

**Progressive web application**

**AON 10**

Jonathan Claessens

Projectomschrijving

In het tweede jaar van de opleiding Toegepaste informatica Applicatieontwikkeling wordt er voor het vak researchproject gevraagd om een webapplicatie te maken in team. Deze vraag werd gesteld door de stagecoördinator van het derde jaar, Marijke Willems. Het doel van het project is om een stagetool te maken die de *workflow* van de stagecoördinator kan vereenvoudigen. Daarom worden er regelmatig meetings met de klant georganiseerd om te kijken of het project voldoet aan alle eisen.

Het onderzoek gaat over het implementeren van een *progressive web application.* Hierin wordt er bekeken wat de voor- en nadelen zijn van een *progressive web application* ten opzichte van een normale webapplicatie met als doel de stagetool te verbeteren. Uit het onderzoek moet blijken of de webapplicatie offline gebruikt kan worden. Verder wordt er onderzocht of een *progressive web application* invloed heeft op de performantie. Uiteindelijk wordt er onderzocht wat de technische moeilijkheden zijn om een *progressive web application* te ontwikkelen en hoe de voordelen opwegen tegen de tijd die gespendeerd moet worden om het te ontwikkelen.

De klant wenst een gemakkelijkere manier om stageopdrachten goed te keuren en hierbij de studenten toe te kennen aan de gekozen stageopdracht. Op dit moment vullen de bedrijven een Google-formulier in. Deze data wordt doorgestuurd naar een Excelsheet waar de klant een overzicht krijgt van alle opdrachten onder elkaar. Verder heeft de klant communicatie met de verschillende reviewers die de stageopdracht bekijken. De reviewers oordelen of deze aansluit en voldoende is voor de opleiding. De communicatie met de reviewers gebeurt via e-mail.

De klant vindt dit onoverzichtelijk en heeft daarom gevraagd om een webapplicatie te maken die alle data van de Excelsheet verwerkt. De gegevens worden lokaal opgeslagen in een database. De klant wil elk jaar een back-up van de gegevens kunnen maken. Hiernaast zal er ook een *restore-*functie zijn om deze back-up naar een vorige versie terug te brengen. De communicatie tussen de docenten verloopt binnen de applicatie.

Om een duidelijk beeld te krijgen van de applicatie wordt er gebruikgemaakt van mock-ups en SUC (System Use Case) - *descriptions*. Daarnaast wordt er een *backlog* opgesteld waar alle userstory’s aan bod komen. Verder worden er ook sprints gemaakt waarin de userstory’sgeplaatst worden zodat elk teamlid weet aan welk deel gewerkt moet worden en hoeveel tijd hiervoor uitgestippeld is. Er wordt gewerkt met de Scrum Agile-methode. Dit wil zeggen dat er iedere week een stand-up meeting plaatsvindt waar besproken wordt wie waaraan bezig is en wie waaraan zal werken de komende week.

Voor de backend wordt er gebruikgemaakt van de programmeertaal Java. Hierbij wordt gedacht aan de synchronisatie van de applicatie met de Gmail-server. Javascript, HTML5 & CSS zijn de programmeertalen die gebruikt worden voor de frontend. Het framework Vue wordt mede gebruikt voor de frontend. Verder wordt er gebruikgemaakt van Adobe XD voor het maken van mock-ups. Ook draw.io wordt gebruikt voor het maken van het SUC-diagram. De sprintplanning, de *backlog,* deuserstory’s en desprintsworden gemaakt en beschreven in JIRA*.* De database die er gebruikt wordt is een *mySQL database.* Github is de tool die gebruikt wordt voor het uploaden van code: dit is handig voor het samenwerken aan dezelfde code.

Inhoudsopgave

[Projectomschrijving ii](#_Toc104747270)

[Inhoudsopgave iii](#_Toc104747271)

[Lijst van gebruikte figuren iv](#_Toc104747272)

[Lijst van gebruikte afkortingen v](#_Toc104747273)

[1 Onderzoeksvraag en hypothese 1](#_Toc104747274)

[1.1 Hoofdvraag 1](#_Toc104747275)

[1.1.1 Welke oplossing of meerwaarde biedt een *progressive web application* ten opzichte van een gewone webapplicatie bij de ontwikkeling van een stageapplicatie? 1](#_Toc104747276)

[1.2 Deelvragen 1](#_Toc104747277)

[1.2.1 Wat is een *progressive web application*? 1](#_Toc104747278)

