

Shishir Singh Timalsina

PROGRESSIVE WEB APPLICATION WITH REACTJS

PROGRESSIVE WEB APPLICATION WITH REACTJS

Shishir Singh Timalina
Bachelor's Thesis
Spring-Autumn 2019
Bachelor's in information technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Bachelor's in information technology

Author(s): Shishir Singh Timalsina

Title of the bachelor's thesis: Progressive Web Application with Reactjs

Supervisor(s): Lasse Haverinen

Term and year of completion: Spring-Autumn 2019

Number of pages: 31

The main aim of this paper was research and implementation of a Progressive web application (PWA) with Reactjs by creating a mobile web application for the Vällkylä tenant committee. PWA is web-application with an additional feature of being installable on a device. PWA is less expensive to develop and simpler compared to a native or hybrid mobile application making them very popular among developers these days.

An application was developed with the help of tools such as Nodejs, a JavaScript run-time environment, Reactjs for user interface and Firebase for a database. Service worker and a Manifest file were used for the implementation of the application to work in offline mode and make the application installable in the home screen resembling a native application. The code for the application is stored in GitHub and the application was deployed using Netlify for hosting.

As a result, there is a working mobile web application that can be used by the Vällkylä tenant committee and the residents of that community. With the help of this application, people can find information regarding the Vällkylä housing community peripherals and opportunities that will be very helpful to make their living experience better in this area. The application functions as expected, a resemblance of a native mobile application, but with a very small size. This demonstrates how powerful of a tool PWA is and its stronghold among the developers nowadays.

Keywords:

Progressive web application, Reactjs, Firebase, Nodejs, Mobile Application, Web Applications, Home screen, Manifest, Service worker file

TABLE OF CONTENT	
1 INTRODUCTION	6
2 TYPES OF MOBILE APPLICATION	8
2.1 Native Mobile Applications	8
2.2 Hybrid Mobile Applications	9
2.3 Web-based mobile application	9
3 PROGRESSIVE WEB APPLICATIONS	10
3.1 Features of a Progressive Web Application	11
3.2 PWA vs Native / Hybrid mobile application	12
3.3 Components of PWA	12
3.4 Manifest File	14
3.5 Service worker	15
4 TECHNOLOGIES USED	17
4.1 Nodejs	17
4.2 Reactjs	17
4.3 Firebase	18
4.4 Lighthouse	18
5 APPLICATION DEVELOPED	19
5.1 Features of the application developed	19
5.2 Development Process	20
5.2.1 Development Environment	20
5.2.2 Reactjs Environment	22
5.2.3 Backend	23
5.2.4 Converting to PWA	23
5.2.5 Lighthouse testing	25
5.2.6 Deploying the PWA	25
6 RESULTS	26
7 CONCLUSIONS	28
REFERENCES	29

VOCABULARY

WORDS

Apps

API

CSS

HTML5

iOS

js

JSON

JSX

PSOAS

PWA

UI

URL

VS code

WIFI

DEFINITION

Applications

Application Programming Interface

Cascading Style Sheets

Hypertext Markup Language

iPhone Operating System

JavaScript

JavaScript Object Notation

JavaScript Syntax Extension

Pohjois-Suomen opiskelija-asuntosäätiö

Progressive Web application

User Interface

Uniform Resource Locator

Visual Studio code

Wireless Fidelity

1 INTRODUCTION

Almost everyone has a smartphone in their pocket and access to the internet. There will be almost 3.8 billion people in the world who will have a smartphone by 2021 (Takahashi, 2018). These smartphones have changed the way how one uses the internet and what it is used for. In recent years research has shown that the average person browses more internet from a phone than a traditional desktop environment. According to Tech-jury (Techjury, 2019), a tech news website, in 2018 52.2% of website traffic was from a mobile device around the globe. The numbers were even greater in some parts of the world as nearly 65% in Asia and 60% in Africa smartphones are accountable for all the internet traffic. These numbers are only increasing as it is estimated that 80% of world internet access is expected from a mobile device by the end of 2019 (Petrov, 2019).

By 2020 mobile apps are expected to generate 188.9 billion US dollars in revenue (Deyan, 2019). There is a great value of a product that seemingly fits on the screen of a smartphone. There are millions of mobile applications and thousands of websites optimized for these mobile devices (Clement, 2019). They have become better and better with time and demands. They are improving with a better design, better user experience optimization, large, able to handle a large amount of data processing and a lot of new extra features with amazing graphical aesthetics. These applications have found their sanctuary in almost every aspect of human society. A mobile application is available for health, business, educations, communication, security and, other many aspects of society.

Around 67% of internet users worldwide visit the web on a mobile device (Deyan, 2019). A mobile website is easier and cheaper to develop. The reachability of a website designed for a mobile device is higher than that of a traditional mobile application. The problem is 85% of consumers prefer mobile apps over mobile websites. Consumer's expectations from a mobile application are much higher than a mobile website. Consumers feel a mobile application is a more convenient, faster, better user experience and easier to browse. (Moth, 2013). That is where PWA comes into the picture. PWA is a website or web application but imitates a mobile application.

PWA (Progressive Web Application) provides many advantages over the traditional method of browsing the internet and make mobile web applications more usable and engaging. PWA is relatively smaller in size and eliminates the requirement of a store such as Google Playstore or Apple

App store is the reason why PWA is a potential way of Web technologies for the future (Progressive Web Applications, n.d.). PWA is a new technology gaining rapid popularity among developers.

For this thesis, a PWA for Vällkylä Tenant committee was developed which is accessible from a web host, installable or in this case, able to be added to the home screen in a device. The application can operate in offline mode as well.

