

HoGent

Faculteit Bedrijf en Organisatie

CMS bovenop PHP framework Laravel biedt het antwoord op de klant gefocuste CMS met behoud van de focus voor ontwikkelaars. Proof of concept.

Dieter Vyncke

Scriptie voorgedragen tot het bekomen van de graad van
Bachelor in de toegepaste informatica

Promotor:
Jens Buysse
Co-promotor:
Joeri Van Steen

Academiejaar: 2015-2016

Tweede examenperiode

Faculteit Bedrijf en Organisatie

CMS bovenop PHP framework Laravel biedt het antwoord op de klant gefocuste CMS met behoud van de focus voor ontwikkelaars. Proof of concept.

Dieter Vyncke

Scriptie voorgedragen tot het bekomen van de graad van
Bachelor in de toegepaste informatica

Promotor:
Jens Buysse
Co-promotor:
Joeri Van Steen

Academiejaar: 2015-2016

Tweede examenperiode

Samenvatting

CMS'en of content management systemen die klant gefocust zijn, bevatten beperkingen voor ontwikkelaars. Laravel is een opkomend populair PHP framework waarvoor reeds enkele CMS'en beschikbaar zijn. Hier is nog maar weinig onderzoek naar gedaan. Via dit werkstuk wordt op zoek gegaan of een CMS gebouwd bovenop PHP framework Laravel een antwoord kan bieden op klant gefocuste CMS'en. Biedt het gekozen CMS een focus voor ontwikkelaars.

De pijnpunten van een klant gefocust CMS worden onderzocht via een literatuurstudie. Er wordt voor beide systemen een website uitgewerkt aan de hand van een vooropgestelde casus (Appendix A). Deze website bevat typerende pagina's en bewerkingen voor een CMS (Appendix B). De structuur en bevindingen van beide systemen worden beschreven.

Testen en bijhorende resultaten benadrukken positieve als negatieve punten voor beide systemen. Factoren die belangrijk zijn voor ontwikkelaars bij het kiezen van een nieuw framework of systeem, worden vergeleken.

Voorwoord

Dit eindwerk is tot stand gebracht in het kader van laatstejaars student in de toegepaste informatica, aan de Hogeschool Gent.

Graag wil ik mijn promotor Jens Buysse bedanken voor de opvolging en advies doorheen dit werkstuk. Bedankt Joeri Van Steen voor de technische hulp.

Ik wil daarnaast ook mijn dank betuigen aan mijn ouders, familie, vrienden en medestudenten die me gesteund hebben bij deze realisatie en afgelopen drie jaar hoger onderwijs aan de Hogeschool Gent.

Inhoudsopgave

1 Inleiding	6
2 Beschrijving van CMS	8
2.1 Definitie van een CMS	8
2.2 Meest populaire CMS'en	9
2.2.1 Wordpress	9
2.2.2 Joomla	9
2.2.3 Drupal	9
3 Voor wie is dit onderzoek bedoeld?	10
4 Pijnpunten Drupal CMS	11
4.1 Leercurve	11
4.2 Backward Compatibility en Updates	11
4.3 Zwaar systeem	12
4.4 Database afhankelijk	13
4.5 Beheersbaarheid	13
4.6 Modules	13
5 Laravel	15
5.1 Wat is Laravel?	15
5.2 Trending	15
5.3 Wat maakt Laravel goed?	18
5.4 Laravel CMS	19
5.5 OctoberCMS	20
6 Casus concreet definiëren	22
7 Methodologie	23
7.1 Drupal	24
7.1.1 Installatie	24
7.1.2 Theming	24

7.1.3	Menu en routing	28
7.1.4	Menu	28
7.1.5	Plugins en modules	30
7.1.6	Opbouw pagina's	32
7.2	OctoberCMS	37
7.2.1	Installatie	37
7.2.2	Theming	38
7.2.3	Menu en routing	40
7.2.4	Plugins en modules	42
7.2.5	Opbouw pagina's	49
8	Vergelijkende testen en resultaat	57
8.1	Leercurve	57
8.1.1	Drupal	58
8.1.2	OctoberCMS	59
8.1.3	Besluit	59
8.2	Documentatie	59
8.2.1	Drupal	60
8.2.2	OctoberCMS	61
8.2.3	Besluit	62
8.3	Community	62
8.3.1	Drupal	62
8.3.2	OctoberCMS	63
8.3.3	Besluit	63
8.4	Performantie	63
8.4.1	Drupal	64
8.4.2	OctoberCMS	67
8.4.3	Besluit	68
8.5	Beheersbaarheid	69
8.5.1	Drupal	69
8.5.2	OctoberCMS	73
8.5.3	Besluit	75
8.6	Database	76
8.6.1	Drupal	76
8.6.2	OctoberCMS	78
8.6.3	Besluit	79
9	Conclusie	80
A	Casus code - Github	82

B Casus - Designs

83

Hoofdstuk 1

Inleiding

Vandaag worden sites steeds uitgebreider en bevatten steeds meer functionaliteiten. Content beheer is één van deze functionaliteiten die steeds meer gevraagd wordt door de klant. Een content beheerssysteem noemen we een CMS (Content Management System). Deze functionaliteit geeft de eindegebruiker (klant) de optie om zelf bepaalde stukken inhoud aan te passen op hun eigen site. Het voordeel is dat klanten snel, eenvoudig en zonder tussenkomst van een ontwikkelaar, hun site kunnen beheren.

Door de vraag naar deze functionaliteit zijn er de laatste jaren steeds meer systemen die zich focussen op CMS. Echter hebben veel van deze systemen hun focus enigszins verlegd. Daar waar de ontwikkelaar een site bouwt met toegevoegde functionaliteit van content beheer, kan de klant nu zijn site volledig zelf opbouwen. Mensen zonder technische kennis betreffende programmeren kunnen op deze manier zelf hun site in elkaar passen.

Doordat de functie van ontwikkelaar verlegd is naar de klant, kan dit struikelblokken opleveren voor de ontwikkelaar zelf. De structuur en de werking van het systeem kunnen anders zijn dan de vertrouwde frameworks die vertrouwd zijn voor ontwikkelaars. Ontwikkelaars die bepaalde kennis en structuren zijn aangeleerd kunnen deze kennis niet ten volle gebruiken in deze CMS'en: ofwel kan het plafond van het CMS bereikt zijn zonder het gewenste resultaat te bekomen, ofwel voldoet het CMS in het algemeen niet aan de verwachtingen.

Er zijn verschillende opties in dit geval. De ontwikkelaar schrijft zelf zijn content beheerssysteem en begint op die manier volledig vanaf nul. Of je gaat als ontwikkelaar op zoek naar een goed alternatief die de focus behoudt op de ontwikkelaar. Hierbij wordt vermeden dat iedere ontwikkelaar steeds opnieuw het wiel gaan uitvinden.

De bedoeling van dit stuk is op zoek te gaan naar een goed alternatief voor de klant

HOOFDSTUK 1. INLEIDING

gefocuste CMS'en. We richten ons op een PHP framework, Laravel. Er wordt op zoek gegaan naar een CMS gebouwd op Laravel dat belangrijke elementen voor een ontwikkelaar kan invullen. Belangrijke elementen bij het kiezen van een nieuw framework is de leercurve, documentatie, community, performantie, beheersbaarheid, databasestructuur enz. Er wordt een vergelijkende testcasus opgesteld met enerzijds een klant gefocust CMS (Drupal), anderzijds een CMS geschreven op Laravel.

Kan PHP framework Laravel een CMS aanbieden dat een gunstig antwoord biedt op alle of op het merendeel van de opgesomde belangrijkste elementen voor ontwikkelaars? Op welke punten schiet een klant-gefocus CMS tekort?

Hoofdstuk 2

Beschrijving van CMS

2.1 Definitie van een CMS

CMS of Content Management Syste(e)m is een term die verschillende definities kan aannemen afhankelijk van de indruk van een persoon of project. CMS kan ook benamingen aannemen als ECM (Enterprise Content Management) of WCM (Web Content Management). Dit zijn alle benoemingen die bedrijven in het leven hebben geroepen om zich te onderscheiden.

Als we het binnen de web ontwikkeling hebben over een Content Management Systeem, spreken we over een CMS. Een CMS biedt de mogelijkheid om met verschillende gebruikers die al dan niet verschillende toegangsrechten hebben, stukken inhoud van een website aan te passen zonder kennis van programmeren. Met inhoud beheren bedoelen we het toevoegen, aanpassen, archiveren en publiceren van informatie. Deze bewerkingen gebeuren via een controle paneel waar de verschillende rechten tot uiting komen. Afhankelijk van de rechten is het al dan niet mogelijk meer bewerkingen uit te voeren. Zo zal bijvoorbeeld een blogger enkel zijn eigen posten kunnen aanspreken, terwijl een beheerder (Admin) de posten van alle geregistreerde gebruikers zal kunnen beheren.

Of een website al dan niet een CMS dient te hebben is afhankelijk van de klant. Het is zo dat de klant een stuk minder afhankelijk is van de web ontwikkelaar, maar is zeker ook niet onafhankelijk. Wanneer updates moeten doorgevoerd worden zal dit moeten gebeuren via de web ontwikkelaar. Aanpassingen buiten de afgebakende beheersbare gebieden, kan de klant niet zelf uitvoeren (Kohan Bernard, 2010).

2.2 Meest populaire CMS'en

We bekijken de drie CMS'en die momenteel het grootste aandeel hebben. We lijsten de systemen op in volgorde van populariteit: WordPress, Joomla, Drupal (BuiltWith, 2015; W3Techs, 2016).

Elk van deze systemen heeft als hoofddoel voor de eindgebruiker een vlot beheersbare site te creëren waarbij hij zelf de content kan beheren. Toch hebben deze systemen elk hun eigen specifieke doel.

2.2.1 Wordpress

WordPress is uitgegroeid tot de grootste globale blog applicatie. Dit systeem is een open source project, waar velen meewerken om dit product iedere dag te verbeteren. WordPress is een bijzonder eenvoudig en makkelijk zelf aan te leren systeem. WordPress werd in eerste instantie vooral gebruikt als blog site. Maar ondertussen is dit systeem verder uitgebreid met verschillende plugins, Widgets en thema's. Waardoor het ook mogelijk is om grotere sites te ontwikkelen.

2.2.2 Joomla

Joomla is een stuk gecompliceerder en beter uitbreidbaar in vergelijking met WordPress, maar blijft toch eenvoudig in gebruik. Qua complexiteit en mogelijkheden ligt het in het midden. Uit deze selectie, is Joomla het meest geschikt voor het opzetten van sociale netwerk en e-commerce sites. Joomla combineert de voordelen van WordPress en Drupal en voegt hierbij zijn eigen features toe.

2.2.3 Drupal

Drupal heeft een wijd aspect van mogelijkheden, het is daarom ook een meer geavanceerd systeem in vergelijking met de andere systemen. Drupal kan gebruikt worden voor het ontwikkelen van eenvoudige informatieve sites tot meer complexe trafiek intensive sites voor grote bedrijven. Doordat Drupal vrij ingewikkeld is, schrikt dit de grootste groep mensen af die geen kennis van programmeren hebben. Het is aangewezen om een minimum kennis van HTML en PHP te hebben als je eraan denkt een website te bouwen via Drupal.

Hoofdstuk 3

Voor wie is dit onderzoek bedoeld?

In dit onderzoek wordt er niet gericht op eindgebruikers die op eenvoudige wijze een site kunnen opstellen via een CMS, maar op ontwikkelaars die op zoek zijn naar een CMS dat de focus houdt zowel op het ontwikkelen als op de custom-built oplossingen.

Uitgaande van deze criteria, komt Drupal het best in aanmerking wanneer we een keuze dienen te maken uit de bovenstaande drie, klant gefocuste CMS'en (W3Techs, 2016). Biedt Drupal de perfecte oplossing of zijn er toch nog nadelen te vinden aan dit systeem? Aan de hand van een vergelijkende studie tussen Drupal en een ander PHP Framework Laravel proberen we een antwoord te vinden op deze vraag.

Het gebruikersdoel van Drupal is vooral gericht op sites voor bedrijven. Het aandeel van bedrijfsdoeleinden torent hoog boven de rest uit (BuiltWith, 2016a). Wanneer we de cijfers met doekeinden voor het PHP framework Laravel erbij nemen zien we dat dit framework net zoals Drupal voornamelijk gebruikt wordt voor bedrijven sites. Het vergelijken van beide systemen uitgaande van de doelstellingen is op deze manier relevant.

Hoofdstuk 4

Pijnpunten Drupal CMS

4.1 Leercurve

Voor nieuwe gebruikers kan het gebruik van Drupal als overrompelend ervaren worden. De vele beschikbare modules die al dan niet overlappend zijn met andere modules, kunnen voor verwarring zorgen. De te uitgebreide functionaliteiten zorgen ervoor dat de essentie verloren gaat.

Iedere nieuwe gebruiker van Drupal komt in aanraking met de term "Drupalisms", een term die vertaald kan worden als: "Do it the Drupal way". Bij het volgen van de Drupal ontwikkelingsmethoden is het niet voldoende om enkel PHP te kennen. Vele methoden en patronen zijn genest in pagina's van pagina's of andere API documenten. Dit vergt tijd om dit als nieuwe Drupal ontwikkelaar onder knie te krijgen. Drupal heeft een geheel eigen methode van werken (MCD partners, 2015).

Echt goede Drupal ontwikkelaars vinden is geen sinecure. Om Drupal volledig onder de knie te krijgen dien je te rekenen op een paar jaar intensieve training.

Ondanks de vele modules kan het gebeuren dat je een feature wil dat niet kant-en-klaar is aangeleverd. Dan is het aangewezen de feature zelf te ontwikkelen aan de hand van een module. Een leercurve die een pak steiler is dan het onder de knie krijgen van de basis van Drupal. Een grondige kennis van PHP en werking van het core is hier essentieel.

4.2 Backward Compatibility en Updates

Bij de start heeft de Drupal Community beslist om niet compatibel te zijn met eerdere versies. Dit wil zeggen dat nieuwe versies geen ondersteuning bieden aan oudere ver-

sies. Reden hiervoor is dat Drupal steeds de laatste technologieën wil ondersteunen en als dusdanig een beter systeem wil afleveren.

Doordat Drupal geen compatibiliteit met oudere versies ondersteunt, levert dit voor bedrijven die steeds up-to-date willen zijn met de laatste technologieën een extra kost op. Voor ontwikkelaars wil dit zeggen dat ze hun modules moeten herschrijven om volledig compatibel te zijn met de nieuwe versie.

Om een Drupal site steeds up-to-date te houden en daarbij volledig beveiligd te zijn, dien je zeer frequent, zo niet maandelijks je modules up te daten. Binnen een bepaalde versie levert dit meestal weinig problemen op, dit in tegenstelling met het upgraden naar een nieuwe core versie van Drupal. Deze actie verloopt meestal niet van een leien dakje (Tzu-Chi Huang, 2015).

4.3 Zwaar systeem

Drupal 7 kan aanzien worden als een zwaar systeem als we dit gaan vergelijken met verschillende andere PHP frameworks. Drupal bezit een grote basiscode met vele functionaliteiten die bijna nooit of uiterst zelden allemaal in één project nodig zullen zijn. Een teveel aan code dat duidelijk voor vertraging zorgt. Drupal 7 maakt gebruik van Hooks, een systeem dat goed werkt om functies automatisch te detecteren en in te laden. Dit systeem komt echter met het nadeel van lagere performantie. Drupal laadt elke geactiveerde Module in bij ieder verzoek, om te zien of er een Hook geïmplementeerd is voor deze bepaalde functie waardoor een piek van activiteiten ontstaat en er performantie verlies optreedt. Dit kan problemen opleveren bij een hosting die niet voldoende RAM ter beschikking stelt. Drupal lost dit op door een overvloed aan caching te voorzien. Hardware op zich is niet duur en daarom niet onoverkomelijk maar het levert een minder efficiënt systeem op.

"The power comes at a great cost to flexibility, simplicity and efficiency. I like the idea of having just the code that is needed – not an extra line or module."

Varun Arora arora Varun (2013)

4.4 Database afhankelijk

Drupal 7 is een database georiënteerd systeem. Bij een bestand georiënteerd systeem is de volledige structuur en opbouw in code is terug te vinden. Drupal slaat de structuur van zijn modellen (Content types) en opbouw van pagina's op in de database. Dit resulteert in een grote afhankelijkheid van de database. Voor statische pagina's die in een ander PHP framework geen database verzoek vergen, is dit bij Drupal 7 wel het geval.

Drupal genereert zeer veel query's, zelfs om de meest eenvoudige data op te halen. Daarnaast heb je ook geen enkel idee welke query's je aanvraagt op je database. Het spreekt voor zich dat de database heel groot kan worden, te groot zelf. Bij complexe structuren waarbij je normaal een 30-tal tabellen nodig hebt, maakt Drupal hiervan het tienvoudige. Drupal gebruikt aparte table joins waarbij er een immens netwerk aan relaties ontstaat tussen tabellen. De tabel is te sterk genormaliseerd om de gewenste flexibiliteit te verkrijgen.

4.5 Beheersbaarheid

Met beheersbaarheid bedoelen we dat Drupal niet 'Developer Oriented' is. Het systeem is niet ontwikkeld om te voldoen aan de wensen web ontwikkelaars maar is dat eerder aan de wensen van eindgebruikers die zelf de sites in elkaar kunnen puzzelen zonder een lijn code te schrijven. Het opbouwen van pagina's, model types en het beheren van items gebeurt allemaal via de back-end van de browser en dus niet via beheersbare code die eenvoudig te debuggen is. Daarnaast bestaan er geen stukken code die later opnieuw van dienst kunnen zijn voor andere projecten. Voor ieder project dienen alle configuratie instellingen opnieuw te worden aangemaakt.

De code die omgezet wordt naar HTML is allesbehalve proper te noemen. Er bestaat een teveel aan overbodige div-tags. Klassen zijn veelal weinigzeggend en moeilijk overzichtelijk te stylen via CSS.

4.6 Modules

Voor elk probleem in Drupal is er waarschijnlijk wel reeds een module gemaakt. Nogtans zijn veel van deze modules geen oplossing voor het daadwerkelijke probleem. Naast de vele afgewerkte en goed werkende modules, zit de module bibliotheek van Drupal vol met slecht werkende en gedateerde modules. Dit komt omdat Drupal open source software is waar iedereen vrij is zijn inbreng te leveren. Slecht werkende modules be-

vatten ernstige bugs waardoor de werking faalt. Gedateerde modules ontstaan omdat Drupal geen comptabiliteit met vorige versies ondersteunt.

Modules waar een sterk team achter staat dat bug reporting opvolgt en de module in stand houdt, zijn veelal goede en betrouwbare modules. Het is aan de gebruiker na te gaan of de module al dan niet bruikbaar is en een oplossing voor het probleem kan zijn. Een goede referentie is Github.

We proberen een antwoord te vinden op elk van deze aangehaalde pijnpunten in het stuk "Vergelijkende testen en resultaat". We bekijken hierin de resultaten alsook gaan we na of er een alternatief systeem bestaat dat deze pijnpunten volledig of deels kan wegwerken.

Hoofdstuk 5

Laravel

Wanneer we kijken naar de tekortkomingen die Drupal heeft, is het geen overbodige investering om op zoek te gaan naar een alternatief dat deze tekortkomingen kan overbruggen. In de zoektocht naar dit alternatief zijn er verschillende criteria waar we rekening mee dienen te houden. Zowel Drupal als Laravel hebben een gelijklopend applicatieplatform en doelpubliek. Bovendien gebruiken beide PHP.

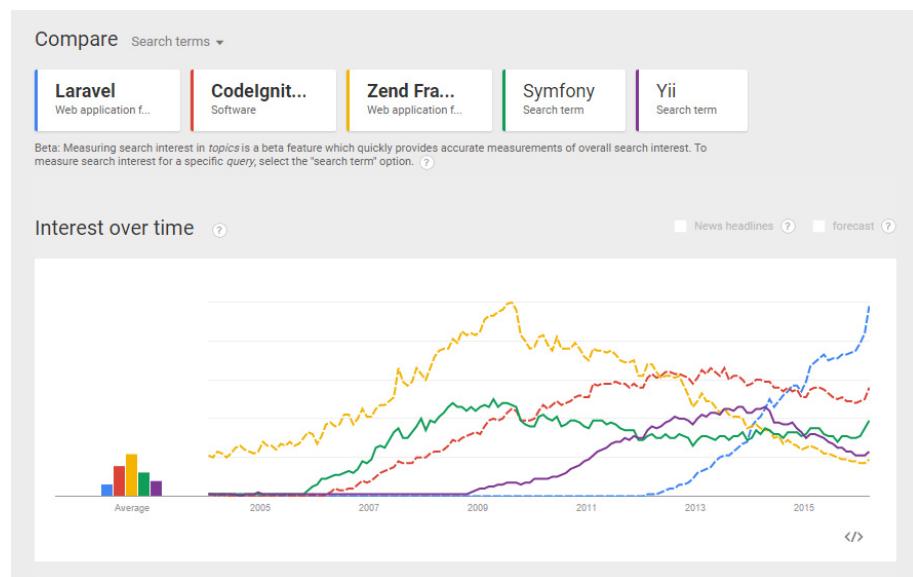
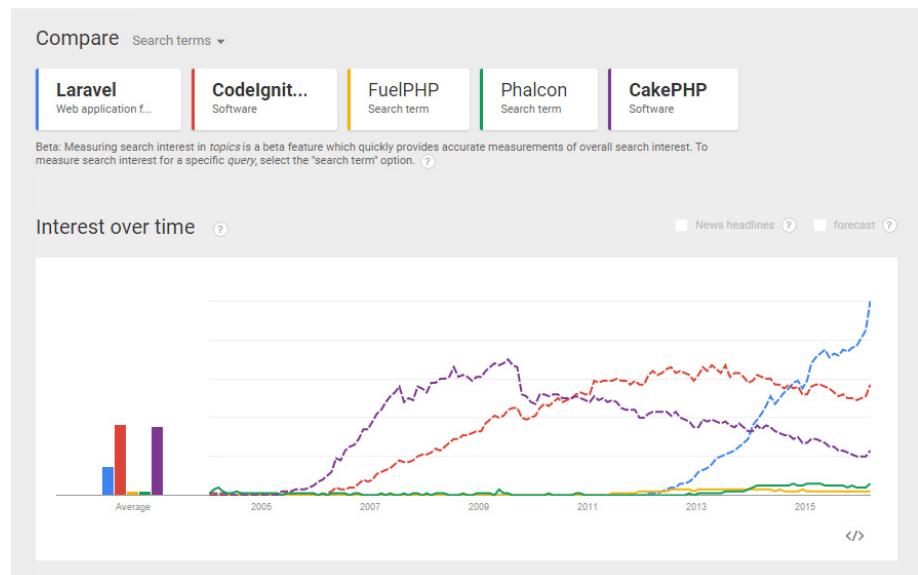
5.1 Wat is Laravel?