[1.2.2 Kan de een PWA nog gebruikt worden als de verbinding wegvalt? 1](#_Toc104747279)

[1.2.3 Welke functionaliteiten biedt een PWA? Wat biedt het op gebied van installatie en hoe gaat een PWA daarmee om? 1](#_Toc104747280)

[1.2.4 Wat zijn de voor- en nadelen van een *progressive web application*? 1](#_Toc104747281)

[1.2.5 Wat zijn de technische moeilijkheden om een *progressive web application* te ontwikkelen? 2](#_Toc104747282)

[1.3 Hypothese 2](#_Toc104747283)

[2 Onderzoeksmethode 2](#_Toc104747284)

[3 Literatuurstudie 3](#_Toc104747285)

[4 Uitvoering 6](#_Toc104747286)

[4.1 Werking PWA 6](#_Toc104747287)

[4.2 Hoe maak je van een website, een PWA. 6](#_Toc104747288)

[4.2.1 Manifest.json 6](#_Toc104747289)

[4.2.2 Service worker 7](#_Toc104747290)

[5 Conclusie 11](#_Toc104747291)

[6 Bibliografie 12](#_Toc104747292)

Lijst van gebruikte figuren

[Figuur 1: PWA vs NATIVE APP 4](#_Toc104747293)

[Figuur 2:App Manifest 6](https://hogeschoolpxl.sharepoint.com/sites/2TIN-ResearchProjectApplicationDevelopment-AON10/Gedeelde%20documenten/AON%2010/AON10_taalversie_researchpaper_KH_aangepast.docx#_Toc104747294)

[Figuur 3: Service worker 7](#_Toc104747295)

[Figuur 4: Eventlistener 8](#_Toc104747296)

[Figuur 5: Link manifest 8](#_Toc104747297)

[Figuur 6: Web applicatie 9](#_Toc104747298)

[Figuur 7: Desktop applicatie 9](#_Toc104747299)

[Figuur 8: PWA online mode 10](#_Toc104747300)

[Figuur 9: PWA offline mode 10](#_Toc104747301)

Lijst van gebruikte afkortingen

|  |  |
| --- | --- |
| **Afkorting** | **Afkorting** |
| PWA | Progressive Web Application |
| SEO | Search Engine Optimization |
| SUC | System Use Cases |
| RAM | Random Access Memory |

# Onderzoeksvraag en hypothese

## Hoofdvraag

### Welke oplossing of meerwaarde biedt een *progressive web application* ten opzichte van een gewone webapplicatie bij de ontwikkeling van een stageapplicatie?

Een stageapplicatie moet eerst en vooral offline functionaliteiten hebben en moet anderzijds snel toegang hebben tot data, zelfs voor gebruikers met een trage verbinding. Er wordt onderzocht of een *progressive web application* deze functionaliteiten kan bieden en er wordt ook gezocht naar welke andere pluspunten het zou kunnen bieden betreffende een stageapplicatie.

## Deelvragen

### Wat is een *progressive web application*?

Wat houdt een *progressive web application* precies in? Wat is een PWA precies en hoe werkt het? Dit wordt onderzocht a.d.h.v. een *proof of concept*. Dit *proof of concept* is een kleine versie van een PWA dat wordt uitgewerkt om bekend te geraken met de werking van PWA’s. Zo kan er worden vastgesteld wat een PWA te bieden heeft alvorens het toe te passen op een grotere schaal in de volledige stageapplicatie. De proof of concept wordt uitvoerig besproken in de uitvoering van deze researchpaper.

### Kan de een PWA nog gebruikt worden als de verbinding wegvalt?

De applicatie moet volledig gebruikt kunnen worden zonder verbinding te hebben met het internet. Als een gebruiker een email wil versturen zonder verbinding te hebben, dan moet deze email toch aankomen bij de ontvanger zonder dat deze email verloren gaat. De email moet worden opgeslagen in een lokale database en zal nadien verstuurd moeten worden zodra het internet weer beschikbaar is. De gebruiker moet zich ook bewust zijn van het feit dat hij offline is. Dit moet worden weergegeven door een melding. Het hele offline gebeuren wordt onderzocht in de literatuurstudie en er wordt dieper op ingegaan in de uitvoering van deze researchpaper.

### Welke functionaliteiten biedt een PWA? Wat biedt het op gebied van installatie en hoe gaat een PWA daarmee om?

Is er de mogelijkheid om een PWA te installeren op een toestel? Blijft het dan nog steeds een webapplicatie of wordt het een native applicatie? Moeten er bestanden worden gedownload voor een installatie? Op deze vragen wordt een antwoord gezocht.