Vällkylä is one of the student residence locations in Oulu city maintained by PSOAS (Pohjois-Suomen opiskelija-asuntosäätiö). PSOAS operates different student housing locations available in different parts of the city. Vällkylä is one of such residence locations. In Vällkylä there is an acting tenant committee with around 5 members. Information regarding different student residence locations is available in PSOAS official website psoas.fi or omapsoas.fi. Different residence locations of PSOAS have different sets of features and activities opportunities. PSOAS official website is very limited with features and services for the acting tenant committee of Vällkylä. To overcome some of the shortcomings of the official PSOAS website Vällkylä Tenant committee decided to have a website.

This document is divided into two distinct parts. The first part will provide highlights to the theoretical study of PWA and its core components than the second part will focus on how the Vällkylä Tenant committee web application was developed and made into PWA. This thesis aims to demonstrate how simple and yet powerful of a tool PWA is.

In this document there will be the description of PWA, it's basic components and core technologies. The application developed during this thesis work will be a demonstration of how a PWA is implemented using Reactjs for designing and developing the user interface and Firebase for implementing a database for storing data. The application developed will be optimized for both desktop and mobile environment viewing. This thesis document objective is to give answers to three important questions:

1. What is a PWA?
2. Why PWA is one of the ways for the future of mobile web applications?
3. Why PWA was best the choice for the tenant committee's application?

2 TYPES OF MOBILE APPLICATION

For understanding the PWA better, it is important to be familiar with what are the types of mobile applications that are prevailing in this industry. There are major three types of mobile applications that can be discovered in the industry:

- Native Mobile Applications
- Hybrid Mobile Applications
- Web-based Mobile Application (What are the popular types and categories of apps, 2019)

2.1 Native Mobile Applications

Native mobile applications are developed exclusively for an environment or operating system, therefore, making them “native” for a specific platform. One application developed for a particular operating system is not executable in other operating system environments.

The most popular operating system for mobile devices is iOS and Android (Mobile Operating System Market Share Worldwide, 2019). Typical iOS mobile applications are developed using Swift programming language (Swift "The powerful programming language that is also easy to learn", 2019) while a mobile application for Android is generally developed using the Java programming language (Java Programming Language, 2019). Another mobile operating system is Windows mobile operating system. It is developed in .Net framework (Introduction to .NET Framework, 2019), but in the present year 2019, they do not have significant market share so it can be comfortably said they are an event of the past (Mobile Operating System Market Share Worldwide, 2019). It is still important to mention that PWA is available for Windows desktop environment and is under continuous development process.

Native mobile applications are developed in completely different programming languages, different theories of design and different techniques from one another giving them a unique set of characteristics and features distinguishing them apart very widely. The main advantages these types of applications provide are a high level of performance, more secure, safe, better accessibility and usage of hardware of the devices. They have better access to a wide range of APIs making them highly optimized for user experience. (What are the popular types and categories of apps, 2019)

2.2 Hybrid Mobile Applications

Hybrid mobile applications are platform-independent mobile applications developed using other web technologies such as HTML5, CSS, and JavaScript. They are designed to be usable in different types of environments and operating systems. A hybrid mobile application may have different levels of support and accessibility for an environment and hardware they are running on.

Some of the most common developer tools for designing and developing hybrid mobile applications are Apache Cordova, Ionic Framework, Meteor, Angular js etc. These tools help the developer to design and develop a mobile application that can be wrapped in a disguise using the Software Development Kit for an environment making them resemble a native mobile application. Hybrid mobile applications are a single codebase making them faster and cheaper compared to investing in developments of a native mobile application. This means one can write the code once and then the same codes can be re-used to convert the application into different platforms using the respective Software development kits.

Some of the limitations of hybrid mobile applications are that they are comparatively slower. The user experience with them is found to be slightly inconsistent in different hardware. (What are the popular types and categories of apps, 2019)

2.3 Web-based mobile application

Web-based mobile applications are browser-based mobile applications designed and optimized for a mobile view. These applications are not installable in a device. They are developed using tools such as HTML5, CSS, JavaScript and many other forms of web technologies.

Web-based mobile applications require a continuous uninterrupted internet connection. Without the internet, they are not functional. A poor connection to the internet can result in a very unpleasant user experience. They require a very minimum amount of device memory and performance. They do not have the same level of access and support for the device they are operating on as of native or hybrid mobile applications. Because of their smaller size and lack of access to many hardware and software components, they may fall behind in large numbers of features that native or hybrid mobile applications prevail in. (What are the popular types and categories of apps, 2019)

3 PROGRESSIVE WEB APPLICATIONS

Progressive web application (PWA) is a web-based application with user experience very similar to that of a native application. It can be considered a new advanced form of a traditional web-based application. PWA is available in both desktop and mobile environments. (Love, 2019). This thesis study will highlight features of PWA for mobile version applications only.

The concept of a mobile application that is browser-based was one of the most key ideas presented by Steve Jobs during the launch of the iPhone in 2007. But because of many constraints, the idea did not become a reality and iPhone application shifted towards Apple SDK based mobile application available in their Apple app store. Seeing many limitations Apple did not pursue the idea of further development, but another giant tech company Google kept persisting the idea of a web-based application. The term 'Progressive web apps' was coined by front-end developer and designer Frances Berriman and Google Chrome engineer Alex Russell at Google. With that new features of service workers and a web app manifest file was described which upgrades a web application to a progressive web application.

Inside a web browser, many mobile web applications behave in the same way as a real mobile application i.e. Native or hybrid mobile applications. The technology of PWA is there to make it happen outside the web browser also by making a mobile web application behave more in manner of a native application in the device it operates in. PWA is very different from a regular mobile application. Unlike native or hybrid applications that are accessed from different app store platforms, PWA is hosted on a website and runs in an internet browser. This means an update can be submitted to the application on the website rather than many different application stores for each platform differently.

