Security 2 Uitleg / Samenvatting

Gemaakt door Bastiaan van der Plaat (0983259)

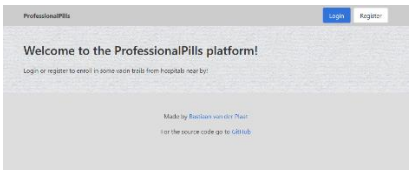
Voor het vak security 2 moesten we een beveiligde website maken met backend systeem voor het fictieve bedrijf Professional Pills. Dit is het verslag van wat ik heb gemaakt en hoe het werkt. Hier staat ook in beschreven welke beveiligingsmaatregelingen ik heb genomen.

Het systeem

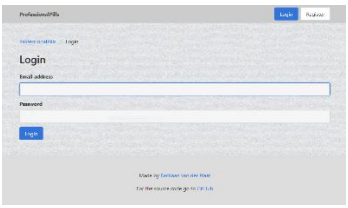
Ik heb het complete systeem gemaakt in PHP met het Laravel webframework aangezien ik daar al eerder mee had gewerkt. Ik kan met deze tools vrij snel iets in me kaar flansen. De site is te bereiken op het domein: professionalpills.ml. Dit is een gratis domein onder het TLD van het Afrikaanse land Mali van Freenom, leuk toch. Ook heb ik een publieke GIT repo waar alle source code staat als je die wilt doorlezen: github.com/bplaat/professionalpills.

De website (user part)

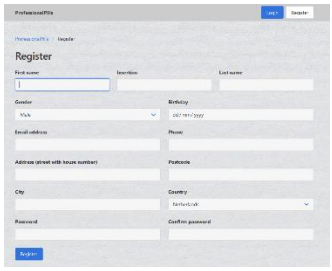
Hier ga ik je kort uitleggen hoe het systeem werkt en aan de hand van screenshots je een korte tuturial geven. Als je naar de website gaat krijg je de optie om jezelf in te loggen of om je te registreren. Als je al bent ingelogd onthoud hij je sessie via een cookie zodat je niet steeds opnieuw hoeft in te loggen.



De home pagina



De inlog pagina

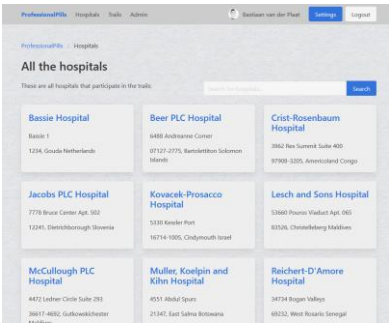


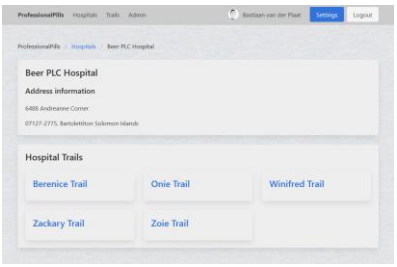
De registereer pagina

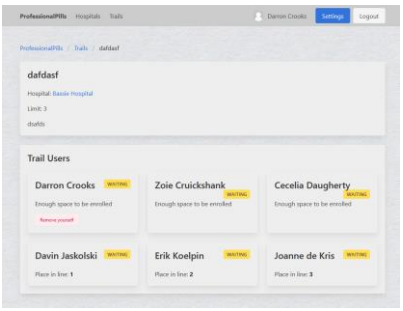
Als je bent ingelogd of je hebt een account aangemaakt dan kom je weer op de home pagina. Op de home pagina staat het lokale ziekenhuis waar je aan verbonden bent. Je kan daarop klikken en je inschrijven voor een onderzoek. Maar je kan ook deelnemen aan onderzoeken van andere ziekenhuizen.

Aan de bovenkant van bij het menu zie je nu twee opties verschijnen Ziekenhuizen en Medische Onderzoeken. Als je naar ziekenhuizen gaat zie je een lijst van alle ziekenhuizen die er zijn met het adres er onder zo kan je zelf kiezen welk ziekenhuis bij je past!

Als je dan vervolgens op een ziekenhuis klikt ga je naar de specifieke pagina van dat ziekenhuis. Daar kan je dan alle lopende medische onderzoeken zien die dat ziekhuis aanbied. Je kan dan op een onderzoek klikken en jezelf inschrijven. Elk onderzoek heeft een maximum aantal inschrijvingen als er meer mensen al staan ingeschreven kom je automatisch in de wachtrij terecht, dit word weergeven.

