

Android Application Development: Working on Native and Progressive Web Application

Jayasimha S R¹, Prashanth K² and Usha J³

¹⁻²RV College of Engineering/MCA Department, Bengaluru, India

Email: jayasimhasr@rvce.edu.in, {prashanthk, ushaj } @rvce.edu.in

Abstract— Android application is a native application program interface for developing user-based applications. Native applications are e-based applications. It is not easy to host applications through native applications. Native apps are operating specific applications. Progressive web application is a solution to host an application through web browser. In this paper, the working of manifest, service workers, caching along with the storage are discussed. Comparing web application to a native application is also carried out and analyzed. In this paper the author discusses the manifest, service workers, cache, and storage services along with comparison of web application to a native application.

Index Terms— Android, Native app, Service worker, Manifest, PWA Cache

I. INTRODUCTION

In this age of ever-changing digital era, the most essential component for communications is through the usage of mobile. In the digital era, the usage of mobile is essential now a day for communication, connection and process. Desktop-based applications are the initial applications used in web browsers in 1980. In the 1990's the web browser turned into an android operating system, black berry [1]. Android is a platform for developing mobile based applications. It is software, operating system, and Linux based kernel. The Android app can be developed through the IDE of android studio. Android is developed through IDE as android studio. There are many features for developing a user interface for the applications developed. The widgets and wireless transitions through the application are one of the focal features available required for the development. The features are developed as a user interface for applications. Widgets and wireless transition through application is one of the features.



Figure1: Android Studio

The Figure 1 shows the android studio version 2021.3.1 and updated 2022.1.1 versions supports all the higher applications development on mobile support upward compatibility for the applications developed Native Mobile Applications can send the push notification and it work offline and it loads on the home screen[2]. Native applications run on secured containers, and it uses standard based technology, which is provided in web. Native applications are constricted to provide an attractive UI environment because lack network connection across the globe. Native application development needs training and experience to provide better user apps. Due to this, Progress Web Application (PWA) is a new technology invented by Google to overcome the native applications and mobile browsing. PWA was designed by Frances Berriman and google chrome engineer Alex Russell named it as progress web apps during 2015 [2]. PWA is launched by clicking on icon on the home screen the device just like the native apps. PWA is loaded instantly on screen regardless of availability of network connectivity. It supports the splash screen, push notifications and background services. PWA supports web app manifest, service workers and users can upgrade web apps to progressive web apps in their native operating system. PWA can work offline, load and send push notifications on the home screen.

II. LITERATURE SURVEY

D. Fortunato and J. Bernardino describe the capability and limitations that exist on developing a web app to run in all devices. The author compared the same solution for each different mobile platform [3]. Thakur, Parbat describe the usage PWA over native applications are discussed [4]. The author Aguirre, V., Delía, L discussed trusted web activities are the complementary to progress web applications. The PWA+TWA applications are analyzed completely, and these apps are the part of web applications.

TABLE 1: ANALYSIS OF MOBILE APPLICATIONS AND PWA

Sl. No	Technological Features	Mobile Web Applications	Progressive Web Applications
1.	Office usage	No	Yes
2.	Accessibility	Mostly web browser	Application launcher
3.	Speed	Very Low	Low
4.	Application usage	More in social media connect	Less usage in social media connect
5.	Cost	Very Low	Low

Table 1 gives the detailed analysis of Mobile web application usage and progressive web applications in the android environment. Author, Huber Stefan &Demetz, Lukas &Federer, Michael discussed the energy usage in the mobile application environment. The native applications are consuming less energy compared to progressive web applications [6]. [10] Thesmall-scale application deployment consumes less energy usage of the cloud compared to the large-scale applications. energy usage in the cloud is completely based on the hardware components and processing capacity of the device [7]. Memory usage and the application consumption memory space are also a factor to consider the energy usage in the mobile devices [8].

TABLE II: MOBILE PLATFORM DIFFERENTIATION

Sl. No	Mobile Platform	Virtual Machine Usage	Language used	Integrated Development Environment	User Interfaces	Devices	Servers
1.	Android	Dalvik	Java	Eclipse	XML files	Heterogeneous	Google
2.	IOS	No	C/Swift	XCode	Cocoa Touch	Homogenous	Apple
3.	Windows	CLR	C#/C++	Visual Studio	XAML Files	Homogenous	Microsoft
4.	Black berry	Enterprise Server VM	JAVA	Plug-in for eclipse	XML files	Homogenous	Research in Motion – Rim

In the table 2 shows the comparative analysis of different mobile platform, with virtual machine its usage and server devices in the IDE environment is discussed [9].

3. Manifest File:

During web development the manifest file will give metadata information on name and version. The Specific aspects of functionality such as background scripts, content scripts and browser actions will be defined. Manifest file is a special JSON file that specifies the basic project information and the icons.