Laravel is een MVC framework geschreven in PHP, om webapplicaties te ontwikkelen. Dit framework is ontworpen om de kwaliteit van de te ontwikkelen software te verbeteren. Door de duidelijke, vooruitstrevende syntax en kernfuncties bespaart dit veel tijd bij het voltooien van een applicatie. De filosofie van Laravel opteert voor conventie over configuratie. Het komt erop neer dat je meestal je doel bereikt met minder code waarbij ernaar gestreefd wordt het midden te houden tussen minimalistisch en functioneel. Het spreekt voor zich dat het begrijpen van leesbare code eenvoudiger is. Het stuurprogramma geeft je de vrijheid om op eenvoudige wijze aanpassingen door te voeren aan caching-, session-, database- en authenticatiefuncties.

Laravel maakt gebruik van de command-line-interface Artisan. Met deze tool is het mogelijk om als ontwikkelaar testen, migraties en periodieke taken uit te voeren.

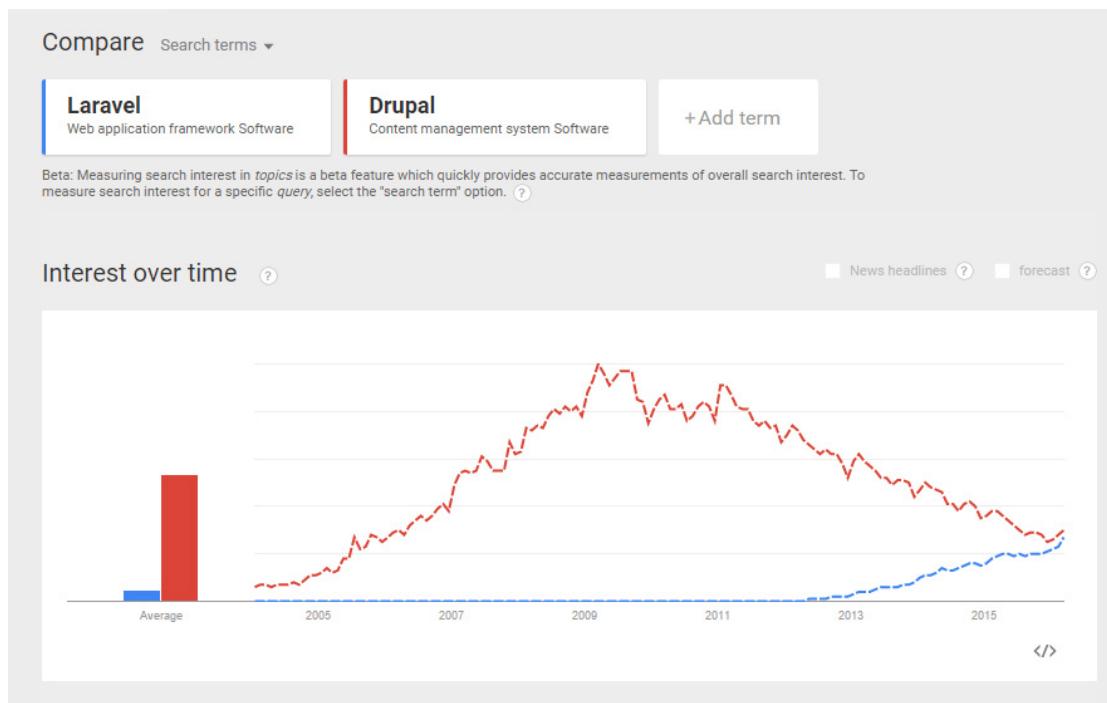
5.2 Trending

Onderstaande link geeft een overzicht van Google Search Words die trending zijn over verloop van tijd. We lijsten de meest populaire PHP frameworks op en zien duidelijk dat Laravel de leiding neemt. Laravel heeft het grootste relatieve aandeel, alsook de sterkste stijging.



Figuur 5.1: Google trends waarbij verschillende populaire PHP frameworks worden vergeleken. Laravel torrent hoog boven de anderen uit.

HOOFDSTUK 5. LARAVEL



Figuur 5.2: Google trends vergelijking tussen Drupal en Laravel framework

De grafiek op figuur 5.2 toont puur informatief aan hoe het relatieve aandeel tussen Laravel en Drupal onderverdeeld is. Je kan hierop twee dingen waarnemen. Enerzijds dat Drupal een sterke negatieve trend vertoont, anderzijds dat Laravel sterk stijgt en bijna evenredig loopt met Drupal. Vermoedelijk zal het aandeel van Drupal binnenkort terug stijgen met de opkomst van Drupal 8. Ook al zijn deze twee systemen op deze manier moeilijk te vergelijken, toch geeft het een beeld weer van de populariteit.

Naast enkele zoektermen afgevuurd te hebben via Google trends biedt Github een zeer betrouwbare en interessante gegevensbron. Laravel staat op de eerste plaats als trending open source PHP framework (Figuur: 5.3).

The screenshot shows the GitHub Trending page for open source repositories. The title 'Trending in open source' is displayed, along with the subtitle 'See what the GitHub community is most excited about today.' Below this, there are two tabs: 'Repositories' and 'Developers'. A dropdown menu shows 'Trending: today'. On the right, there is a sidebar with language filters: 'All languages', 'Unknown languages', 'CSS', 'HTML', 'Java', 'JavaScript', and 'PHP' (which is highlighted with a blue background). Other language options like 'Python' and 'Ruby' are also visible. The main content area lists five repositories:

- AutumnsWind/Front-end-tutorial**: Description: '最全的资源教程-前端涉及的所有知识体系'. Stars: 110. Built by 4 contributors.
- php-pm/php-pm**: Description: 'PPM is a process manager, supercharger and load balancer for PHP applications.'. Stars: 108. Built by 4 contributors.
- paralax/lfi-labs**: Description: 'small set of PHP scripts to practice exploiting LFI, RFI and CMD injection vulns'. Stars: 34. Built by 1 contributor.
- getgrav/grav**: Description: 'Modern, Crazy Fast, Ridiculously Easy and Amazingly Powerful Flat-File CMS'. Stars: 25. Built by 5 contributors.
- laravel/laravel**: Description: 'A PHP Framework For Web Artisans'. Stars: 21. Built by 5 contributors.

Figuur 5.3: Github trending repos

5.3 Wat maakt Laravel goed?

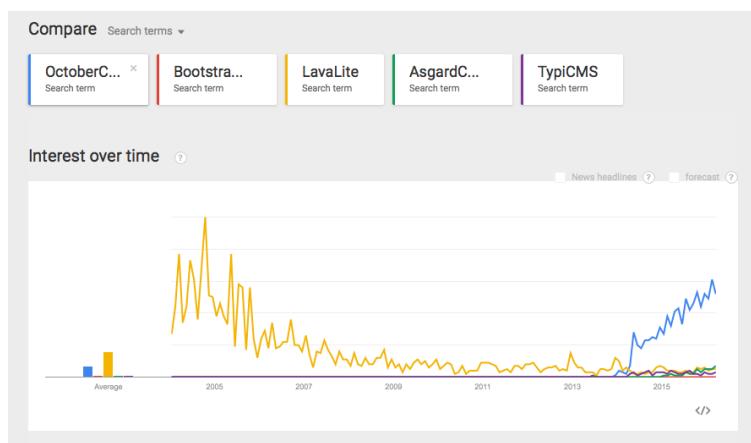
Laravel kan webapplicaties aan van alle grootte en van alle complexiteit. Dit dankzij een aantal doorslaggevende voordelen dat het systeem biedt. Laravel is gebouwd om eenvoudig en snel te leren. Die leercurve wordt ondersteund door een goed uitgebouwde documentatie die neergeschreven staat als tutorials. Naast de standaard documentatie is er ook een aparte website die zich toelegt op video tutorials. Laracast.com stelt honderden video's ter beschikking.

5.4 Laravel CMS

Laravel op zich is een PHP framework en geen CMS. Het is natuurlijk mogelijk om zelf een CMS te schrijven aan de hand van Laravel. Echter, dit is niet de doelstelling van dit werk. Daarom wordt er op zoek gegaan naar een bestaand CMS dat geschreven is via Laravel. Deze zoektocht geeft een lijst terug van een vijf-tal mogelijke CMS'en. Uit deze lijst wordt er slecht één CMS gekozen. Wegens de beperktheid van dit werk, is het onderzoeken van meerdere CMS'en niet mogelijk.

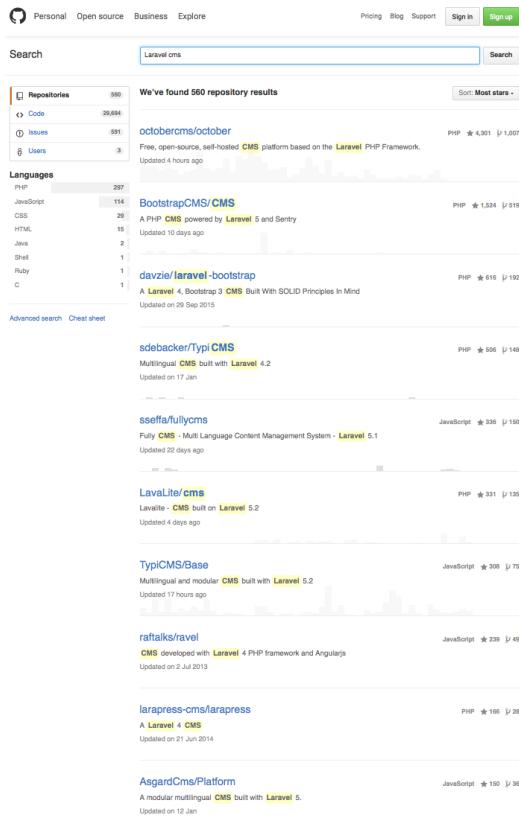
Om tot een keuze te komen werden de CMS'en onderworpen aan een aantal zaken. Zo werd gekeken welke zoekterm het beste was voor kwam (Figuur: 5.4). Daaruit bleek duidelijk dat OctoberCMS als grootste speler naar voor komt. Naast zoektermen kan GitHub een zeer betrouwbare bron zijn voor allerhande ontwikkelprogramma's. Een zoekactie op 'Laravel CMS' leverde na filtering op aantal sterren, een grotendeels gelijklopend resultaat met Google Trends.

OctoberCMS komt ook hier het beste naar voren. Meer nog, er is een duidelijke activiteit merkbaar bij dit systeem. De hoeveelheid en frequentie van updates, als ook het aantal 'Forks' ligt hoog. Personen die een 'Repository Fork' halen experimenteren of verbeteren het systeem. Forks worden in de meeste gevallen gebruikt om voorstellen tot aanpassingen te doen aan de projecteigenaar.



Figuur 5.4: Google Trends voor Laravel CMS

Kortom, er leeft een community achter OctoberCMS, meer dan welk ander Laravel CMS.



Figuur 5.5: Github Laravel CMS

5.5 OctoberCMS

Elk CMS heeft zijn eigen specifieke eigenschappen en eigen systeem opbouw. De hoofdidee van dit CMS is terug te gaan naar de 'Basics', een licht systeem zonder complexe structuren en een duidelijke flow van begin tot einde. Dit systeem belooft een snelle leercurve waardoor de opstart van elk project binnen een aantal minuten klaar kan zijn. OctoberCMS op zich is geen framework, aangezien het als het ware op de schouders zit van een bestaand framework, Laravel. OctoberCMS zit op een hoger niveau en kan steunen op alle voordelen van het Laravel framework. Door te steunen op deze voordelen is het eenvoudiger en sneller ontwikkelen.

De hoofgedachte die terug te vinden is op de website van OctoberCMS, is een duidelijk standpunt die ze innemen omtrent een CMS.

“Clients don’t build websites, developers do.”

Een team dat deze gedachte als belangrijk ervaart, zal de pijnpunten die bijvoorbeeld Drupal heeft, teniet doen. Wanneer een CMS met de intentie is gebouwd om de eindgebruiker het roer van het schip te geven, zal de ontwikkelaar hoogstwaarschijnlijk snel stoten op het bereiken van het plafond van het systeem. De ontwikkelaar heeft zich niet volledig kunnen ontplooien in het project en grijpt terug naar een alternatief waar alles opnieuw uitgevonden moet worden. Dit heeft als gevolg dat ontwikkelen terug duurder wordt.

Met OctoberCMS, met de doelgroep van ontwikkelaars, zullen deze sneller en goedkoper een project van a tot z opleveren. Met kennis van HTML, CSS en PHP kan een standaard website ontwikkeld worden. Een grondige kennis van PHP is vereist wanneer de site bijkomende plugins nodig heeft. Plugins kunnen gezien worden als de belangrijkste elementen voor het uitbreiden van het OctoberCMS.

OctoberCMS biedt een systeem met oog op ontwikkelen vanuit het perspectief van de ontwikkelaar, maar biedt ook een eenvoudig te beheersen systeem voor de eindgebruiker. Aanpassingen die de eindgebruiker doorvoert worden gedaan via de back-end in de browser.

Hoofdstuk 6

Casus concreet definiëren

De analytische site BuiltWith (BuiltWith, 2016b) geeft aan dat Laravel vooral gebruikt wordt voor bedrijfsdoeleinden, net zoals Drupal. Het framework wordt met andere woorden het vaakst gekozen om sites voor bedrijven te ontwikkelen. Nieuws- en ontspanningssites komen op de tweede en derde plaats.

De casus bevat een voorgedefinieerde template die exact zal uitgewerkt worden in twee systemen. De template wordt zowel gebouwd in Drupal 7 als een CMS gebruikmakend van het framework Laravel, OctoberCMS. Op deze manier kunnen vergelijkingen op een correcte manier gebeuren (Appendix B).

Tijdens het opbouwen van de template worden aantekening gemaakt van problemen, oplossingen en bedenkingen. De vooruitgang wordt in grote lijnen beschreven.

Beide systemen worden uitgetest op een bestaand design voor een bedrijfswebsite. Deze website bevat de meest typerende en meest belangrijke elementen als voorbeeld voor een CMS. Pagina's die worden geïmplementeerd zijn:

- Een index pagina met nieuwsberichten
- Beheerpagina, overzicht- en detail pagina voor realisaties
- Overzichtspagina van het team
- Overzicht- en detail pagina voor testimonials

Hoofdstuk 7

Methodologie

Dit deel beschrijft het doorloopp proces van de vooropgestelde template. De opbouw van de site in beide systemen heeft een bepaalde volgorde die doorlopen moet worden. Drupal en OctoberCMS zijn twee verschillende systemen die naar hetzelfde resultaat ijveren. Toch zal duidelijk worden dat er veel verschillen zitten in opbouw, structuur en werking.

De verschillende hoofdstukken, hieronder beschreven zijn toepasbaar op beide systemen, maar de invulling ervan zal duiden op de verschillen en/of gelijkenissen die beide hanteren. Volgende punten worden toegelicht en besproken:

- Installatie
- Theming
- Menu en routing
- Plugins en Modules
- Opbouw pagina's

7.1 Drupal

7.1.1 Installatie

Het installeren en opstellen van een nieuwe Drupal 7 core is kinderspel. De minimale systeemvereisten voor een installatie van Drupal 7 zijn verwaarloosbaar. De belangrijkste parameter om te controleren is de versie van PHP. Drupal 7 draait op een PHP versie van 5.2.5 of hoger.

De installatie gaat enkel via een Wizard installatie. Deze kan gedownload worden op de officiële website van Drupal. Download de laatste versie van de core en pak deze uit op een plaats waar naar kan gelinkt worden via je lokale server. Maak nu eerst een lege database aan en surf vervolgens via de browser naar het installatiebestand. Dit is de root directory gevuld door /install.php. De Wizard begeleid je door het volledige installatieproces. Na de standaard installatie en de taal geselecteerd te hebben, wordt de configuratie van de database ingesteld. Kies welke soort database je wil gebruiken, vul de naam, gebruikersnaam en wachtwoord in. Test de connectie. Aangezien we lokaal werken hoeven geen geavanceerde opties worden ingesteld. We gaan verder en zien dat Drupal wordt geïnstalleerd net zoals alle standaard modules.

Nadat de installatie voltooid is dienen nog enkele parameters te worden ingesteld. De nieuwe site moet nog een naam en e-mail adres meekrijgen. Er moet nog een Admin worden aangemaakt met bijhorende naam en wachtwoord, alsook een tijdszone van de server. Dit zijn allemaal aanpassingen die niet definitief zijn en kunnen later worden bijgesteld.

We kunnen nu onze nieuwe Drupal site bezoeken en beheren. De nieuwe site bevat een aantal standaard elementen opgemaakt met een default thema.

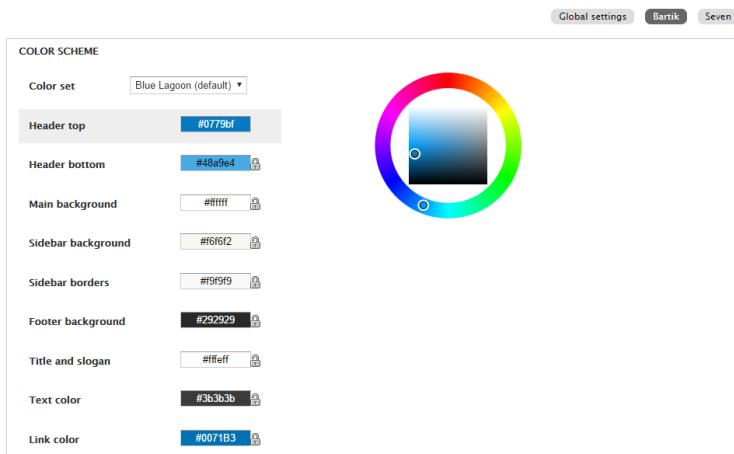
De installatie neemt hooguit 5 minuten in beslag en geeft geen technische moeilijkheden. De installatietijd kan gestaafd worden aan de hand van volgende screencast (Learn By The Drop, 2011).

7.1.2 Theming

Drupal is een CMS waarbij de aanpassingen grotendeels in de browser gebeuren. Om die reden heeft Drupal twee verschillende thema's actief. Het eerste thema geeft de look en feel van de site. Het tweede is een administratief thema. Dit thema is enkel voor site beheerders. Bij het administratief thema is overzicht en duidelijkheid de grootste troef. Hierbij hoeven geen flashy kleuren gebruikt te worden, maar moet een omgeving te vinden waar de administrator alles terug weet te vinden.

Een standaard installatie van Drupal krijgt voor beide thema's een default toegewezen. In de meeste gevallen zal je bij deze standaard thema's geen genoegen nemen. Indien je opteert af te stappen van een standaard thema heb je drie opties om deze te vervangen.

Drupal heeft een grote community die ondertussen een lange tijd bestaat. Het aantal thema's dat zowel te verkrijgen is op de Drupal site, als daarbuiten, is immens. Thema's te verkrijgen op de Drupal site zijn allemaal gratis. Externe sites zoals Templatemonsters.com bieden niet minder dan 450 thema's waarvan het grootste deel echter tegen betaling. Deze thema's zijn ready-to-go en hoeven in de meeste gevallen geen aanpassing. Bij het installeren van een thema is het wel mogelijk een aantal parameters in te stellen afhankelijk van het thema.



Figuur 7.1: Parameters zoals kleuren zijn in te stellen bij sommige thema's.

Een tweede manier van werken is de totaal tegenovergestelde manier. Maak een nieuw thema aan door vanaf nul te beginnen. Maak een folder aan onder volgend pad: projectnaam/sites/all/themes/. Er zijn een aantal stappen en documenten verplicht bij het creëren van een eigen thema. Zorg voor een .info file. Dit document beschrijft je thema.

```
name = yourthemename
description = Description of yourtheme.
core = 7.x
engine = phptemplate
stylesheets[all][] = style.css
stylesheets[all][] = layout.css
regions[left] = Left sidebar
regions[right] = Right sidebar
```

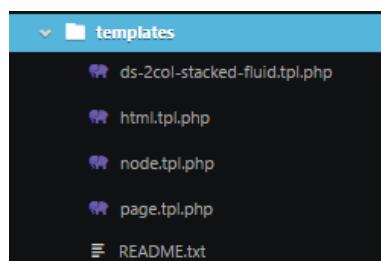
```

regions[content] = Content
regions[header] = Header
regions[footer] = Footer

```

Een thema heeft een naam en beschrijving nodig en draait op een bepaalde Drupal versie. Verder kunnen we CSS files, scripts en regions toevoegen. Regions worden verder in dit hoofdstuk besproken. Via de .info file is het mogelijk het thema te selecteren in de browser. Een site en zeker een thema is niets zonder een opmaak. Daarom voorzie je CSS files in een aparte folder.

Van hieraf heb je een thema dat bruikbaar is om mee aan de slag te gaan. Een Drupal thema is gebaseerd op template files die als extensie .tpl.php meekrijgen. Deze files bevatten de html van je thema. Wanneer deze template files niet gedefinieerd zijn in een bepaald thema, worden deze aangesproken vanuit de Drupal core. Wanneer je templates aanmaakt in een thema, overschrijf je met andere woorden de template uit de Drupal core. Templates komen in een aparte folder. In de template page.tpl.php zal je het snelst aanpassingen doen. Deze template bevat de body van de pagina en dus algemene structuur.



