IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101259

Wall-et PWA Crypto Wallet

Aditya Sachin Patil[1], Vaibhav Singh Rawat[2], Haresh Raju Kaneshan[3] Yashita Agarwal[4]

Dr. Sonal Sharma[5]

Department of Computer Science and Engineering, FET- Jain University Bangalore, Karnataka,INDIA[1-5]

Abstract - To explore the contents of any industry, users access them using a website or an app. They both have their disadvantages but google has developed a progressive web app that has the advantage of both a website and an app, i.e its a website which operates as an app but it doesn't require to be installed. Wall-et is a PWA that can be used to trade cryptocurrencies effectively. It can be downloaded on a desktop or a smartphone can work offline and is discoverable through search since it's also a website. All the transactions are secure using SSL, google auth is being used to reduce impersonators and it's easy on one's eyes because of its seamless beautiful interface made by using a cluster of HTTPS and laravel.

Key Words: Progressive, Offline, Service Worker, AppShell, App Manifest

1. INTRODUCTION

Designer Frances Berriman and Google Chrome engineer Alex Russell coined the term "progressive web apps" in 2015 to describe apps that take advantage of new features supported by modern browsers, such as service workers and web app manifests, to allow users to upgrade web apps to progressive web apps in their native operating system.Push alerts, offline functionality, and loading on the home screen are all features of native mobile applications. Mobile Web Apps viewed using a mobile browser, on the other hand, haven't done such things in the past. With new Web APIs and design approaches, Progressive Web Apps address this. Progressive Web Apps offer native-app-like functionality to the mobile browser. It employs standards-based technology and operates in a secure container that anybody on the internet may access. Even with fully built mobile web apps, it still fails to give a visually appealing and satisfying experience to consumers, owing to a lack of robust network connections in various places. As a result, PWA (Progressive Web Apps) is a new Google technology designed to overcome the limitations of mobile browsing and native apps.PWAs are opened by clicking on an icon on the device's home screen, similar to how native applications are launched. PWAs appear on your screen right away, regardless of whether or not you have internet access. They provide support for the splash screen by sending push notifications. The service worker (a collection of APIs) works in the background, allowing developers to cache and preload content programmatically and manage data using a notion known as push notifications. The Service Worker module is responsible for providing universal entry points via which PWA may handle background tasks. It operates on its thread. PWAs have a completely responsive and secure URL that can be linked to. Progressive Web Apps begin as Chrome tabs that gradually become more "app-like" as more people use them, to the point where they may be pinned to a phone's home screen or in the app drawer and have access to app-like features like alerts and offline use.

In 2014, the number of people using mobile devices to access the internet overtook those using desktop computers. This indicates how important it is to make your online apps mobile-friendly today more than ever. To overcome the restrictions that the web as a platform imposes on mobile devices, companies frequently feel the need to build native or hybrid apps. In many circumstances, they will need to create an app for the web, iOS, and Android. In most cases, a native application is written in a device-specific programming language and integrated development environment (IDE). Native Android apps are typically written in Java using the Android Studio IDE. These programs are typically downloaded from app stores and have extensive access to device hardware via platform-specific APIs.These programs may be downloaded from the respective operating system's app store and operated in a native environment, with all of the functionality that a native app has. When these programs are removed from their natural context, they are unable to provide this experience owing to browser limitations. PWAs (Progressive Web Applications) may be able to overcome this issue. A hybrid application is produced with web technologies but may seem and operate like a native application using hybrid development frameworks (e.g., Apache Cordova). As a result, when designing a mobile application, the three most frequent options have typically been to construct native, hybrid, or mobile web apps; however, progressive web apps (PWAs), announced by Google in 2015, might be considered a fourth option. A Progressive Web App (PWA) is a web application that intends to provide a mobile device with a native-like user experience, such as offline support and push notifications.

2. FEATURES

- **Progressive** – Regardless of the browser that you are using or even where you are situated in the world,Progressive Web Apps work for every user. So, if a user uses Chrome or Opera or whether the user is living in India, UK or even North Korea, it doesn't matter! Progressive Web Apps will work just as well because they're built with progressive enhancement as a core tenet.

