



UNIVERSITEIT VAN AMSTERDAM

PROJECT WEBDB

Florum

2 februari 2018

Studenten:

Skip Harks
12002038

Robin Wacanno
11741163

Chris Bras
11917318

Bodi Boelé
11904089

Tutor:

W.R.J. Vermeulen

Practicumgroep:

WebDB2018IN01

Docent:

Robert Belleman

Cursus:

Webprogrammeren en Databases

Vakcode:

5062WEDA6Y

Inhoudsopgave

1	Introductie	3
2	Ontwerp en Implementatie	3
2.1	Server-opzet	3
2.2	Dubbele code	3
2.3	Beveiliging	4
2.4	Tokens en e-mail	4
2.5	Captcha's	4
2.6	Database	4
2.7	Gravatar	5
2.8	Markdown	6
2.9	Registreren en verificatie	6
2.10	User pagina	6
2.11	Karma	7
2.12	Beheerder Tools	7
2.13	Inbox en Notificaties	7
2.14	Zoekfunctie	8
2.15	Thread-pagina	8
2.16	Lay-out	9
3	Resultaten	10
4	Discussie	13
4.1	Bronvermeldingen	13

1 Introductie

Het Florum

Deze website dient als een discussieforum waar alle plantenliefhebbers samen kunnen komen om te converseren en vragen te stellen over al wat maar met flora te maken heeft.

De website moet alle basisfunctionaliteit van een forum bevatten, denk aan: threads maken, reacties plaatsen, accounts maken en kunnen onderhouden. Er moeten beheerders zijn die het forum kunnen onderhouden zonder handmatig de database in te hoeven gaan.

Het Florum bevat tevens wat extra functionaliteiten die diens gebruik aangenamer maken. Een voorbeeld van deze extra functionaliteit is de inbox, hierin kunnen gebruikers berichten ontvangen over de status van hun threads.

2 Ontwerp en Implementatie

Hoewel dit misschien twee aparte onderdelen lijken, is tijdens het project gebleken dat deze in vele gevallen hand in hand gaan. Hierom zullen een aantal aspecten van de site en van de werkwijze vanuit zowel een ontwerp- als een implementatie-oogpunt belicht worden.

2.1 Server-opzet

Het was van belang om ontwikkeling aan dit project met een goede basis te starten. Dit begon bij het nadenken over de manier waarop de bestanden van de website in mapjes geordend zouden worden en hoe deze vervolgens vanaf de server aan de gebruiker gepresenteerd dienden te worden. Allereerst moest de `.git`-map — welke in de *root* van het project zat wegens het gebruik van *git* om bestanden op de server te laden — voor bezoekers van buiten verborgen worden. Er waren twee manieren waarop dit gedaan kon worden: de map kon met behulp van een `.htaccess`-bestand worden afgeschermd of het hosten van de website werd niet gedaan vanuit de *root* van het project, maar vanuit een hoger-gelegen map. Voor dit tweede werd gekozen omdat dit zowel makkelijker te implementeren was als een robuustere oplossing leek. Het hosten van de *web root* werd dus gedaan vanuit de map `www/`. Onderdelen als *CSS*-, *JavaScript*-bestanden en plaatjes werden respectievelijk in de mappen `static/css/`, `static/js/` en `static/images/` gezet. Om deze mappen voor externe bezoekers bereikbaar te maken werd de map `static/` binnen de *virtual host* van de *Apache*-configuratie als een alias gehost op `[domein]/static`. De achterliggende *PHP*-bestanden die niet vanuit *AJAX* aangeroepen dienden te worden en dus ook niet vanaf het web beschikbaar moesten zijn werden in de map `php/` gezet. Als laatste werden de menubalk en *footer* van de pagina's, als zijnde veelvoorkomende elementen, in de map `elements/` gezet.

2.2 Dubbele code

Waarom er voor de zojuist genoemde pagina-onderdelen een aparte map was heeft naast de roep om orde alles te maken met het voorkomen van herhaling in de code. Zoals misschien wel te begrijpen is, heb je bij een website bestaande uit meerdere pagina's op ieder van deze pagina's bijvoorbeeld een kopie van het menu nodig. Nou kun je in ieder *PHP*- of *HTML*-bestand datzelfde menu kopiëren, maar dit zou herhaling en problemen bij het willen maken van aanpassingen opleveren. Daarom is gekozen om de zich in het `elements`-mapje bevindende elementen met behulp van *PHP* in te voegen. Ook achter de schermen is zowel bij vaak gebruikte stukken *PHP*- als *JavaScript*-code geprobeerd deze in één enkel bestand te zetten, zodat deze vervolgens in andere bestanden kan worden gevoegd. In het geval van *JavaScript* was dit een ietwat minder elegant proces, omdat er binnen deze taal geen universeel ondersteunde manier is om bestanden in andere bestanden op te nemen. De wijze waarop dit is opgelost is door het benodigde bestand in het *HTML*-document te laden

voorafgaand het bestand met de code die het vereiste.