Figuur 7.2: Template folder van een thema. Overschrijf zoveel template files als nodig.

Een laatste optie om een thema in te stellen, is een combinatie van de twee voorgaande manieren. Drupal.org heeft een aantal thema's die een basis structuur hebben uitgewerkt. Deze thema's gemaakt met het oog op uitbreiding. Ze bevatten een .info file, een gedeeltelijke of volledige opbouw van een CSS structuur, bijhorende overschreven template files en eventuele andere folders met bijvoorbeeld scripts. Dit soort thema heeft als voordeel dat je niet helemaal opnieuw moet starten maar geeft je een basis om verder op te bouwen.

Bij het uitwerken van de casus werd gebruik gemaakt van het basisthema Zen. Dit thema wordt aanzien als populair op Drupal.org. Zen is een eenvoudig HTML5 starters thema met een responsief en mobile-first grid.

De opbouw van een pagina loopt gedeeltelijk anders in Drupal in vergelijking met een

HOOFDSTUK 7. METHODOLOGIE

PHP framework. Zoals reeds aangehaald worden in elk thema regions gedefinieerd in de info file. Regions bepalen de structuur en gebieden op een pagina. Vaak voorkomende regions zijn een header, footer, een zijbalk, enz. Wanneer de verschillende gebieden zijn aangemaakt kunnen deze worden ingevuld met allerhande content. Regions worden ingevuld met blocks. Blocks zijn aanpasbare blokken die verschillende vormen van inhoud kunnen aannemen. Standaard maakt Drupal enkele block elementen aan zoals menu's, een zoekformulier, shortcuts, enz.

Home » Administration » Structure

+ Add block + Add group + Add menu block

BLOCK	REGION	WEIGHT	OPERATIONS
Header			
Hoofdmenu	Header	-11	configure

Figuur 7.3: Blocks bevatten verschillende bouwstenen van een pagina. Deze zijn terug te vinden onder structuur > blocks.

Navigation	- None -	-6	configure
Powered by Drupal	- None -	-5	configure
Recent comments	- None -	-4	configure
Recent content	- None -	-3	configure
Search form	- None -	-2	configure
Shortcuts	- None -	0	configure
Social Media Links	- None -	0	configure
Subscription Block	- None -	-15	configure
Switch user	- None -	0	configure
Syndicate	- None -	4	configure
System help	- None -	5	configure
User login	- None -	6	configure
User menu	- None -	7	configure
View: Index Page – Pictures: Index page block first image	- None -	0	configure

Figuur 7.4: Standaard worden reeds enkele blocks voorzien, nieuwe kunnen gecreëerd worden.

7.1.3 Menu en routing

Routing

Routing in Drupal is een stuk ingewikkelder dan een PHP Framework. Het volledig in detail uitwerken van routing in Drupal valt buiten dit werkstuk. Daarom wordt de algemene werking hier toegelicht.

Drupal hanteert de database tabel menu_router om alle verschillende URL's in op te slaan. Wanneer een URL wordt aangevraagd, wordt de volledige tabel overlopen op zoek naar de correcte URL. URL's krijgen steeds een machine naam en alias toegewezen. De machine naam wordt opgeslagen in de database tabel en heeft een weinig zeggende naam voor de gebruiker. De machine naam wordt voorafgegaan door 'node' gevolgd door een cijfer. Een node in Drupal is elk stukje individuele content, zoals een pagina, artikel, blog post, enz. Een alias naam kan per node zelf bepaald worden door de gebruiker. Dit is een haalbare zaak wanneer een site uit een beperkt aantal pagina's bestaat. Indien een site tientallen pagina's bevat is het beter URL's automatisch te laten creëren. De module Pathauto genereert automatische aliassen voor URL's met een beschrijvende naam.



Figuur 7.5: URL aliassen kunnen automatisch of manueel ingevuld worden per node.

7.1.4 Menu

Een menu in Drupal kan volledig beheerd worden via de back-end in de browser. Onder structuur > menu's worden alle bestaande menu's opgeliist. Standaard maakt Drupal een aantal menu's aan die verder aanpasbaar zijn. Deze menu's zijn eerder gericht op de configuratie van het systeem en van de gebruiker. Links verwijzen naar een gebruikersaccount of naar administratieve taken. Het aanmaken van een menu vraagt weinig kennis. Via de knop 'Add menu' voeg je een menu met een naam en beschrijving toe. Nu kunnen links toegevoegd worden via de 'Add link' knop. Een nieuwe link bevat

HOOFDSTUK 7. METHODOLOGIE

een naam, pad naar de content en eventuele omschrijving. Dit is de meest eenvoudige opstelling van een menu.

Home » Administration » Structure » Menus

+ Add link

MENU LINK	ENABLED	WEIGHT	OPERATIONS	
/ Home /	<input checked="" type="checkbox"/>	-50 ▾	edit	delete
/ Projecten /	<input checked="" type="checkbox"/>	-49 ▾	edit	delete
Index	<input checked="" type="checkbox"/>	-48 ▾	edit	delete
/ Het Team /	<input checked="" type="checkbox"/>	-47 ▾	edit	delete
/ Testimonials /	<input checked="" type="checkbox"/>	-46 ▾	edit	delete

Figuur 7.6: Aangemaakt menu met bijhorende menu items.

Een menu kan uitgebreid worden aan de hand van verschillende modules om tot gewenste resultaat te komen. Zo kan een menu bestaan uit submenu's met dropdown functie, kan een link vervangen worden door een afbeelding of kan een gewicht worden toegekend aan een link.

Menu settings
/ Het Team /

URL redirects
No redirects

URL path settings
Alias: team

Revision information
No revision

Comment settings
Closed

Authoring information
By admin on 2015-07-28 12:34:40
+0200

Publishing options
Published

Provide a menu link

Menu link title
/ Het Team /

Description

Shown when hovering over the menu link.

Parent item
<Hoofdmenu>

Weight
-47 ▾

Menu links with smaller weights are displayed before links with larger weights.

Figuur 7.7: Wanneer een pagina wordt aangemaakt is het mogelijk direct een menu item te voorzien in een geselecteerd menu.

7.1.5 Plugins en modules

Drupal 7 gebruikt modules als bouwstenen om sites verder uit te breiden. Drupal is een open source community die een groot aantal gebruikers kent. Elke ontwikkelaar heeft de kans om mee te helpen de community verder te verbeteren en uit te breiden. Meewerken kan door bijvoorbeeld mee te werken aan de core of patches te maken voor modules. Patches zijn verbeteringen aan de code of oplossingen voor bugs.

Het aanbod van bestaande modules is immens. Maar liefst 11,894 modules zijn beschikbaar voor Drupal 7 (geraadpleegd op 07/05/2016 (Drupal.org, 2016)). Modules worden opgedeeld in categorieën, dit zorgt ervoor dat het zoeken naar de gewenste module eenvoudig en snel verloopt.

Installeren van modules is een eenvoudig proces waarbij je op de site van Drupal.org op zoek gaat naar een module. Deze module kan je downloaden als gezippt bestand en uitpakken in de folder project/sites/all/modules. De installatie en activatie van een Module wordt gedaan in de back-end van de browser. Onder de tab modules ga je op zoek naar de nieuwe module en zet je deze op actief. Bij de installatie worden de nodige tabellen aangemaakt en wordt de module gebruiksklaar gemaakt. Afhankelijk van de module kunnen nu de nodige configuratie instellingen gebeuren.

Modules kunnen verschillende verhoudingen aannemen. Een module kan een kleine uitbreiding bevatten zoals een extra veld bij content types (content types wordt uitgelicht bij het volgende hoofdstuk). Maar evengoed een uitgebreide module die op zijn beurt nog uitgebreid kan worden met modules. De mogelijkheden met dit soort combinaties is aanzienlijk. In vele gevallen is een grondige kennis van dit soort modules aan te raden.

De uitleg, installatie en configuratie omtrent een module is steeds terug te vinden op de bijhorende pagina. Deze uitleg voldoet meestal bij kleine modules, grotere modules voorzien in vele gevallen een eigen tutorial waarbij de deze verder wordt toegelicht.

Door de hoeveelheid aan modules is de kans dat je voor een bepaald probleem geen oplossing vindt, eerder klein. Toch bestaat de kans dat iets 'custom made' moet gemaakt worden. Het aanmaken van een nieuwe module vergt een grondige kennis van PHP en de werking van de core. De implementatie van een module valt buiten de grenzen van dit stuk.

HOOFDSTUK 7. METHODOLOGIE

11,894 Modules match your search

Maintenance status [- Any -]
Development status [- Any -]
Module categories [- Any -]
Core compatibility 7.x
Status Full projects
Search Modules
Sort by Most installed ▾
Search

Extend and customize Drupal functionality with contributed modules. If a module doesn't quite do what you want it to do, if you find a bug or have a suggestion, then [join forces](#) and help the module maintainer. Or, share your own by [starting a new module](#).

Chaos tool suite (ctools)

Posted by merlinofchaos on December 5, 2008 at 7:37pm

This suite is primarily a set of APIs and tools to improve the developer experience. It also contains a module called the Page Manager whose job is to manage pages. In particular it manages panel pages, but as it grows it will be able to manage far more than just Panels.

For the moment, it includes the following tools:

- Plugins -- tools to make it easy for modules to let other modules implement plugins from .inc files.
- Exportables -- tools to make it easier for modules to have objects that live in database or live in code, such as 'default views'.
- AJAX responder -- tools to make it easier for the server to handle AJAX requests and tell the client what to do with them.
- Form tools -- tools to make it easier for forms to deal with AJAX.
- Object caching -- tool to make it easier to edit an object across multiple page requests and cache the editing work.
- Contexts -- the notion of wrapping objects in a unified wrapper and providing an API to create and accept these contexts as input.
- Modal dialog -- tool to make it simple to put a form in a modal dialog.
- Dependent -- a simple form widget to make form items appear and disappear based upon the selections in another item.
- Content -- pluggable content types used as panes in Panels and other modules like Dashboard.

Downloads

Recommended releases

 This module is currently being ported to Drupal 8, but is not usable yet. Help us by following this [issue](#).

Version	Download	Date
7.x-1.9	tar.gz (419.27 KB) zip (646.63 KB)	2015-Aug-19

Other releases

Version	Download	Date
8.x-3.0-alpha26	tar.gz (64.65 KB) zip (135.65 KB)	2016-May-06
6.x-1.15	tar.gz (369.32 KB) zip (550.44 KB)	2015-Dec-22

Figuur 7.8: Het aanbod aan modules voor Drupal 7 is immens.

7.1.6 Opbouw pagina's

Dit hoofdstuk beschrijft de werkwijze waarmee de casusopdracht werd verwezenlijkt. Hierin worden de structuur en het proces in grote lijnen toegelicht.

Eenmaal de structuur is bepaald via regions kunnen we beginnen nadenken over de verschillende content types. Een content type is een voorgedefinieerde collectie van data types en velden die samen horen. Elke node wordt gelinkt aan een content type. Om de pagina Team te implementeren van de casus, werd een content type aangemaakt. Het aanmaken van het content type teamlid wordt stap voor stap uitgelegd. Via de link structuur > content types kan je een nieuw type aanmaken. Standaard geef je hier de naam en omschrijving mee als parameters. Onder de tab 'Manage Fields' kan je verschillende velden definiëren. Elk veld heeft een label met afgeleide machine naam. Het 'Field type' bepaalt het data type van dit veld. Een Widget is de manier waarop een veld kan worden ingevuld en getoond.

LABEL	WEIGHT	PARENT	MACHINE NAME	FIELD TYPE	WIDGET	OPERATIONS
Naam Medewerker	0	- None -	title	Node module element		
Job - Afdeling	1	- None -	field_job_afdeling	Text	Text field	edit delete
Aanwerkingsdatum	2	- None -	field_aanwerkingsdatum	Date	Pop-up calendar	edit delete
Profielfoto medewerker	3	- None -	field_profielfoto_medewerker	Image	Image	edit delete
Jaren in dienst	4	- None -	field_jaren_in_dienst	Computed	Computed	edit delete
URL redirects	5	- None -	redirect	Redirect module form elements		
URL path settings	6	- None -	path	Path module form elements		
Add new field						
<input type="text"/> Label	7	- None -		- Select a field type -	- Select a widget -	Type of data to store. Form element to edit the data.
Add existing field						
<input type="text"/> Label	8	- None -	- Select an existing field -	- Select a widget -		Field to share. Form element to edit the data.

[Save](#)

Figuur 7.9: Defineer de gewenste velden met bijhorende data types voor iedere content type.

De tab 'Manage Display' bepaalt welke velden getoond worden en op welke manier. Velden kunnen al dan niet een label bevatten om het veld te duiden. Een format bepaalt de manier waarop de inhoud getoond wordt. Deze parameters zijn verschillend afhankelijk van het datatype.

Standaard kan je kiezen uit een aantal data types. Voldoen deze niet, bestaan er verschillende modules om de data types uit te breiden.

Nu het content type voor teamlid is aangemaakt kunnen er leden worden toegevoegd. Een lid kan worden toegevoegd via content > add content > Teamlid. Vul de velden in die eerder werden gedefinieerd in de configuratie van het content type. Bepaal manueel een URL of laat deze automatisch genereren.

Alle nieuwe teamleden kunnen bekijken worden in een overzicht via de content link. De volgende stap is het visualiseren van de verschillende leden op een aparte pagina. De leden moeten in een grid van drie kolommen komen te staan. De views module is een zeer uitgebreide module die perfect dienst doet bij problemen zoals deze. Views kan voor vele doeleinden gebruikt worden. Voorstellen van content in een specifieke lijst is één van de meest gebruikte. Elke lijst kan getoond worden als block of als aparte pagina.

The screenshot shows the 'Add new view' configuration page. At the top, there are tabs for 'LIST' and 'SETTINGS'. Below the tabs, the breadcrumb navigation shows: Home > Administration > Structure > Views. The main form has the following fields:

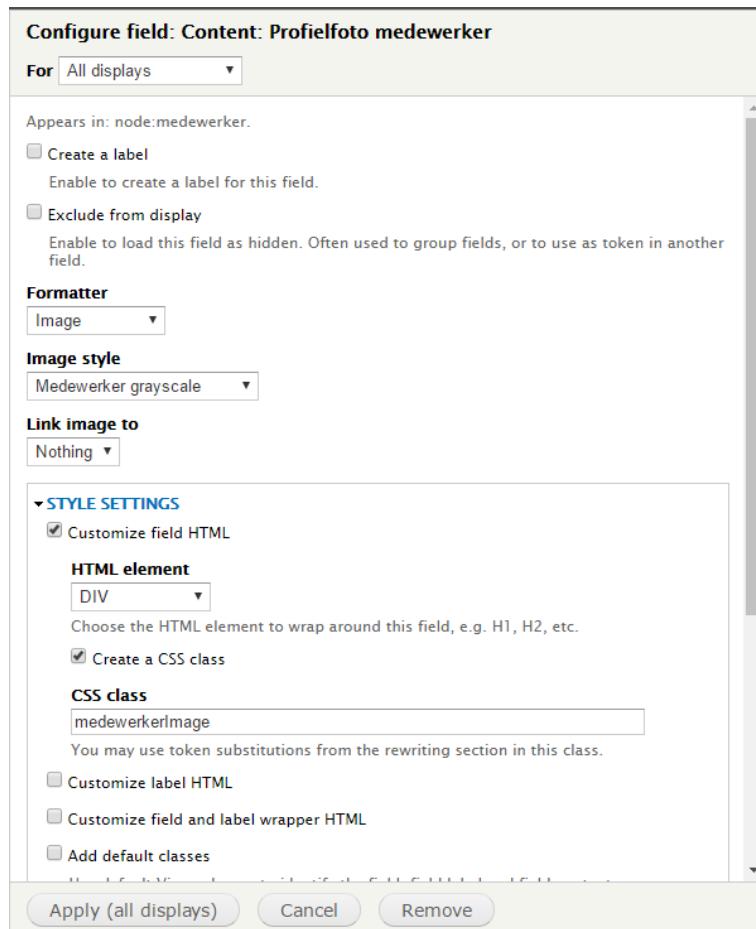
- View name ***: Teammembers. A note below says "Machine name: teammembers [Edit]".
- Description**: An optional field.
- Show**: Content. **of type**: Medewerker. **sorted by**: Newest first.
- Create a page**: An optional checkbox.
- Create a block**: A checked checkbox. Sub-fields include:
 - Block title**: Teammembers.
 - Display format**: Unformatted list or titles (linked).
 - Items per page**: 5.
 - Use a pager**: An optional checkbox.

At the bottom of the form are three buttons: 'Save & exit', 'Continue & edit', and 'Cancel'.

Figuur 7.10: Instellingen bij het aanmaken van een nieuwe view.

De view wordt aangemaakt en zal bestaan uit content met als content type Teamlid, aflopend gesorteerd. We zullen de lijst tonen als een block. Later wordt duidelijk waarom. Alle andere instellingen laten we default staan. Alle instellingen die nu gedaan zijn, kunnen achteraf nog aangepast worden. De configuratiemogelijkheden van views zijn oneindig. Daarom is het niet slecht eerst een reeks tutorials te volgen om de belangrijkste opties te leren kennen. Hoewel de meeste opties grotendeels voor zichzelf spreken, kunnen instellingen onder de 'Advanced' tab een view zeer gecompliceerd maken.

De formattering van de lijst zal gebeuren in een horizontaal grid met drie kolommen. We kiezen ervoor om velden apart te selecteren en in te stellen. Elk gekozen veld kan uitgebreid geconfigureerd worden. Een veld kan bepaalde stijlinstellingen bevatten of herschreven worden.



Figuur 7.11: Configuratie van een veld in views Team.

HOOFDSTUK 7. METHODOLOGIE

Inhoud kan een eenvoudige of een gecombineerde filtering bevatten die gesorteerd kan worden op elk type veld. Onderaan wordt een preview gemaakt van de opgehaalde inhoud en hoe deze zich gaat gedragen.

Home > Administration > Structure > Views
Modify the display(s) of your view below or add new displays.

Displays

View Medewerkers [+ Add](#) [edit view name/description](#)

+ View Medewerkers details

Display name: **View Medewerkers** [clone View Medewerkers](#)

TITLE
Title: **None**

FORMAT
Format: Grid | [Settings](#)
Show: Fields | [Settings](#)

FIELDS [Add](#)
Content: Profielfoto medewerker
Content: Job - Afdeling
Content: Jaren in dienst
Content: Titel

FILTER CRITERIA [Add](#)
Content: Published (Yes)
Content: Type (= Medewerker)

SORT CRITERIA [Add](#)
Content: Post date (desc)

BLOCK SETTINGS
Block name: **None**
Access: [Permission](#) | [View published content](#)

HEADER [Add](#)

FOOTER [Add](#)

PAGER
Use pager: [Display a specified number of items](#) | 5 items
More link: No

CONTEXTUAL FILTERS [Add](#)

RELATIONSHIPS [Add](#)

NO RESULTS BEHAVIOR [Add](#)

EXPOSED FORM
Exposed form style: **Basic** | [Settings](#)

OTHER
Machine Name: **block**
Comment: **No comment**
Use AJAX: **No**
Hide attachments in summary: **No**
Hide contextual links: **No**
Use aggregation: **No**
Query settings: [Settings](#)
Field Language: **Current user's language**
Caching: **None**
Link display: **None**
CSS class: **None**
Theme: **Information**
Block caching: **Do not cache**

Auto preview [Preview with contextual filters](#) [Update preview](#)
Separate contextual filter values with a "/". For example, 40/12/10.

Content [@](#)

Content @

anton
plaatsing - 27 werkjaren bij Foré

peter
plaatsing - 14 werkjaren bij Foré¹

elsje
afwerking - 17 werkjaren bij Foré

Figuur 7.12: Overzicht van de view Team

De view is aangemaakt en staat klaar in een block om getoond te worden op een pagina. Volgens het ontwerp bestaat de pagina naast een oplijsting van teamleden ook uit een banner met bijhorende afbeelding. Voor de banner maken we een apart content type en volgen hetzelfde proces als voor teamleden. Er wordt een content type aangemaakt opdat een banner op verschillende pagina's komt te staan. Om de twee blokken inhoud nu in elkaar te smelten gebruiken we de modules page manager in combinatie met panels.

Ook panels is een uitgebreide module waar hier slechts het topje van de ijsberg wordt van gebruikt. Wij gebruiken het om twee views samen te laten komen op één pagina.

Op de homepagina werd gebruik gemaakt van een eigen gecreëerde layout. Volgens het ontwerp kan je zien dat de homepagina bestaat uit een grid systeem dat niet alledaags is en geen standaard layout keuze is bij panels.

The screenshot shows the Drupal administration interface for a page titled "Page Het Team". The left sidebar has a "Variants" section selected under "Panel Het Team". The main content area displays "Content" items. There are two views: "Center" and "View Het Team". The "Center" view contains one item labeled "Het Team". The "View Het Team" view contains one item labeled "No title". At the bottom, there are buttons for "Update", "Update and save", and "Update and preview".

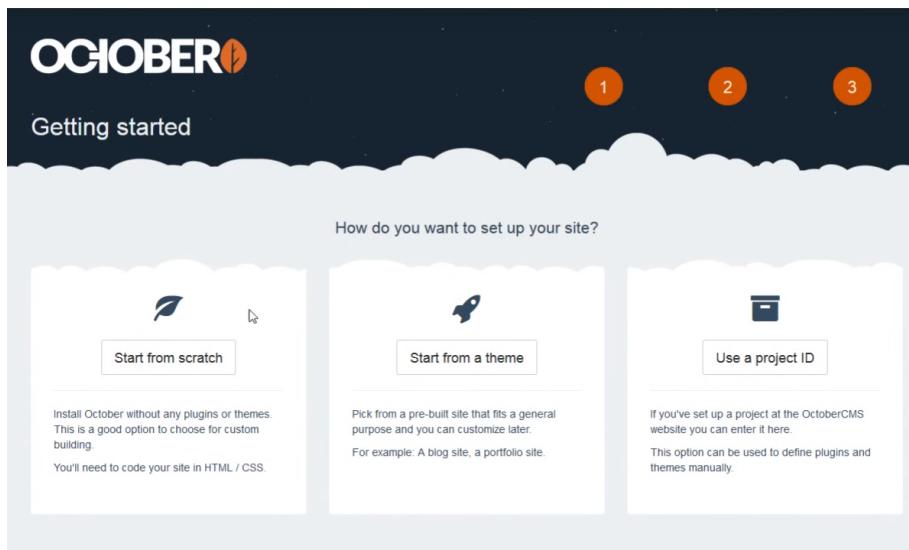
Figuur 7.13: Team panel pagina, samenvoegen van twee views op één pagina.