- **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.

- **Application feel**— Due to the app-style interactions and navigation in Progressive Web Apps, they give a feel of an application to users.

- **Independent of connectivity**—Service workers help Progressive Web Apps in such a way that they work offline, or even on networks with low quality.

- **Installation**—Users can keep the widely used PWAs on their home screen of their device without the interference of app stores.

- **Discoverable** - Progressive Web Apps are recognizable as applications. The service worker and W3C manifest registration scope allows search engines to find them easily.

- **Engageable**—Feature like push notifications makes Progressive Web Apps more engaging. These apps are installable and live on the user's home screen, without the need for an app store. Users can specify icons for the home screen and splash screen when the app is loading. Page to be loaded when the app is launched and screen orientation.

- **Safe**—Progressive Web Apps are served via HTTPS, which ensures that without authentication it cannot be tampered by anyone.

- **Fresh**- Progressive Web Apps are always up-to-date all thanks to the service worker update process.

3. METHODOLOGY

SERVICE WORKERS:

Service Workers are a key component of Progressive Web Apps. The following are some of the benefits supplied by service workers::

- Offline Access.
- Push Notifications
- Background content updating
- Content caching

Service workers perform the following functionalities:

- 1. Caches the App Shell.
- 2. Updates the Content in background.
- 3. Gets the push notification id from the user to send the notification.
- 4. Invalidates the cache when needed

APP SHELL:

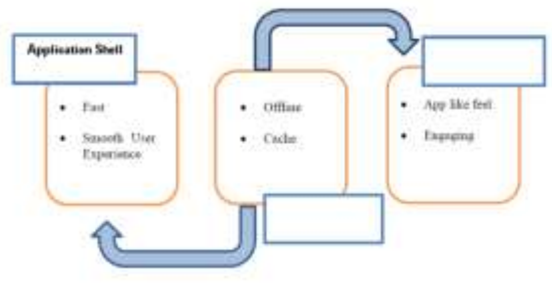
ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101259

The Service Worker serves up the Application Shell Architecture before delivering the content. The service worker frequently caches them from its source via API queries. People who visit a site frequently will be able to save the last material they saw while waiting for the network to dynamically load the most recent update. The focus of the App Shell architecture is on keeping the app UI shell and the content inside of it distinct, and they are cached independently. When a user visits and returns later, App Shell should ideally be cached so that it loads as rapidly as feasible. The user's perception of speed and usefulness of the programme should increase if the shell and content load independently..



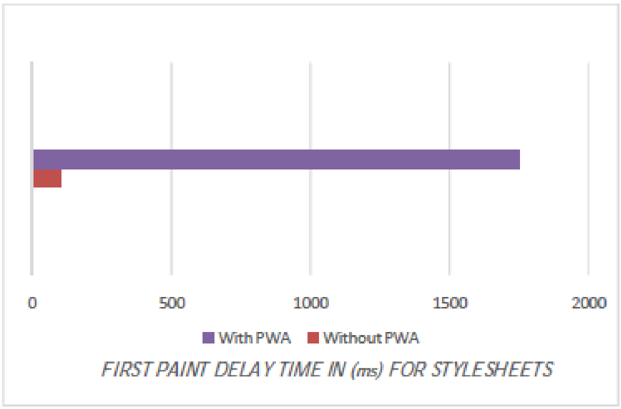
WEB APP MANIFEST:

In a JSON text file, the web app manifest provides information about a programme (such as its name, creator, icon, and description). The manifest must provide information about websites that are put on a device's home screen, allowing users to access them faster and have a more complete experience. Web app manifests are a component of progressive web applications, which allow websites to be deployed to a device's home screen without the need of an app store, as well as other features such as functioning offline and getting push notifications.