### Wat zijn de voor- en nadelen van een *progressive web application*?

Er wordt opzoek gegaan naar welke voordelen een PWA kan bieden in het ontwikkelen van een stageapplicatie. Daarentegen wordt er gekeken naar welke nadelen het met zich brengt zodat deze mogelijks vermeden kunnen worden. Er wordt niet enkel rekening gehouden of het mogelijk is om een PWA offline te gebruiken, maar er wordt ook onderzocht of een PWA data-efficiënt is. Verder wordt er aandacht gegeven waarom andere bedrijven hun website ombouwen tot een PWA. Dit wordt gedaan om conclusies te trekken uit projecten van anderen en het toe te passen in de stageapplicatie.

### Wat zijn de technische moeilijkheden om een *progressive web application* te ontwikkelen?

Komt er veel bij kijken om een PWA te ontwikkelen? Er wordt gekeken of er veel extra tijd moet worden gespendeerd om een webapplicatie *progressive* te maken en of die gespendeerde tijd de moeite loont t.o.v. de voordelen die het kan bieden.

## Hypothese

Een *progressive web application* zal een aantal meerwaardes bieden voor een stageapplicatie te ontwikkelen. In het bijzonder voor een webapplicatie offline zijn functionaliteit te laten behouden zal dit een pluspunt zijn. Eenmaal als de gebruiker de applicatie offline gebruikt zal er automatisch verbinding gemaakt worden met een lokale databank om de gegevens op te halen. Als er tijdens het offline gebruik een email verstuurd wordt, dan zal deze email ook lokaal worden opgeslagen en verstuurd worden wanneer het internet weer beschikbaar is. Verder wordt er verwacht dat de applicatie gebruiksvriendelijker gemaakt kan worden door de positieve aspecten die een *progressive web application* met zich meebrengt. Het implementeren van een *progressive web application* is moeilijker dan een *native* applicatie omdat er meer research zal moeten plaatsvinden.

# Onderzoeksmethode

De onderzoeksmethode voor deze researchpaper is een vergelijkend onderzoek.

Er wordt een vergelijking gemaakt tussen een gewonewebapplicatie en een *progressive web application.* Ook een *native* app wordt in mindere mate besproken en vergeleken.

Er zal ook in kleine maten gekeken worden naar welke oplossingen *progressive web applications* bieden voor andere projecten en welke bedrijven hiervan gebruikmaken. Dit is interessant om te zien hoe andere bedrijven de sprong maken naar een PWA en hoe dit hun applicatie kan verbeteren. De stageapplicatie die in ontwikkeling is kan dan worden vergeleken met *use cases* van andere projecten en op die manier kan worden bepaald welke aspecten ook een meerwaarde kunnen bieden voor deze stageapplicatie.

Er wordt gezocht naar bronnen op het internet om uit te zoeken hoe p*rogressive web applications* precies werken en hoe er met deze informatie een demo applicatie kan worden gemaakt. Het gebruiken van gratis tutorials op YouTube speelt hier ook een belangrijke rol.

# Literatuurstudie

Volgens Y. Ribbers is een *progressive web application* of PWA het volgende: “Een Progressive Web App (PWA) is een webapplicatie die beschikt over functionaliteiten die voorheen alleen waren weggelegd voor *native* desktop- of mobiele applicaties.” [1]

Er wordt gezegd dat een PWA een ervaring nabootst die gelijkaardig is aan die van een *native* applicatie. Hij omschrijft een *native* app als volgt: “Denk hierbij aan kenmerken van de ‘gewone’ apps die je vanuit de App Store of Google Play Store kunt installeren op je telefoon of tablet. Deze apps hebben push-notificaties, kun je offline gebruiken en ze hebben vaak een heel ander uiterlijk dan een website. Progressive Web Apps hebben deze functies ook en lijken dus heel erg op ‘normale’ of *native* apps.” [1]

Ze hebben technologieën gecombineerd om de gebruiker het gevoel te geven dat ze op een *native* app zitten: ”Een van deze nieuwe technologieën is de Service Worker. Een soort proxy, die tussen de server en de client draait om data te *cachen* en continu op de achtergrond kan luisteren naar events zoals push-notificaties.” Deze *service worker* zorgt ervoor dat de gebruiker de applicatie deels offline kan gebruiken en dat gebruikers niet elke keer de volledige app moeten laden. [1]

