**PROGRESSIVE WEB APP (PWA)**

PWA (Progressive Web Apps) is one of the most talked about technology shifts in the web and has gained unparalleled momentum among the practitioners in the IT world. If you are building for the web, I’m sure that PWA is the latest ‘buzzword’ that has been added to your work vocabulary. It’s not surprising because PWA has made the far fetched dream of installing web apps on phone for real.

**WHAT IS A PROGRESSIVE WEB APP ?**

Progressive Web Apps (PWA) combine new technologies with established best practices for creating reliable, accessible, and engaging experiences. They give users a native-like experience with a user friendly opt-in installation flow.

**A Progressive Web App** (PWA) is a web app that uses modern web capabilities to deliver an app-like experience to users. It is the next big thing in web development as they bring **mobile-app-like experiences** to your users without requiring them to install an app from the app store/ play store.

These apps meet certain requirements, are deployed to servers, accessible through URLs, and indexed by search engines.

**The Progressive Web App Features**

1. The essential features of the progressive web apps will be discussed below while outlining its benefits. The features like no app submission, push notifications, faster loading time, and SEO benefits, reduce the cost of development and so on.

2. In addition to these, the PWAs are known for their responsiveness as they can fit into any shape such as desktop, mobile, tablet.

3. They can work with any browser the user chooses due to the fact that they are created with progressive enhancement as a core tenet.

4. In the security point of view, they are highly secured with the HTTPS.

**WHAT IS REQUIRED IN PWA ?**

To be considered a Progressive Web App, the app must be:

* **Progressive** - Work for every user, regardless of browser choice, because they are built with progressive enhancement.
* **Responsive** - Fit any form factor, desktop, mobile, tablet, or whatever is next.
* **Connectivity independent** - Enhanced with service workers to work offline or on low quality networks.
* **App-like** - Use the app-shell model to provide app-style navigation and interactions.
* **Fresh** - Always up-to-date thanks to the service worker update process.
* **Safe** - Served via HTTPS to prevent snooping and ensure content has not been tampered with..
* **Re-engageable** - Make re-engagement easy through features like push notifications.
* **Installable** - Allow users to “keep” apps they find most useful on their home screen without the hassle of an app store.
* **Linkable -** Easily share via URL and not require complex installation.
* **Offline Support -** Apps should be able to work offline. Whether that be displaying a proper “offline” message or caching app data for display purpose.

**WHY BUILD A PROGRESSIVE WEB APP (PWA) ?**

The cost of developing, testing, and maintaining applications for several platforms is unreasonable for those companies who build apps for their internal use. Thus, they believe that PWA will become a more viable alternative to them.

If ever the best of web and the best of apps had a clone child — it is PWA. Or else, it’s just that the web page that can behave more like an app downloaded from the App Store/ Play Store. It starts as a normal web page in a browser, and as a user explores the webpage, they get the prompt if they would like to “Add to Home Screen”. Once the user gives the thumbs up to this prompt, then PWA gets added to their home screen. Once open from the home screen, it can even hide the browser UI controls and appear as an app.

PWA has managed to bridge the gap in the web. The web has always been thirsty for reliable performance at par with the native apps. It has always yearned for a place in the notification tray and in the home screen just like an app. More than 40% of the users bounce from the websites that take more than 3 seconds to load. PWA is a solution for this faced by the users. It is all about removing friction and making it easy for the users to get to what they want.The entire credit for this seamless experience should be given to the **Service Worker**.

**SERVICE WORKER**

A service worker is a script that your browser runs in the background, separate from a web page, opening the door to features that don't need a web page or user interaction. It is the backbone of every PWA. They enable

* content caching
* background content updating
* push notification
* offline function to prior visited sites (that means, after the first visit to a website, the site and app will be reliably fast even on flaky networks).

**Things to note about a service worker:**

* It's a JavaScript Worker, so it can't access the DOM directly. Instead, a service worker can communicate with the pages it controls by responding to messages sent via the postMessage interface, and those pages can manipulate the DOM if needed.
* Service worker is a programmable network proxy, allowing you to control how network requests from your page are handled.
* It's terminated when not in use, and restarted when it's next needed, so you cannot rely on global state within a service worker's onfetch and onmessage handlers. If there is information that you need to persist and reuse across restarts, service workers do have access to the IndexedDB API.
* Service workers make extensive use of promises, so if you're new to promises, then you should stop reading this and check out Promises, an introduction.