A PWA can be installed or added to a home screen with an icon identical to a regular mobile application. From this shortcut icon, the application will be launchable and usable. It will resemble more of a regular application. PWA is a Web application which means it needs to be connected to the internet but that is where the PWA unique ability comes into action. There is offline support, or in case of lack of network, a PWA is still functional as a native or hybrid application. The components

of the application are installed or cached for offline use. PWA is not close to native or hybrid applications in functionality but has a huge advantage in the memory footprint as PWA is very small in size.

Progressive web application is also developed using web technologies such as HTML5, CSS, and JavaScript same as hybrid mobile application. But instead of developing for an operating system environment PWA is designed for web browsers. The most important factor that makes PWA powerful is platform independence over a hybrid mobile application and native mobile applications. (Nath, 2017)

3.1 Features of a Progressive Web Application

Some of the key features of a PWA are:

- **Responsive:** A PWA fits in all different screen size options available for a mobile device.
- **Installable:** A PWA is added to app drawer in mobile devices which makes them grouped among other applications in the devices in view to the user. Among all the neighbors of native application icons, a PWA will blend in giving then a resemblance of a mobile application rather than a web application.
- **Splash Screen:** During startup, PWA will display a splash screen for a very small amount of time. This technique is very common among native applications.
- **Highly engage-able:** A PWA keeps the user engaged. PWA supports some of the features of the native applications such as push notification, home screen icon, fullscreen in the viewport and offline-first app approach. (Halder, 2018)

3.2 PWA vs Native / Hybrid mobile application

PWA is a web application in the cloak of a native or hybrid application. The main target of PWA resembling as a native or hybrid application. A comparison of a progressive web application and a traditional native or hybrid mobile application are listed below in the table:

TABLE 1. Comparison between PWA and Native/Hybrid mobile applications (What are the popular types and categories of apps, 2019)

Progressive Web Application	Native/Hybrid Mobile Application
· Very small in size.	· Comparatively very large
· No need for a digital distribution platform	· Require a digital distribution platform such as Appstore or Play Store.
· Support different operating system environment	· Must be specifically designed to run on a particular operating system environment
· Lacks access to many APIs of a device	· Has a large APIs to select making them more powerful, they provide more user engagement features and many accessible hardware components of the device they operate on.
· First-time use requires a network connection.	· Work during lack of connection to the network

3.3 Components of PWA

Any web application which is optimized for a mobile screen can be upgraded to a PWA. The concept of PWA is very simple, caching the application on the device or storing basic entities of the

application on the device. When a user opens a link from the PWA only those opened by the user are cached. The parts of the application that are not used or loaded by the user from the network are not cached making the application much smaller.

The most important part of PWA is the similar resemblance of a native application. This means the application will have a logo or home icon for the mobile device. PWA will load in the user's device's screen (Fullscreen if optimized) without displaying any web browser's UI components. This makes the experience of using a PWA very engageable. A very important phenomenon to notice is that PWA itself is running in a browser but without showing the browser's UI components in the viewport. This creates an illusion of a native application to the users.

To do this it is required to make a Manifest JSON file in the project folder which provides information regarding splash screen, icons, application's name and many more. The application is cached using a JavaScript file known as service worker. (Ghimire, 2019)

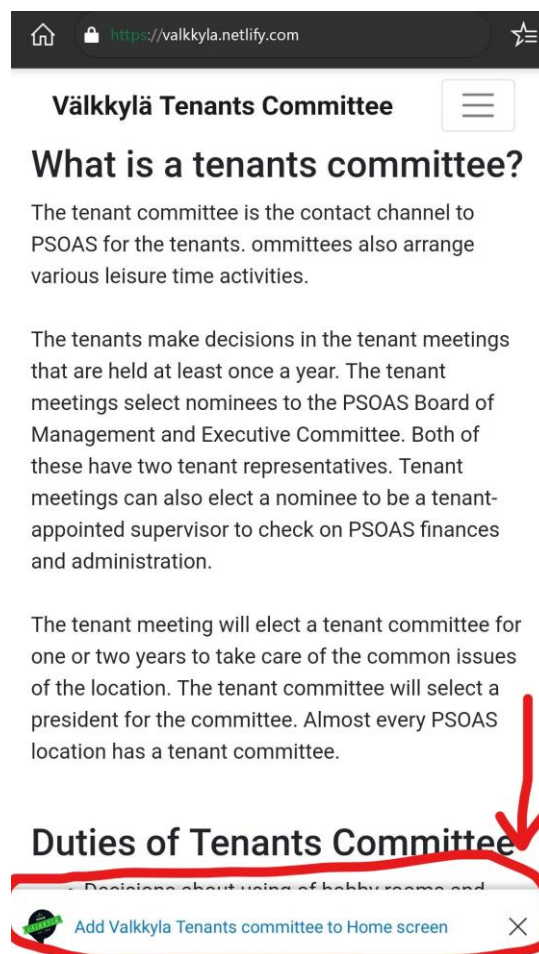


FIGURE 1. Screen snapshot of a PWA in a mobile web browser

FIGURE 1 gives a closer look at how a PWA behaves in a mobile device. When the URL is opened, in this case, `valkkyla.netlify.com`, a small prompt appears on the bottom of the screen. Defined within the parameter of the red color, the application provides an option to add or install the application to the home screen. On the left side of the prompt, there is the logo of the application and on the middle section of the prompt, the name of the application is mentioned, Väkkylä Tenant committee.