In de code is ook in de *CSS* op een zekere manier dubbele code voorkomen. Er is bij het gebruik van veel voorkomende kleuren gebruik gemaakt van de `var()`-functie in *CSS*. Deze stelt je in staat eenmalig een kleur aan een variabele te koppelen, waarna vervolgens die variabele op de benodigde plekken gebruikt kan worden. Op deze manier is het wijzigen van kleuren ook vele malen eenvoudiger.

2.3 Beveiliging

Het is voor iedere website van belang om tegen kwaadwillende gebruikers beschermd te zijn. Hierbij zijn, in het geval van een dynamische website als deze, *SQL*-injecties en *Cross Site Scripting* twee van de meest kwaadaardige. Tegen de eerste was bescherming vrij eenvoudig; door gebruik te maken van de *PDO*-functies `prepare()` en `execute()`. De eerste de *query* definiërend met enkel de plaatsing van de variabelen, de tweede deze gepositioneerde variabelen een waarde toekennend.

Tegen *Cross Site Scripting*, ookwel het injecteren van *HTML*-code, werd beschermd door alle gebruikersinvoer die in de database diende te worden opgeslagen eerst door de *PHP*-functie `htmlspecialchars()` te halen. In de eerste plaats van gekozen om dit met de functie `htmlentities()` te doen, maar dit bleek problematisch bij de implementatie van Markdown — welke verderop behandeld zal worden.

2.4 Tokens en e-mail

Een andere manier van beveiligen, in dit geval van gebruikers tegen andere gebruikers met wat meer dubieuze motieven, kwam tot stand met het per e-mail versturen van *tokens*. Dit vindt allereerst plaats bij het registreren. Nadat een gebruiker zijn gegevens heeft ingevuld, wordt er een e-mail naar het door hem of haar opgegeven adres gestuurd met daarin een *token*-bevattende activatie link. Na het drukken op deze link wordt achter de schermen het account van de gebruiker geactiveerd en kan deze beginnen met het gebruik van de website. Dit gebeurt om zowel het bezit van het e-mailadres vast te stellen als om te voorkomen dat gebruikers herhaaldelijk accounts kunnen aanmaken. Soortgelijk is het mechanisme bij het opvragen van een wachtwoordsverandering en bij het veranderen van het e-mailadres. Bij die eerste is gekozen voor per e-mail verstuurd tokens omdat de enige betrouwbare manier is om de identiteit van een gebruiker te verifiëren, bij de tweede logischerwijs omdat ook van dit nieuwe e-mailadres vastgesteld moet worden dat de gebruiker hier de eigenaar van is.