R. Hendriksen heeft de volgende definitie voor een PWA: “Het is ‘Progressive’ omdat het zichzelf *lazy-load* op basis van het gedrag van de gebruiker. Het is ‘Web’ omdat het is geschreven in de taal voor het web: HTML, CSS en JavaScript. Het is een ‘App’ omdat het code draait en installeert op het apparaat van de gebruiker.” [6]

Een PWA biedt heel wat voordelen. Jammer genoeg zit er hier en daar een nadeel aan. Om te beginnen kan een PWA als app en website gebruikt worden. Dit biedt meerdere mogelijkheden omdat zo de functionaliteiten van de browser en het apparaat zelf kunnen worden benut. Het kan dus de camera of locatie van de mobiele telefoon gebruiken. Alhoewel veel functies van het toestel kunnen worden gebruikt, moet er wel rekening worden gehouden met het feit dat het iets beperkter is dan de functionaliteiten van een *native* applicatie. Via een PWA is er geen toegang tot de contacten en de kalender. Wanneer een applicatie deze functies nodig heeft, is het beter om een *native* app te ontwikkelen in tegenstelling tot een PWA. [2]

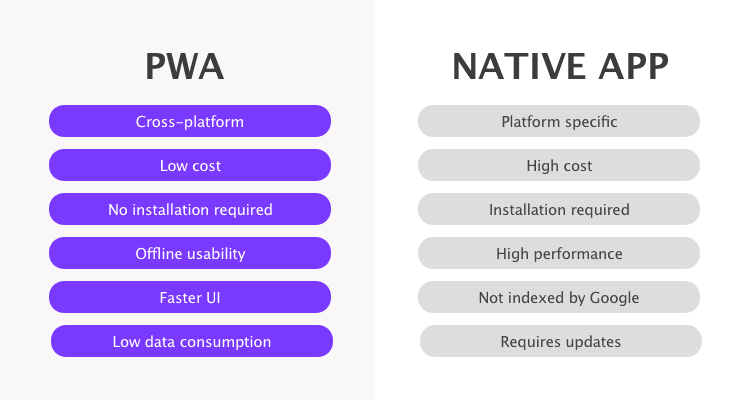
Er is geen installatie nodig omdat een PWA gewoon via de browser beschikbaar is. In tegenstelling tot een *native* app, die enkel toegankelijk is op het toestel zelf, zijn PWA’s via de browser beschikbaar op zowel computers, mobiele telefoons als tablets. Een PWA kan wel worden toegevoegd als een icoon op een toestel, maar hiervoor moet de gebruiker de website zelf eerst bezoeken. Dit wil zeggen dat de app niet vindbaar is in de appstores maar enkel via de browser. Het voordeel hiervan is dat er geen updates zijn omdat de PWA verbonden is met het web. Dus wanneer de gebruiker onlineverbinding heeft, begeeft hij zich op een app die up-to-date is. [2]

Een PWA heeft weinig opslagruimte nodig en gebruikt zelfs minder resources dan een *native* app. De performantie blijft zelfs op de meeste gebieden hetzelfde dan bij een *native* app. Maar toch ligt in het algemeen de performantie van *native* apps hoger. [2]

PWA’s bieden eengoede ondersteuning voor *Search Engine Optimization* (SEO), in tegenstelling tot *native* apps. Op deze manier kan de content van PWA’s gewoon worden gegoogeld. [2]

Het is zelfs zo dat het goedkoper is om een PWA te ontwikkelen dan een *native* app. Dit komt doordat het ontwikkelingsproces sneller is. Daarbovenop is het een ideale manier voor doorontwikkeling van een applicatie. [2]

Figuur 1 somt de voordelen van een *progressive web application* nog eens kort op t.o.v. een native app.



Figuur 1: PWA vs NATIVE APP

Een *progressive web application* wordt gebouwd met veel voorkomende webtechnologieën zoals HTML, CSS, JavaScript en WebAssembly. Het is gemaakt om te werken op elk platform dat gebruikmaakt van een webbrowser die de technologie ondersteunt. In 2021 worden eigenschappen van een PWA ondersteund door Google Chrome, Apple Safari, Firefox voor Android en Microsoft Edge. Firefox wordt niet ondersteund voor Desktop. Apple loopt achter op de ondersteuning van PWA’s. [3]