7.2 OctoberCMS

7.2.1 Installatie

De installatie van OctoberCMS kan op twee manieren. Clone via Github het project of download een installer pakket. De eenvoudigste manier van installatie is via de installer te werken. Deze manier levert hetzelfde resultaat op, echter met een minimum aan inspanning. Start een lokale server op en maak een database aan. Zorg ervoor dat je over de nodige rechten beschikt vooraleer de installatie van start te laten gaan. Dit kan anders problemen geven bij de connectie van de database. Via de `install.php` file in de browser wordt je begeleid door de installatie van OctoberCMS. Via deze stapsgewijze configuratie is het CMS in geen tijd geïnstalleerd en klaar voor gebruik.

Een belangrijke configuratie om in acht te nemen is de opbouw van de site. Er zijn namelijk drie manieren om de site te configureren. De minst vooraf geconfigureerde optie is te starten van 'scratch'. Deze optie wordt geïnstalleerd met een basis thema dat minimale opmaak heeft. Een demo plugin zorgt voor wat opvulling van de site. Een tweede mogelijkheid is te kiezen voor vooraf geconfigureerd thema met daar bijhorende plugins. Deze optie is een goed startpunt om OctoberCMS te leren kennen of om snel een blog of portfoliosite op te stellen. De laatste optie is te kiezen om een bestaand project te importeren. Handig indien je bestaande plugins en thema's wil importeren vanaf de installatie.



Figuur 7.14: OctoberCMS site configuratie

Als ontwikkelaar is het in de meeste gevallen de beste optie om te kiezen om te starten van 'scratch'. Hierbij wordt er gestart van een 'schone lei' en bevat de installatie enkel

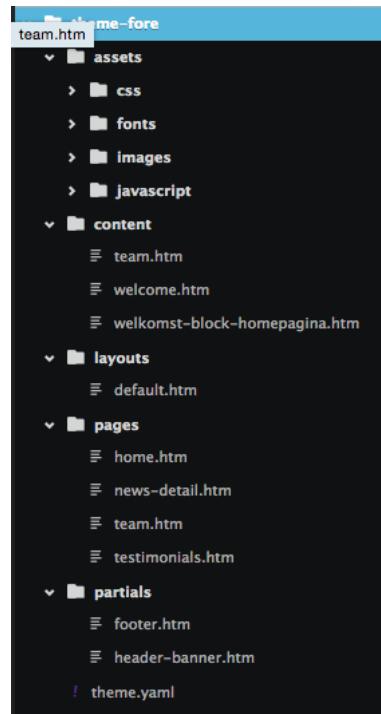
code die nodig is.

Nadat de installatie is afgerond, is het belangrijk om veiligheidsredenen de installatiebestanden te verwijderen.

Het doorlopen van de installatie neemt hooguit 5 minuten in beslag. Het doorlopen van dit proces kan gestaafd worden aan de hand van volgende screencast (Learn Together, 2016).

7.2.2 Theming

Een thema is de basis van output naar de gebruiker toe. Elk thema bevat verschillende sub folders die de website zijn huidige look en feel zal geven. Een standaard installatie bevat een bestaand thema. Dit thema later kan worden overschreven of vervangen kan worden door een eigen thema. Er kan eenvoudigweg van thema worden geswitcht via de back-end in de browser of door de 'activeTheme' instelling aan te passen in het cms.php bestand. Dit bestand is terug te vinden in de 'config' folder. De opbouw van een thema ziet er als volgt uit:



Figuur 7.15: OctoberCMS folder structuur

Elk thema heeft sub folders voor assets, content, layouts, pages en partials. Het theme.yaml bestand bevat de thema configuratie. Zaken als naam, omschrijving, auteur en code van het thema, worden in dit bestand beschreven. Bestanden krijgen de extensie .htm en maken gebruik van Twig markup. Twig is één van de vele PHP template engines. Twig richt zich op propere en overzichtelijke code en wordt bij het uitvoeren omgezet in geoptimaliseerde PHP code.

Elke template pagina kan bestaan uit drie onderdelen. Een blok voor configuratie, PHP code en Twig code. Elk van deze blokken worden van elkaar onderscheiden door twee '=' symbolen. De configuratieblok bestaat uit een route of URL naam en een eventuele layout template. Een layout template specificeert de layout van een pagina. Deze pagina zorgt ervoor dat gelijke stukken code die over verschillende pagina's heen gebruikt worden, slechts één keer aangemaakt moeten worden. Het PHP code blok is optioneel en bevat functies die steeds uitgevoerd worden alvorens de pagina gerenderd wordt. Deze code wordt omgezet naar een aparte PHP file bij het compileren. Het derde blok bevat de Twig markup code die gerenderd en gecompileerd wordt naar html in de browser. Het Twig gedeelte bevat alle code die de zichtbare elementen op de pagina zullen weergeven aan de gebruiker.

```
1   title = "Fore"
2   url = "/"
3   layout = "default"
4   ==
5
6   function onStart()
7   {
8       $this['first_newsItem'] = Db::table('dieter_news_newsitem')->first();
9       $this['newsItems'] = Db::table('dieter_news_newsitem')->skip(1)->take(2)->get();
10    }
11
12 ==
13
14 <figure class="banner-image-wrapper">
15     <div class="image">
16         
17     </div>
18     <figcaption class="banner-image">
19         <div class="container">
20             <div class="desc">
21                 {% content 'welcome.htm' %}
22             </div>
23         </div>
24     </figcaption>
25 </figure>
```

Figuur 7.16: Home template pagina

Naast een layout template onderscheiden we ook partials. Herbruikbare elementen die beschreven worden in een aparte pagina. Deze kunnen toegevoegd worden aan andere pagina's. Vaak voorkomende elementen die in een partial worden gestopt zijn een footer en header. Het toevoegen van een partial op een pagina is eenvoudig. Roep via de Twig annotatie de partial op en de daarbij horende bestandsnaam. Er wordt automatisch gezocht in de partial folder van het thema naar de bestandsnaam.

```
<!-- Footer -->
<footer class="page-footer">
    {% partial "footer" %}
</footer>
```

Figuur 7.17: Partial implementatie

Content blocks bevatten een statische inhoud en kunnen bestaan uit tekst, HTML of Markdown. Markdown is een bestandstype dat tekst zonder opmaak bevat en wordt omgezet naar HTML (Gruber John, 2004). Een content block kan onafhankelijk van de pagina waarin het is toegevoegd, veranderd worden. Een content sectie kan heel eenvoudig door de eindgebruiker in de back-end van de browser aangeroepen en aangepast worden. Een content block kan op twee manieren worden aangemaakt. Ofwel via de back-end van de browser, ofwel door een .htm document aan te maken in de 'content' folder van het thema.

De folder met assets kan allerhande subfolders en documenten bevatten. De standaard opbouw van een assets map bevat een folder voor alle CSS documenten, een images folder voor alle gebruikte afbeeldingen en een javascript folder als verzamelmap voor alle script documenten.

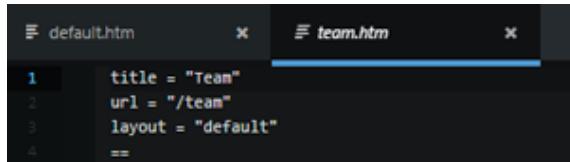
7.2.3 Menu en routing

Routing

Routing in OctoberCMS is opgesplitst in een routing voor back-end controllers en CMS pagina's. De routing van de back-end controllers gebeurt afzonderlijk per plugin. Plugins worden hieronder uitgebreid besproken. Daarom is het momenteel voldoende te weten dat de controller klasse communiceert met de views pagina's via de naam toegewezen aan de views pagina. De routing voor de standaard pagina's index, create, preview en update gebeuren automatisch. Het is uiteraard mogelijk deze automatische routing te overschrijven.

De routing van de CMS pagina's zit logisch en eenvoudig in elkaar. Een pagina krijgt

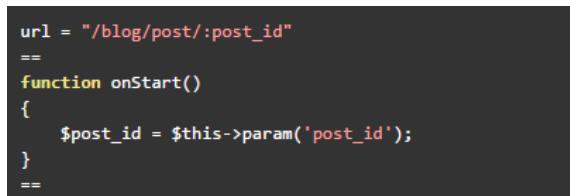
een URL toegewezen in het configuratie gedeelte van elke pagina. Elke pagina URL wordt voorafgegaan door een slash (/). Het voorbeeld hieronder weergegeven is de meest eenvoudige vorm van routing.



```
default.htm          *      team.htm          *
1 title = "Team"
2 url = "/team"
3 layout = "default"
4 ==
```

Figuur 7.18: URL routing

Het is mogelijk om een URL op te bouwen aan de hand van bijgevoegde parameters. Deze parameters worden aangeroepen in de PHP sectie van de pagina. Op deze manier kan de pagina dynamisch aangemaakt worden afhankelijk van de waarde van de parameter. Een parameter kan optioneel zijn wanneer die zich op het einde van de URL bevindt. Een optionele parameter wordt aangegeven via een vraagteken achter de parameter.



```
url = "/blog/post/:post_id"
==
function onStart()
{
    $post_id = $this->param('post_id');
}
==
```

Figuur 7.19: URL routing met parameter
(Bobkov Alexey, 2016)

Menu

In de meeste gevallen is een menu een statisch gegeven. Menu items behoren niet tot de dagelijkse aanpassingen van een site. OctoberCMS opteert dan ook standaard een menu statisch te implementeren. Een menu wordt het beste rechtstreeks in de layout pagina gezet of aangemaakt als partial.

Twig beschikt over tags, filters en variabelen die aangeroepen kunnen worden binnen een pagina. Bij het linken van een pagina gebruik je de '|page' filter. Deze filter maakt een link gebruikmakend van de pagina naam. De gegeven naam linkt naar de opgegeven URL in het configuratie gedeelte van de pagina (zie Routing).

```
<nav class="main-nav" role="navigation">
    <span class="trigger"><i class="icon-hamburger" aria-hidden="true"></i><span>Menu</span></span>
    <ul class="nav-wrap">
        <li class="({% if this.page.id == 'home' %}active{%- endif %})"><a href="{{ 'home'|page }}"/> Home </a></li>
        <li class="({% if this.page.id == 'projects' %}active{%- endif %})"><a href="{{ 'projects'|page }}"/> Projecten </a></li>
        <li><a href="{{ 'home'|page }}></a></li>
        <li class="({% if this.page.id == 'team' %}active{%- endif %})"><a href="{{ 'team'|page }}"/> Team </a></li>
        <li class="({% if this.page.id == 'testimonials' %}active{%- endif %})"><a href="{{ 'testimonials'|page }}"/> Getuigenissen </a></li>
    </ul>
</nav>
```

Figuur 7.20: Code voor het menu, geïmplementeerd in de layout pagina.

Het is mogelijk een vaste parameter door te geven aan de link om bijvoorbeeld te linken naar een post met bepaalde ID. Parameters worden tussen haken na de '|page' filter geplaatst.

```
<a href="{{ 'post'|page({ post_id: 10 }) }}>
    Blog post #10
</a>
```

Figuur 7.21: Menu met parameter
(Bobkov Alexey, 2016)

Voor gevallen waarbij een menu aanpasbaar moet zijn voor de eindgebruiker heeft OctoberCMS een oplossing geboden. Een plugin genaamd MenuManager zorgt ervoor dat het managen van uw menu in de back-end via de browser kan beheerd worden. Omdat dit eerder een specifiek en minder voorkomend gebeuren is, wordt dit niet verder getest en besproken.

7.2.4 Plugins en modules

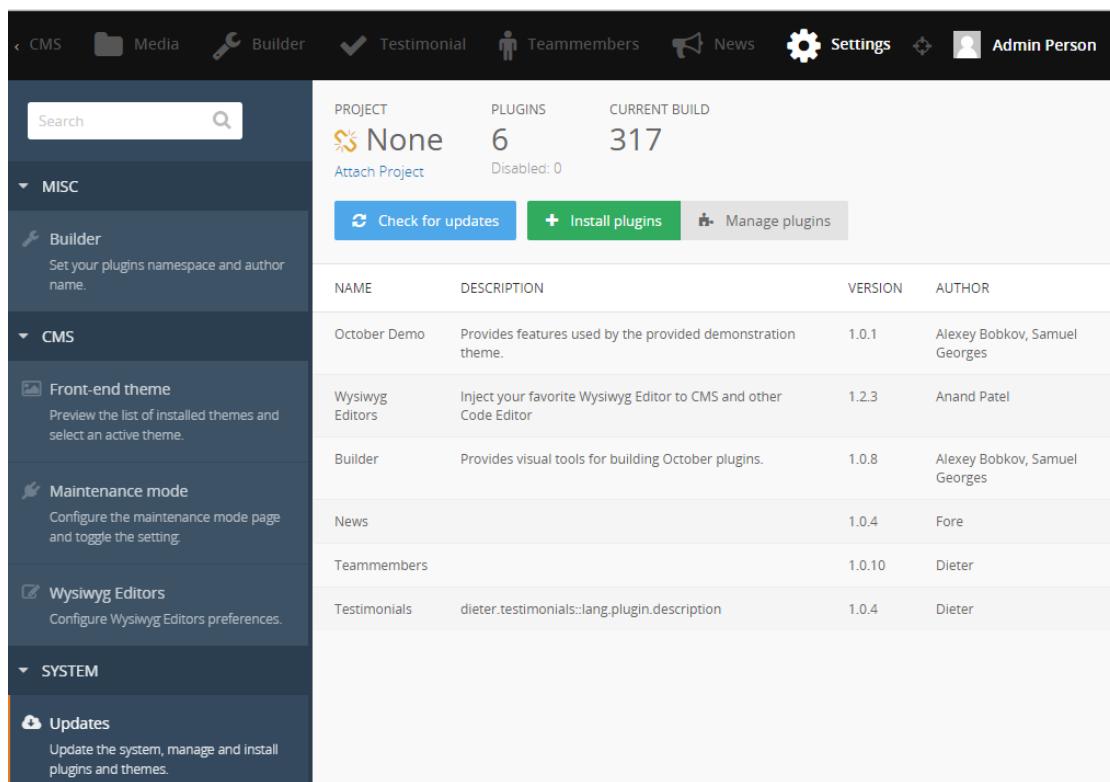
Plugins kunnen gezien worden als de belangrijkste elementen voor het uitbreiden van OctoberCMS. Elke plugin komt in de map '/plugins' en heeft een vaste mappenstructuur. Een plugin kan verschillende elementen definiëren. Back-end pagina's en menu's, interactie met andere Plugins, components, enz.

Organisatie Plugins

Overzicht

Standaard wordt bij een installatie van OctoberCMS, ook bij een installatie 'from scratch', een plugin geïnstalleerd. Deze plugin doet enkel dienst als demo. Een overzicht van alle geïnstalleerde plugins kan je zien aan de hand van je mappenstructuur. De visuele weergave in de back-end van de browser biedt meer overzicht. Een oplijsting in tabelvorm waarbij je alle plugins met hun belangrijkste informatie kan bekijken. Deze lijst is terug te vinden op de Update pagina onder instellingen.

HOOFDSTUK 7. METHODOLOGIE



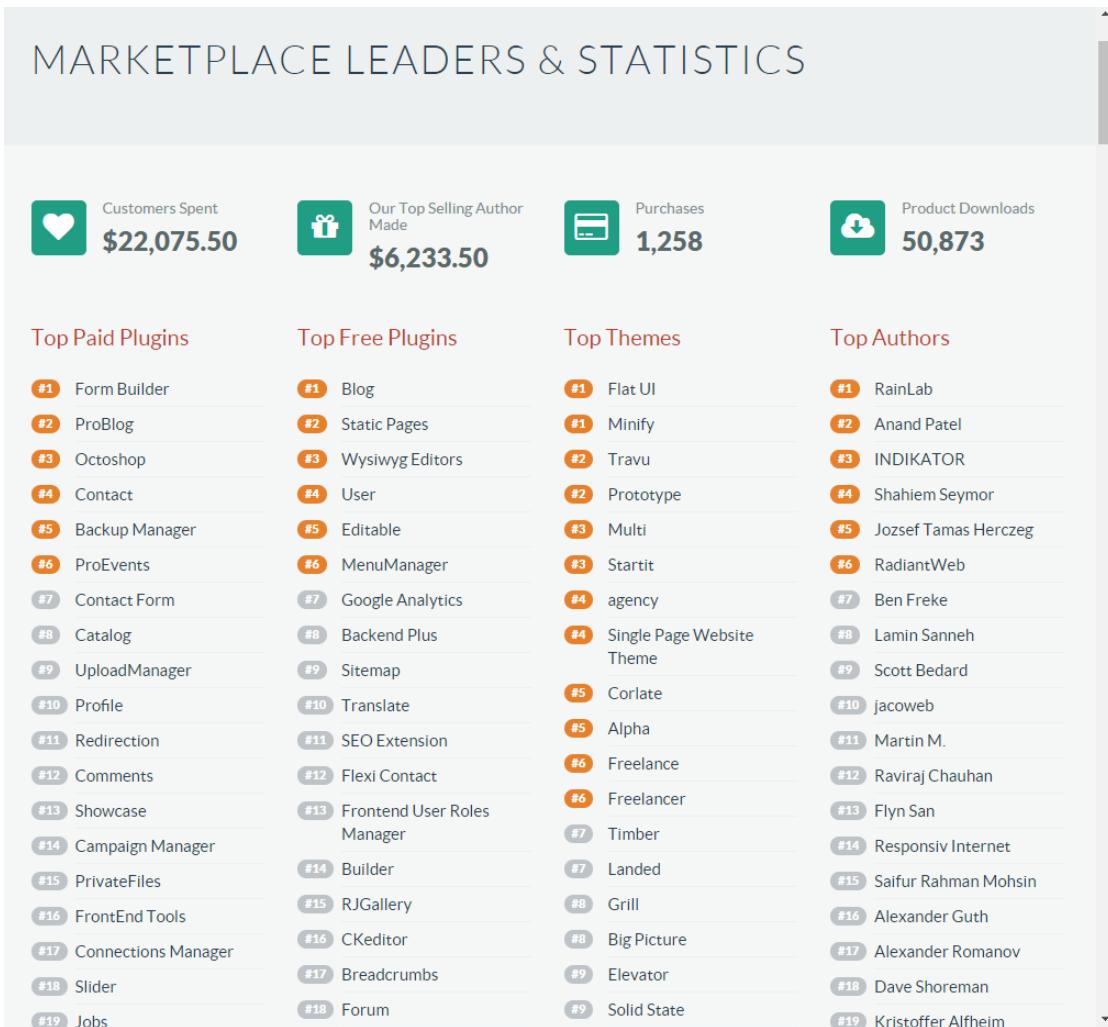
The screenshot shows the OctoberCMS admin interface for managing plugins. At the top, there's a navigation bar with links for CMS, Media, Builder, Testimonial, Teammembers, News, Settings, and Admin Person. Below the navigation is a search bar and a sidebar with sections for MISC, CMS, and SYSTEM. The main content area displays a summary of the project (None), 6 installed plugins, and a current build count of 317. It includes buttons for 'Check for updates', 'Install plugins', and 'Manage plugins'. A table lists the installed plugins with columns for NAME, DESCRIPTION, VERSION, and AUTHOR.

NAME	DESCRIPTION	VERSION	AUTHOR
October Demo	Provides features used by the provided demonstration theme.	1.0.1	Alexey Bobkov, Samuel Georges
Wysiwyg Editors	Inject your favorite Wysiwyg Editor to CMS and other Code Editor	1.2.3	Anand Patel
Builder	Provides visual tools for building October plugins.	1.0.8	Alexey Bobkov, Samuel Georges
News		1.0.4	Fore
Teammembers		1.0.10	Dieter
Testimonials	dieter.testimonials::lang.plugin.description	1.0.4	Dieter

Figuur 7.22: Overzicht geïnstalleerde plugins

Marketplace

De Marketplace van OctoberCMS is een opslagplaats van alle gepubliceerde plugins en thema's. Elke ontwikkelaar kan zich registeren op de Marketplace en op die manier thema's en plugins publiceren. Bij het publiceren van een plugin of thema wordt deze eerst gescreend en onderworpen aan testen die op zoek gaan naar bugs of fouten. Een publicatie kan je gratis of tegen betaling ter beschikking stellen. De Marketplace is een groeiend platform waar er momenteel 317 plugins te verkrijgen zijn (geraadpleegd op 05/05/2016). Als ontwikkelaar kan je ook een aardig centje bijverdienen indien je plugin goed gebouwd en een nuttige aanvulling is. Als we kijken naar statistieken die te raadplegen zijn op de Marketplace, zien we een klein overzicht van de populairste plugins, thema's en auteurs.



Figuur 7.23: Overzicht populaire plugins, thema's en auteurs op de MarketPlace (ge-raadpleegd op 05/05/2016).

Installeren

Alvorens een plugin te installeren dien je eerst een project aan te maken op je account in OctoberCMS. Dit project kan je koppelen met een unieke sleutel aan je site. Dit proces doorlopen is niet nodig wanneer je lokaal werkt.

Installeren van een bestaande gratis plugin kan vanuit de site zelf. Via de knop 'Install Plugins' is het mogelijk om plugins te zoeken en te installeren die gestockeerd zijn op de Marketplace. Plugins die betalend zijn moeten eerst aangekocht worden. Achteraf kunnen deze worden toegevoegd aan een project.