The process of making a PWA or converting an existing application to PWA can be summarized into three steps:

1. Design and develop web application optimized for mobile view
2. Add a manifest file for application information
3. Add a service worker file to cache the application

3.4 Manifest File

The manifest file is JSON (JavaScript Object Notation) file which holds some information regarding the application. This information is used when running the application on a mobile device. This information helps to identify the application while saving or installing them on the device's home screen. The most important information they hold is the name of the application, different types of icons for the application used in respective places and environment, default orientation of the application, background screen color, theme color, and many more. (Ghimire, 2019)

```

public > {} manifest.json > abc background_color
1  {
2    "short_name": "Valkkyla Tenants committee",
3    "name": "Välkkylä Tenants Committee ",
4    "icons": [
5      {
6        "src": "favicon.ico",
7        "sizes": "64x64 32x32 24x24 16x16",
8        "type": "image/x-icon"
9      },
10     {
11       "src": "icon.png",
12       "sizes": "512x512",
13       "type": "image/png"
14     }
15   ],
16   "start_url": ".",
17   "display": "standalone",
18   "theme_color": "#000000",
19   "background_color": "#ffffff"
20 }
21

```

FIGURE 2. Example code of a manifest file

FIGURE 2 is an example code of a manifest file. In this code, the name of the application is mentioned "Välkkyla Tenants committee". So when an application is installed or added to the home screen in a mobile device, it would have that as the name of the application. Also, there are icons mentioned in this piece of code with different sizes that will be auto-selected according to the device requirements. Icon optimized for an iOS device is mentioned which will have a resolution of 512x512. Furthermore, theme color is mentioned and background color. The background color will be used for displaying the splash screen for a very short time when the application is loaded.

3.5 Service worker

Service worker can be considered the backbone of a Progressive Web application. It is responsible for caching the application in the devices, push notification, background synchronization, and handling network traffic (Gaunt, 2019).

Web applications are very simple. When a device is connected to a network the application will load in the browser but when there is no network a web application would not load. This is where service worker will make a difference in the application and make them more capable. For the first-time load of the application service worker stores the required data assets of the application in the browser cache. And when a user visits the application next time the service worker will load the

previously cached data in the browser's viewport even before checking the network. This results in the capability for the application to load and operate in a lack of network. (Halder, 2018)

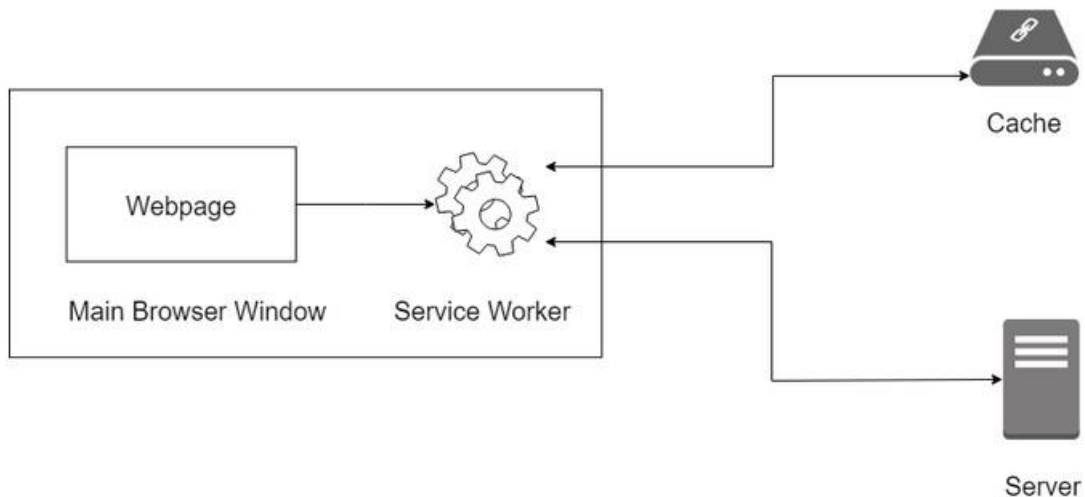


FIGURE 3. The basic concept of a Service Worker (Ghimire, 2019)

From FIGURE 3, it can be analyzed that the service worker works in a very different manner despite loading in the same browser. Rather than in the main windows of the browser, it works in the background. From there it makes a connection with the network from where it loads and caches all the required data assets of an application. When the device is connected to a network, the service worker loads data from the server and displays them in view shell. When the device is not connected to a network service worker simply loads and displays all the cached assets of the application from the previous run session, which results in a seamless user experience despite poor or no network connection. This is how a service worker makes the PWA behave more alike to a native application. (Ghimire, 2019)

4 TECHNOLOGIES USED

4.1 Nodejs

Node.js is a JavaScript runtime environment that includes everything required for executing a program written in JavaScript language. JavaScript was designed for running a program in a web browser. It was discovered that JavaScript has much more potential than only being used in web applications so node.js came into existence making JavaScript available outside a web browser. With node.js JavaScript can do more than just making websites interactive.

Both web browser JavaScript and node.js run on the V8 JavaScript runtime engine. This engine converts the JavaScript code in machine code which can be executed without interpreting first by the computer. (Priyesh, 2018)

4.2 Reactjs

Reactjs is a JavaScript library developed by Facebook used for developing user interfaces, especially for a single page application. Reactjs is used for handling multiple view layers of an application utilizing many different reusable components. An application developed using Reactjs can make changes to data in the viewport without reloading the whole page itself. This makes Reactjs very popular among modern web and application developers. (Pandit, 2017)

Reactjs is component-based which means when a page is loaded in viewport all the elements displayed on the screen are divided into many components. When one of the elements needs to update or make the changes, the other components are not affected during this process. Each time an element needs to change in the viewport the page does not have to reload. This effect makes a web-based application appear less of a webpage and acts more comparable to an application. Developers can work with one component at a time instead of designing and writing down all the codes for the entire page that needs to be loaded. This further improves the expandability of the application as more components can be added to add new feature sets.