WebAssembly biedt een manier om verschillende talen code op native snelheid uit te voeren. Het is ontworpen op JavaScript aan te vullen en ernaast te draaien. Met behulp van de WebAssembly JavaScript API’s kunnen WebAssembly-modules in een JavaScript-app ingeladen worden en de functionaliteiten tussen beide gedeeld worden. Hierdoor kan er gebruik gemaakt worden van de prestaties en kracht van WebAssembly en de flexibiliteit van JavaScript in dezelfde apps. [7]

L. Morinigo zegt dat gebruikers meestal een website verwachten die enkel online werkt. Maar vaak komt een gebruiker in een scenario terecht waar er geen onlineverbinding is. PWA’s maken het mogelijk dat de website deels offline werkt door gebruik te maken van *service workers*. Een *service worker* is een script geschreven in JavaScript dat gebruikt wordt om internetverkeer te onderscheppen. Hierdoor kunnen snelle webpagina’s ontwikkeld worden en hebben gebruikers een goede offline-ervaring. *Service workers* hebben toegang tot netwerkverzoeken. Dit is gevaarlijk voor scriptinjectie en kan worden gebruikt om de hele site te vervangen door mensen met slechte bedoelingen. Daarom wordt er best gebruikgemaakt van HTTPS zodat iemand anders geen code kan injecteren in de webapplicatie. [4]

Er zijn ook heel wat grote bedrijven die de overgang hebben gemaakt naar een PWA. Voor deze literatuurstudie worden er drie besproken. Er wordt gekeken waarom er gekozen is voor een PWA en wat de voordelen zijn.

K. Grzybowksa heeft een kijkje genomen naar een aantal bedrijven die gebruikmaken van een PWA. Ze schrijft dat de mobiele versie van de e-commerce gigant AliExpress driemaal sneller groeit dan de webversie. Daarom heeft AliExpress besloten om een *cross-browser* PWA te ontwikkelen om het beste uit hun *native* applicatie te combineren met het bereik van een webapplicatie. Er worden hierdoor 104% meer nieuwe gebruikers aangetrokken over alle browsers. De conversiegraad heeft een stijging van 82% op iOS-toestellen. Tweemaal meer pagina’s worden bezocht per sessie per gebruiker over alle browsers. Ten slotte is er een stijging van 74% van de tijd dat de gebruikers spenderen per sessie over alle browsers. [5]

Twitter wordt grotendeels gebruikt op mobiele toestellen. Daarom ontwikkelde Twitter een PWA die sneller en meer data-efficiënt toegang tot de app kon bieden. De app laadt ook sneller voor gebruikers met een trage verbinding. De introductie van de Twitter PWA verhoogde het aantal pagina’s bezocht per sessie met 65% en het aantal verstuurde tweets met 75%. [5]

Trivago is een Duits bedrijf dat zich specialiseert in een hotelzoekmachine die prijzen vergelijkt. *Native* apps zijn niet ideaal voor gebruikers die reizen omdat die gebruikers vaak mobiele data willen besparen en dus niet graag zaken downloaden op hun mobiele telefoon. Daarom heeft Trivago een PWA gelanceerd. Het biedt hun offlinefunctionaliteiten en *push notifications.* De PWA biedt performantie gelijkaardig aan die van een *native* app. [5]

Het komt er dus op neer dat een *progressive web application,* of PWA afgekort, een toepassing is die via het web beschikbaar is op zowel een desktop als mobiele toestellen. Het wordt gemaakt met technologieën zoals HTML, CSS, JavaScript en WebAssembly. De PWA benut het volledige scherm waardoor het meer lijkt op een website dan op een app. Het brengt eigenlijk alle voordelen van een *native* app met zich mee maar dan in webversie.

Deze studies wijzen erop dat een *progressive web application* ideaal is voor de ontwikkeling van een stageapplicatie. Dit komt doordat een *progressive web application* offlinefunctionaliteiten en push notificaties biedt d.m.v. *service workers*.

Het is een trend dat grote bedrijven een overgang maken naar een *progressive web application* door tal van voordelen dat het met zich meebrengt voor de eindgebruikers. Voor vele bedrijven is er een aanzienlijke stijging op gebied van conversie en nieuwe gebruikers.

Uit het voorbeeld van Twitter kunnen we besluiten dat de *progressive web application* ervoor zorgt dat de app sneller en meer data-efficiënt werkt zodat de applicatie sneller laadt voor gebruikers met een slechte verbinding. Voor Trivago is het vooral de bedoeling dat hun gebruikers de applicatie offline kunnen gebruiken omdat hun klanten reizen. Deze voordelen zijn ook van belang bij een stageapplicatie die snel data moet kunnen inladen en offlinefunctionaliteit moet behouden. Er kan vanuit gegaan worden dat grote bedrijven met een groot kapitaal ook vertrouwen hebben in deze technologie om hun applicaties te verbeteren.