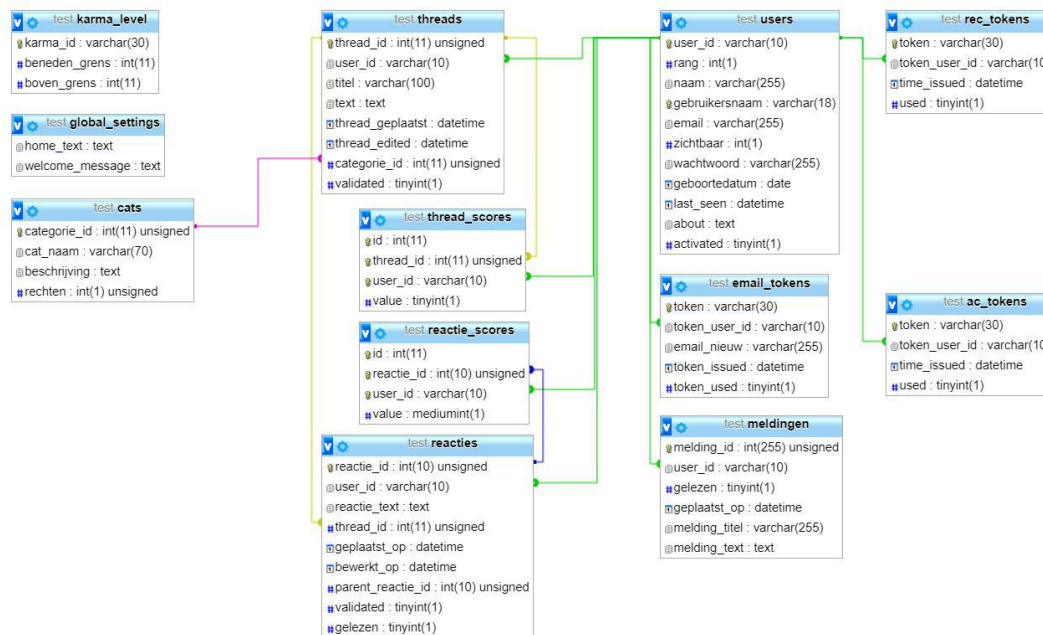
2.5 Captcha's

Onze opdrachtgever vond het van belang dat gebruikers bepaalde aspecten, namelijk het registreren, het *resetten* van wachtwoorden en het plaatsen van een bericht, niet door gebruikers geëxploiteerd kunnen worden door er binnen te korte tijd herhaaldelijk gebruik van te maken. Om deze *spam* te voorkomen is gebruik gemaakt van de *reCAPTCHA* van Google. Deze *CAPTCHA* is toegevoegd aan de betreffende formulieren en moet door de gebruiker opgelost worden voordat het formulier succesvol verwerkt zal worden door de server. De opdracht van de *CAPTCHA* wordt bij herhaaldelijk gebruik steeds ingewikkelder en maak het gebruik van automatische scriptjes onmogelijk. Een belangrijke reden voor het gebruik van de *CAPTCHA* is tevens dat bij het wachtwoordreset een e-mail naar de gebruiker wordt gestuurd en er bij herhaaldelijke aanvraag ook een grote hoeveelheid e-mails gestuurd kunnen worden. Dit is met deze toevoeging dus niet meer mogelijk.

2.6 Database

Onze database is opgezet met een datamodel welke logisch gekoppeld kan worden aan de opdrachtomschrijving. Ons datamodel heeft als belangrijkste tabel de **users** tabel, waarin

de `user_id` de belangrijkste *key/index* is.



Figuur 1: Grafische weergave van het toegepaste datamodel.

Bovenstaand figuur geeft een grafische weergave van de opbouw van ons datamodel. Zoals eerder gemeld, is de `user_id` de belangrijkste *key*, welke alle tabellen verbindt. Er zijn slechts twee tabellen die niet verbonden zijn aan het `user_id`. Het gekozen datamodel scheidt goed verschillende entiteiten, door bijvoorbeeld de `thread_scores` tabel niet te integreren bij de `threads`. De keuze hiervoor is mede te verantwoorden omdat een score door verschillende gebruikers gegeven kan worden op een enkele thread/reactie. De keuze om de tabellen `reactie_scores` en `thread_scores` te scheiden is bewust gemaakt zodat het onderscheid tussen threads en reacties gehandhaafd blijft. De database voldoet aan alle normaalvormen;

1. De volgorde van rijen en kolommen in de tabel heeft geen betekenis.
2. Entiteiten die niet bij elkaar horen zitten in verschillende tabellen.
3. Er bestaan geen transitieve afhankelijkheden.

In eerdere ontwerpen van de tabel bestond er in de `users` tabel een kolom genaamd `Karma`, deze kolom was de som van alle door de gebruiker verdiende scores, volgens de derde normaal vorm is dit niet correct vanwege het feit dat dit makkelijk berekend kan worden.

Verder is de keuze gemaakt om drie token tabellen te maken, voor de `email_tokens` tabel is deze keuze makkelijk te verantwoorden omdat in deze tabel de token gekoppeld wordt aan het nieuwe, nog te wijzigen, e-mailadres. De keuze om ook de tabellen `ac_tokens` en `rec_tokens` te scheiden heeft te maken met de tweede normaalvorm, de entiteiten hebben dan wel een zelfde structuur, maar zijn wezenlijk anders.

2.7 Gravatar

Het was van belang dat gebruikers zich duidelijk op het forum kunnen onderscheiden. Het kunnen instellen van een avatar is hierbij een belangrijk hulpmiddel. Er moest echter rekening worden gehouden met het feit dat deze avatars de server niet te veel mochten belasten.

Tevens moesten ze makkelijk in gebruik zijn. Hierom is gekozen om de Gravatar te implementeren. Op deze manier hoeven de avatars niet op de server opgeslagen te worden en is de avatar voor mensen die al gebruik maken van Gravatar gesynchroniseerd over verschillende websites.