Ontwikkelen

Naast de vele plugins die te vinden zijn op de Marketplace kan het voorkomen dat iets 'custom built' moet gemaakt worden. Wanneer beschikbare plugins niet voldoen aan de wensen van de klant, moet een nieuwe plugin ontwikkeld worden.

Ontwikkelen van een nieuwe plugin kan op drie manieren aangepakt worden. Elke plugin wordt in een folder aangemaakt met een unieke auteursnaam.

Een plugin kan volledig manueel aangemaakt worden. Dit is de meest arbeidsintensieve manier en zeker niet de eenvoudigste. Hierbij maak je elke file manueel aan met de daarbij horende code. Een interessantere manier is te werken via de command-line-interface die gebruikt wordt bij Laravel. Artisan bevat een lijst met handige commando's die het ontwikkelen eenvoudiger en sneller maakt. Het aanmaken van een plugin met een standaard opmaak van folders en files gaat eenvoudig via volgend commando:

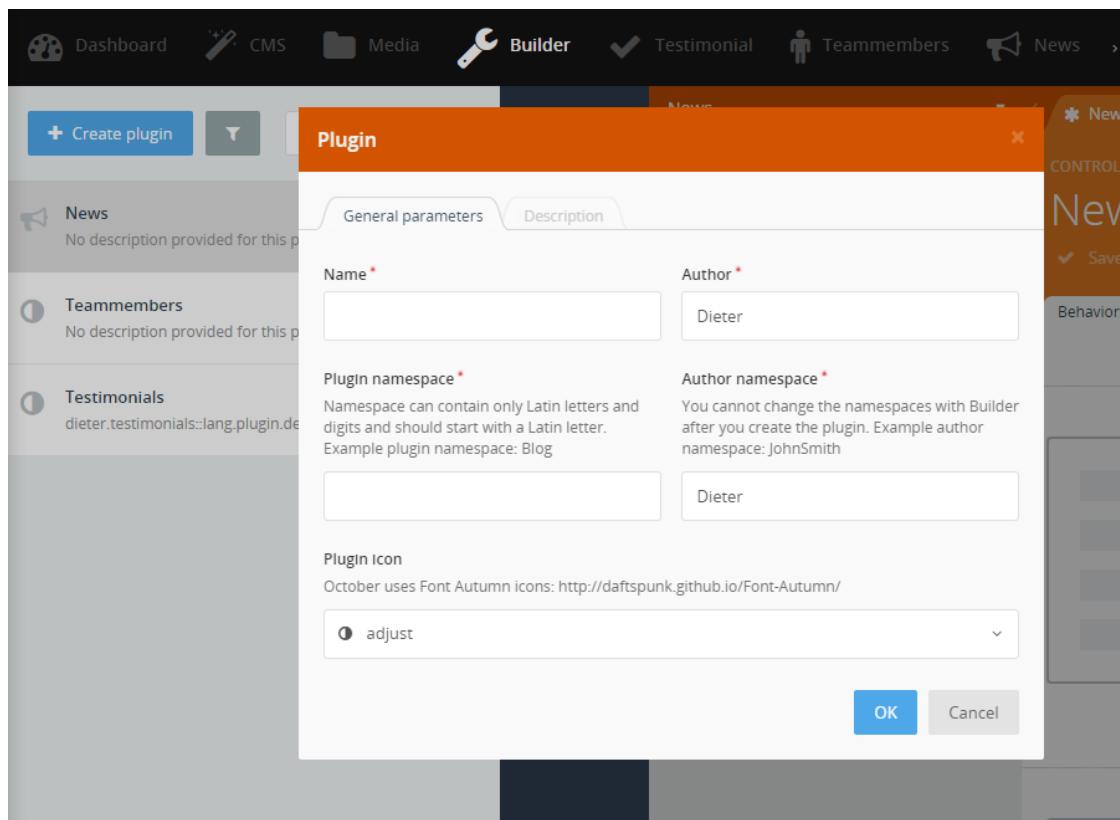
```
php artisan create:plugin dieter.teammembers
```

Hierbij wordt een upgrades folder aangemaakt met een version.yaml bestand. Deze folder en file houden de veranderingen van de plugin bij. Daarnaast hebben we een bestand Plugin.php waarin de algemene configuratie van de plugin in vermeld staat. Een plugin kan verder opgebouwd worden met models-, controllers-, language-, en classes folders, afhankelijk van de functie van een plugin.

Het aanmaken van deze mappen en files kunnen allemaal via Artisan. Artisan biedt commando's voor elk van deze opdrachten.

De laatste manier om een plugin aan te maken en te configureren is via een bestaande plugin. De Builder plugin is een relatief nieuwe plugin die gelanceerd is op 10 februari 2016 door de makers (Rainlab) zelf. Het verkort en vereenvoudigt het ontwikkelen van een plugin. Het is een ideale manier om een plugin snel op te zetten en achteraf aanpassingen te doen in code of via de Back-end in de browser. De Builder plugin is ontwikkeld om de repetitieve activiteiten te automatiseren zonder de controle te verliezen. Op deze manier is er meer tijd vrij voor de belangrijkere business logica.

Via een chronologische structuur van de Builder plugin doorloop je de verschillende stappen en heb je in geen tijd een tabel opgebouwd die gelinkt is aan een model. Controllers geven je de mogelijkheid data in te voeren en te beheren. Het invoeren gaat via een formulier. Het beheren en visualiseren is via een lijst. Beide controllers kunnen volledig geconfigureerd worden. Op ieder moment is het mogelijk om te schakelen naar een editor om de nodige aanpassingen te doen in code.



Figuur 7.24: Het aanmaken van een nieuwe plugin via de Builder plugin is eenvoudig. Via het pop-up venster geef je een unieke naam en de auteur in. Personaliseer met een icoon.

De Builder plugin biedt ook een permissie systeem waarbij de toegang wordt beschreven voor verschillende gebruikers of gebruikersgroepen. De tab back-end menu laat je toe de plugin weer te geven bovenaan de taakbalk. Een menu kan bestaan uit verschillende sub items.

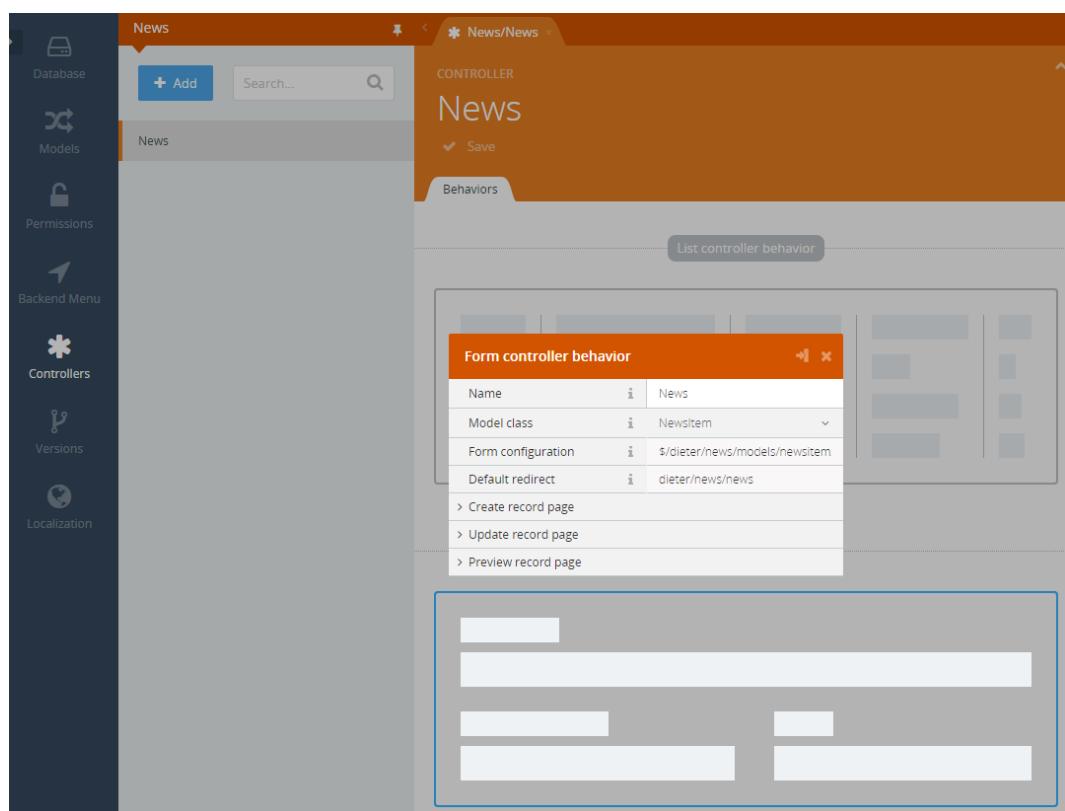
HOOFDSTUK 7. METHODOLOGIE

The screenshot shows the DIETER CMS interface for creating a new database table. The left sidebar has icons for Database, Models, Permissions, Backend Menu, Controllers, Versions, and Localization. The main area is titled 'News' and shows a table named 'dieter_news_newsitem'. The 'Columns' tab is selected, displaying a grid of columns with their types, lengths, and various settings like UNSIGNED, NULLABLE, AUTOINCR, PK, and DEFAULT. The columns listed are: id (Integer, 10, UNSIGNED, PK), title (String, 255), short_des... (String, 255), body (Text, 65535), created_at (Timestamp), updated_at (Timestamp), and slug (String, 255).

Figuur 7.25: Maak via de tabel de nodige kolommen aan. Eenmaal opgeslagen wordt een query uitgevoerd op de database.

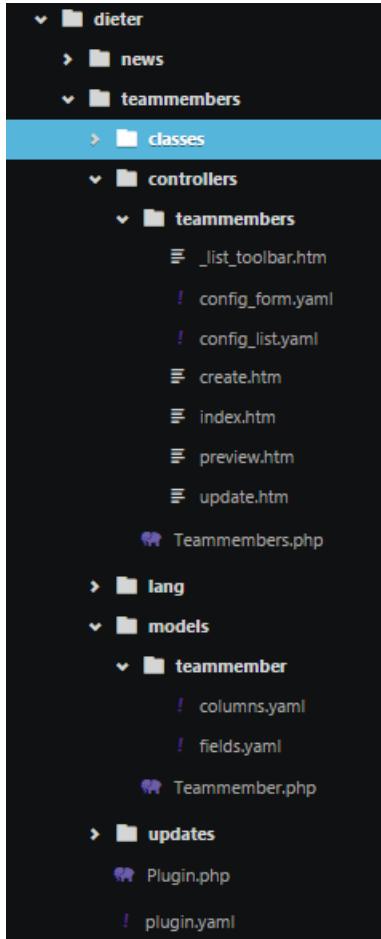
The screenshot shows the DIETER CMS interface for configuring a model field. The left sidebar includes Localization. The main area is titled 'News' and shows a file named 'fields.yaml' under 'Newsitem/forms'. The 'Title' field is being configured, showing a 'Text' control type with various properties like Field name (title), Label (dieter.news:lang.newsitem.title), Comment, Comment position (Auto), Span (Auto), Placeholder, Default, CSS class, Disabled, Hidden, Required (checked), Context ([Create, Update, Preview]), and Advanced. A 'Single line text box' input field is also visible.

Figuur 7.26: Configureer een model. Vul de nodige velden aan en link ze met tabelnaam uit de database. Configureer waar nodig.



Figuur 7.27: Configureer en link de models aan de controller klasse.

Wanneer het volledige proces van de Builder plugin doorlopen is, verkrijgen we bijvoorbeeld volgende mappenstructuur:



Figuur 7.28: Mappenstructuur plugins

Plugins kunnen ook components of componenten bevatten. Components kunnen gezien worden als intelligente configureerbare blokken. Deze componenten kunnen aan allerhande pagina's worden toegevoegd waardoor er een extra functionaliteit aan de pagina wordt toegevoegd.

7.2.5 Opbouw pagina's

De opbouw van pagina's geeft een korte toelichting van de werkwijze die gehanteerd werd bij de uitwerking van de casus. Het is niet de bedoeling iedere pagina volledig gedetailleerd uit te schrijven. Daarom worden enkel volgende pagina's besproken. Zo wordt de volgorde en manier van werken duidelijk.

- Opbouw van de layout pagina, partials en content blocks
- Aanmaken van een pagina met bijhorende plugin
- Beheersbaarheid van pagina's

Opbouw van de layout pagina, partials en content blocks

Zoals eerder werd aangehaald is een layout pagina een handige tool om de basisstructuur van een pagina op te bouwen. Zo hoef je niet elke keer opnieuw weerkerende elementen op iedere pagina te definiëren. De default.htm layout pagina begint met een 'head' sectie waarin onder andere een paginatitel, meta-tags en CSS stijlen kunnen aangeroept worden.

```
<head>
    <meta charset="utf-8">
    <title>{{ this.page.title }}</title>
    <meta name="description" content="{{ this.page.meta_description }}">
    <meta name="title" content="{{ this.page.meta_title }}">
    <meta name="author" content="OctoberCMS">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="generator" content="OctoberCMS">
    <!-- <link rel="icon" type="image/png" href="{{ 'assets/images/october.png'|theme }}" / -->
    {% styles %}
    <link href="{{ [
        'assets/css/style.css'
    ]|theme }}" rel="stylesheet">
</head>
```

Figuur 7.29: De 'head' structuur van de layout pagina

Het meest interessante stukje code uit deze sectie zal de injectie van de CSS zijn. Stijlen kunnen rechtstreeks worden toegevoegd. Wanneer een stijl op een bepaalde pagina voorkomt, kan deze in de PHP sectie van de pagina worden geïmplementeerd. Een andere manier is gebruik te maken van Twig tags.

```
function onStart()
{
    $this->addCss('assets/css/hello.css');
```

Figuur 7.30: Injectie van CSS via de onStart functie in de PHP sectie van de pagina.
(Bobkov Alexey, 2016)

```
{% put styles %}
  <link href="/themes/demo/assets/css/page.css" rel="stylesheet" />
{%- endput %}
```

Figuur 7.31: Injectie van CSS door gebruik te maken van Twig markup.
(Bobkov Alexey, 2016)

Verder onderscheiden we een navigatie die eerder al besproken is. Het actief zetten van een navigatie item is aan de hand van een IF-statement. Het controleert voor ieder menu item of het al dan niet de pagina's huidige ID is. Als dit het geval is, wordt de klasse 'active' toegevoegd.

```
<li class="{% if this.page.id == 'home' %}active{% endif %}"><a href="{{ 'home'|page }}"/> Home /</a></li>
```

Figuur 7.32: If-statement om na te gaan of een menu item actief is.

Binnen de 'main' sectie wordt de inhoud van de desbetreffende pagina ingeladen. Deze inhoud zal dus verschillend zijn voor elke pagina. Hoe deze inhoud geïmplementeerd wordt is zeer eenvoudig. In de configuratie sectie van iedere pagina is het mogelijk een template in te geven aan de hand van zijn document naam. Zo geef je voor elke gewenste pagina bijvoorbeeld default als layout template. De pagina zal opgebouwd worden aan de hand van de default layout pagina. Inhoud van deze pagina wordt opgevuld daar waar de Twig tag 'page' staat.

```
<!-- Content -->
<section class="main">
  {% page %}
</section>
```

Figuur 7.33: De Twig tag zal vervangen worden door de inhoud van desbetreffende pagina.

Bij de pagina footer wordt gebruik gemaakt van een partial. Zoals eerder werd besproken is dit een herbruikbaar stuk code die in een aparte file wordt bewaard. Het opdelen van code zorgt ervoor dat een pagina overzichtelijk blijft.

Aanmaken van een pagina met bijkomende plugin

Er wordt slechts één proces doorlopen voor het aanmaken van een pagina met bijkomende plugin. De team pagina geeft een overzicht van de verschillende werknemers met bijgevoegde foto, naam, functie en aanwervingsjaar. Verder bevat de pagina een statische banner met een content block, die in de back-end via de browser kan worden aangepast door de eindgebruiker.

Het proces begint met het aanmaken van een plugin. Dit kan via verschillende manieren, hier werd geopteerd om de Builder plugin te gebruiken. Sommige stappen werden reeds via woord en beeld duidelijk gemaakt in vorig hoofdstuk, daarom zullen reeds aangehaalde stappen slechts kort besproken worden.

Begin met het aanmaken van een plugin door het een naam, auteur, beschrijving en icoon te geven. Vervolgens kan je voor de geselecteerde plugin een database tabel aanmaken. Iedere keer je de tabel gaat opslaan komt een pop-up venster tevoorschijn met genereerde query die uitgevoerd wordt op de database. Zo worden alle aanpassingen op deze tabel gedaan.

COLUMN	TYPE	LENGTH	UNSIGNED	NULLABLE	AUTOINCR	PK	DEFAULT
id	Integer	10	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
name	String	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
job	String	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
hiredate	Date		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
imageurl	String	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figuur 7.34: Voeg de nodige rijen en vul de kolommen in waar nodig.

De volgende stap is het aanmaken van de model klasse. Geef het model een naam en selecteer de database tabel. Het model is opgedeeld uit 'Form fields' en 'List fields'. De 'Form fields' of formuliervelden, zijn velden die weergegeven en geconfigureerd worden voor de invoer van nieuwe data. De 'List fields' of lijstvelden geven een overzicht van alle aangemaakte data weer in een lijst.

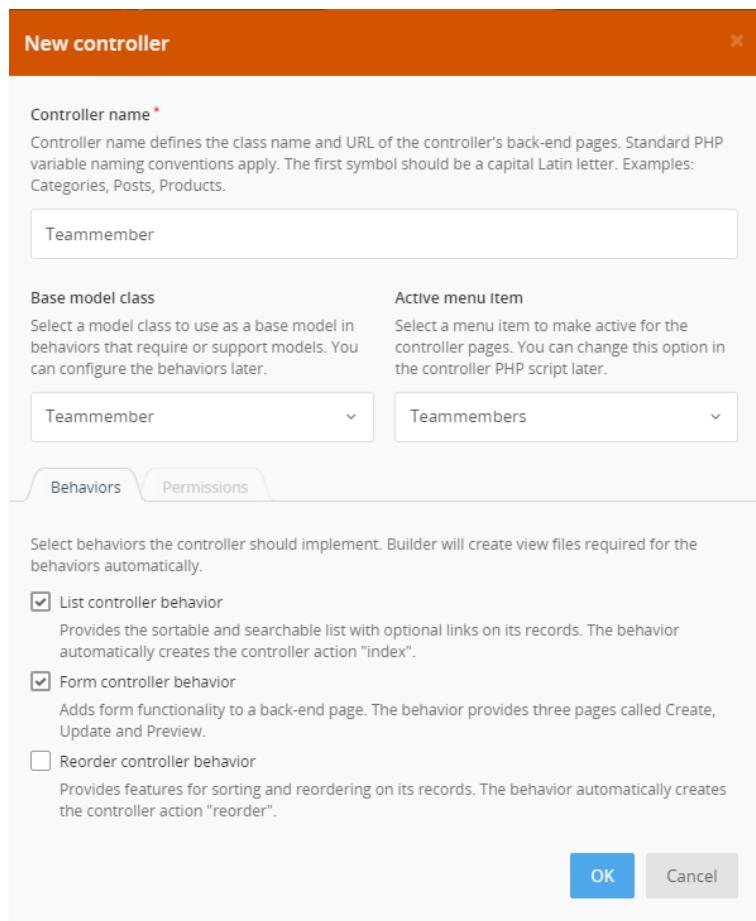
Bij het aanmaken van de formuliervelden kan je allerhande velden toevoegen. Gaande van tekstvelden, dropdown velden tot file upload velden. Eenmaal een type geselecteerd kan de configuratie van dit veld gebeuren. Ieder veld heeft een 'Field name' die een naam krijgt gerelateerd aan de tabel. Andere velden zijn veelal verschillend, afhankelijk van het data type. De meeste velden spreken voor zich, zo niet zorgt het informatie icoontje voor bijhorende informatie.

```
default.htm          / fields.yaml
1   fields:
2     name:
3       label: 'dieter.teammembers::lang.teammember.name'
4       oc.commentPosition: ''
5       span: auto
6       type: text
7     job:
8       label: 'dieter.teammembers::lang.teammember.job'
9       oc.commentPosition: ''
10      span: auto
11      type: text
12    hiredate:
13      label: 'dieter.teammembers::lang.teammember.hiredate'
14      oc.commentPosition: ''
15      mode: date
16      span: auto
17      type: datepicker
18    imageurl:
19      label: 'dieter.teammembers::lang.teammember.image_url'
20      mode: image
21      span: left
22      type: mediafinder
23
```

Figuur 7.35: De configuratie van alle velden is terug te vinden in de 'fields.yaml' file.

De werking voor lijstvelden is gelijklopend met deze van de formulervelden. Specifieke permissies werden niet toegevoegd, aangezien we slechts beschikken over één gebruiker. Maak nu een back-end menu item aan met een gepaste naam en icoon. De URL vullen we pas aan eenmaal de controller is aangemaakt.

We voegen een controller toe en geven hem een naam. We selecteren een model klasse als ook een menu item. Onder de 'Behaviors' tab selecteren we de checkboxen 'List- en Form controller behaviour'. Deze opties zorgen voor de aanmaak van de standaard controller pagina's: creëren, updaten, preview en overzicht.



Figuur 7.36: Het pop-up venster voor de aanmaak van een controller.

Nu de controller is aangemaakt kunnen we de URL instellen bij het menu item. Dit waren de stappen bij het opstellen van een plugin via de Builder plugin. Hoewel dit een standaard configuratie van een plugin is, kan nu in code via een editor nog business logica worden toegevoegd. Er kunnen bijvoorbeeld nog aanpassingen volgen aan de modelklasse, controller klasse en eventuele toevoegingen van andere back-end pagina's.