# Uitvoering

## Werking PWA

Een app wordt beschouwd als PWA als het:

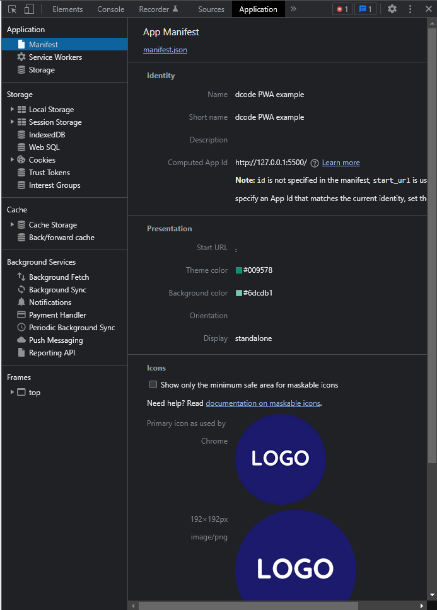
* Gebruik maakt van HTTPS of localhost
* Beschikt over een manifest.json bestand
* Beschikt over een service worker.
* Responsive, smooth en ‘app-like’ is.
* Een goede Progressive Web App zal in staat zijn om offline gebruikt te worden door kerndelen te cachen (HTML, CSS, JS, images…).

Om te controleren of een website effectief compatibel is als *Progressive Web Application* kan dit door te navigeren naar de lighthouse tab (bij chrome inspect), selecteer vervolgens ‘Progressive Web App’ en klik dan op ‘Generate report’. In dit verslag staat wat de website nog nodig heeft om zich als PWA te kunnen gedragen. Lighthouse is ook een gemakkelijke tool om de performance en SEO-score te testen van een website.

## Hoe maak je van een website, een PWA.

### Manifest.json

Om het mogelijk te maken om een website te installeren maken moeten we een "manifest.json” bestand toevoegen in de hoogste directory van de website. Dit bestand wordt gebruikt om bepaalde eigenschappen te definiëren. Dit zijn de standaard eigenschappen die toegekend kunnen worden aan de PWA applicatie:

* name: De volledige naam van de applicatie.
* short\_name: Bepaalde systemen zoals een app store willen de mogelijkheid hebben om een verkorte naam te kunnen weergeven.

Figuur 2:App Manifest

* Start URL: Hier staat gespecifieerd welke map de PWA moet gebruiken. Gebruik ‘.’ om de map te gebruiken waar het manifest.json-bestand zich in bevindt.
* Background color: Hier kan de kleur gespecifieerd zijn die gebruikt zal worden tijdens het laden van de applicatie
* Theme color: Hier kan de kleur gespecifieerd zijn die gebruikt zal worden in de statusbalk van de applicatie.
* display: Hier wordt ingesteld hoe de applicatie standaard wordt weergegeven. In het voorbeeld maken we gebruik van de ‘standalone’-waarde, dit zorgt ervoor dat de URL van de pagina wordt verborgen en dat de website meer aanvoelt als een applicatie.  
  <https://developer.mozilla.org/en-US/docs/Web/Manifest/display>
* icons: Hier wordt een lijst met icons meegeven die worden gebruikt in de applicatie. In deze lijst moet er per object een “src”, een “size” en een “type” eigenschap ingevuld zijn. “src” verwijst naar de locatie waar de icon zich bevindt en “size” naar de grootte van het icon en “type” het type van het icon. Het is ook belangrijk om bij minstens 1 van de iconen de “purpose” eigenschap toe te voegen zodat de browser weet dat alles buiten de ‘safe zone’ genegeerd mag worden. Zie <https://developer.mozilla.org/en-US/docs/Web/Manifest/icons>

### Service worker

Het is belangrijk om eerst te checken of dat de gebruikte browser service workers ondersteund. Dit kan gedaan worden door het volgende in index.js te plaatsen.

In deze if-blok wordt de serviceworker-file geregistreerd.  
Het is hierna de bedoeling dat het serviceworker bestand wordt aangemaakt in de hoogste map van de applicatie omdat de service worker naar alle files moet kunnen kijken onder zich.