2.8 Markdown

Het opmaken van tekst kan een bericht op een forum dermate aantrekkelijker en doeltreffender maken. Het is echter onverstandig om gebruikers hun berichten direct met *HTML* op te laten maken, zowel uit gemak- als veiligheidsoverwegingen. *Markdown* is hier een goed alternatief voor. Het is makkelijk in gebruik en stelt de (kwaadwillende) gebruiker niet in staat gevaarlijke code in de pagina te injecteren. Naast berichten kan ook de tekst op de home-pagina door gebruikers worden aangepast met behulp van *Markdown*.

2.9 Registreren en verificatie

Een niet-geregistreerde bezoeker kan zich registreren om meer functies op de website te kunnen gebruiken. Een gebruiker heeft slechts een unieke gebruikersnaam, uniek e-mailadres en een wachtwoord nodig om te registreren. Na het indienen van de registratie dient de gebruiker zijn account/e-mailadres te bevestigen/verifiëren door middel van een ontvangen link in de e-mail.

Registreren wordt verwerkt in 2 verschillende *PHP*, beide bestanden zijn te vinden in de folder `www`; `registreer.php` en `verwerk_input.php`. Naast deze *PHP* bestanden worden ook nog enkele *JavaScript* bestanden gebruikt om de user te helpen bij het kiezen van een goed wachtwoord en te voldoen aan sommige eisen aan de gebruikersnaam. De gebruikte *JavaScript* bestanden zijn te vinden in de folder `static/js/`. De structuur van de code is zo dat `registreer.php` gebruik maakt van *HTML*, *JavaScript* en een klein beetje *PHP*, om onschuldige gebruikers te helpen bij het correct invullen van de velden. In `verwerk_input.php` wordt vooral gebruik gemaakt van *PHP* in combinatie met blinde fora, om gebruikers bij een foute invoer terug te sturen naar de registreer pagina, alwaar de gegevens weer ingevuld zijn en de locatie van de fout is weergegeven.

De registratie dient te voldoen aan verschillende eisen, waarvan één een wachtwoord is van een zekere sterkte (volle wachtwoord sterkte meter/'oke' als commentaar), de volgende is het invullen van een reCAPTCHA alvorens de registratie voltooid kan worden.

De wachtwoordmeter is opgebouwd uit een *JavaScript* database van woorden in combinatie met verschillende algorithmes om zo gebruikers te waarschuwen voor het gebruik van hele eenvoudige wachtwoorden. Verder is er ook een verificatie per mail ingebouwd, hiervoor wordt een 30 karakters lange token gegenereerd, om hiermee de gebruiker zijn account te laten bevestigen. Dit dwingt een gebruiker een geldig e-mailadres te gebruiken alvorens hij zijn account kan gebruiken, wat ervoor zorgt dat er zeer weinig 'lege' accounts gemaakt worden.

2.10 User pagina

Een niet-geregistreerde bezoeker kan op de user-pagina evenveel als een geregistreerde gebruiker, beide kunnen gegevens en recente threads van de gekozen gebruiker zien. Een geregistreerde gebruiker kan op zijn eigen user pagina ook gegevens bewerken, zo kan deze zijn naam en geboortedatum invullen, tevens kan de gebruiker ervoor kiezen om zijn e-mailadres openbaar te maken. Een 'Super beheerder' kan op de user pagina de rang van een gebruiker/beheerder aanpassen en/of deze verbannen.

Voor de userpagina worden de volgende bestanden gehost in de `www` folder; `user.php` en `userData.php`. `user.php` kan door iedereen worden gezien, `userData.php` kan alleen door een geregistreerde gebruiker worden opgehaald, waar deze slechts zijn eigen pagina kan zien

(en wijzigen). De bestanden die nodig zijn voor beide pagina's zijn te vinden in de folders: `static/js/` & `php/userpage/`.

Het wijzigen van je wachtwoord werkt aan de hand van dezelfde functies als op de registreer pagina. Bij het wijzigen van een e-mailadres zal het oude e-mailadres worden gebruikt tot het moment dat de gebruiker zijn nieuwe e-mailadres heeft bevestigd.

Verder kan een 'Super beheerder' door naar de persoonlijke pagina van een gebruiker te gaan de rang van een gebruiker wijzigen in; verbannen/gebruiker/beheerder. In deze functie wordt gebruik gemaakt van *jQuery* om de wijzigingen direct aan te passen in de database.

2.11 Karma

Karma is een systeem waarin gebruikers elkaar threads en reacties kunnen beoordelen. Beoordelen kan met een +1 of een -1. Door Karma te verdienen kunnen gebruikers verschillende rangen verkrijgen, deze rangen zijn een weerspiegeling van de bijdrage van de gebruiker op het forum. Gebruikers beginnen als een Grassprietje en kunnen daarvandaan doorgroeien tot een Sequoia of afzakken tot onkruid.

