Web Apps are called future of Web Technology and Mobile Apps. Progressive Web Apps bridges the gap between websites and native apps. Progressive Web Apps provides the best user experience than websites and requires less memory space than native apps. Application Shell and Service Workers are the main elements in Progressive Web App. Also Progressive Web Apps are giving the more features like push notifications, splash screen and symbol on home screen. When Progressive Web Apps are launched from home screen, they mixed in environment.

Progressive Web Apps are top-level, full screen and also work on offline mode. In future, Progressive Web Apps are the most forward look mobile apps. Hence, in this paper, we propose a system which gives an innovative approach for organization system along with Progressive Web App.

In background of PWA the service worker is working which is a set of API that allows developer to programmatically cache and preloaded assets and manages the data through a concept called push notifications. Service Worker is a module which runs its own thread. It is responsible to provide generalized entry points by which PWA can process the background task. As smart phones get used in large amount, the number of apps are also increased. But the problem with these native web apps is it cannot work properly when network is poor or unreliable. Also these apps require large memory space and they cannot work on offline mode. That motivates us to develop PWA Application..

II. REVIEW OF LITERATURE

A. Present system work on Progressive web app is a middle approach for native app and web application. It reduces lots of problems of user about poor network connectivity and rich interface just like native app. The app loads quickly, even when the user is having low network connectivity. PWA can send push notifications to the user and has an icon on the home screen and loads as top-level, full screen experience. Progressive web apps are an interesting forward look into the future of mobile apps. It will become an important factor in the world of apps[1].

- B. This provides a state-of-the-art overview of the development strategies and technologies for developing mobile apps, each of them with its own advantages and drawbacks. In this context, the use of web technologies is discussed as a promising investment for moving forward one of the most intriguing challenges in the world of mobile apps: its fragmentation with respect to mobile platforms. A discussion of research challenges, and thus opportunities, closes the talk[3].
- C. Propose a system which gives an innovative approach for organization system along with Progressive WebApp. In background of PWA the service worker is working which is a set of API that allows developer to programmatically cache and preloaded assets and manages the data through a concept called push notifications. Service Worker is a module which runs its own thread. It is responsible to provide generalized entry points by which PWA can process the background task[4].
- D. This paper intends to understand which evolutions, capabilities and limitations exists on developing a web app to run in all devices. Present the new concept of Progressive Web App, created by Google, in a way to normalize all web developments. It will be introduced the major advantages on developing the apps centralized as a Progressive Web App, comparing on developing the same solution for each different mobile platform[5].
- E. This paper, argue for progressive web apps as a possibly unifying technology for web apps and native apps. After an introduction of features, system scrutinize the performance. Two cross-platform mobile apps and one Progressive Web App have been developed for comparison purposes, and provided in an open source repository for results' validity verification[6].
- F. This paper, System present an empirical study that evaluates the impact of service workers on the energy efficiency of PWAs, when operating in different network conditions on two different generations of mobile devices. Method. System designed an empirical experiment with two main factors: the use of service workers and the type of network available[8]
- G. This paper discussed 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 smart phone 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[10].

III. PROPOSED WORK

A. System Overview

A progressive web application is essentially a concept of how to build a mobile web app that is progressively enhanced with modern web technologies. PWAs are initially served from a remote web server similar to mobile web apps, but can when visited through a browser app be installed on devices as well. Another core feature they have is that they can be used regardless of network availability with the help of service workers that, among other things, enables caching and preloading resources. PWAs aim to bridge the gap in user experience between web and native/hybrid applications. Owner will upload his tier related product on App. User will view that product and buy this product. To access the system user have to register to system.

- 1) *Service Worker*: The Service Worker is responsible for most of the core features associated with progressive web apps. A PWA cannot properly work in browsers without Service Worker support. The worker is registered on a user's first page visit. It consists of a JavaScript file embodying lifecycle hooks for business logic and cache control. It can be used to handle tasks such as background synchronization.
- 2) *Application Shell*: The application shell is defined by the Google Web Fundamentals group as "[...] the minimal HTML, CSS, and JavaScript powering a user interface. They list three criteria for the shell: fast loading time, cached, and displaying dynamic content. Data is pulled from external APIs.
- 3) *Web App Manifest*: The purpose of the manifest file is to expose certain modifiable settings to app developers. These settings include such as logo image path, app name, splash screen and more. In short, the manifest can be used to modify behavior and style of PWA applications.
- 4) *HTTPS*: Hypertext Transfer Protocol Secure(HTTPS) denotes a protocol where http pages are sent with an encrypting transport layer security (TLS), or formerly secure socket layer (SSL), protocol. This can improve the security of transferring web pages over a computer network(e.g., the Internet) by preventing man-in-the-middle attacks. Man-in-the-middle attacks can be achieved by someone spoofing their identity to act as an intended receiver between two communication parties.