Afbeelding met tekst, scherm, schermafbeelding

Automatisch gegenereerde beschrijving

Figuur 3: Service worker

In het service worker bestand kunnen EventListeners worden toegevoegd. In ons voorbeeld wordt er gebruik gemaakt van twee EventListeners. Bij de eerste EventListener wordt de install-EventListener gebruikt. De parameter die wordt meegegeven wordt dan vervolgens gebruikt om een cache te openen en te registreren wat er allemaal lokaal moet worden opgeslagen.

Bij de tweede EventListener wordt de fetch-EventListener gebruikt. Met de parameter die hierbij wordt meegegeven wordt er gekeken of er gefetched wordt naar iets dat in de cache staat en of dit al opgehaald is of niet.

Afbeelding met tekst, schermafbeelding, scherm

Automatisch gegenereerde beschrijving

Figuur 4: Eventlistener

Als er wijzigingen worden uitgevoerd in de Service Worker is het belangrijk om in het applicatie tabblad van de browser de nieuwe service worker aan te klikken en hierna op ‘skip waiting’ te klikken.

Het is ook belangrijk om de manifest.json file te linken in de HTML-pagina, dit wordt gedaan door *‘<link rel=”manifest” href=”manifest.json”>’* toe te voegen in de *head*-blok. Dit kan gecontroleerd worden door de html pagina in de browser te inspecteren, naar applicatie te navigeren, op de manifest tab te klikken. Hier zou de manifest.json file terug te vinden moeten zijn, samen met de eigenschappen die hier ingesteld staan.

Het is ook belangrijk om een theme-color te specificeren in ons *head*-blok. Gebruik dezelfde kleur gespecificeerd bij de “theme\_color” eigenschap in het manifest.json-bestand.

Specificeer ook een link element voor het ‘apple-touch-icon’ en gebruik hier de icon voor. ‘*<link rel=”apple-touch-icon” href=”images/logo.png”>’* . Dit is het icon dat verschijnt op het ‘home screen’ als een gebruiker de PWA installeert op zijn toestel. <https://web.dev/codelab-apple-touch-icon/>

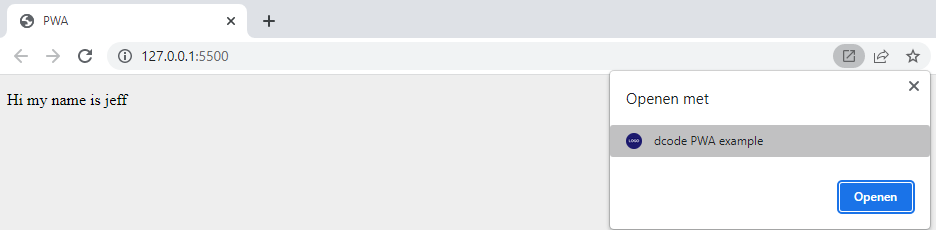
Afbeelding met tekst

Automatisch gegenereerde beschrijving

Figuur 5: Link manifest

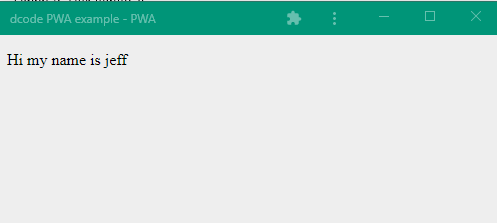
Na al deze stappen is het mogelijk om de website te installeren en het offline als programma te gebruiken.

De demoapplicatie toont aan dat de PWA gebruikt kan worden in een webbrowser. Figuur 6 geeft dit weer.



Figuur 6: Web applicatie

Daarnaast toont de demoapplicatie ook aan dat een applicatie toegevoegd kan worden aan het startscherm van het toestel. Hiervoor hoeft niets gedownload te worden. Dit wordt mogelijk door de manifest.json die in de code van de demoapplicatie is toegevoegd. Wanneer een applicatie toegevoegd wordt op een toestel dan blijft dit nog steeds een browserapplicatie. Dit wordt weergegeven in figuur 7.