Karma wordt geïmplementeerd door middel van twee verschillende tabellen in de database, een voor karma op reacties en een voor karma op threads. De keuze voor twee tabellen hiervoor is bewust genomen om het onderscheid tussen reacties en threads duidelijk te houden en er voor te zorgen dat reactie en thread-id's niet door elkaar gaan lopen. Een gebruiker kan per reactie of thread maar een keer een score geven, dit kan +1 of -1 zijn, wordt er twee keer gestemd dan wordt de oude entry uit de database verwijderd en vervangen door de nieuwe waarde. Ook is het mogelijk voor de gebruiker om zijn stem weer in te trekken, hiermee wordt dan ook de entry verwijderd uit de database om overbodige informatie te voorkomen.

2.12 Beheerder Tools

Een van de vereiste van het forum was het mogelijk maken voor beheerders om bepaalde posts eerst te kunnen screenen voordat deze zichtbaar worden voor andere gebruikers. Om dit makkelijk en gebruiksvriendelijk te maken, is er een aparte pagina voor de beheerders gemaakt waarop goed te keuren berichten verschijnen. Vanaf hier kunnen beheerders op hun gemak berichten doorlezen en bepalen of deze geschikt zijn voor het forum. Ook is er voor de beheerders een aparte pagina vanaf waar globale site instellingen geregeld kunnen worden zoals het instellen van de tekst op de homepage en de tekst die nieuwe gebruikers ontvangen in hun inbox na het activeren van hun account. Overige beheerder tools kunnen gevonden worden op handige plekken, zo kunnen nieuwe categorieën worden toegevoegd direct op de homepage, de rechten voor reeds bestaande categorieën kunnen hier ook makkelijk worden aangepast door middel van een selectie menu. Ten slot kunnen gebruikers kunnen worden verbannen of gepromoveerd vanaf hun gebruikerspagina ook door middel van een selectie menu die de database direct update.

2.13 Inbox en Notificaties

Door onduidelijkheden bij gebruikers bij het verwijderen van hun berichten die niet goedgekeurd worden, was er een manier nodig om gebruikers belangrijke informatie toe te sturen. Hierdoor is er een inbox in het leven geroepen. In de inbox ontvangt de gebruiker berichten wanneer er nieuwe reacties geplaatst zijn bij een van hun posts of wanneer zij een bericht hebben van de beheerder wanneer bijvoorbeeld een bericht niet goedgekeurd is. De gebruiker krijgt ook een bubbel te zien bij hun naam die aangeeft hoeveel berichten in de inbox nog niet gelezen zijn. Ook kan de gebruiker bepaalde filters instellen om alleen bepaalde berichten te laten zien, zoals bijvoorbeeld alleen ongelezen berichten.

De mededelingsberichten van beheerders maken gebruik van een aparte tabel in de database, hierin staat het bericht en de ID van de gebruiker aan wie het bericht gericht is. Ongelezen

berichten worden bijgehouden in de berichten tabel zelf door een kolom in deze tabel. Wanneer de plaatser van een thread zijn post bezoekt, worden hier automatisch alle ongelezen reacties als gelezen gemarkeerd, zo hoeft er niet een voor een geklikt te worden op notificaties. De teller van de bubbel wordt in realtime bijgewerkt, dit wordt gedaan door elke paar seconden een PHP bestand te pollen die vervolgens alle ongelezen berichten uit reacties en mededelingen optelt, evenals alle niet gevalideerde threads en reacties uit de respectievelijke tabellen om dit voor de beheerder beschikbaar te maken.

2.14 Zoekfunctie

Doordat een forum al gauw onoverzichtelijk wordt naarmate het aantal berichten groeit, is er een zoekfunctie ontwikkeld voor het Forum. Deze zoekfunctie kan zoeken op gebruikers, threads en reacties op threads.