De team pagina heeft verder geen aanpassingen nodig, waardoor we rechtstreeks een front-end pagina kunnen aanmaken. Voeg een .htm file toe aan de 'pages' folder en vul de configuratie sectie in met naam, URL en Layout. De PHP sectie krijgt een onStart functie mee waarin we alle team leden gaan ophalen uit de database.

```
$this['members'] = Db::table('dieter_teammembers_teammembers')->get();
```

Het is nu mogelijk om de verzameling van leden te overlopen in de Twig sectie van de

HOOFDSTUK 7. METHODOLOGIE

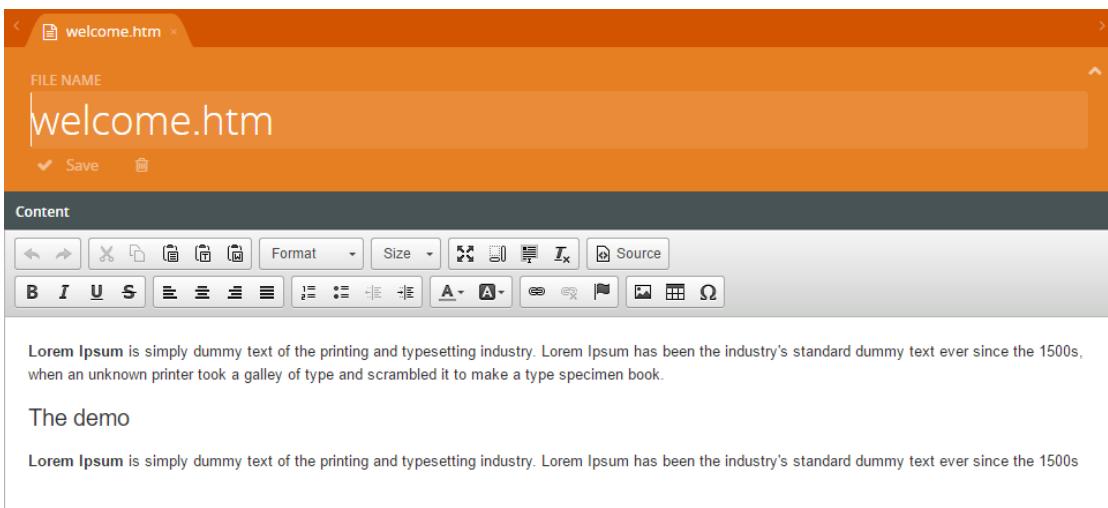
pagina. Overlopen van een collectie of verzameling doen we met een for-lus. Via de dot notatie (.) kunnen we een attribuut aanspreken van een teamlid.

```
<div class="team">
  <div class="row g0">
    {% for member in members %}
      <div class="one-third">
        <div class="member" style="background-image: url('{{ member.imageurl|media }}');">
          
          <div class="overlay-wrapper">
            <div class="desc">
              {% if member.hiredate != '0000-00-00' %}
                <p class="bold">
                  {{ member.name }}
                </p>
                <p>
                  {{ member.job }} - bij Foré sinds <span class="bold">{{ member.hiredate|date('Y') }}</span>
                </p>
              {% else %}
                <p>
                  <span class="bold">{{ member.name }}</span> - {{ member.job }}
                </p>
              {% endif %}
            </div>
          </div>
        </div>
      {% endfor %}
    </div>
  </div>
```

Figuur 7.37: Overloop de collectie via een for-lus

Bovenaan de pagina bevindt zich een banner met statische afbeelding en inhoud die aangevuld is via een content block. Het content block kan de eindgebruiker eenvoudig beheren via een teksteditor in de back-end van de browser.

```
<div class="desc">
  {% content 'welcome.htm' %}
</div>
```



Figuur 7.38: Eenvoudig tekst aanpassen via de browser, zonder enige technische kennis.

Beheersbaarheid van pagina's

We zagen net dat het aanpassen van een content block heel eenvoudig in zijn werk gaat. We overlopen hier hoe we teamleden kunnen aanmaken en beheren.

Via het menu item 'Teammembers' in de taakbalk komen we op een overzicht met alle bestaande teamleden. Via de Create link komen we op de formulierpagina waar we eerder alle velden voor hebben ingesteld. We vullen een naam, job, aanwervingsdatum in en voegen een afbeelding toe via de Media Manager van OctoberCMS. Eenmaal opgeslagen wordt het nieuwe lid weergegeven in de lijst. Een lid updaten kan door erop te klikken, aanpassingen door te voeren en terug op te slaan.

Er is een mogelijkheid om pages, partials, layouts en components aan te passen via de browser, toch is dit iets dat je beter aanpast via een editor en eerder verborgen staat voor de eindgebruiker.

Hoofdstuk 8

Vergelijkende testen en resultaat

Nu de structuur en algemene werking van beide systemen is voorgelegd komen de vergelijkende testen. Deze testen zullen op verschillende punten beide systemen evalueren en vergelijken. Volgende punten zijn belangrijk voor systemen die 'Developer Friendly' zijn. Met andere woorden: punten waar ontwikkelaars belang aan hechten bij het kiezen van een framework of CMS voor hun nieuw project (Siddharth, 2009).

Onderzoeken en staven van volgende punten:

- Leercurve
- Documentatie
- Community
- Performantie
- Beheersbaarheid
- Database structuur

8.1 Leercurve

Elk systeem heeft zo zijn eigen gewoontes en creëert hierdoor een eigen mini-wereld. Zo verschillen systemen in de opbouw van structuur en mappen, naamgevingsconventies, enz. Sommige zijn zeer strikt in gewoontes, andere zeer flexibel. Bij het kiezen van een systeem is het uiteraard belangrijk te kijken naar deze met de minst steile leercurve.

8.1.1 Drupal

Drupal heeft de naam een steile leercurve te hebben. Dit is te nuanceren wanneer we de top van de berg gaan definiëren. Drupal is een relatief eenvoudig systeem om mee van start te gaan. De installatie loopt vlot en na enkele tutorials te volgen moet het mogelijk zijn voor een ontwikkelaar om een eigen thema te creëren, als ook het opstellen van een eenvoudige website.

De eerste overwinning om Drupal onder de knie te krijgen is het begrijpen van de verschillende, meestal nieuwe benamingen. Drupal beschikt over een eigen arsenaal van naamgevingsconventies die uniek zijn voor het systeem (Drupal, 2016a). Deze lijst bevat een volledig overzicht van alle termen die Drupal gebruikt. Velen hiervan zijn algemene termen die verspreid over de IT-wereld voorkomen. Andere zijn zeer specifiek en nieuw voor de meesten. Termen zoals taxonomy, content type, views, blocks en regions zijn typische specifieke termen.

Drupal 7 is opgebouwd uit modules. De volgende stap in het leerproces is het begrijpen en configureren van deze modules. De meeste modules spreken voor zich. Toch bestaan er modules die uitgebreide configuratiemogelijkheden hebben. Modules zoals views en rules zijn modules waar configuratie tot in het oneindige loopt. Het beheren en configureren van deze modules vergt enige kennis en ervaring. Kennis kan worden uitgebreid via talloze boeken en tutorials die online te vinden zijn.

Het bekomen van een resultaat en de weg die gevolgd is, kan steeds op verschillende manieren. Dit is niet anders voor Drupal. Het combineren van verschillende modules tot een werkend en snel geheel kan een uitdaging zijn. Het is geen overbodige luxe op voorhand na te denken welke functies er op de site moeten worden geïmplementeerd en welke module te gebruiken voor welke functie.

Wanneer we een stuk hoger willen komen op de Drupal berg kan dit inderdaad steil worden. Een ontwikkelaar kent zijn systeem pas echt als hij de werking ervan begrijpt, waarom gebeuren dingen op de manier waarop ze gebeuren?

Hoewel er duizenden modules bestaan die een oplossing kunnen bieden voor een bepaald systeem, bestaat de kans dat iets 'custom built' gemaakt moet worden. Hiervoor dien je zelf een module te schrijven of een aftakking van een bestaande module. Het schrijven van een module in Drupal vergt een grondige kennis van de core en werking van het systeem.

Het creëren van een module ligt buiten de algemene werking voor het opbouwen van een site in Drupal. De ontwikkelaar stapt met andere woorden in een andere wereld.

Een volledige oplijsting van een stapsgewijze toer die te volgen is bij het ontwikkelen van een module, is zeer uitgebreid (Drupal (2015)).

8.1.2 OctoberCMS

In tegenstelling tot Drupal 7 heb je voor het starten aan een eenvoudige website met OctoberCMS wel een basiskennis nodig van HTML, CSS, PHP, SQL en de command-line. Met het bezitten van deze genoemde fundamenten is de leercurve van OctoberCMS eerder klein.

OctoberCMS is gebouwd op het PHP framework Laravel. Daarom is het belangrijk de beginselen van Laravel onder de knie te krijgen alvorens van start te gaan met OctoberCMS. Hoewel de folderstructuur en het gebruik van plugins uniek is voor OctoberCMS, is de syntax en werking grotendeels gelijk aan die van Laravel.

De opstart van een project met OctoberCMS is zoals eerder aangehaald, snel en eenvoudig. OctoberCMS is modulair opgebouwd en biedt de ontwikkelaar de keuze zijn site op te bouwen via zelf gemaakte plugins, of bestaande plugins terug te vinden op de Marketplace. Met kennis van MVC (Model-View-Controller) is het bouwen van een eigen plugin eenvoudig. Een plugin kan uiteraard veeleisend en erg uitgebreid worden waardoor een grondige kennis van PHP nodig is. Via de Builder plugin wordt het mogelijk om in de back-end van de browser een eenvoudige plugin op te zetten en deze later te configureren in code.

8.1.3 Besluit

De implementatie van Drupal 7 en OctoberCMS zijn amper te vergelijken. Hoewel ze allebei streven naar hetzelfde resultaat, is de manier waarop geheel anders. Drupal heeft een steile leercurve omdat het algemene werking die een ontwikkelaar normaal hanteert niet aanwezig is bij Drupal 7. Hij kan zijn opgedane kennis in het verleden met andere PHP of Javascript frameworks niet uitspelen in Drupal. Bij OctoberCMS daarentegen kan een ontwikkelaar met kennis van MVC, opgedaan bij andere frameworks, wel aan de slag. De drempel en leercurve voor een ontwikkelaar ligt lager bij OctoberCMS dan bij Drupal 7.

8.2 Documentatie

De helling van de leercurve van een systeem kan veel te maken hebben met de al dan niet aanwezige documentatie. Bij een systeem met een gebrek aan goede documentatie kan het uitzoeken van een probleem te veel tijd en moeite in beslag nemen. Daarom

is het belangrijk alvorens te kiezen voor een bepaald systeem, onderzoek te verrichten daaromtrent. Een goed gedocumenteerd systeem heeft vele voorbeelden, snippet code, artikels en tutorials. Tegenwoordig is het gebruik van screencasts zeer populair. De ideale methode om een leerling visueel en audiovisueel informatie te verschaffen omtrent een onderwerp.

8.2.1 Drupal

Door de jaren heen heeft Drupal een uitgebreide en gestructureerde documentatie kunnen aanleggen (Drupal, 2016b). Deze documentatie is terug te vinden op de eigenlijke site van Drupal. Maar ook via andere kanalen kan je je kennis verruimen. Het spreekt voor zich dat niet alle bronnen even betrouwbaar zijn, een nuchtere en kritische geest is belangrijk.

Online leermogelijkheden:

- Drupal.org (Drupal, 2016b)
- Drupalize (Drupalize-me)
- BuildAModule (BuildAModule, 2016)
- Udemy (Udemy, 2016)

Gratis alternatieven zijn:

- Youtube
- Wunderkraut (Falk Johan, 2015)

Het bijschaven van je kennis kan naast deze online tools ook via tientallen boeken. Let er wel op dat informatie die je verzamelt van vorige versies van Drupal 7 gedeeltelijk of geheel anders kunnen zijn.

Offline leermogelijkheden: Drupal (2016d)

- Programmer's Guide to Drupal, 2nd Edition
- Pro Drupal as an Enterprise Development Platform
- The Definitive Guide to Drupal 7

8.2.2 OctoberCMS

In tegenstelling tot Drupal kan OctoberCMS geen beroep doen op jarenlang opgebouwde documentatie. De bèta versie werd gelanceerd op 15 mei 2014. Een officiële release kwam er iets minder dan een jaar later. Op 1 april 2015 werd OctoberCMS als stabiele versie aangeboden (Bobkov Alexey).

Door het korte bestaan van dit systeem kan het toch rekenen op een uitgebreide documentatie met een mooie onderverdeling per segment. Elk pagina bevat bijgevoegde snippet code die de uitleg duidelijk maakt (Bobkov Alexey, 2016).

The screenshot shows the official OctoberCMS documentation site. On the left, there's a sidebar with icons and links for various sections: SETUP, CMS, THEMES, PLUGINS, BACKEND, DATABASE, SERVICES, CONSOLE, and EXTRAS. The main content area has two main sections:

- # Rendering content blocks**: This section explains how to use the `{% content 'file.htm' %}` tag to render a content block. It includes a code example:

```
url = "/contacts"
==
<div class="contacts">
    {% content 'contacts.htm' %}
</div>
```
- # Passing variables to content blocks**: This section explains how to pass variables to a content block from external code. It states that while content blocks don't support Twig markup, they do support basic syntax for variables. It includes a code example:

```
{% content 'welcome.htm' name='John' %}
```

Below these sections, there's a note about accessing variables inside content blocks using curly braces.

Figuur 8.1: OctoberCMS biedt een duidelijke documentatie aan op hun officiële site.

OctoberCMS beschikt naast documentatie ook over een klein aantal screencasts en tutorials (OctoberCMS, 2016c) die eerder oppervlakkig zijn en dienst doen als inleiding tot het systeem. Naast de officiële documentatie, screencasts en tutorials die te vinden zijn op de website van OctoberCMS, is er momenteel weinig te vinden. Met uitzondering van twee websites. Sitepoint (SitePoint, 2016), een verzamelpunt voor ontwikkelaars waarbij ze over verschillende technieken kunnen bijleren aan de hand van tutorials. Sitepoint bevat een bescheiden aantal tutorials die vooral gericht zijn op het ontwikkelen van plugins. De andere site is een blog die de naam Octo-Help (Pacurar

Filip, 2015) draagt maar niet rechtstreeks verbonden is met OctoberCMS. Als laatste biedt Youtube nog enkele screencasts aan.

8.2.3 Besluit

Het is duidelijk dat Drupal uitblinkt in de hoeveelheid documentatie en de verschillende kanalen waarop deze te vinden is. Toch kan OctoberCMS als relatief nieuw systeem rekenen op goede documentatie. Al blijft het aantal kanalen nog beperkt. Als built-on systeem van Laravel kan OctoberCMS op vele vlakken rekenen op verdere documentatie van Laravel als terugvalbasis.

8.3 Community

Bij het stoten op een probleem kan een goede community de snelle oplossing bieden. Naast goede documentatie is een goede community cruciaal voor een systeem. Bij het ontwikkelen van een website stoot je gegarandeerd ooit op een probleem dat niet terug te vinden is in documentatie. Dit kan je een heleboel kopzorgen bezorgen wanneer je dit helemaal alleen moet uitzoeken. Bij een goede community kan je via verschillende kanalen je probleem verspreiden. Iedereen kan hierop reageren en helpen mee zoeken naar een oplossing. Indien je probleem reeds voorkwam bij andere personen is de kans groter om sneller een oplossing te vinden. Het toegangsniveau van een community kan ook een belangrijk struikelblok vormen. Bij sommige community's kan de vraagstelling en moeilijkheid hoger liggen dan bij andere. Zo wordt je bijvoorbeeld met een eenvoudige vraag als beginner geconfronteerd met boze blikken en reacties. Krijg je woorden naar het hoofd als RTFM (Read The Fucking Manual). Anderen sturen je door naar de exacte pagina met documentatie.

8.3.1 Drupal

De Community rond Drupal is onevenaarbaar. Naast een forum waarbij je met alle vragen terecht kan, bestaan er ook Drupal evenementen, sociale media groepen, een Drupal associatie en nog zoveel meer (Drupal, 2016e). De kans dat je op nieuw probleem stoot binnen de Drupal wereld is eerder klein te noemen. Zowel via het officiële forum als sites zoals Stack Overflow vind je duizenden posten opgedeeld volgens categorie (Drupal, 2016c). Via een goede zoekterm of omschrijving van het probleem, kom je ongetwijfeld terecht op een post met hetzelfde probleem en de daarbij horende oplossing(en). Het toegangsniveau van de community is laag. Zowel gebruikers met weinig technische kennis als ervaren ontwikkelaars kunnen er terecht met hun vragen, bedenkingen, ideeën en oplossingen. De onderwerpen gaan van installatieproblemen tot problemen bij het ontwikkelen van modules.

8.3.2 OctoberCMS

OctoberCMS is niet te vergelijken qua omvang met Drupal. Hoewel dit systeem nog maar een relatief kort bestaan heeft is de community er rond sterk gegroeid de afgelopen maanden. Oorspronkelijk werd dit systeem door twee personen opgebouwd. Nu zijn er maar liefst 141 personen die deelnemen aan het verbeteren van de core van OctoberCMS (OctoberCMS, 2016a) en 38 personen die de documentatie verder aanvullen en verbeteren (OctoberCMS, 2016b) (Geraadpleegd op 14-05-2016).

Het support forum van OctoberCMS bevat verschillende onderverdelingen. Gaande van 'Best Practices', help en ondersteuning, tot nieuws en aankondigingen. De support bevat een kleine 1500 posten die al dan niet een antwoord bieden op de gestelde vraag. Vragen kunnen onbeantwoord blijven omdat er momenteel nog geen eenduidige oplossing voor gevonden is. De support op andere sites zoals Stack Overflow blijven beperkt. De Live Chat biedt een snelle respons voor specifieke vragen.

8.3.3 Besluit

Zoals reeds aangehaald is de Community rond Drupal onevenaarbaar. De oplossing voor vragen en problemen zijn ongetwijfeld te vinden op het internet. Alleen kan het moeilijk worden de juiste oplossing binnen een aanvaardbare tijdspanne te vinden. Het aantal posten omtrent sommige onderwerpen kunnen oplopen in de honderden. Het ingeven van specifieke zoektermen is de boodschap. OctoberCMS daarentegen komt op dit vlak soms te kort. Ik haalde reeds aan dat het aantal posten beperkt blijft tot 1500 stuks., al dan niet beantwoord. Dit is aan de lage kant aangezien dit een verzameling is van allerhande soorten onderwerpen. Dit is te verklaren door het eerder korte bestaan en het beperkt aantal gebruikers van OctoberCMS. Het vinden van een oplossing voor een probleem bij OctoberCMS kan ook buiten de aanvaardbare tijdspanne vallen waardoor dit een gebrek kan zijn voor bijkomende oplossingen voor gelijkaardige problemen.

8.4 Performantie

Performantie is een vanzelfsprekend belangrijk element bij het kiezen van een systeem. Hoe snel wordt een pagina opgehaald en visueel voorgesteld aan de eindgebruiker, wat en hoe verbetert een systeem zijn performantie. Dit zijn factoren die belangrijk zijn voor de ontwikkelaar maar zeker ook voor de eindgebruiker/klant.

Drupal en OctoberCMS werden beide getest op performantie. Deze testen werden lokaal uitgevoerd waarbij beide systemen dezelfde functies (pagina's) hebben. De per-

formantie testen zijn uitgevoerd via de ontwikkel extensie in Google Chrome. Via deze extensie is het onder andere mogelijk HTML en CSS elementen te inspecteren, bronbestanden op te vragen en netwerkgegevens op te halen. Via de tab netwerkgegevens kan je de performantie van elke pagina onder de loep nemen. Nagaan welke bestanden worden ingeladen en hoelang het effectief duurt vooraleer de volledige pagina ingeladen is. Er kan gesimuleerd worden voor verschillende netwerksnelheden en types. De homepage zal onderworpen en geëvalueerd worden aan performantie prestaties. Naast de front-end bekijken we of de laadtijd verschillend is met de back-end. We onderwerpen een content pagina aan de performantie test.

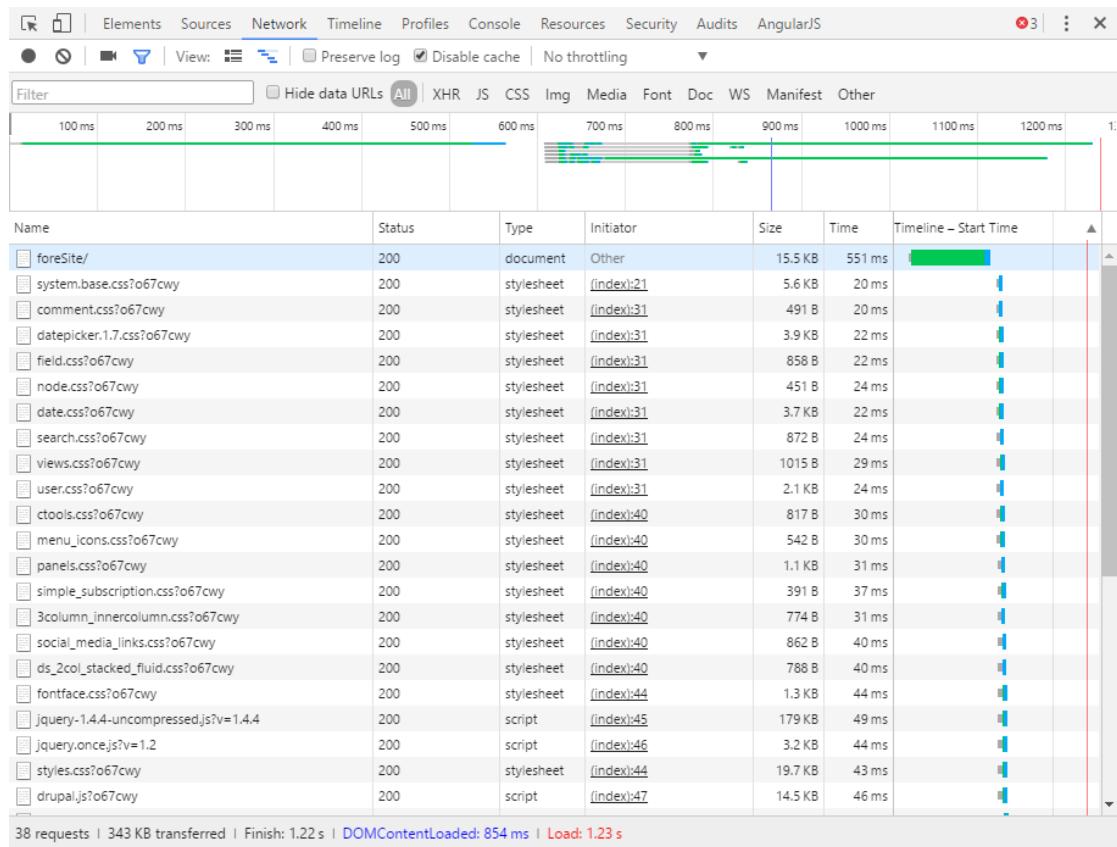
8.4.1 Drupal

Drupal is nooit de beste performer geweest uit het pak. Het willen voldoen aan ieders noden om een zo wijd mogelijk toepassingsgebied te creëren resulteert in een te zware en grote 'codebase' of basiscode. Een project gebruikt nooit de volledige capaciteit van wat het systeem aankan maar bevat wel de overbodige functionaliteiten die niet gebruikt worden. Deze ongebruikte functionaliteiten vertragen derhalve. De architectuur van het systeem om deze ongebruikte functionaliteiten te gaan aanspreken zullen ook een impact geven op de performantie.