Reactjs is written in JSX which is an extension to JavaScript language. They look very similar to JavaScript. Anyone familiar with HTML and JavaScript can easily learn how to write them. They are very human-readable which makes them very easy to work with. (Pandit, 2017)

4.3 Firebase

Google Firebase is a backend tool for the development of high-performance and scalable applications. Firebase is prominent among developers for web applications and mobile applications as they are very low cost and less time-consuming. James Templon and Andrew Lee founded Evolve in 2011 which developed Firebase. Firebase was purchased by Google in 2014. It was remodeled by Google for simpler to grasp for developers and made the services cheaper with many new additional features. Many basic level functionalities can be acquired from Firebase for free. Some of the most important services provided by Firebase are:

- Analytics
- Cloud Messaging
- Authentication
- Realtime Database
- Cloud Firestore
- Storage
- Hosting
- Machine Learning Kit
- Crashlytics
- Test Labs (Patel, 2019)

4.4 Lighthouse

Lighthouse is an open-source automated tool developed for improving the performance and quality of a web page. Lighthouse is a Google Chrome browser extension tool. The development of PWA is heavily influenced by Google. Their internet browser, Google Chrome, provides the best support for the development of a PWA. Lighthouse helps developers to determine if the web-application developed is functional enough to be a PWA or not.

From a simple test run, one can have an idea of how well the application is performing as a PWA. A test run from Lighthouse tool of a web page can provide a report on its performance, accessibility, SEO and most importantly how well the practice is performed for a website to be a PWA. (Lighthouse, 2019) A score is provided to let the developer know how much of a well-optimized web page is. Higher the score, better the website and its performance as a PWA.

5 APPLICATION DEVELOPED

5.1 Features of the application developed

Välkkylä tenant committee conducts many events and activities regularly. The committee is also responsible for taking care of some of the public spaces available in the peripheral of Välkkylä. Some of the common issues the committee faces are:

- Tenants unable to find contact information of committee members
- Hard to deliver information about events resulting in the inadequacy of participation from the tenant living in the locality.
- Common questions are frequently asked by the tenant.
- A new tenant living in Välkkylä lack knowledge of all the opportunities and spaces available in the locality.
- Some repetitive mistakes and common complains go unheard.

To resolve these issues Välkkylä tenant committee decided to have a personal web site or web application accessible for all the tenant living in this community. Some of the important features of the application are:

- Information regarding the committee and its responsibilities
- Contact information concerning committee members and their responsibilities
- Information regarding common spaces:
 - TV room
 - Workshop room
 - Billiard room
 - Study room
 - Sewing room
 - Grill place
 - Parking space
- Provide answers to frequently asked questions
- User profile for posting news and information

- Store files and pictures
- Post news for tenant to view:
 - Meeting information
 - Actions taken by the committee
 - Information about the decisions made by the committee
 - Change in some systems around the public spaces
 - Events information organized by the committee
 - Information regarding malfunctioning or broken item

5.2 Development Process

The development of the Vällkylä tenant committee application went through different phases.

The phases of development of the applications are listed below:

- Development environment
- Reactjs environment
- Backend
- Converting to a PWA
- Lighthouse testing
- Deploying the PWA

5.2.1 Development Environment

The application was developed in a Windows operating system environment. VS Code was used as a text editor for writing and modifying codes. An extension program called ES7 React/Redux/GraphQL/React-Native snippets was installed in VS Code for some shortcut keys which makes writing JSX codes faster.

NodeJS was installed therefore JavaScript runtime environment is available. As the application is a web-based application so it was important that to use JavaScript. JavaScript runs mainly only in a browser and with the assistance of NodeJS, it is possible to develop the PWA in a desktop environment. Npm was installed alongside Nodejs. Npm provides access to thousands of free packages to develop the application.

```

1  {
2    "name": "project",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "bootstrap": "^4.3.1",
7      "firebase": "^6.6.1",
8      "react": "^16.8.6",
9      "react-bootstrap": "^1.0.0-beta.8",
10     "react-dom": "^16.8.6",
11     "react-router-dom": "^5.0.1",
12     "react-scripts": "3.0.0"
13   },
14   "scripts": {
15     "start": "react-scripts start",
16     "build": "react-scripts build",
17     "test": "react-scripts test",
18     "eject": "react-scripts eject"
19   },
20   "eslintConfig": {
21     "extends": "react-app"
22   },
23   "browserslist": {
24     "production": [
25       ">0.2%",
26       "not dead",
27       "not op_mini all"
28     ],
29     "development": [
30       "last 1 chrome version",
31       "last 1 firefox version",
32       "last 1 safari version"
33     ]
34   }
35 }

```

FIGURE 4. Screenshot of package.json file

FIGURE 4 above is a screenshot of package.json file. It contains the application's required dependencies information. The most important dependencies are React, Firebase and React-bootstrap. Other dependencies that assist in the development of the react application are React-Bootstrap, React-Router-Dom, React-Dom, and React-Scripts.

5.2.2 Reactjs Environment

The application was developed using Reactjs. The reason to choose Reactjs was that it simplifies the development of the application and it is very simple to grasp. A Web application designed in Reactjs does not need to reload every time some changes are made.