Bij het zoeken op gebruikers wordt er gebruik gemaakt van *SQL LIKE*-statements en zoekt op zowel de echte naam als de gebruikersnaam van de gebruiker, er is hier gekozen voor een *LIKE* zodat mensen ook een stuk midden in een gebruikersnaam kunnen zoeken zonder dat dit geen resultaten geeft zoals bij een *FULLTEXT* zoekopdracht. Threads en reacties worden gezocht met de eerder genoemde *FULLTEXT* search in *SQL*. Dit is in dit geval gekozen om rekening te houden met eventuele groei van het forum terwijl de zoekresultaten dan niet te langzaam opgehaald worden aangezien *FULLTEXT* nadeel is echter, zoals eerder vermeld, dat elke zoekopdracht wordt gezien als begin van een woord en je dus niet stukken tekst midden in een woord kan zoeken. Dit is echter geen probleem voor deze toepassing aangezien een reactie of een thread beide uit veel woorden bestaan en gebruikers van zichzelf al geneigd zijn om op volledige woorden of zinnen te zoeken als ze een bepaalde reactie of thread willen vinden.

2.15 Thread-pagina

De Thread-pagina is de kern van elk forum, de meeste gebruikers van het Forum zullen de meeste tijd op deze pagina doorbrengen. Om deze reden is ervoor gekozen om het ontwerp van de thread-pagina overzichtelijk te houden, dit is gedaan door net als op de rest van de site groentinten en ronde vormen te gebruiken.

Op de thread-pagina's kan elke gebruiker de post en reacties van de gekozen thread bekijken. Door dit voor iedereen toegankelijk te maken wordt niemand geforceerd om een account te maken en is de basis van het Forum dus voor iedereen beschikbaar.

Geregistreerde gebruikers kunnen reacties plaatsen en threads beginnen in de meeste categorieën. Er zijn enkele categorieën waar alleen beheerders en super beheerders nieuwe threads mogen beginnen. Deze categorieën zullen basisinformatie en regels van het Forum bevatten, daar is het dus ook niet nodig voor normale gebruikers om threads te kunnen beginnen. Ook is het mogelijk om als gebruiker op reeds bestaande reacties te reageren. Deze reacties bevatten de tekst van de reactie waarop ze reageren in een aparte bubble met grijze achtergrond. Zo kan elke gebruiker makkelijk zien welke reactie waarbij hoort. Deze dubbele reacties kunnen geplaatst worden door op een aparte reageer knop te klikken die bij elke reactie staat. Met *JavaScript* wordt het reactieformulier dan zichtbaar gemaakt en de parent-reactie-id wordt meegegeven om de juiste text en gebruikersnaam uit de database te halen.

Op de thread-pagina worden *SQL*-queries gebruikt om alle data van een thread, met alle reacties en gebruikersinformatie uit de database te halen. Met deze data wordt de pagina met de juiste informatie weergegeven aan de gebruiker.

Geregistreerde gebruikers die hun eigen threads bekijken zien een aantal extra knoppen waarmee ze de thread kunnen aanpassen of verwijderen. Dit is handig om simpele spelfouten op te lossen, maar kan ook gebruikt worden wanneer bepaalde informatie in de thread niet meer van toepassing is. Beheerders kunnen alle threads van gebruikers verwijderen, dit kan bijvoorbeeld nodig zijn bij aanstootgevende threads.

2.16 Lay-out

Voor de lay-out van het Florum heeft de opdrachtgever vrij spel gegeven. De enige vereiste was dat het duidelijk een plantenforum zou worden. Om echt het gevoel van de natuur te krijgen is er gebruik gemaakt van verschillende tinten groen en ronde vormen. Het idee was om de lay-out simpel en duidelijk te houden. Om dit te bereiken is er naast groen enkel grijs en wit gebruikt. Door deze rustige elementen te gebruiken valt de focus niet op de lay-out van de website, maar vooral op de technische elementen die erin verwerkt zijn. De zijkanten van de pagina's zijn ook bewust leeg gelaten, hier is voor gekozen omdat de inhoud van de pagina's nu langwerpiger van vorm is. Dit leest veel prettiger dan wanneer de inhoud over de hele breedte van het scherm is verdeeld.

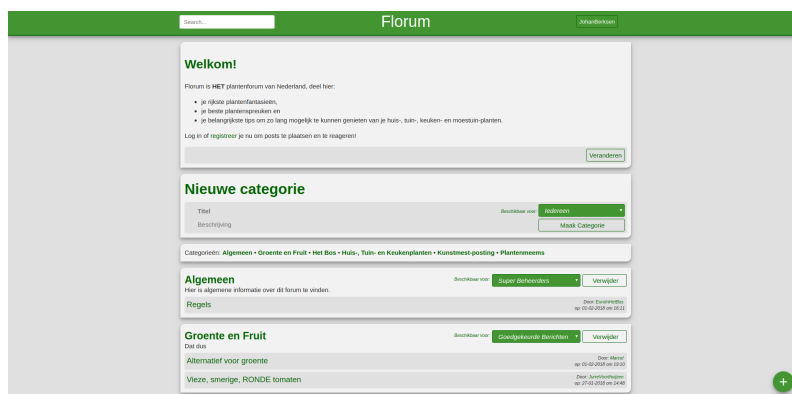
Als de website op een mobiel apparaat wordt bekeken vallen de grijze vlakken aan de zijkant weg, dit gebeurt omdat een mobiel al langwerpiger is en de grijze vlakken hier slechts onnodig ruimte in zouden nemen. Ook past de grootte van andere elementen zich aan wanneer de site op een mobiel apparaat wordt bekeken. Hiervoor is gebruik gemaakt van de *Media Query*-techniek in *CSS*. Deze techniek past elementen aan wanneer aan de voorwaarden wordt voldaan; in dit geval is de voorwaarde de scherm breedte, als deze smaller dan een bepaald aantal pixels wordt passen elementen zich aan.