4. RESULTS

Google Lighthouse may be utilised programmatically as a Node module, as a Chrome Extension, or via the command line. Lighthouse focuses on PWA quality and performance indicators. The basic Lighthouse report presents data on a wide range of topics in an easy-to-understand style. The following are some important website performance inclusions:

- HTTP/2 — Displays a list of all internal resources that were not delivered via HTTP/2.
- First Meaningful Paint - The time it takes for the browser to start rendering content.
- Time to Interactive — A metric indicating when a page has loaded sufficiently for a user to engage with it.



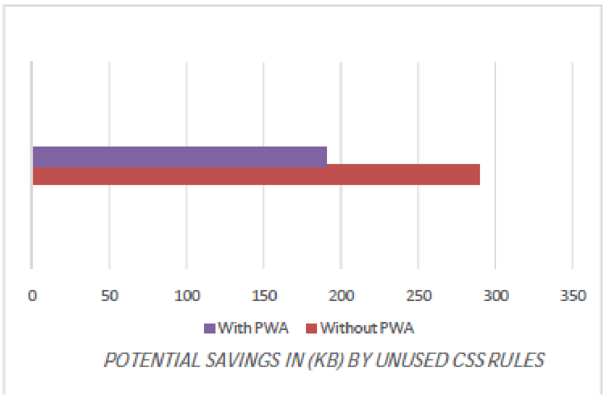
The initial page is when the browser has all of the information it requires to render the page. This refers to the first time the browser renders a picture of the produced page on the screen. Style sheets are a sort of template file that contains font and layout settings to give publications a consistent appearance. When compared to a webpage without PWA, the time it takes for the first paint from the style sheet of a PWA enabled web page is shorter (85 ms) (1769 ms).

ISSN (O) 2278-1021, ISSN (P) 2319-5940

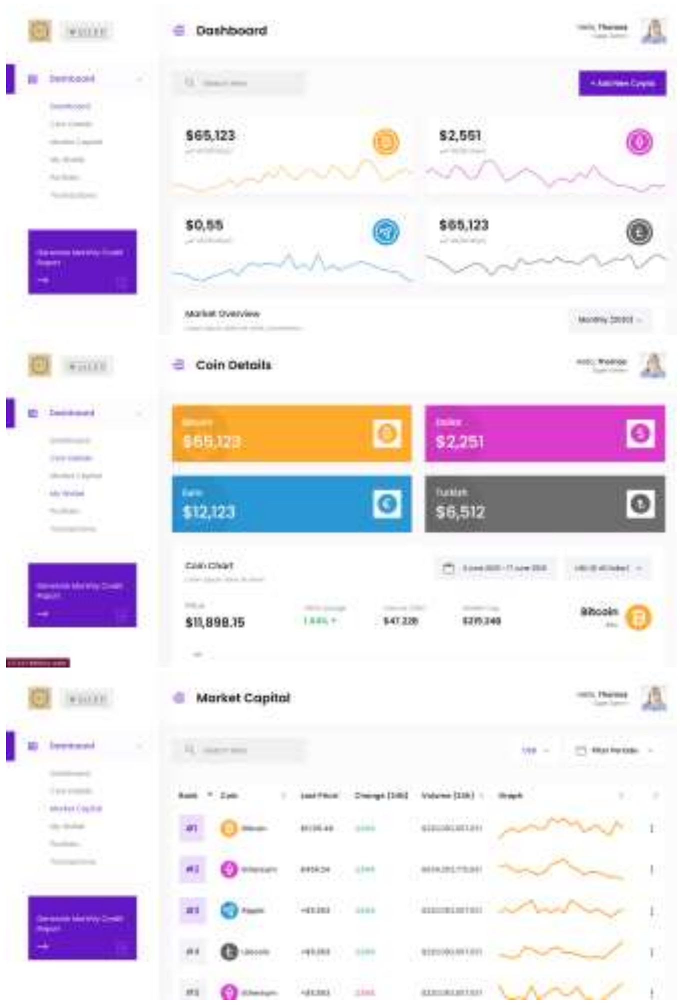
International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101259



A browser must first download and interpret any style sheets necessary to lay out a web page before it can begin rendering it. Even though many of the rules set in it don't apply to the current page, most websites reuse the same external CSS file for all of their pages. Cutting down on the CSS footprint is the greatest strategy to reduce latency caused by stylesheet download and rendering time. When compared to a webpage without PWA, the potential savings of a PWA equipped web page is lesser (187 KB) (283 KB).