De manier waarop Drupal hiermee omgaat om de performantie op te drijven is via caching. Drupal cached overal waar het kan. Gaande van de database query resultaten, de front-end resources, enz. Dit is op zich geen probleem bij sites waar enkel en alleen content getoond moet worden door anonieme gebruikers. Wanneer de site intensief gebruikt wordt door geautoriseerde personen die content inladen en aanpassen, biedt caching niet de ideale oplossing.

We onderwerpen de homepagina aan een performantie test. We zien dat een eerste lading van de pagina, volledig geladen (Load) is op 1.23 seconden. De laadtijd voor de initiële opmaak (DOMContentLoaded) bedraagt slechts 854 milliseconden. De totale grootte die ingeladen moet worden voor de index pagina is 1.5MB. Via het overzicht kunnen alle files gedetailleerd bestudeerd worden.

HOOFDSTUK 8. VERGELIJKENDE TESTEN EN RESULTAAT



Figuur 8.2: Drupal performantie test: overzicht van het resultaat.

Laden pagina	Laadtijd opmaak	Laadtijd volledige pagina
1ste	0.854s	1.23s
2de	0.905s	1,44s
3de	0.844s	1.23s
4de	0.842s	1.36s
5de	0.819s	1.19s
6de	0.842s	1.22s
Gemiddelde	0.851s	1.28s

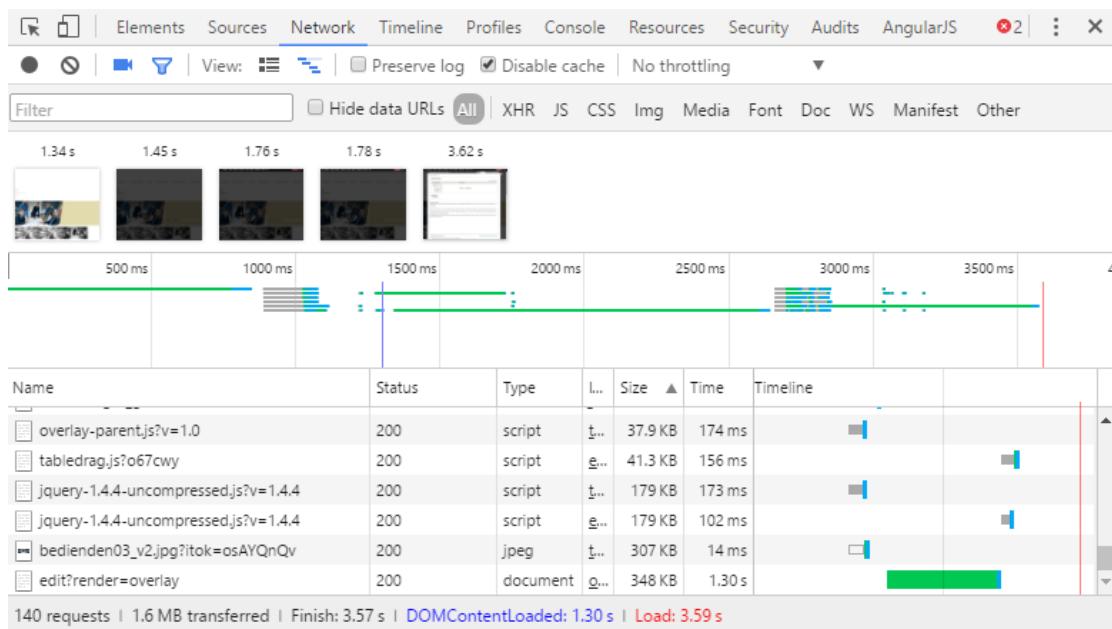
Tabel 8.1: Een overzicht van de laadtijden van de index pagina.

De index pagina wordt geladen met een gemiddelde snelheid van 1.28 seconden. Dit is op zich behoorlijk veel als we dit vergelijken met sommige andere frameworks die een laadtijd hebben van 20 tot 100ms. We bekijken of de front-end laadtijd verschillend is met de back-end laadtijd.

Laden pagina	Laadtijd opmaak	Laadtijd volledige pagina
1ste	0.897	3.00s
2de	1.67s	3.82s
3de	1.10s	2.93s
4de	0.891s	2.95s
5de	1.43s	3.11s
6de	1.05s	4.35s
Gemiddelde	1.188s	3.36s

Tabel 8.2: Een overzicht van de laadtijden van de back-end pagina.

Het wordt duidelijk dat de laadtijden van de front-end een stuk lager liggen dan deze aan de back-end zijde. Het verschil tussen de laadtijd van de initiële opmaak en de volledige laadtijd is immens. De back-end van Drupal wordt bovenop de front-end afgebeeld als een soort van laag. Dit verklaart het verschil tussen de laadtijd van de initiële opmaak en de volledige pagina. De laadtijd van de initiële opmaak is de tijd die nodig is om de front-end af te beelden. De overige tijd is nodig om de back-end in de laden bovenop de front-end pagina.



Figuur 8.3: Drupal performantie test: overzicht van het resultaat van de back-end pagina.

8.4.2 OctoberCMS

OctoberCMS is zeker geen uitschieter als het over performance gaat. Nochtans is de structuur en files volledig beheersbaar. Aangezien OctoberCMS bovenop Laravel draait en dit systeem een grote codebase heeft, vertraagt dit het volledige systeem. Standaard wordt slechts weinig in de cache opgeslagen, dit kan manueel beheerd worden waardoor performance prestaties kunnen verbeterd worden.

Wanneer we de homepagina van OctoberCMS onderwerpen aan performantie prestaties kunnen we zien dat de laadtijd van de volledige pagina (Load) 1.88 seconden bedraagt. Het verschil tussen laadtijd van de volledige pagina en de initiële opmaak (DOMContentLoaded) is te verwaarlozen. De homepagina heeft een grootte van 576KB die verdeeld wordt over 15 bestanden. Via het overzicht kan voor elk bestand de grootte, laadtijd en volgorde van uitvoering, uitvoerig bestudeerd worden.

Laden pagina	Laadtijd opmaak	Laadtijd volledige pagina
1ste	1.88s	1.90s
2de	2.02s	2.02s
3de	2.08s	2.11s
4de	1.94s	1.96s
5de	1.88s	1.89s
6de	2.02s	2.05s
Gemiddelde	1.97s	1.98s

Tabel 8.3: Een overzicht van de laadtijden van de index pagina.

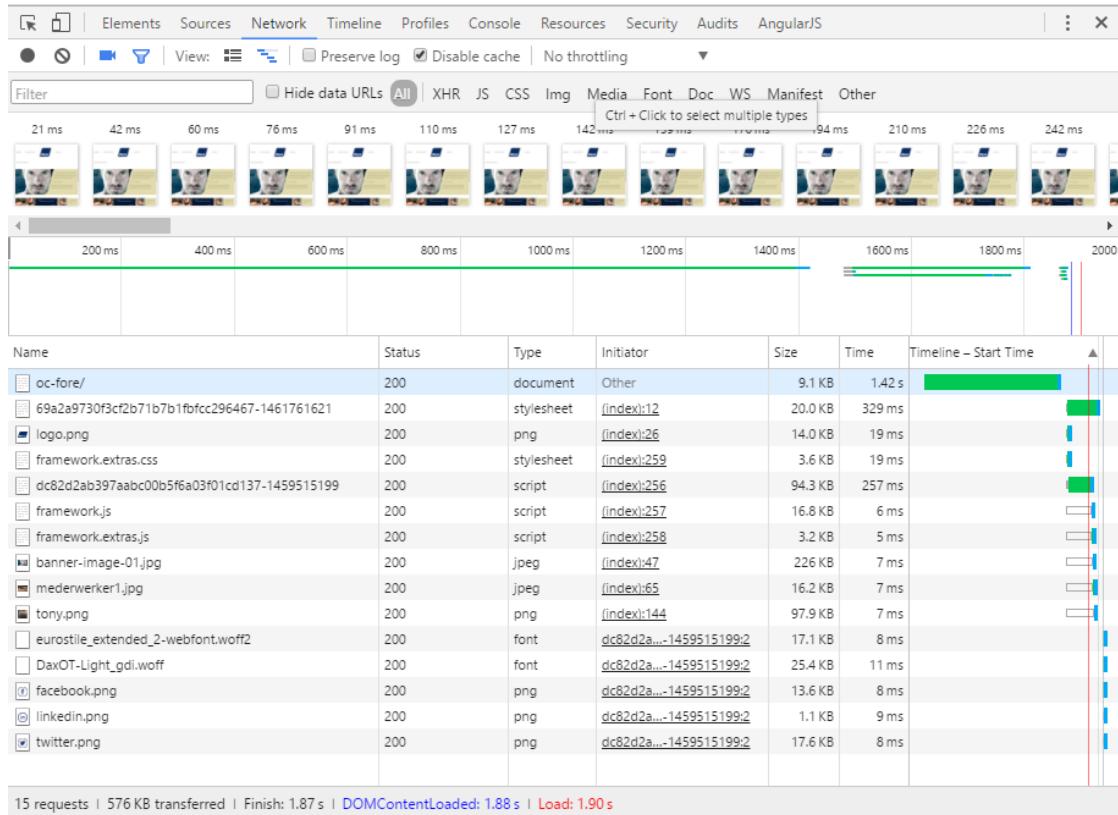
De index pagina wordt ingeladen met een gemiddelde snelheid van 1.98 seconden. Dit is eerder aan de trage kant. We vergelijken de laadtijden van de front-end nu met deze van de back-end.

Laden pagina	Laadtijd opmaak	Laadtijd volledige pagina
1ste	2.31s	2.39s
2de	2.96s	3.03s
3de	2.39s	2.43s
4de	2.27s	2.30s
5de	2.36s	2.40s
6de	2.37s	2.42s
Gemiddelde	2.44s	2.49s

Tabel 8.4: Een overzicht van de laadtijden van de back-end pagina's.

De gemiddelde laadtijden voor de back-end pagina's liggen ongeveer een halve second

hoger. Hoewel dit op het eerste zicht weinig lijkt, kan dit beduidend trager aanvoelen voor de gebruiker.



Figuur 8.4: OctoberCMS performantie test: overzicht van het resultaat.

8.4.3 Besluit

We kunnen vaststellen dat beide systemen niet de beste laadtijden neerzetten. De verschillen tussen het laden van de front-end en de back-end pagina's zijn bij Drupal groot. Bij OctoberCMS is er ook een duidelijk verschil maar dat is niets in vergelijking met Drupal.

Mits een goede caching kan de laadtijd van OctoberCMS een stuk verminderd worden. Maar een laadtijd neerzetten onder de 100 milliseconden, zoals heel snelle systemen, zit er niet in.

8.5 Beheersbaarheid

De beheersbaarheid van een systeem is de mate waarin een systeem 'Developer Oriented' is. In welke mate is het systeem ontworpen voor ontwikkelaars. Veelal worden CMS systemen met oog op de eindgebruiker gemaakt. Dit zijn mensen zonder enige of geringe technische kennis omtrent programmeren. Deze systemen zijn ontworpen om de mogelijkheid te scheppen voor deze mensen om een eigen basis site op te stellen. Ontwikkelaars kunnen hiermee ook aan de slag om verder te gaan dan een basis site. Beheersbaarheid staat gelijk aan de controle die er al dan niet is in verband met coding, theming, foutberichten, enz.

8.5.1 Drupal

Drupal is ontworpen met de intentie om minder technische mensen ook de kans te geven een eenvoudige website op te bouwen via hun systeem. Ontwikkelen via Drupal is niet altijd een plezier. Ontwikkelen van een basissite in Drupal is niet moeilijk te noemen qua programering, maar het kan veel moeite kosten het concept en de flow ervan onder de knie te krijgen. Het is ook niet altijd duidelijk welke aanpassingen welke gevolg hebben.

Belangrijk bij het starten van een project met Drupal is vooraf het project 100 procent te definiëren, zodat de structuur kan vastgelegd worden. Veranderlijke projecten waarbij elementen doorheen de implementatie veranderen zijn geen opdracht voor Drupal.

Coding

Coding, of de manier waarop code geschreven en beheerd wordt in een systeem. Bij Drupal 7 is het mogelijk een volledige site op te stellen zonder ook maar één regel code te schrijven. Alles verloopt hier via de back-end van de browser. Zoals eerder aangehaald kan de structuur van verschillende pagina's en nodes worden aangepast door template bestanden te overschrijven van de core bestanden. Qua beheersbaarheid is dit niet ideaal. Elke module kan één of meerdere template bestanden bevatten die elk apart overschreven moeten worden. Natuurlijk is het niet verplicht alle template bestanden te gaan overschrijven. Toch is het aan te raden zeker de page.tpl.php te overschrijven om de structuur van je 'body' aan te passen.

Een andere reden om de template bestanden te overschrijven is om propere code te verkrijgen. Drupal 7 levert default niet bepaald propere code af. Een teveel aan div-tags die diep genest zitten, meestal zonder enige meerwaarde aan je code. Daar-

naast voegt Drupal 7 een heel arsenal klassen toe aan deze div-tags. Klassen die veelal weinig concrete info verstrekken over de inhoud van de sectie. Het stijlen van deze klassen en daarbij het overzicht bewaren, is een hele opdracht. In de meeste gevallen is het wel mogelijk om manueel een klassennaam toe te voegen aan het gewenste element. Er is een mogelijkheid om aan ieder veld van een content type een unieke klasse toe te voegen. Dit verandert uiteraard niets aan het uitzicht van de code, maar geeft wel een mogelijkheid om dit item uniek te gaan stijlen.

```

▼<div id="headerContainer"> == $0
  ▼<div class="group-left">
    ▼<div class="field field-name-field-image field-type-image field-label-hidden">
      ▼<div class="field-items">
        ▼<div class="field-item even">
          
        </div>
      </div>
    </div>
    ▼<div class="group-right">
      ▼<div class="field field-name-field-tekst-boven-titel field-type-text-with-summary
        field-label-hidden">
        ▼<div class="field-items">
          ▼<div class="field-item even">
            "Moloremque dolum nem earuntis aut qui in commis volores equiste odit,
            cuptatiam, accum faccus sedit vollate ctibus quo optae ea qui beatibeatim
            repudis quuntinimpos dicabo. Elis aut ut doluptius.

            It la esectibus, eum vel int essin resecte cestios atempore ped que quam endio.
            Eribus dolorum quiatam, eiuntus consecatem fugit de qui omnism venis
            doloratum vendem dem nus."
          </div>
        </div>
      </div>
      ▼<div class="field field-name-title field-type-ds field-label-hidden">
        ▼<div class="field-items">
          ▼<div class="field-item even" property="dc:title">
            <h2>Vakmanschap since 1904</h2>
          </div>
        </div>
      </div>
      ▶<div class="field field-name-field-tekst-onder-titel field-type-text-with-summary
        field-label-hidden">...</div>
    </div>
  </div>

```

Figuur 8.5: Drupal's HTML code om de Header te tonen is allesbehalve proper en overzichtelijk te noemen.

Via het overschrijven van template files kunnen een teveel aan div-tags verwijderd worden om zo de code minimaal op te poetsen. De overbodige en weinig omschrijvende klassen kunnen vervangen of verwijderd worden door concrete, eigen gekozen namen.

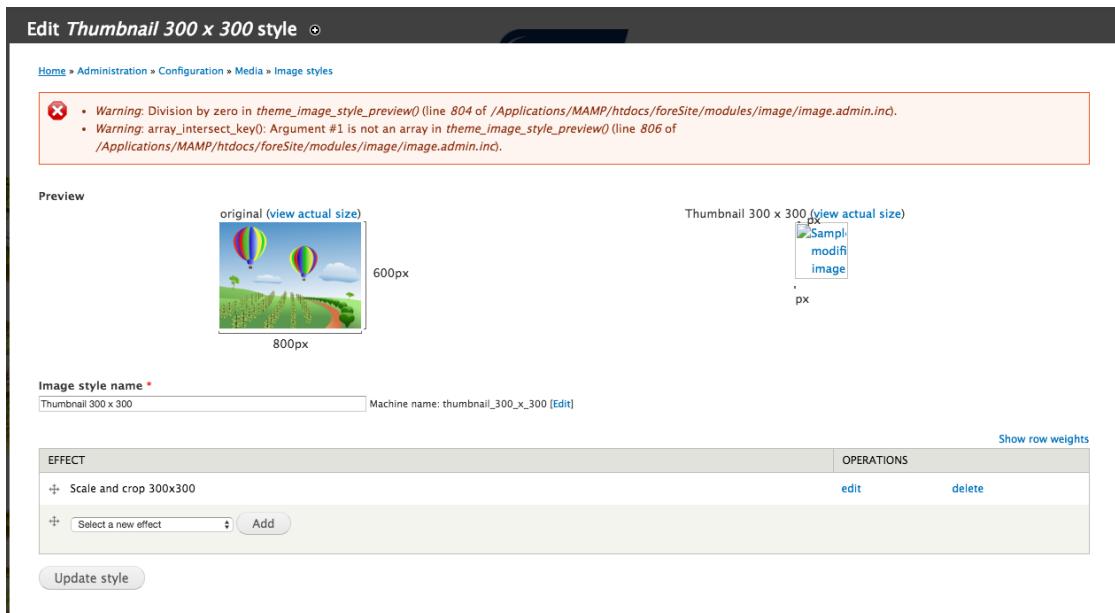
Inhoud kan enkel worden aangemaakt en gecontroleerd via de back-end van de browser. Dit kan beperkingen opleveren. Er is te weinig controle over de structuur die de inhoud aanneemt. Deze controle kan gedeeltelijk opgevangen worden door modules die inhoud kunnen groeperen, filteren en herschrijven. De module display suite geeft de mogelijkheid velden te groeperen en ordenen. De views module zorgt ervoor dat er gefilterd kan worden en dat velden herschreven kunnen worden. Hoewel dit in de meeste gevallen een oplossing kan bieden, blijft dit een omslachtige manier van werken en is alles behalve gunstig als ontwikkelaar. In het andere geval is het zoeken naar een module die hopelijk een oplossing kan bieden.

Een ander nadeel van het database georiënteerde systeem is het gebrek aan duplicatie van code. Aangezien bijna alles via de database verloopt is het moeilijk om een bepaalde actie of configuratie op net dezelfde manier uit te voeren. Een configuratie die in de ene site naar behoren werd afgeleverd kan niet eenvoudigweg gekopieerd worden naar een ander project die een gelijkaardige configuratie vraagt.

Foutberichten

Foutberichten tijdens het programmeren kunnen een cruciale bron van informatie zijn. Zonder concrete aanwijzing van het probleem kan dit uitdijken in een zoektocht. Drupal beschikt over drie opties als het op foutberichten komt. Wanneer je site in productiemodus draait kies je best om geen foutberichten weer te geven. Dit komt allesbehalve goed over naar de eindgebruiker. In ontwikkelmodus kan je kiezen om alle berichten weer te geven of enkel fouten en gevaren.

Over het algemeen wordt je amper slimmer bij het bekijken van foutberichten in Drupal. Veelal zullen deze database gericht zijn, aangezien zowat alle acties die uitgevoerd worden connectie met de database vragen. Foutberichten worden op het eerste zicht gedetailleerd getoond met een beschrijving, lijnnummer en bestandsreferentie. Als we kijken waar deze file zich bevindt, merken we op dat dit uit een module komt. Een geïnstalleerde module die simpelweg geïnstalleerd is en waar verder geen aanpassingen aan gedaan zijn. Even de cache legen kan hopelijk het probleem oplossen. Anders kan je met deze info hooguit aan de slag door deze foutbericht in te geven als zoekterm in Google.



Figuur 8.6: Een foutbericht in Drupal die weinig beschrijvende info bevat.

Een ander frustrerend fenomeen binnen Drupal is het WSOD (white screen of death). Het kan gebeuren dat je als gebruiker staat op een volledig witte pagina. Deze pagina bevat geen inhoud, foutberichten of waarschuwingen. De oorzaak van dit probleem kan vele redenen hebben. Indien je site in productie staat kan het zijn dat fouten uitgeschakeld zijn waardoor de foutbericht niet zichtbaar is. Wanneer dit niet het geval is en er nog steeds niets verschijnt, kan de foutbericht(en) geforceerd getoond worden door het index.php bestand aan te passen. Voeg net na de opening van de PHP tag volgende code toe.

```
error_reporting(E_ALL);
ini_set('display_errors', TRUE);
ini_set('display_startup_errors', TRUE);
```

Dit zal alle mogelijke foutberichten op het scherm tonen. Enkel geheugenfouten zullen niet zichtbaar zijn.