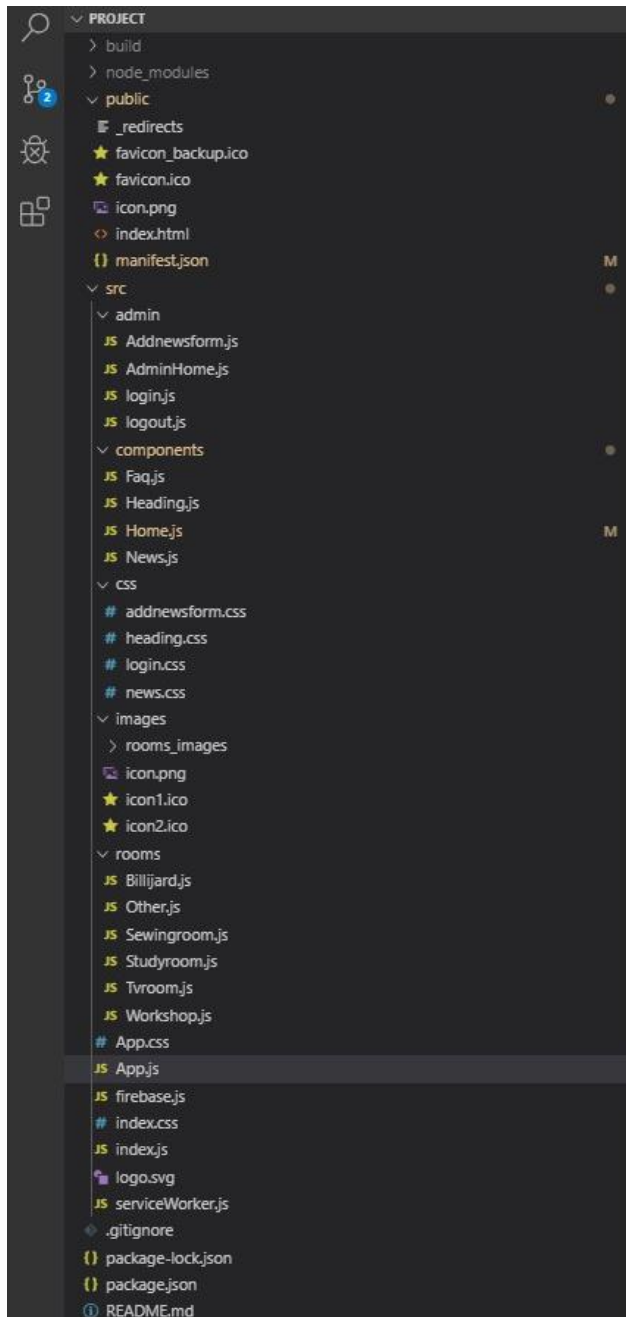


FIGURE 5. The file structure of the project

FIGURE 5 is a screenshot of the project file structure. Components are available in three distinct folders. A folder named component contains components for all the major pages for the application such as header and main menu options. The room folder contains components related to only rooms for the applications. Admin folder holds components required for login authentications and adding news feature of the application.

5.2.3 Backend

Firebase was chosen for the backend for its simplicity and easiness of use. Firebase makes it very easy to make database, authentication, storing data and hosting. For this project, Firebase was chosen for the database only. The key features of the application are storing data and storing files and pictures. These activities are handled by Firebase very easily and efficiently which is very important for learning developers.

For the number of features and complexity of the application developed, Firebase fits perfectly with its features, its simplicity and no cost for operation. One of the most important aspects of Firebase is its capability for expansion. If one wants to expand the application Firebase offers great possibilities of services at a very reasonable price.

For the application developed, only one user profile was created which can access the login feature. After logging in the user can post a piece of news with title and description which will be stored in the database. The data is retracted from the application's latest news page where the user can view all the listed news.

5.2.4 Converting to PWA

The process of converting an existing web application to PWA was done by taking two actions. Make a manifest JSON file and a Service worker file. In the manifest file, important elements of the application mentioned are the name of the application, its background color, and the icon for the application. From that, a splash screen is acquired which displays for a very small amount of time before the application loads in the viewport. The splash screen displays the application's name and icon of the application.

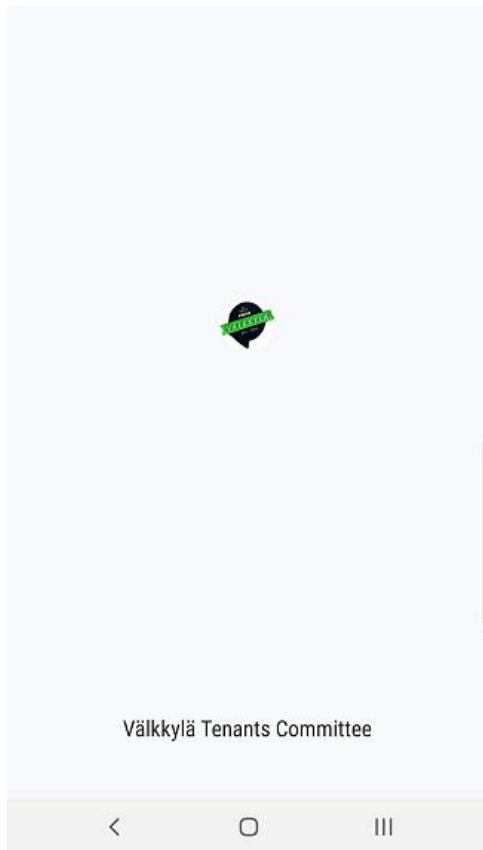


FIGURE 6. Screenshot of splash screen of the application

From *FIGURE 6* it can be observed that the name of the application is Välkkylä Tenant Committee with a small icon in the middle. It can be observed that the background color is white as mentioned in the manifest file. All this information is well mentioned in the manifest file.

The next step is to make a Service worker. Reactjs by default provides a Service worker file. It does all the necessary tasks of caching the application with the necessary assets of the application. When a Reactjs project is created, it exists as a file in the project directory but in an unregistered or unused state. Changes to a single line of code and a running service worker can be obtained as demonstrated in *FIGURE 7*. The screenshot was taken from the project files.



FIGURE 7. Registering service worker.

