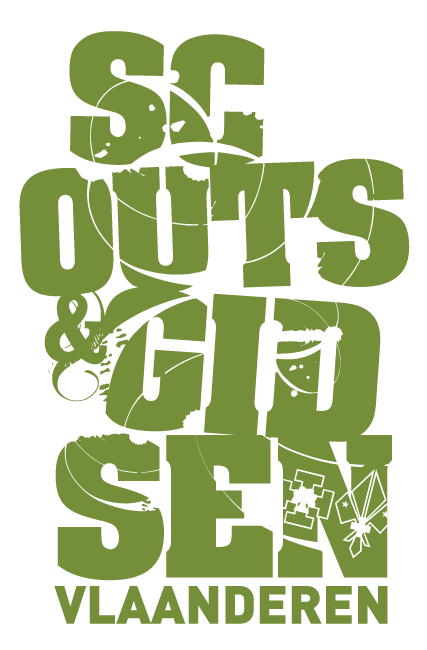
|  |  |
| --- | --- |
| FIIW | Afbeelding met tekst, illustratie  Automatisch gegenereerde beschrijvinglogo FIIW.png |

Cloud Computing & Toepassingen

Project: Dashboard Scouts & Gidsen Vlaanderen

Datum



+



Auteurs: Lowie Deferme & Siemen Vandervoort

Groep: Master Elektronica-ICT

Prof: Kris Aerts & Stijn Schildermans

# Doelstellingen en motivatie project

Lowie is al een tijdje hoofdleiding van de Scouts en moet dus wekelijks activiteiten maken en bijhouden. Deze activiteiten moeten dan weer gecommuniceerd worden met de leden en andere leiding. Bij elke activiteit horen enkele kenmerken zoals locatie, tijdstip en het weerbericht voor die dag. Voor sommige activiteiten moeten leden zichzelf inschrijven zodat de leiding op voorhand weet hoeveel kinderen er komen. Dit geldt voor alle Scoutsgroepen van Vlaanderen. Een toepassing gebaseerd op Laravel maakt het bijhouden van activiteiten veel gemakkelijker. Alles wordt weergegeven in een overzichtelijk interface met duidelijke tabbladen en tabellen.

# Realisaties

Het project is gemaakt aan de hand van het Laravel Framework en is ontwikkeld in vier verschillende Docker containers:

* Het Laravelproject
* De MariaDB database voor de gebruikers
* De SQLITE database met SOAP interface voor de activiteiten
* De SOAP service voor info over het weer.

Op deze manier is het gemakkelijk op het project op te starten vanuit de docker-compose.yml in de frontend repo: ‘docker compose up’. Het project zal op elke computer op dezelfde manier gestart worden wat het gemakkelijk maakt om er met meerdere mensen aan te werken en het project te tonen aan derden.

De website van het Laravelproject bevat verschillende tabbladen:

* Home

Dit is de startpagina en toont de Facebookfeed van Scouts & Gidsen Vlaanderen.

* Login

Op deze pagina kan je inloggen met Google of Facebook, het project maakt gebruik van ‘Socialite’ om authentication te doen met OAth providers. Leden kunnen dus gemakkelijk inloggen met hun e-mail adres of Facebook account. De gebruikers worden opgeslagen in de MariaDB database en moeten door een admin toegelaten worden tot de activiteitenpagina. Op deze manier kunnen er geen derden toegang krijgen tot de activiteiten.

* Logout

Met deze knop kan je jezelf uitloggen.

* Activiteiten

Deze pagina haalt de activiteiten op uit de database (via SOAP) en toont de extra informatie. De pagina vraagt toegang tot je locatie omdat deze nodig is voor enkele API’s. Als eerste toont de pagina de route van jouw locatie naar de activiteit, samen met de tijdsduur en afstand. De route wordt opgehaald van een zelfgemaakte REST-API met Python. De service is terug te vinden op Heroku. De pagina geeft ook het weerbericht weer wat opgehaald wordt via een REST service van ‘weerlive.nl’. Een SOAP service geeft wat extra informatie over het weer gebaseerd op de temperatuur en de kans op regen. Leden kunnen zichzelf inschrijven voor een activiteit waarna hun ID opgeslagen wordt in de MariaDB database. Een zelfgemaakte REST-API geeft op basis van je locatie, de datum het uur van zonsopgang of zonsondergang terug, dit wordt dan gebruikt om te bepalen of het tijdens de activiteit donker is. Na zonsondergang is het donker en verwacht de leiding dat ouders hun kinderen uit veiligheid in het Scoutslokaal komen ophalen.

* Inschrijvingen (na inloggen als admin)

De pagina met inschrijvingen toont de lijst met activiteiten en als je een activiteit aanklikt worden de ingeschreven leden getoond.

* Admin (na inloggen als admin)

Via de admin pagina kunnen admins activiteiten toevoegen en verwijderen. Ze kunnen ook andere leden hun admin en toegangsrechten aanpassen.

* Contact

De contact pagina voert een POST-request uit naar ‘formsubmit.co’. Deze website verstuurt de ingevulde velden door naar het ingegeven e-mail adres.

# Project opstarten

1. Zorg voor onderstaande mappenstructuur via het clonen van de andere repositories.  
   Rootfolder

* Sgvd-activity-db
* Sgvd-frontend
* Sgvd-weather

1. Voer volgende commands en tasks uit:

* SOAP service met weer info builden vanuit sgvd-weather met ‘docker-build: release’ task in Visual Studio Code (Terminal -> Run Task)
* SOAP service met de activiteitendatabase builden gelijkaardig als hierboven.
* ‘docker compose up’ en in de Laravel container php artisan migrate:refresh en php artisan db:seed

1. Ga naar http://localhost:3000/start
2. Afsluiten

* ctrl -c om de containers te stoppen.

# Documentatie en links

* SQLITE database met activiteiten

Is een SOAP service en heeft dus WSDL

* SOAP weerinfo

Eigen documentatie: <https://sgvd-maps.herokuapp.com/documentation_weather>

* REST API routebeschrijving

Website: <https://openrouteservice.org/>

Eigen documentatie: <https://sgvd-maps.herokuapp.com/documentation>\_maps

* REST API zonsondergang/zonsopgang

Eigen documentatie[[1]](#footnote-1): <https://sgvd-sunset.herokuapp.com/>

* REST API weerbericht

<http://weerlive.nl/delen.php>

* Docker

<https://docs.docker.com/>

* Project files

<https://github.com/sgv-dashboard> (extra readme aanwezig bij de frontend)

1. De werking van de API key staat hierin niet uitgelegt wegens tijdsgebrek [↑](#footnote-ref-1)