Fast first load reliable performance happens when **Accelerated Mobile Apps (**AMP) meets the Service Worker. AMP makes the web fast for users and simple for developers. It provides reliable fast web components for first load. These components are much faster to load and less data hungry. The websites which uses the combo of AMP and service worker will provide a reliable speed has of native apps.Once the page is loaded the site sets-up the service worker and assets are cached intelligently. This will always keep the PWA up-to-date thereby freeing the users from the frequent updates to be done from the App Store.

**THE THREE COMPONENTS OF PWA :**

The progressive web app basically comprised of three elements, which is also the core of its functionality. It includes:

• A service worker

• An app shell

• An app manifest

**A Service Worker**

The Service Worker can rightly be said as the backbone of the progressive web apps. It is the major technology that fuels the progressive apps and is known for its robustness and power. The progressive apps are able to offer offline navigation and content caching, push notifications and updating content facility due to the presence of the service worker. This technology also helps in bringing connectivity changes and operates backstage to run in response to a network.

**App Shell**

The App Shell is another important component of the progressive web apps and is a design concept, which initially allows the loading of the app shell before the content. It caches the design and the content on separate notes and helps in improving the performance of the app.

**App Manifest**

The App Manifest is a wonderful tool that allows the installation of the app from the browser and then pins it into the device’s home screen, similar to what we do on your personal computers and laptops. You can create a shortcut for any web app and install it on the home screen with this support. In order to specify the fact that the web app can be installed as an app, the developers write a manifest.json file and it is connected to the main HTML page.

**THE BENEFITS OF PROGRESSIVE WEB APPS :**

Once we have discussed the definition and the major components of the progressive web apps, let’s now focus on what are the key benefits of PWA.

**PWA supports the Offline Mode as Well**

One of the main advantages of Progressive Web Apps is that it supports the browsing even if there is no or very limited internet connection. This is something which you will not find in the case of websites.

Thus, it ensures more and more user engagement and also allows you to save the information automatically. The offline page contains a logo, some info, and few advanced features. For instance, people can view the product list displayed on the e-commerce app.

**Makes Use of Low Data**

The PWAs also make use of low internet data as compared to the native apps. It has been seen that you can save a lot of monthly data, thus reducing your bill on internet connection.

**Push Notifications**

They are a great asset tool for mobile marketing today helping to bring more target customers on the board. And the progressive web apps are blessed to have this vital weapon where you can send specific messages in bulk related to discount offers, rewards, and coupons. This helps in boosting the sales to higher up the order.

**Downloads are awesomely Fast**

Unlike the native apps, the users don’t have to go to the Play Store to download the progressive web apps. And they also do not take much time to get downloaded. They can be directly downloaded onto the device. It has its own icons on the device.

**No Need for App Submission**

The developers with progressive web apps don’t need to submit into the Google Play Store or App Store. This reduces their workload and also tends to save a lot of time and money.

Plus, they also do not need any approval or consent to update the app and can push for it at regular intervals. The updates are automatically downloaded when the app is relaunched.

**Best Option if Budget is less**

The Progressive Web Apps can be a viable option for any business enterprise or a startup, which is having a tight or low budget estimate. The cost of developing PWA is much lower than the native apps.

As such the cost estimate of developing a native app would be around $25k-$80k, whereas the PWA costs around a fraction of that i.e. $6k to $10k. Moreover, it also offers good overall results for the business with the best ROI.

**Shows Positive SEO Results**

Since the PWA has great loading speed, it tends to boost the accessibility and searching power of the app. It offers a great overall user experience, adding to the success story of your business. It is a positive indication for marketing strategy.

**Some other Benefits**

* The progressive web apps just act like websites providing the same features and functionalities including accessing the database.
* They are known to deliver improved performance.

**The Popular Clientele**

Even though progressive apps are build on the new technologies, they are already widely used in the industries. PWA has some of the most prestigious clientele including Alibaba, Flipkart, Twitter, Ola, Pinterest, OLX, Bookmyshow and much more.



**HOW TO CREATE A PROGRESSIVE WEB APP :**

The creation here consists of five different process. They are,

* Analyse
* Prioritize
* Choose Tools
* Implement
* Measure and Evaluate

**Analyse**

We need to know the quanlitative and qualitative snapshots of the current situation of our website. It is must to understand the standing point before you add new layers of complexity to your code.

If starting a project from start, you will eventually find yourself iterating over it again and again. And in each of the iteration it is should start with a proper analysis.