5.2.5 Lighthouse testing

FIGURE 8 below shows the result of the lighthouse test conducted on valkkyla.netlify.net. A simple test of the application developed was conducted using Lighthouse extension tool in Google Chrome browser.

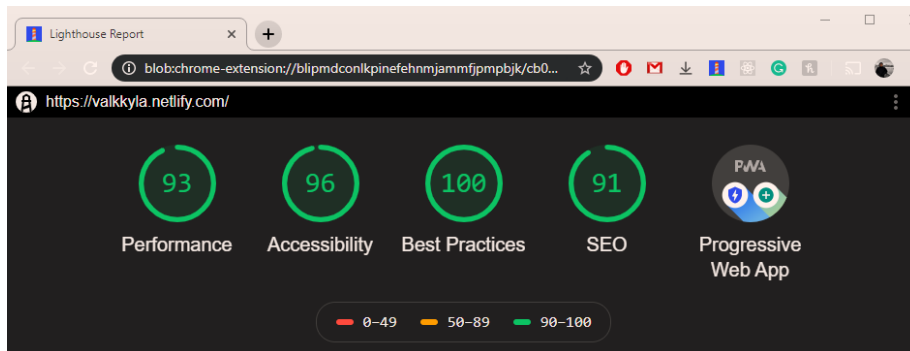


FIGURE 8. Lighthouse test run of valkkyla.netlify.net on chrome browser

The test score of the application developed is very high ie. above 90. The application developed can be well defined within parameters of a PWA requirement. Any user desiring to use this application will have a seamless experience of a mobile application while the application developed will remain a web application.

5.2.6 Deploying the PWA

For deploying the application, code was stored in GitHub. GitHub is a web platform that hosts code repositories. Almost every developer uses it for storing codes and version control. The application was hosted on Netlify. Netlify is a free web hosting platform with very fast and reliable performance. Netlify gives great offers and an option to expand or make an improvement to the application. Netlify offers options for versioning and rollbacks in case it is necessary to revert to the older version of the application. (Pelser, 2017)

6 RESULTS

After theoretical studies and development, a progressive web application was developed and operating. There was a successful configuration of the environment, coding, debugging the codes and hosting the application successfully. Bootstraps were used during the development process of the application providing a well-optimized view for both desktop and mobile environments.

The PWA performs like a regular website in desktop mode and has reached capabilities resembling a native application while viewed in a mobile environment. The application can work in offline mode or poor network quality. The application developed is installable on the home screen with an icon for the mobile screen.

By using the application developed, users can find information regarding the public spaces in Vällkylä community, information regarding the tenant committee and actions being taken by the committee and it also provides answers to some frequently asked questions by the tenant.

This application can be accessed from the link <https://valkkyla.netlify.com>.

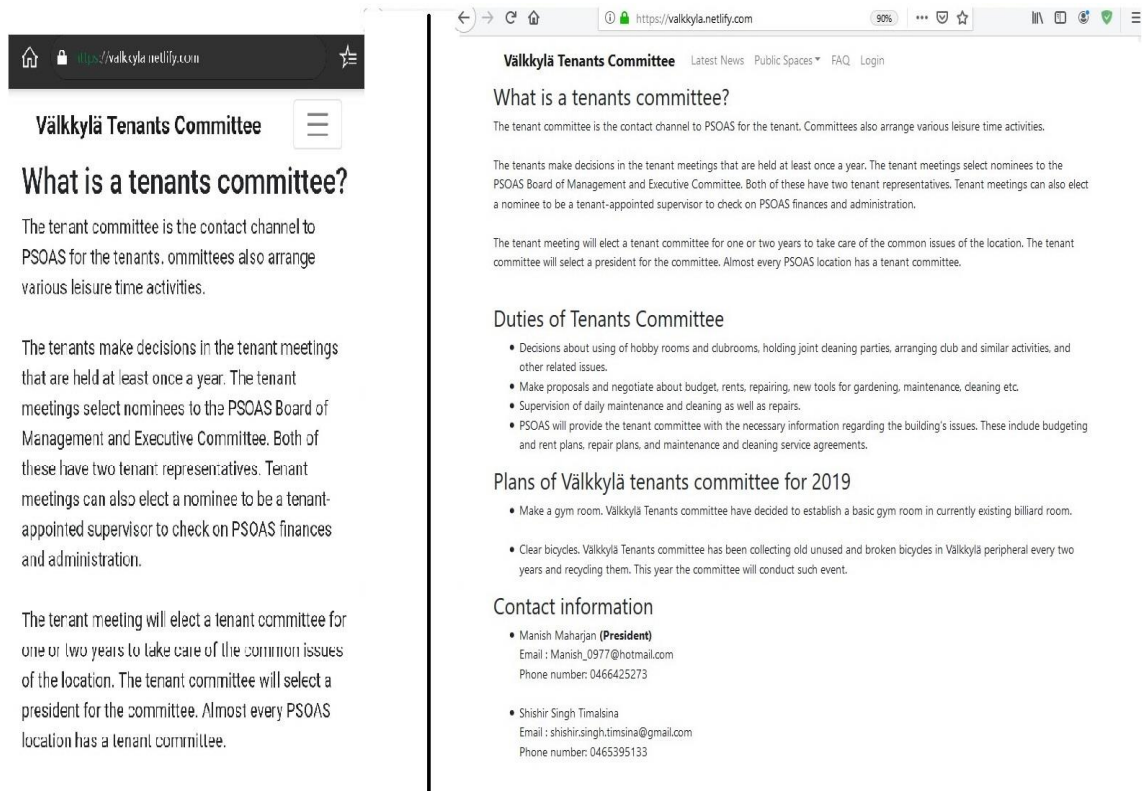


FIGURE 9. PWA Väikkylä tenant committee application in both mobile and desktop view

FIGURE 9, is the screenshot of the application developed in both mobile view and desktop view. The content is same in both, but the view is optimized differently for different screen sizes.