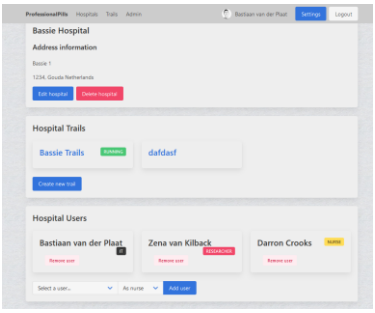


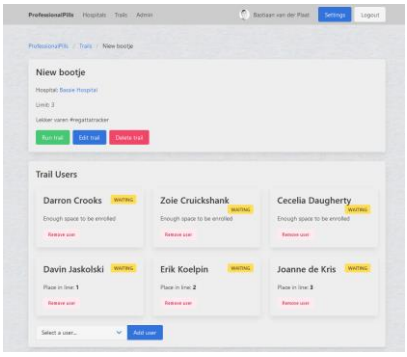


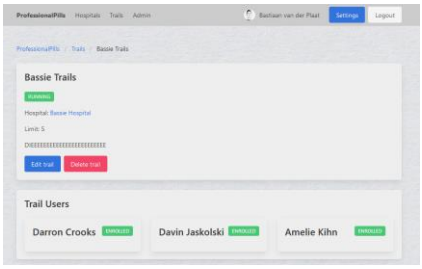


De ziekhuizen pagina *De pagina van een specifiek ziekhuis* *Een trail pagina met 3 wachtende mensen*

Op de website kun je dus mensen toevoegen aan onderzoeken maar je kan ook mensen koppelen aan ziekhuizen met een rol. Dit zijn de verschillende rollen: Vervoerder, Verpleegkundige, Doctor, Researcher en IT. Hoe hoger de rol hoe meer je dingen kan aan passen op de zieken huis pagina. Alleen mensen met de IT rol kunnen mensen toevoegen, rol veranderen en verwijderen van een ziekenhuis. Mensen die een researcher zijn kunnen onderzoeken aanmaken, starten, aanpassen en verwijderen. en ook mensen toevoegen aan het onderzoek. Als een onderzoek word gestart kan je je niet meer inschrijven voor het onderzoek. En de mensen die niet in de wachtrij stonden worden automatisch ingeschreven en doen mee met het onderzoek.

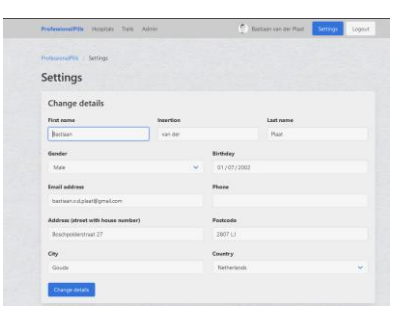


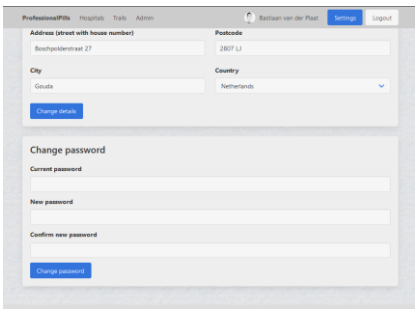




Ziekenhuis pagina met extra *Onderzoek pagina met extra* *Een gestart onderzoek*

Als laatste kan je als je bent ingelogd ook nog je eigen informatie updaten of je wachtwoord veranderen door naar de settings pagina te gaan.

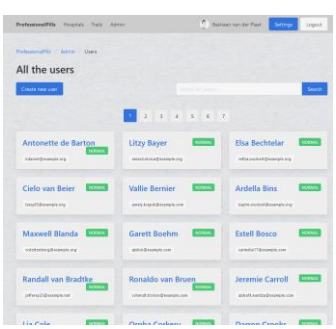


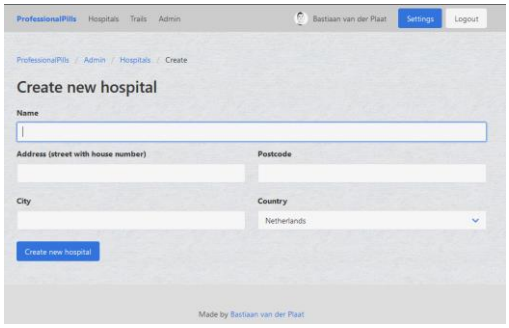


De settings pagina

De website (admin part)

Het hierboven uitgelegde gedeelte is het user gedeelte elke user heeft ook nog een aparte rol die normaal of admin kan zijn. Als je zo'n super admin account hebt komen er nog een aantal pagina's beschikbaar waar je alles kan doen wat je met een normaal account kan doen als je een IT rol had bij een ziekenhuis. En je kan ook nog users beheren en nieuwe ziekenhuizen toevoegen aan het systeem.