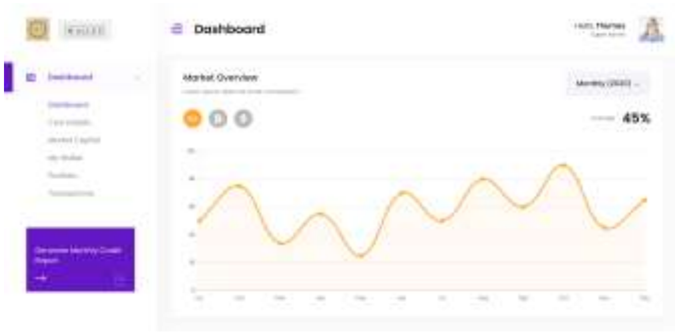


ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101259



5. CONCLUSIONS AND FUTURE SCOPE

Our Wall-app PWA is fast, lightweight and accurate because the nature of a progressive web app is that it's not very resource, native require a different set of libraries to run made on a different platform for every OS but this is not the case with PWAs, native apps use libraries which takes time to process and is heavy on the storage. Even though our Wall-et is a crypto trading app, it can be used offline to look at the recent trading history, coin trends, statistics and analysis. It analyzes trades and generates real-time reports on profit and loss. Realized and unrealized gains. It is easy to access, easier than a website on a web application because it uses much lower resources to load the data, so even with a slower internet connection i.e a 2G connection, the contents will easily load. In the future , the PWA will also support API integration by allowing different applications to use the service of our PWA. It can vastly increase the use case by allowing different crypto trading platforms to share information. Other applications can use our services and data through proper APIs to enrich the experience for the user. Finally, during these trying moments of COVID, an online progressive web software like Wall-et can assist regulate the situation by removing the need for users to engage physically.

REFERENCES

[1]. Parbat Thakur, “Evaluation and Implementation of Progressive Web Application”, bachelor’s thesis, Helsinki Metropolia University of Applied Sciences, 2018.
[2]. Mikael Wahlström, “Exploring Progressive Web Applications for Health Care”master's thesis, Umeå University, 2017.
[3]. Thomas Steiner, “An Analysis of Progressive Web App Features When the Means of Web Access is not a Web Browser”,2018.
[4]. Pavel Břoušek,” Evaluation and usage of Google Progressive Web Apps technology”, bachelor's thesis, Masaryk University Faculty of Informatics, 2017.
[5]. Bob Frankston,” Progressive Web Apps”, IEEE Consumer Electronics Magazine, vol.7, no 2. pp.106, 2018.
[6]. Rebecca Fransson and Alexandre Driaguine,” Comparing Progressive Web Applications with Native Android Applications”, bachelor’s thesis, Linnaeus University,2017.
[7]. Ivano Malavolta, Giuseppe Procaccianti, Paul Noorland andPeter 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.
[8]. Andreas Biørn-Hansen,Tim A. Majchrzak and Tor-Morten Grønli, “Progressive Web Apps: The Possible Web-native Unifier for Mobile Development”, SCITEPRESS, pp.345,2017.
[9]. Farrugia Kevin, “A Beginner’s Guide to Progressive Web Apps,” Smash Magazine,2016.
[10]. Teplý Jan Bc., “Hybrid mobile application for project”, Czech Technical University, Prague, 2016.
[11]. P. Kinlan, “Installable web Apps with the web App manifest in chrome for Android”, Google Developers, 2017.
[12]. J. Parsons, “What are progressive web Apps? “, The Official Ionic Blog,2016
[13]. David Fortunato and Jorge Bernardino,”Progressive web apps: An alternative to the native mobile Apps”, IEEE/13th Iberian Conference on Information Systems and Technologies (CISTI), 2018.