As a result of this work, a progressive web application was developed which can be used by tenants living in the Väikkylä community.

7 CONCLUSIONS

The main goal of this thesis study was implementing and evaluating a Progressive web application by developing a PWA for Vällkylä Tenant committee in Oulu which can be used by both the committee members and tenant living in the community.

To sum up, this project can be considered partially successful. Most of the features listed for the application are completed. The application still misses the feature of storing files and the action that can be performed by the committee is very limited to only being able to post news. Apart from this lack of features, the application is a PWA. It can be installed on the home screen. By making it a PWA the complexity of development and cost for running the application is very low. Since it is hosted on a web server changes can be made very simple manner compared to a native or hybrid mobile application. The application can be further developed by adding more features such as messaging, booking options of items and public chatroom for the tenant. At the current moment, the application is hosted in a free hosting service. It can be shifted to an appropriate hosting service and a custom domain name if the committee feels that the application makes a difference among the tenant.

After the study and completion of the application, lots of new knowledge are acquired about PWA. An important concept that needs to be understood is that PWA would not be replacing a native or hybrid mobile application. However, PWA does provide an alternative approach for developing mobile applications. PWA is cheaper and faster to develop due to its simplicity. But PWA is also very limited with the number of features it can provide and there is a limitation on their functionality. The main reason it is very popular among developers is once the application is developed, it can be deployed anywhere strategy. This is what makes PWA a very powerful tool and one of the prevailing methods of developing a mobile application. PWA will be more prominent among small business firms as where cost and time saving are very important.

REFERENCES

1. Clement, j. (2019, August 01). *Mobile app usage- Statistics & Facts*. Retrieved from Statista: <https://www.statista.com/topics/1002/mobile-app-usage/>
2. Deyan, G. (2019, March 28). *61+ Revealing Smartphone Statistics for 2019*. Retrieved from Techjury: https://techjury.net/stats-about/smartphone-usage/#How_Much_Money_Does_the_Mobile_Ecosystem_Generate
3. Gaunt, M. (2019). *Service Worker: an Introduction*. Retrieved from Developers Google: <https://developers.google.com/web/fundamentals/primers/service-workers/>
4. Ghimire, S. (2019, September 13). *Progressive Web App with ReactJS and Service Workers*. Retrieved from Medium: <https://medium.com/swlh/create-a-progressive-web-app-with-reactjs-and-service-workers-f79f14d2e449>
5. Halder, M. (2018, August 18). *What is a PWA and why should you care?* Retrieved from Medium: <https://blog.bitsrc.io/what-is-a-pwa-and-why-should-you-care-388afb6c0bad>
6. *Java Programming Language*. (2019). Retrieved from Oracle Java Documentation: <https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html>
7. *Lighthouse*. (2019). Retrieved from Developers google: <https://developers.google.com/web/tools/lighthouse>

8. Love, C. (2019, October 28). *Desktop Progressive Web Apps (PWA) - How they Work and Advantages over Native Apps*. Retrieved from Love2dev: <https://love2dev.com/blog/chrome-desktop-pwa/>
9. *Mobile Operating System Market Share Worldwide*. (2019). Retrieved from StatCounter: <https://gs.statcounter.com/os-market-share/mobile/worldwide>
10. Moth, D. (2013, March 12). *85% of consumer favour apps over mobile websites*. Retrieved from Econconsultancy: <https://econconsultancy.com/85-of-consumers-favour-apps-over-mobile-websites/>
11. Nath, D. S. (2017, March 24). *4 important points to know about Progressive Web Apps*. Retrieved from Medium: <https://medium.com/@deepusnath/4-points-to-keep-in-mind-before-introducing-progressive-web-apps-pwa-to-your-team-8dc66bcf6011>
12. Pandit, N. (2017, May 26). *What is Reactjs and Why should we use Reactjs?* Retrieved from C#Corner: <https://www.c-sharpcorner.com/article/what-and-why-reactjs/>
13. Patel, P. (2019, September 24). *Introduction to Google Firebase*. Retrieved from C#Corner: <https://www.c-sharpcorner.com/article/introduction-to-google-firebase/>
14. Pelser, J. (2017, December 05). *Why I like Netlify*. Retrieved from Jerrie Pelser: <https://www.jerriepelser.com/blog/why-i-like-netlify/>
15. Petrov, C. (2019, March 12). *52 Mobile vs. Desktop Usage Statistics For 2019 [Mobile's Overtaking!]*. Retrieved from Techjury: <https://techjury.net/stats-about/mobile-vs-desktop-usage/>
16. Priyesh, P. (2018, April 18). *What exactly is Node.js?* Retrieved from FreecodeCamp: <https://www.freecodecamp.org/news/what-exactly-is-node-js-ae36e97449f5/>

17. *Progressive Web Applications*. (n.d.). Retrieved from Wikipedia:
https://en.wikipedia.org/wiki/Progressive_web_applications
18. *Swift "The powerful programming language that is also easy to learn"*. (2019). Retrieved from Developer apple:
<https://developer.apple.com/swift/>
19. Takahashi, D. (2018, September 11). *Newzoo: Smartphone users will top 3 billion in 2018, hit 3.8 billion by 2021*. Retrieved from Venture Beat:
<https://venturebeat.com/2018/09/11/newzoo-smartphone-users-will-top-3-billion-in-2018-hit-3-8-billion-by-2021/>
20. *Techjury*. (2019). Retrieved from Techjury: <https://techjury.net/>
21. *What are the popular types and categories of apps*. (2019). Retrieved from Think Mobiles: <https://thinkmobiles.com/blog/popular-types-of-apps/>