Progresive web apps

Jesus Armando Gomez Leyva

January 27, 2024

1 Concepts

1. Progressive Web Application (PWA) is a type of web app that can operate both as a web page and mobile app on any device. It is a great solution for poor mobile UX and low conversion rates in your online store. Using standard technologies, PWA is aimed at delivering native-like user experience, with speedier conversion and cleaner browsing even with a poor Internet connection. ([StoreFront, 2023]).

2. Progressive Web Applications (PWAs) are apps built with web technologies that we probably all know and love, like HTML, CSS, and JavaScript. But they have the feel and functionality of an actual native app. Wait a minute! Native Apps, what do we mean by this? ([Nyakundi, 2023]).

2 Caracteristics

- **Responsiveness** - Different companies produce gadgets with different screen sizes, and as a developer it's your responsibility to ensure all the different users enjoy the product regardless the device they are using. So it's a good idea to make sure your app can be used on any screen size and it's content is available at any view-port size.

- **Installable** - Research has shown that users tend to engage more with installed apps compared to visiting the official sites. Having a PWA as your product gives the users the look, feel and engagement of a normal app.
- **Independent Connectivity** - By keeping a user engaged to your app even while they are offline, provides a more consistent experience than dropping them back to a default offline page
- **Discoverability** - Since most PWAs are converted websites, it is fair to make them discoverable on the search engines, this will help generate extra traffic to your app. This also acts as an advantage over native apps which can't be discovered over the search engines.
- **Appearance** - The appearance of the app should feel and look like that of a normal app, so be sure to include things like an app icon, this will help make it easily recognizable also things like splash screen will add the touch and feel of an app.
- **Cross Platform** - PWAs are developed as web app first, which means that they need to work on all browsers/systems and not just a selected few. Users should be able to use them in any browser before they decide to install them.

  [Nyakundi, 2023].

3 Differences between SOA and PWA

1. **User Experience**
   - **Service-Oriented Web Page**: Web pages are often more static and require a constant internet connection to load-real time content.
   - **Progresive Web App**: PWA's offer a more interactive and dynamic experience. They can function offline or with weaker network connections.
2. **Updates**

- **Service-Oriented Web Page**: Updates occur on the server, and users get the latest features and improvements by refreshing the page.
- **Progresive Web App**: Updates for a PWA can happen automatically in the background, allowing users to enjoy the latest features without manually installing updates.

3. **Access and Distribution**

   - **Service-Oriented Web Page**: Typically accessed through a web browser. Users can visit the page using any device with a browser and an internet connection.
   - **Progresive Web App**: A web application that can be accessed through a web browser but can also be installed on the user's device for a more app-like experience. This allows users to access the PWA from their home screen, even without an internet connection.

[Gill, 2023].

4 Benefits of PWA

1. **They work offline:** Unlike standard web applications, PWAs are network independent, which allows them to work even when users are offline or have an unreliable network connection. On a very simplistic level, this is accomplished by using Service Workers and APIs to revisit and cache page requests and responses, thereby allowing users to browse content they previously viewed.

2. **They're Search Engine Optimized:** PWAs are intended to be more discoverable and compatible with search engines. To support that goal, PWAs adhere to certain global standards and formats that make it easier to catalog, rank, and surface content in search engines. For instance, to become a PWA, a website must have HTTPS support and include a web app manifest.

3. **They're installable:** Unlike traditional web apps, PWAs have the ability to be installed on a device. This empowers users to access

that app via an app icon and ultimately creates a more seamless and integrated user experience. When users wish to access the application, they can click the shortcut on their Home Screen rather than opening a web browser window and typing out a URL. PWAs don't need to be installed to function. However, providing this option to users enables the app to perform and feel more like a native application, thereby reducing friction from the re-engagement process.

4. **They're linkable:** Unlike native mobile applications, traditional web applications are accessible via a direct URL without requiring setup or installation. That URL structure makes it easy to encourage users to engage with specific content by linking directly to a page you'd like them to view and even anchoring to specific text. For native mobile applications, driving specific user actions within the app can require more UI development and strategic in-app communication.

5. **They're responsive:** Today's users expect a seamless, omnichannel experience across desktop, mobile, and brick-and-mortar touchpoints. Responsive design — the ability of a web application interface to automatically adapt to a device's layout and individual user's behavior — is table stakes for delivering on these heightened expectations. PWAs leverage modern technologies to ensure a web application UI is flexible and responsive.

6. **They provide built-in security benefits:** PWAs are built using HTTPS, which encrypts data shared between the app and the server. This protocol makes it inherently more challenging for hackers to access sensitive data. Additionally, PWAs rely on Service Workers to enable app functionality and require an app manifest that controls how an app may be launched and displayed. Compared with native applications, PWAs have more limited permissions, which typically reduces exposure to security threats. These combined technologies ultimately help prevent attacks and fortify app security.

7. **They're more cost-effective to develop than native mobile apps:** PWAs can be built using web technologies such as HTML, CSS, and JavaScript, which makes them a more cost-effective alternative to developing a native mobile application for each operating system (iOS,

Android, etc.). Compared to web apps, native mobile applications typically take more time and resources to build, have unique requirements for each operating system, and have associated app store maintenance fees.

8. **They outperform standard web apps:** Because PWAs are designed to be lightweight from a data consumption standpoint, they have better load times, impeccable responsiveness, and more seamless animations than traditional web apps. This all equates to a more delightful, scaleable, and flexible user experience across various devices. PWAs are also known for their ability to use progressive enhancement techniques to work across different browsers and devices even when those browsers don't share the same capabilities, making them inherently more compatible than standard web apps and native mobile apps.

[Kremer, 2023]

References

Navdeep Singh Gill. What is service-oriented architecture (soa)?, 2023. URL `https://www.xenonstack.com/insights/service-oriented-architecture`. Accessed: January 5, 2024.

Meryl Kremer. What is a pwa?, 2023. URL `https://onesignal.com/blog/what-is-a-pwa/`. Accessed: January 5, 2024.

Hillary Nyakundi. What is a pwa? progressive web apps for beginners, 2023. URL `https://www.freecodecamp.org/news/what-are-progressive-web-apps/`. Accessed: January 5, 2024.

Vue StoreFront. What is pwa? progressive web apps in ecommerce explained, 2023. URL `https://vuestorefront.io/blog/pwa`. Accessed: January 5, 2024.