Here are the few steps how the manifest file can be added to the html page.

Step 1: create a file first.html.

```
<html lang="en">
<head>.....<link rel="manifest" href="/manifest.json">/head>
<body>.....</body></html>
```

Step 2: create manifest.json

```
{
"name": "first_manifest",
"short_name": "first_manifest",
"start_url": "/first.html",
"display": "fullscreen",
"background_color": "#ffff00",
"description": "A simple application for manifestation",
"orientation": "portrait-primary",
"theme_color": "#2196F3",
}
```

Step 3: Install the Live Server



Fig2: Install of the Live Server on VS Code IDE

Step 4: Procedure to run file on server

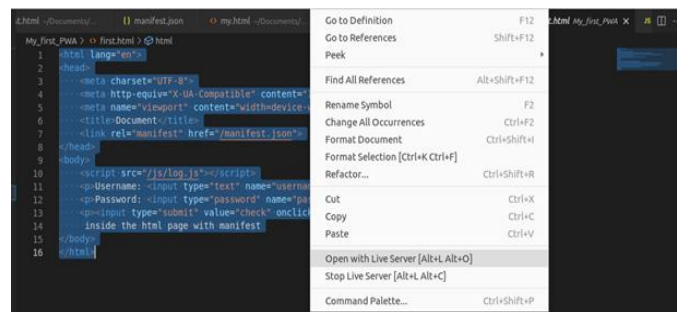


Fig 3: Display the process of running the HTML page on Live Server

Step 5: Display of the webpage with the manifest file

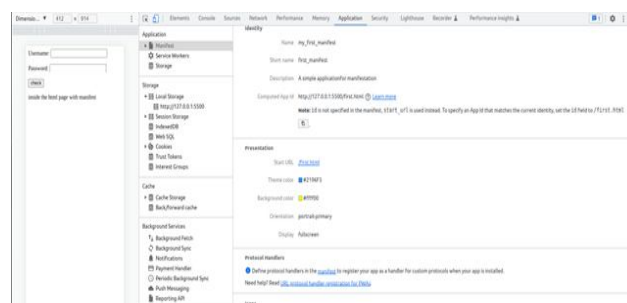


Fig 4: Display of the manifest file loaded on the application in developer tools

Progressive Web Application is a technology for modernizing the web sites to leverage user's expectations. User needs in how PWA provides is fast, installable, reliable, and engaging. Based on the user experience during the application usage there are different opinions like users hate delays and stress caused due to delays. Secondly fifty percent of the time will go to loading the contents. Third, due to limited storage not more applications can be installed.

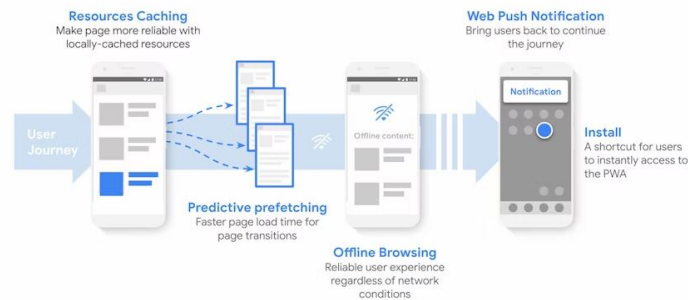


Fig 5: User Journey on PWA

In Figure 5 the PWAs leverage the web capabilities to next level like, predictive loading of the contents what users are seeing and cache the contents during the page transition. Adding a push notification as native app can show. PWAs make a website shortcut on the display to instantly access. Figure 6 is taken from the web.dev which shows the user journey

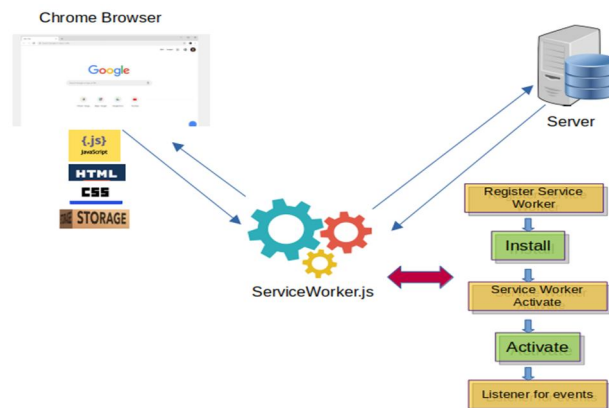


Fig 6: Service Worker Architecture

Figure 6 gives the service worker architecture, these are the basic parts of PWA which enables process to be fast loading, offline access, push notifications, and other capabilities. The Service Worker should be registered first under the scope of the project. Service worker analyses the installation process and completes. Sample code for registration is discussed below.