Figuur 7: Desktop applicatie

Als de gebruiker online is wordt de data ingeladen en getoond. Zie figuur 8.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Figuur 8: PWA online mode

Wanneer er geen internetverbinding is wordt er geen foutmelding getoond en werkt de PWA nog perfect. Als de gebruiker offline is zal de data die getoond wordt mogelijks niet up-to-date zijn. Dit komt doordat de 3 benodigde files niet ingeladen kunnen worden (rode markering). Een verstuurde mail zal wel gecachet worden totdat er terug verbinding is. Zie figuur 9.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Figuur 9: PWA offline mode

# Conclusie

In dit vergelijkend onderzoek is er opzoek gegaan naar een oplossing op de vraag of *progressive web applications* een meerwaarde kunnen bieden ten opzichte van een gewone webapplicatie voor de ontwikkeling van een stageapplicatie die offlinefunctionaliteiten biedt. Uit nader onderzoek is gebleken dat een *progressive web application* de meest logische keuze is voor de ontwikkeling van een stageapplicatie die gebruikmaakt van offlinefunctionaliteiten. De stageapplicatie moet werken in een browser, maar moet offline nog steeds bruikbaar blijven.

Deze studie heeft uitgewezen dat *native* apps het meeste van hun functionaliteiten behouden als de applicatie offline is. PWA’s combineren de offlinefunctionaliteiten van de native apps en gieten dit in een webversie. De stageapplicatie zal moeten synchroniseren met de lokale database en de gebruiker moet berichten kunnen sturen naar reviewers. Het gebruik van de applicatie mag niet in het gedrang komen wanneer de verbinding wegvalt en het mag zeker niet crashen. *Progressive web applications* vegen al deze problemen van de tafel. Door middel van *service workers* zal de stageapplicatie grotendeels zijn functionaliteit behouden als een gebruiker offline is. Zo kan de stagecoördinator een email sturen naar een reviewer, zelfs wanneer hij offline is. Deze mail zal tijdelijk gecachet worden en verstuurd worden wanneer er terug verbinding is. De demoapplicatie heeft aangetoond dat de gegevens nog steeds beschikbaar blijven ook als de gebruiker offline is.

De demoapplicatie toont ook aan dat een PWA geen installatie nodig heeft. De gebruiker kan naar een PWA surfen en heeft daar de mogelijkheid om de PWA aan het startscherm van het apparaat toe te voegen. Dit kan door middel van het installeerknopje in de zoekbalk van je browser. Dit bootst het gevoel van een *native* app na en maakt het ook makkelijk in gebruik, maar behoudt de functionaliteit van een website.

De stageapplicatie zal dus altijd up-to-date zijn zolang er verbinding is. Het is niet mogelijk om een PWA te installeren via de appstore, maar dat is niet van belang voor de stageapplicatie die enkel een webversie nodig heeft. Een native applicatie die op een toestel geïnstalleerd wordt kan veel RAM-geheugen verbruiken. Doordat een PWA gedownload kan worden neemt het maar weinig geheugen in. Alhoewel PWA’s minder resources gebruiken, zijn native applicaties wel nog vaak iets performanter.

Verder is het een groeiende markt. PWA’s worden steeds populairder en de ontwikkeling van dit soort applicaties wordt meer en meer ondersteund. Opmerkelijk is de overgang die bedrijven massaal maken op gebieden van hun applicaties om te zetten naar een PWA. Dit komt doordat PWA’s heel wat kunnen bieden voor hun eindgebruikers omdat het unieke kenmerken heeft dat gewone of native webapplicaties niet kunnen evenaren.

# Bibliografie

|  |  |
| --- | --- |
| [1] | Y. Ribbers, „Emmerce,” [Online]. Available: https://www.emerce.nl/achtergrond/de-progressive-web-app-alles-wat-je-moet-weten. [Geopend 31 maart 2022]. |
| [2] | Onbekend, „DTT,” [Online]. Available: https://www.d-tt.nl/artikelen/pwa-progressive-web-apps-voordelen-nadelen. [Geopend 30 maart 2022]. |
| [3] | Onbekend, „Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/Progressive\_web\_application. [Geopend 31 maart 2022]. |
| [4] | L. Morinigo, „Medium,” [Online]. Available: https://medium.com/samsung-internet-dev/pwa-series-service-workers-the-basics-about-offline-a6e8f1d92dfd. [Geopend 25 maart 2022]. |
| [5] | K. Grzybowksa, „Divante,” [Online]. Available: https://www.divante.com/blog/companies-that-use-pwa. [Geopend 2 maart 2022]. |
| [6] | R. Hendriksen, „frankwatching,” [Online]. Available: https://www.frankwatching.com/archive/2020/03/10/pwa-progressive-web-app-voordelen-nadelen/. [Geopend 22 05 2022]. |
| [7] | Onbekend, „WebAssembly,” [Online]. Available: https://developer.mozilla.org/en-US/docs/WebAssembly. [Geopend 25 May 2022]. |