3 Resultaten

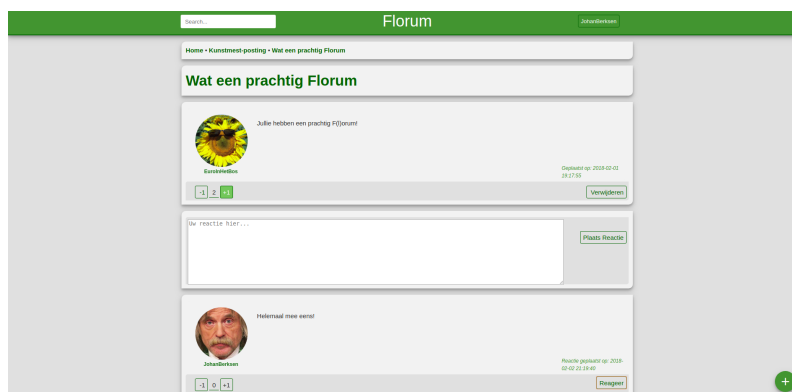
Hier zijn enkele screenshots van het Forum.



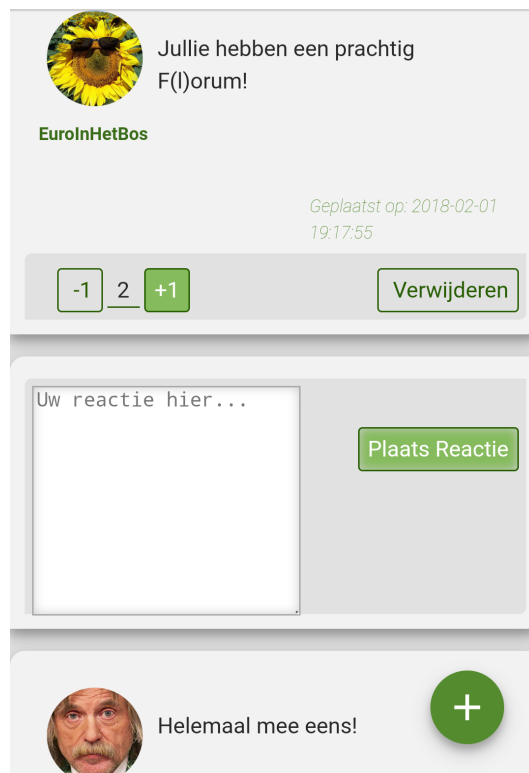
Figuur 2: De home-pagina zoals een gast deze te zien krijgt.



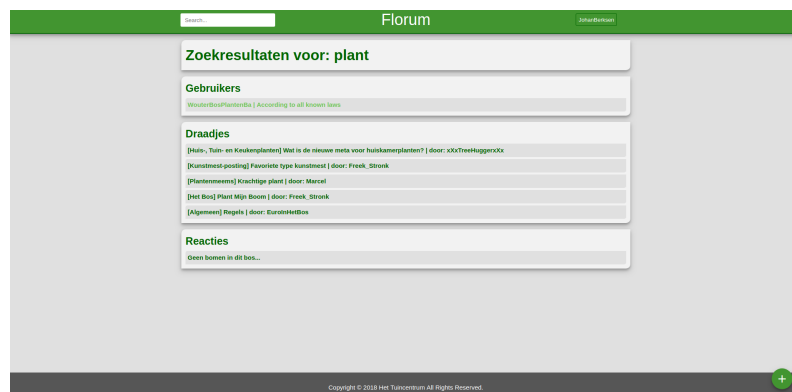
Figuur 3: De home-pagina zoals een beheerder deze te zien krijgt.