The tool which tremendously helps in analysising a website is the "LIGHTHOUSE" tool.

It is directly integrated into the DevTool of the recent chrome, sos one can run the audit directly from the chrome.

LIGHTHOUSE

Lighthouse is a chrome extension and a comment line tool that runs over hundreds of auidts on the site to identify how one can improve the sites's performance, accessibility and overall progressiveness.

Lighthouse runs the site or app through a series of tests covering important areas for progressive web apps. It gives an idea of what is working well and what to improve.

Chrome => DevTools => Audits => Start the test

Audit reloads the page several times and captures the traces on each loads that when it becomes interactive, how much time each loads take and what happens when there is no network etc... Audit takes almost 60 seconds for each test. Then, it shows the scores (progressive web apps, performance, accessibility and best practice), report as the results. It also gives us suggestions or proposal to fix and rectify the issues.

There is also a download button which allows us to save the generated report to be used for later. It helps to periodically observe and enjoy the progress also to watch out for the possible regressions throughout the project. When download we can use the lighthouse viewer to display it in a browser tab. From there one can also share and export it to different formats like PDFs.

Though Lighthouse is a fantastic tool, it is not the end of the destination. In addition, one should try the website across other browsers, other devices and netwok conditions.

**Prioritize**

**smart display**

1. PWA ready

The site should be lean, smooth-working and optimized as reasonably possible. Small changes can result into a big improvements like:

* Optimized images
* Remove unnecessary code
* Leverage browser caching
* Avoid blocking code
* Divide big, monolithic javascript

2. Safety and Security

It means HTTPS. Keeping the users safe is hugely important in world of PWA.

3. Attentive

We must need to keep the attention of the user and need to serve the content quickly.

4. Service Worker and Offline

Users may be on 3G or LieFi, so we must focus on offline.

5. Push notifications and Add to Homescreen

This helps to integrate better with user's devices, easier to bring them back and turn them to customers.

6. Payments

Payments API brings values for your business.

**UX**

UX, the core of PWA. Whatever one may choose, wrap it under user experience.

It is important that how one implement the features of their site.

**Choose Tools**

Do you want your apps to be hard-hitting and innovative? To do that, you need the right tools. Here we use a tool called WORKBOX, a javascript library for PWA. Especially if one writes in offline, they experience end up writing a lot of biolerplate.

But with workbox, one can avoid the lot of error-prone codes and it also helps in smooth-up the development process.Workbox gives access to all the power of complex caching strategies via a combination of tools.

* Precaching
* Runtime caching
* Strategies
* Request routing
* Background sync
* Helpful debugging
* Greater flexibility and feature set than sw-precache and sw-toolbox

It is also a set of generators that helps one with asset management. When implementing caching, this can get really easy to handle all the URLs. Workbox helps you trade complexity for flexibility.

**Implement**

Ideas are important, yet most of all they are starting points. Planing things are nothing untill it is implemented. Here, we implement all the ideas we have planned, analysed. But on achieveing the good experience, one must take a look at these points:

* **Don't block transitions on network** - Users should always feel in constrol of what is happeneing. Janky transitions from one group to another can destroy that feeling. So, instead of blocking the transitions of the page on thenetwork, you can use the skeleton screens.
* **Prevent content jumping - Unstable load** - When you load or read an article or a piece of content, it jumps out from your eyes because of some additional content loaded at the top of it. It is especially annoying and really an anti-pattern. So, in order to avoid this issue we must ensure a stable load, specify the size of images beforehand and use dynamic elements.
* **Cache the right thing at the right time** - Saving resources into cache can make the site much better and ready able but sometimes cacheing the entire site is simply not viable or possible. So, one must concentrate on what to cache ? and when to cache ?.
* **Avoid scrolling glitches** - Instead scroll fast with virtualized lists.

**Measure and Evaluate**

Measuring things at the end of the project provides jusstification for the next steps. So, go again through audit (Lighthouse) and compare the outcome with the previous version and see how much the site has improved from that of the previous one.

Geeting 100% result in lighthouse is better but nothing is better than checking the reaction of the users to the website.

In evaluating the changes, one need to pay attention to analutics and to the words of the users both to track improvements and to look out for possible regressions on the way.

**CONCLUSION :**

The Progressive Web Apps is getting a good positive response from the web and mobile app development Industry. Despite its drawbacks, it is not getting support from Apple iOS, it has a number of advantages and features that can take any business to a result-oriented position. They also make the job of the app developers much more easily reducing the overall cost and time of development.