Create start.js inside the js folder

```
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register("/sw.js")
    .then((reg)=>console.log("registered",reg))
    .catch((err)=>console.log("err",err));
} else {
  console.log('No service worker support in this browser');
}
```

Inside an html page script tag to load the start.js to load the serviceworker.

```
<scriptsrc="/js/start.js"></script>
```

Lifecycle of ServiceWorker starts from the ‘install’ which runs only once during the registration of it, and till the change in serviceworker file is identified.

```
Create sw.js ---->ServiceWorker
self.addEventListener('install', _event => {
    console.log ('inside the install', _event);
});
```

‘Activate’ will be fired as the old serviceworker is gone and new service worker has got registered and the activate will be triggered.

```
self.addEventListener('activate', _event => {
    console.log ('inside the activate', _event);
});
```

Now the service worker is ready for fetch.

```
self.addEventListener('fetch', async (event) => {
    console.log ('inside the fetched', _event);
});
```

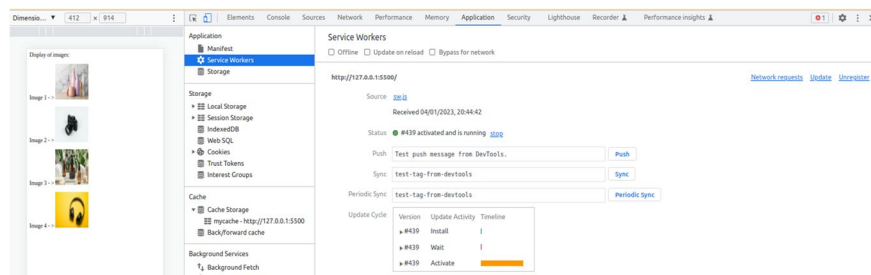


Fig 7: Service Worker

Cache API: Caching is a process of storing the web contents on browser side to respond for the respective request whenever the connection is not stable. To carry out this the Cache API is being used. The idea behind the caching is to load the contents back offline. The process of caching can be done in steps:

Define the name for cache and array to which the files to be added to cache.

Step1: Add the assets to caches in the install phase.

Step2: in fetch respond to the request when there is a need of contents to be displayed.

III. CONCLUSIONS

In the android application development, the usage of native and progress web application has discussed in this paper. PWA is fully supported with offline loading through service worker, and it allows push notification to the mobile applications. PWA provides a very good user interface and complete support of background synchronization. In this paper manifest, service worker working environment is discussed in brief. The working of service worker architecture along with its running status is discussed in detail.

REFERENCES

- [1] Reetu Malhotra, Deepak Kumar, D.P Gupta, “An Android Application for Campus Information System”, Procedia Computer Science, Volume 172,2020, PP 863-868, ISSN 1877-0509
- [2] Tandel, Sayali& Jamadar, Abhishek. (2018). Impact of Progressive Web Apps on Web App Development. 10.15680/IJIRSET.2018.0709021.
- [3] D. Fortunato and J. Bernardino, "Progressive web apps: An alternative to the native mobile Apps," 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), 2018, pp. 1-6, doi: 10.23919/CISTI.2018.8399228.
- [4] Thakur, Parbat. "Evaluation and implementation of progressive web application." (2018).
- [5] Aguirre, V., Delia, L., Thomas, P., Corbalán, L., Cáseres, G., Sosa, J.F. (2020). PWA and TWA: Recent Development Trends. In: Pesado, P., Arroyo, M. (eds) Computer Science – CACIC 2019. CACIC 2019. Communications in Computer and Information Science, vol 1184. Springer, Cham. https://doi.org/10.1007/978-3-030-48325-8_14
- [6] Huber, Stefan &Demetz, Lukas &Felderer, Michael. (2021). PWA vs the Others: A Comparative Study on the UI Energy-Efficiency of Progressive Web Apps. 10.1007/978-3-030-74296-6_35.

- [7] S. R. Jayasimha, J. Usha and S. G. Srivani, "Efficient Power Management using Fuzzy Logic for Cloud Computing Environment," 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), 2018, pp. 35-40, doi: 10.1109/CSITSS.2018.8768764.
- [8] Usha, J. & S R, Jayasimha and Srivani, S G. (2018). Automata Approach to Reduce Power Consumption in Smart Grid Cloud Data Center. 10.1007/978-981-10-9059-2_23.
- [9] OluwatofunmiAdetunjia*, ChigozirimAjaegbub,NzechukwuOtunemec, Olawale J. Omotoshod, (2021). Dawning of Progressive Web Applications (PWA): Edging Out the Pitfalls of Traditional Mobile Development, SSN (Print) 2313-4410, ISSN (Online) 2313-4402.
- [10] Jayasimha, S. R., J. Usha, and S. G. SrivaniIyengar. "A comprehensive review of energy efficiency in cloud computing environment." *Int. J. Eng. Technol* 7 (2018):249-252.