B. System Architecture

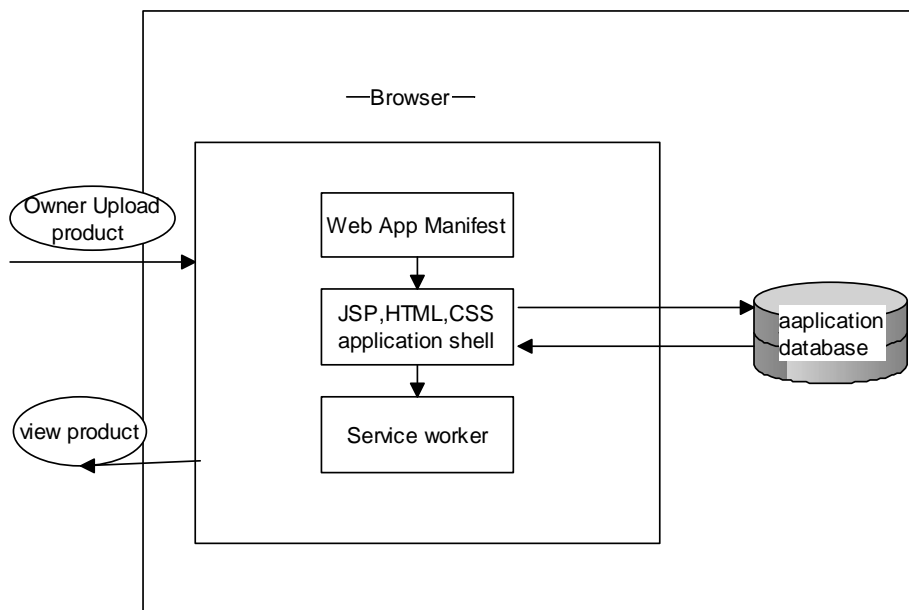


Fig. 1 System Architecture of Progressive Web App for Organization System

C. System Working

1) **Service Worker:** Service workers are the main part of PWA which runs in background separately from the web pages. All the response to events, network request made from server and client is managed by service workers. The lifetime of Service workers are generally kept for a minimum time. It wakes up when it gets an event and runs only as long as it needs to process it. This is standard for workers on the web. The DOM Structure of a web page is not accessible by a Service worker can't access the DOM but can access things like the network request, fetch API and Cache API, The Indexed DB API and postMessage() are also available to use for data persistence and messaging between the service worker and pages it controls. A service worker can intercept network requests made from a page (which triggers a fetch event on the service worker) and return a response retrieved from the network, or from a local cache, or constructed programmatically. Effectively, it's a programmable proxy in the browser. The neat part is that, regardless of where the response comes from, it looks to the web page as though there were no service worker involvement. Service Workers are a way to increase Web app performance by helping to cache and deliver content and background functionality (like push notifications). Service workers can make sites work offline or help speed up the content by, "intercepting network requests to deliver programmatic or cached responses."

Service Worker Lifecycle

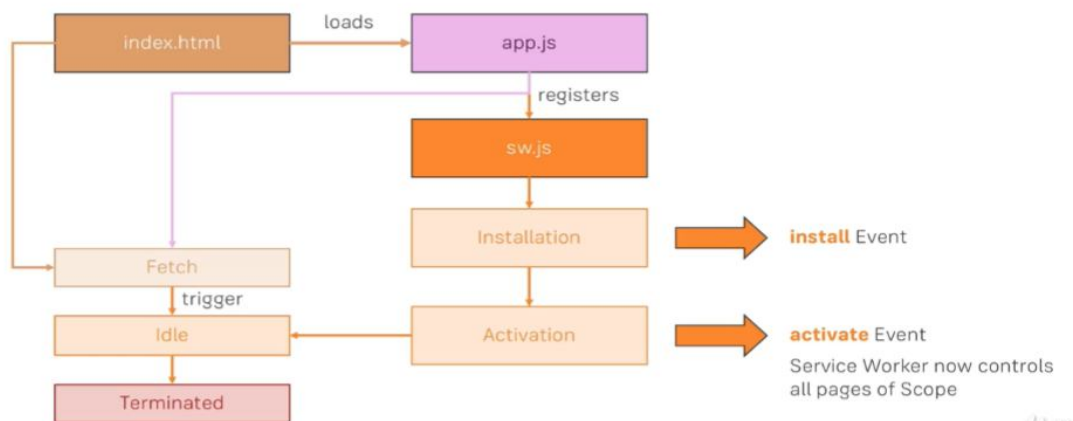


Fig. 2 Service Worker Lifecycle

- 2) **Background Sync:** Background synchronization is a new web API that let you defer actions until the user has stable connectivity. This is useful for ensuring that whatever the user wants to send, is actually sent.