Alle users pagina *Nieuw ziekenhuis aanmaken pagina*

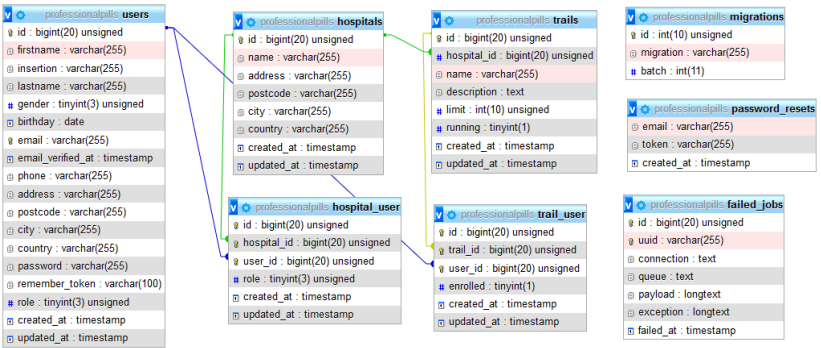
Deze pagina's zijn natuurlijk erg krachtig en moeten goed beschermd worden. Dit word dus gedaan door die extra rol property die op de user model zit.

De REST API

Ik heb ook nog een REST API gemaakt waarmee je alles van de site kan uitlezen en ook nog mensen kan toevoegen aan onderzoeken. Zo kan in theorie ook bijvoorbeeld een native Android / iOS app gemaakt kunnen worden die dan met die REST API communiceert. Ik heb de API niet beveiligd met certificaten omdat ik dit een lelijke manier vond en doordat ik dus toch al een punt misloop omdat ik het niet op de goede manier wouw beveiligen heb ik hem helemaal niet beveiligd 🎉. De API stuurt gewoon alles in JSON terug en is searchable en gepaged zodat je altijd max 20 results krijgt. In een markdown bestandje heb ik een beetje documentatie geschreven over hoe het werkt:

github.com/bplaat/professionalpills/blob/master/docs/api.md

ERD voor de duidelijkheid

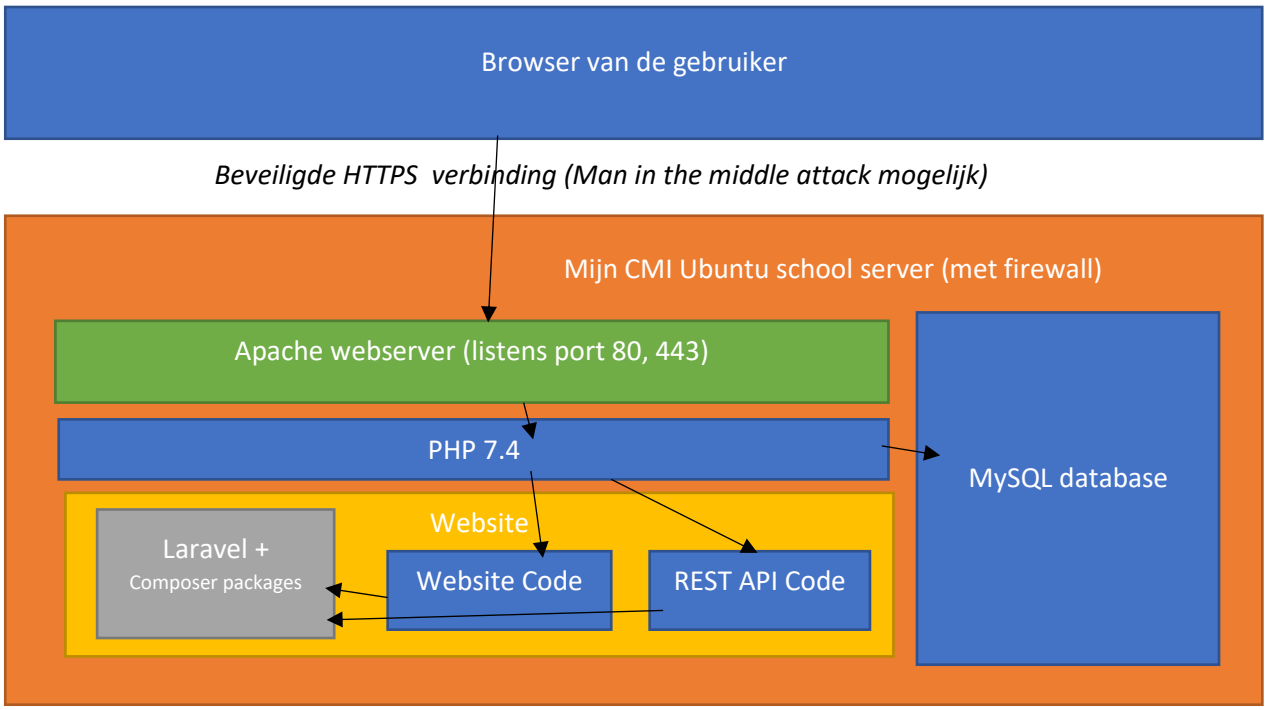


Je hebt natuurlijk de users, hospitals en trails tabellen. En de koppel tabellen tussen hospitals users en trails users. De laatste 3 tabellen zijn van Laravel. En de password_resets en failed_jobs tabellen worden helemaal niet gebruikt.

De beveiligingen maatregelen

- De website is beveiligd met een Let's Encrypt HTTPS certificaat, dus officieel veilig!!!
- Op alles wat je op de website doet moet je de goede rechten hebben. Je kan die rechten krijgen door in te loggen met een account met de juiste rechten. Of die rechten krijgen van een persoon die al wel die rechten heeft.
    - Dus om een ziekenhuis aan te passen moet je een IT-er zijn bij dat ziekenhuis of een Admin account hebben
    - Om een onderzoek aan te maken of aan te passen moet je minstens een researcher zijn bij dat ziekenhuis
    - ~~Maar dit maak allemaal niet zo heel veel uit aangezien de REST API compleet open en bloot leesbaar is maar dat zien we even door de vingers…~~
- Je blijft ingelogd door een sessie cookie maar om je wachtwoord aan te passen moet je wel eerst je oude wachtwoord intypen.
- De admin gebruiker kan ook het account van mensen hijacken door met een druk op de knop in te loggen op iemand anders account zonder wachtwoord te hoeven indrukken.

Threat model



Risicoanalyse

Problem & Rating	**Man-in-the-middle-attack op de website (7/10 score)**
Damage potential	Men zou via een man-in-the-middle-attack code kunnen injecteren in de site om zo andere computers te hacken.
Reproducibility	Het is mogelijk als de site puur op HTTP werkte om een man-in-the-middle-attack uitvoeren maar de site is beveiligd met HTTPS.
Exploitability	Dit is niet mogelijk want de site is beveiligd met HTTPS en deze aanval is dus ook niet mogelijk. Ook worden alle HTTP verbindingen automatisch geredirect naar een HTTPS verbinding.
Affected users	Alle gebruikers van de site zouden in theorie kwetsbaar zijn als ze op een onveilige verbinding zitten zoals in een café of restaurant waar mensen een fake sterker WIFI netwerk kunnen opzetten.
Discoverability	Men moet weten dat deze site bestaat en dat hij veel gebruikt wordt en dat is hij niet.

Problem & Rating	**Geen input validatie en SQL injectie gevaar (9/10 score)**
Damage potential	Als de site geen input validatie zou gebruiken zou een kwaadwillende SQL codes bij de input velden kunnen zetten en zo de complete site kunnen hacken.
Reproducibility	Dit is mogelijk als de site geen input validatie gebruik, gelukkig word elk input veld van de site streng gevalideerd.
Exploitability	Het zou een groot gevaar zijn als dit mogelijk was aangezien de gehele site hierdoor aan te vallen was.
Affected users	Alle gebruikers zijn kwetsbaar want alle informatie uit de database kan gestolen worden.
Discoverability	Dit is een van de eerste dingen die hackers uitproberen op een site dus als het mogelijk is het vrij snel bekend en wel verspreid.

Problem & Rating	**Bekende bugs / expliots in de software stack die ik gebruik (5/10 score)**
Damage potential	De website maakt gebruik van verschillende software waar natuurlijk ook bugs in kunnen zitten. Een kleine handgreep aan de software die we gebruiken: Apache, PHP, MySQL, Laravel…
Reproducibility	Als er een bekende exploit is de is er een kans dat deze ook op deze site gebruikt kan worden.
Exploitability	De kans dat dit gebeurd is trouwens wel erg laag en ik geloof dat in mijn code eerder bugs zitten dan in de code van deze standaard open-source projecten.
Affected users	Alle gebruikers zijn kwetsbaar want alle in theorie kan alle informatie uit de database kan gestolen worden.
Discoverability	Deze open-source software wordt goed onderhouden dus het vinden en / of maken van zo'n bug is erg lastig.