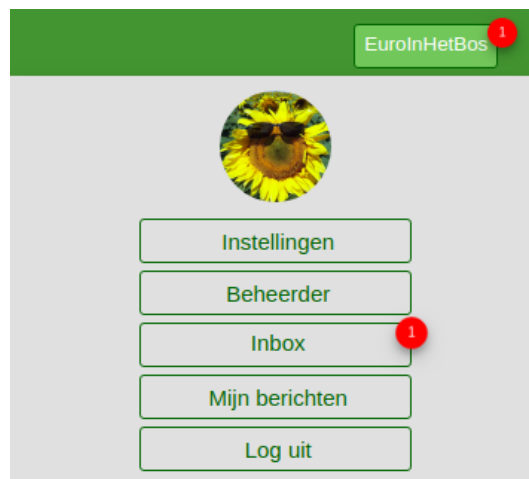
Figuur 4: De thread-pagina weergegeven op een pc (ingelogd als beheerder).



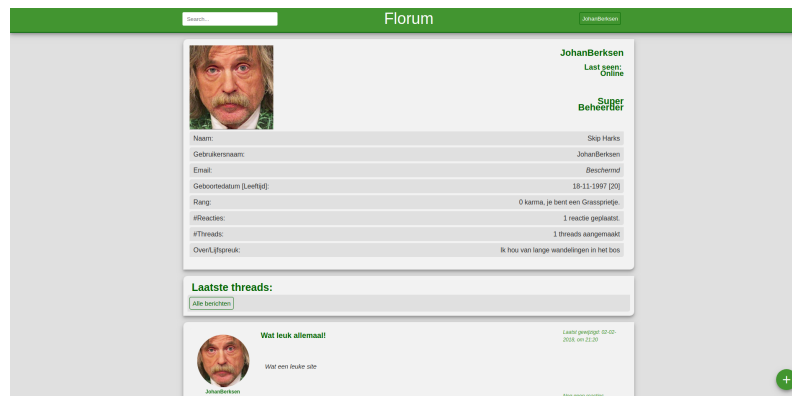
Figuur 5: De thread-pagina weergegeven op een mobiel (ingelogd als beheerder).



Figuur 6: De zoekresultaten wanneer een gebruiker zoekt naar het woord "plant".



Figuur 7: Het *drop-down*-menu van een ingelogde beheerder met een bericht in de inbox.



Figuur 8: De userpagina voor JohanBerksen (beheerder).

4 Discussie

Alle geplande functionaliteit is eerder geïmplementeerd dan verwacht, hierdoor zijn veel extra's toegevoegd die niet van te voren bedacht waren. De resultaten zijn daarom ook beter dan in eerste instantie verwacht werd met alle functionaliteit werkend naar behoren. Er zijn echter ook dingen die een volgende keer anders en beter kunnen. Een goed voorbeeld hiervan is de opmaak van de website met *CSS*. Veel dingen schalen niet geheel netjes en de algemene opmaak van de website laat nog wat te wensen over om over te komen als een moderne, responsieve website. Ook zijn er enkele pagina's die niet geheel met *AJAX* zijn geïmplementeerd omdat deze eerder in het project gemaakt zijn en er later pas bleek dat *AJAX* implementeren niet veel extra werk had geweest. Ook de naamgeving van functies, variabelen en bestanden had beter gekund want deze is niet eenduidig. Ook dit is een punt dat meegenomen zal worden naar toekomstige projecten.

Natuurlijk is er ook nog ruimte voor verdere uitbreidingen van het forum. Zo zou een *direct messaging*-systeem tussen gebruikers een goede toevoeging zijn. Dit zou met de huidige opzet niet moeilijk zijn om te implementeren aangezien de inbox al naar behoren werkt en hier makkelijk uitbreidingen aan toe te voegen zijn.

4.1 Bronvermeldingen

als gebruik is gemaakt van code die niet door de groep is ontwikkeld, vermeld dat dan hier.

- De wachtwoordsterktemeter (`static/js/zxcvb.js`) is afkomstig van: <https://css-tricks.com/password-strength-meter/>
- Parsedown (`php/parsedown.php`) is afkomstig van: <https://github.com/erusev/parsedown>
- De captcha-verwerking (`php/postCaptcha.php`) is afkomstig van: <http://makingspidersense.com/tutorials/tut-recaptcha.txt>
- Het CSS shake-effect (`static/css/style.css`) is afkomstig van: <https://teamtreehouse.com/community/shake-effect-with-javascript-only>
- De url get *JavaScript* functie (`static/js/getGet.js`) is afkomstig van: <https://css-tricks.com/snippets/javascript/get-url-variables/>