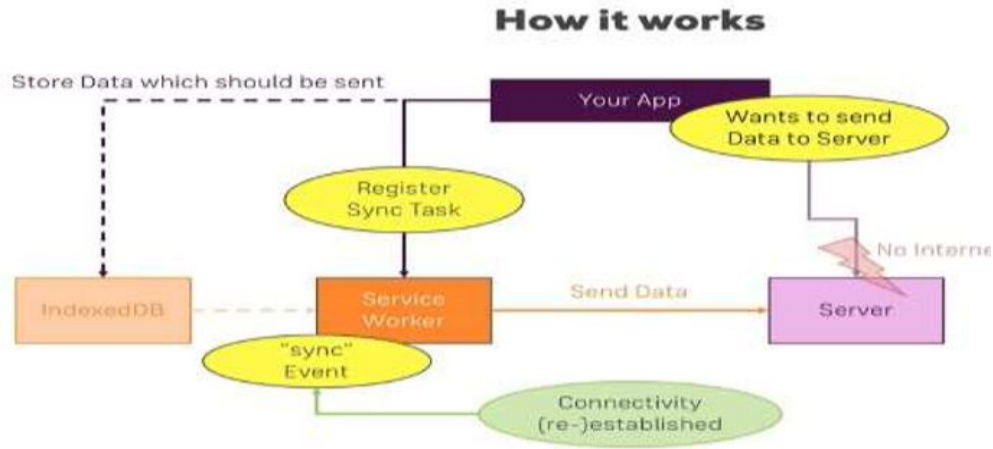


Fig. 3 Working of Background Synchronization

- 3) **Push Notifications:** Push messaging notifications provides a simple and effective way to re-engage with your users and in this code lab you'll learn how to add push notifications to your web app.

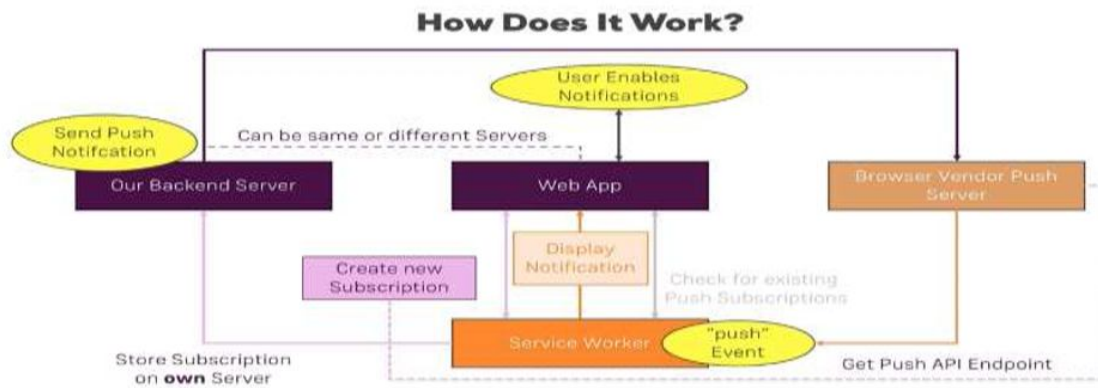


Fig. 4 Working of Push Notifications

IV. CONCLUSION

Proposed system is developed for PWA App with online tier shopping system for user and owner. As an end-user, the PWA installation process becomes more similar to regular apps through new advancements in user experience aspects. Web apps can look, feel and act similar to native, hybrid and interpreted apps. 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.

REFERENCES

- [1] Amit Mhaske, Aditya Bhattad, Priyanka Khamkar, Radhika More Progressive Web App for Educational System International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue: 01 Jan-2018
- [2] William Jobe Native Apps Vs. Mobile Web app 2013
- [3] Ivano Malavolta Beyond Native Apps : Web Technologies to the Rescue! 2016
- [4] Rahul Surendra Mishra Progressive WEBAPP : Review 2016
- [5] David Fortunato, Jorge Bernardino PWA : An Alternative To the Native Mobile Apps 2016
- [6] Andreas Biorn-Hansen, Tim A. Majchrzak and Tor-Morten Gronli Progressive Web Apps: The Possible Web-native Unifier for Mobile Development 2017
- [7] Andriy Luntovskyy Advanced Software-Technological Approaches for Mobile App Development 2017
- [8] Ivano Malavolta, Giuseppe Procaccianti, Paul Noorland Assessing the Impact of Service Workers on the Energy Efficiency Of Progressive Web App 2017
- [9] Harry Alvin Waidan Kefas Building Progressive Web Apps Using Polymer Library 2017
- [10] Abhi Gambhir, Gaurav Raj Analysis of Cache in Service Worker and Performance Scoring of Progressive Web App 2018
- [11] Kashish Behl, Gaurav Raj Architectural Pattern of Progressive Web and Background Synchronization 2011.