Versiebeheersysteem

In de meeste gevallen worden IT-projecten zelden ontwikkeld door slechts één persoon. Zo zullen bij het ontwikkelen van websites de taken meestal verdeeld worden tussen front-end en back-end: twee of meerdere personen die een stuk van de opbouw van een website voor hun rekening nemen. De beste manier om samen te werken in groep aan één of meerdere projecten is door gebruik te maken van een versiebeheersysteem.

Code wordt extern op een server bewaard waar desbetreffende personen toegang tot hebben. Het meest bekende versiebeheersysteem is zonder twijfel Git.

Versiebeheersystemen kunnen verschillende doeleinden hebben. Gaande van kleine statische of dynamische websites tot grotere applicaties.

Een versiebeheersysteem maakt gebruik van een bestandsgebaseerde opslagmethode. Drupal slaat de meest belangrijke data op in een relationele database, waardoor versiebeheer een stuk moeilijker wordt. Alle inhoud en configuratie van deze inhoud (content types), views, menu's en nog zoveel andere informatie, worden via de browser ingegeven en opgeslagen in de database. Bij een versiebeheer met een bestandsgebaseerde opslagmethode kan er bij een fout eenvoudig worden teruggegrepen naar vorige code. Aanpassingen doorvoerd in Drupal zijn permanent doorgegeven aan de database.

Het is uiteraard mogelijk om te werken met een versiebeheersysteem, alleen wordt de kans op conflicten in de database een stuk groter en is de werking behoorlijk omslachtig. Exporteer de database en sla deze op in de map projectnaam > sites > default. Nu kan deze worden gepusht naar de externe server. Importeren werkt op dezelfde manier. Deze manier van werken kan problemen geven wanneer twee of meer personen op dezelfde moment aanpassingen doen en op deze manier de database gaan beïnvloeden. Aangezien bijna alle taken via de database verlopen, is de kans groot dat dit zal voorkomen. Deze methode kan werken wanneer de ene persoon zich enkel bezighoudt met het stijlen van de website via code in CSS, en de andere, andere structurele taken uitvoert die de database beïnvloeden.

8.5.2 OctoberCMS

OctoberCMS werd volledig met het oog op ontwikkelaars gemaakt. Het systeem is volledig bestandsgebaseerd en biedt daardoor een volledige controle aan de ontwikkelaar. De back-end in de browser wordt getoond in een mooi en duidelijk design. Het gebruik ervan is intuitief en bevat geen overdrive aan elementen.

Coding

Alle code die geschreven is, is aanpasbaar naar eigen wens. De controle over HTML, CSS, Javascript ligt volledig in de handen van de ontwikkelaar. Het gebruik van partials en layout pagina's bevordert de herbruikbaarheid van code en behoudt het overzicht bij grote projecten. Content blocks kunnen beheerd worden door de eindgebruiker alsook de ontwikkelaar. Aanpassingen kunnen in code of in de back-end van de browser via een WYSIWYG. Dit is een teksteditor waarbij de eindgebruiker de opgemaakte tekst direct ziet en de kans heeft deze aan te passen.

Toevoegen van bestaande plugins kan simpelweg via de back-end in de browser. Eenmaal geïnstalleerd kunnen componenten gesleept worden in de editor en wordt er automatisch snippet code aangemaakt. Dit is de meest eenvoudige manier van werken. Voor geavanceerd gebruik kan de code ook worden getoond en aangepast waar nodig. Het aanmaken van nieuwe plugins kan op verschillende manieren (eerder besproken) en hoeft niet moeilijk te zijn. Een plugin kan uiterst geavanceerd worden afhankelijk van de functies waaraan hij moet voldoen.

Het aanmaken van een nieuw thema is centraal beheerd. Alle mappen en bestanden die behoren tot het thema, bevinden zich in dezelfde map. Geen gedoe met uitzoeken waar welke stukken code worden aangemaakt en beheerd. Soms kan het eenvoudig zijn eerst de site statisch te maken. Achteraf wordt de site dynamisch gemaakt door plugins toe te voegen. Voor OctoberCMS is dit geen probleem. Een ontwerp kan met volledige controle overgezet worden in HTML en CSS. Ook het gebruik van CSS frameworks vormen geen probleem.

Foutberichten

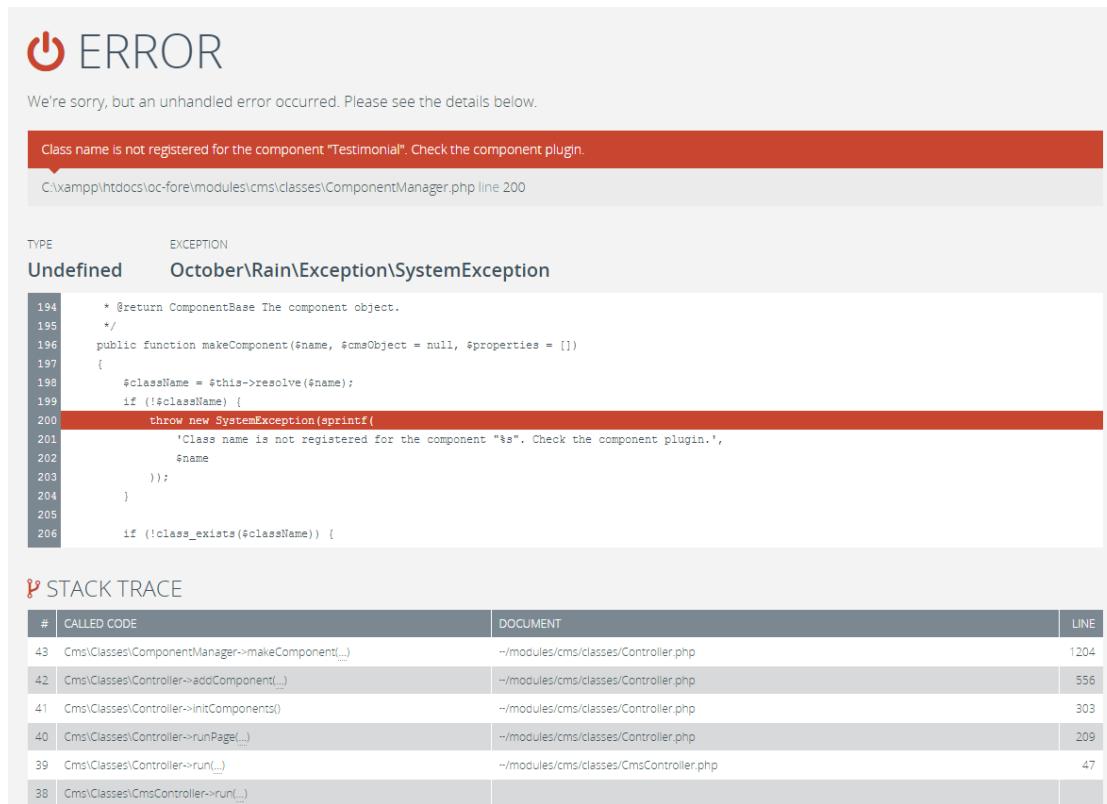
Met oog op ontwikkeling zijn foutberichten een bron van informatie. We verwachten van een systeem als OctoberCMS die zich definieert als Developer Oriented systeem, duidelijke, gestructureerde foutberichten toont.

Indien een fout optreedt in code wordt men doorverwezen naar een aparte foutpagina. Deze geeft een korte boodschap weer met bijhorende file waarvan de fout werd gegooid. Een tweede blok geeft het soort fout terug met bijhorende exception. Een visuele representatie van het blok code die de fout gooit, wordt getoond om de fout te duiden. Als laatste wordt de stack trace getoond. Een overzicht van alle stappen uitgevoerd door het systeem.

Via dit overzicht moet het duidelijk worden voor de ontwikkelaar waar en waarom deze fout werd gegooid. Verdere stappen kunnen nu ondernomen worden om de fout op te lossen of de code te debuggen.

Versiebeheersysteem

OctoberCMS laat zich heel eenvoudig beheren door een versiebeheersysteem. Doordat het systeem bestandsgebaseerd is geeft dit geen conflicten met betrekking tot de database. Front-end en back-end ontwikkelaars kunnen tegelijkertijd werken aan éénzelfde project zonder probleem. Eventuele conflicten waarbij elkaars code overschreven werd kan via een goede IDE (Integrated Development Environment) of editor worden samengevoegd en aangepast.



Figuur 8.7: De foutbericht van OctoberCMS bevat voldoende informatie om het probleem te lokaliseren en verdere stappen te ondernemen.

8.5.3 Besluit

Doordat bij Drupal alle acties in de back-end van de browser gebeuren, wordt dit ervaren als overrompelend. Een teveel aan verschillende opties en items maakt het menu overdreven groot en vergt ervaring alle knoppen en opties te leren kennen. De toevoeging van modules zorgt voor een uitbreiding van het menu, als ook opties binnen andere menu-items. OctoberCMS houdt de back-end in de browser overzichtelijk en implementeert enkel de nodige functies. Alle andere functionaliteiten worden in code aangepast.

OctoberCMS is een flexibel systeem die zowel grote als kleinere projecten voor zijn rekening kan nemen. Bij een installatie 'from Scratch' is enkel de minimum functionaliteit en code aanwezig om het systeem te laten draaien. Geen gedoe met overschrijven van huidige code zoals het geval bij Drupal.

Foutberichten worden in OctoberCMS gedetailleerd en gestructureerd opgebouwd

als aparte pagina, waarmee je verder aan de slag kan. Drupal toont enkel een korte foutbericht bovenaan de pagina die weinigzeggend is.

Er kan besloten worden dat OctoberCMS op gebied van beheersbaarheid een stuk beter scoort dan Drupal 7.

8.6 Database

Bijna iedere applicatie komt in contact met een database, al is het nu om gewoon data op te halen of weg te schrijven. Een CMS maakt sowieso gebruik van een database. Verschillende elementen moeten in rekening gebracht worden bij het kiezen van een systeem en eventuele selectie van het soort database. Efficiëntie van een database kan een heikel punt vormen. De hoeveelheid query's die verstuurd worden en hoe lang deze query's er over doen. Of er al dan niet gebruik wordt gemaakt van caching, enz. Allemaal items om rekening mee te houden bij het evalueren van een systeem.

8.6.1 Drupal

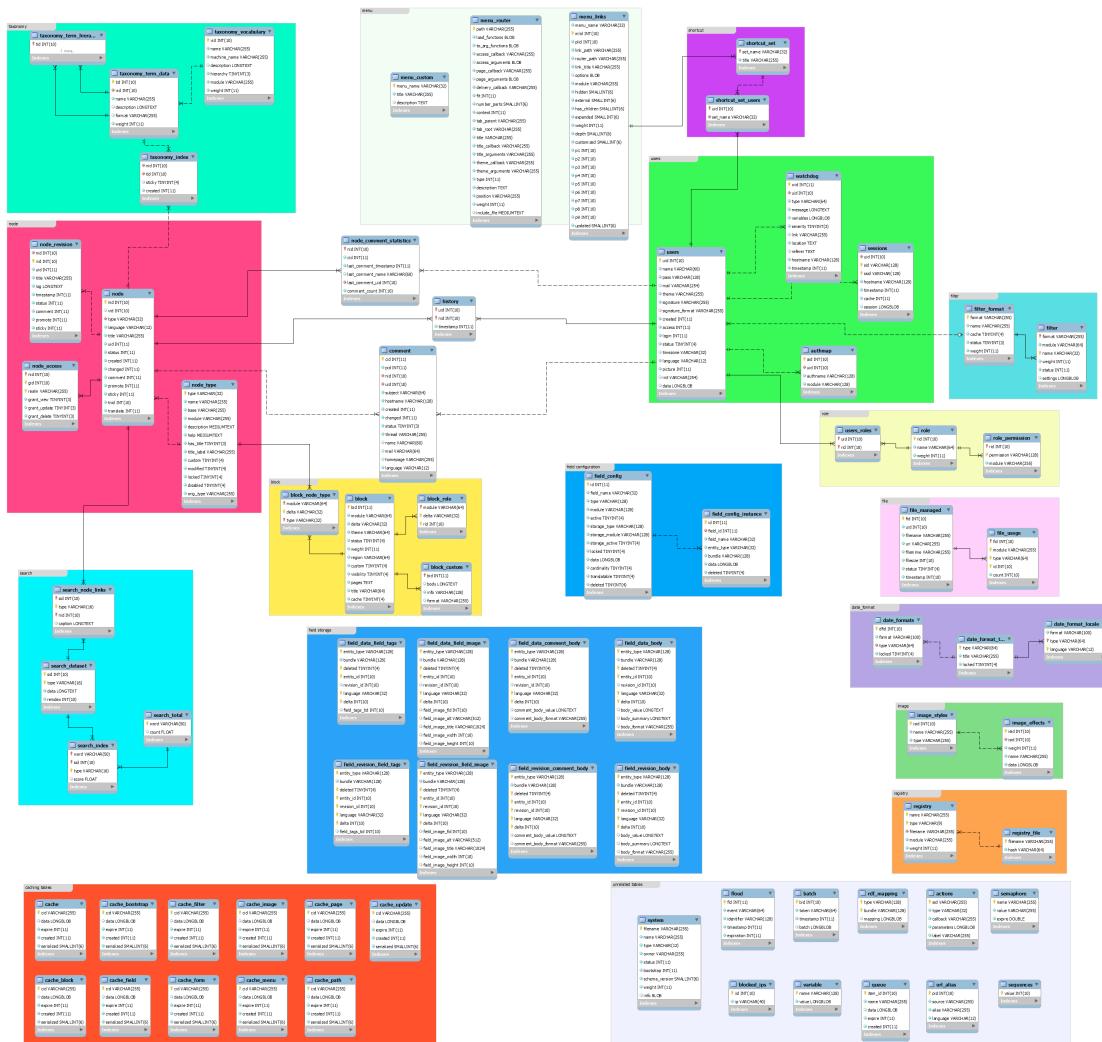
Een installatie van Drupal 7 wordt meegeleverd met een verzameling van maar liefst 73 tabellen die een bestandsgrondte van 4.5MB innemen, waarvan 10 tabellen zullen dienst doen voor caching. Deze tabellen slaan alle wederkerende informatie op, zodat niet alle data telkens opnieuw volledig opgehaald moet worden. Op deze manier behaalt Drupal zijn gewenste performantie. 11 van de 73 tabellen hebben geen relatie met andere tabellen en bestaan dus op zichzelf. Overige tabellen kunnen opgedeeld worden in verschillende categorieën.

- Field gerelateerde tabellen
- User gerelateerde tabellen
- Node gerelateerde tabellen
- Tussentabellen die User en Node tabellen linken

Deze tabellen zijn gekoppeld met elkaar via vreemde sleutel relaties.

De hoeveelheid tabellen die enkel en alleen dienen om het systeem op te starten en te laten werken, is eerder aan de hoge kant. Drupal is modulair en zal voor het opbouwen van een site tientallen modules toevoegen. Aangezien deze modules communiceren met de database zal voor iedere module minimum één tabel worden aangemaakt. Deze nieuwe tabellen kunnen op hun beurt relaties bevatten met andere tabellen. Het gevolg van deze acties leidt tot een groter, ingewikkelder ERD (Entity-Relationship

HOOFDSTUK 8. VERGELIJKENDE TESTEN EN RESULTAAT



Figuur 8.8: Bij de installatie van Drupal worden 73 tabellen aangemaakt.

diagram). Een schema met alle entiteiten en relaties tussen deze entiteiten. Probeer daarom aandacht te vestigen op de hoeveelheid modules en de combinatie ervan die je installeert. Zet alle onnodige modules uit om de belasting op de database zo laag mogelijk te houden.

Qua flexibiliteit en hergebruik van data is Drupal verre van ideaal. De database-structuur van Drupal is te ver genormaliseerd en bevat onnodige tabellen waardoor de volledige database niet de gewenste flexibiliteit bevat. Wanneer we de database van Drupal op een andere manier willen hergebruiken zien we dat dit zo goed als onmogelijk blijkt te zijn.

8.6.2 OctoberCMS

Een nieuwe installatie van OctoberCMS maakt 23 tabellen aan die een bestandsgrootte innemen van 495KB. Bij het creëren van nieuwe plugins kunnen extra tabellen worden toegevoegd. Het aantal toegevoegde tabellen is afhankelijk van de complexiteit van de plugin. De toename van tabellen blijft beperkt net zoals de relaties tussen tabellen. Deze tabellen en hun onderlinge relaties worden in code aangemaakt. Tabellen die aangemaakt worden bij installatie kunnen grotendeels opgesplitst worden in categorieën.

- Systeem gerelateerde tabellen
- User gerelateerde tabellen

De overige tabellen zijn op zichzelf staande tabellen en kunnen niet gecategoriseerd worden. Over het algemeen zien we dat de structuur van het ERD eenvoudig is opgebouwd. Er is slechts één tussentabel waarbij een relatie loopt. Dit is de tabel 'backend_users_groups'. Er is geen sprake van ingewikkelde relaties met vreemde sleutels.

Zoals reeds aangehaald kunnen er bij het creëren van plugins één of meerdere tabellen worden aangemaakt. Deze tabellen worden allemaal voorafgegaan door de unieke naam van de ontwikkelaar gevolgd door de naam van de plugin en de tabelnaam.

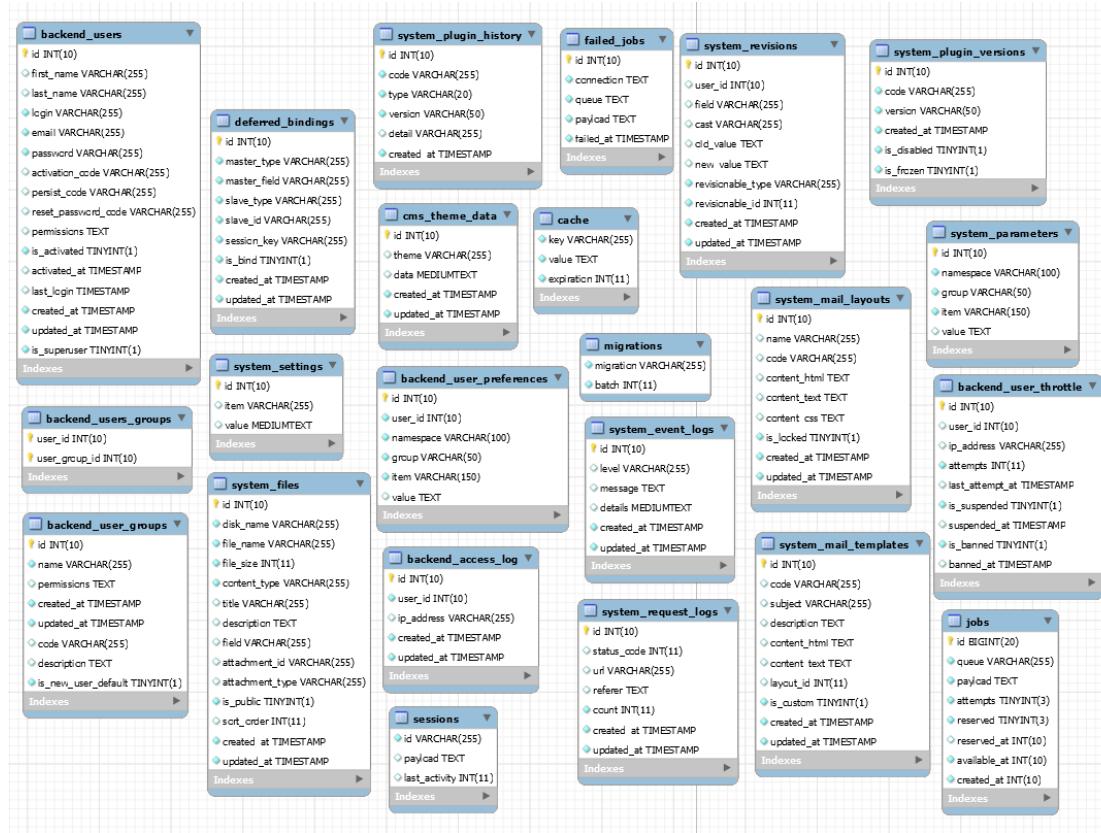
OctoberCMS voorziet één tabel voor caching. Hoewel de meeste cache gerelateerde acties worden voorzien door OctoberCMS zelf, is het mogelijk ook manueel functies aan te maken om eigen data te cachen.

```
Cache::put('key', 'value', $minutes);
$value = Cache::get('key');
```

Via deze eenvoudige methoden kan je via de 'put' functie elementen voor een bepaalde tijd (\$minutes) opslaan in de cache en vervolgens terug ophalen via de 'get' methode.

De databasesstructuur van OctoberCMS is flexibel. De database kan op eenvoudige wijze gebruikt worden door andere applicaties. Het doorzoeken van tabellen levert een snel, duidelijk en volledig resultaat op.

HOOFDSTUK 8. VERGELIJKENDE TESTEN EN RESULTAAT



Figuur 8.9: Bij de installatie van OctoberCMS worden 23 tabellen aangemaakt.

8.6.3 Besluit

De bestandsgrootte en het aantal tabellen bij installatie van de database van Drupal is groot. Met een enkel toenemende database door modules, is het belangrijk de controle van modules in het oog te houden. OctoberCMS start met een select aantal tabellen om het systeem te laten werken. Geen overbodige tabellen waarvan het systeem misschien nooit gebruik maakt.

De database van Drupal is te ver genormaliseerd waardoor de duidelijke structuur en flexibiliteit niet ideaal is. Het hergebruiken of extraheren van gegevens uit de database is onbegonnen werk. Dit in tegenstelling tot OctoberCMS. Als ontwikkelaar is het interessanter werken met een database zoals deze aangeleverd wordt bij OctoberCMS.

Hoofdstuk 9

Conclusie

Het doel van dit stuk was te onderzoeken of er al dan niet een beter alternatief systeem bestaat voor Drupal CMS, dat zich focust op ontwikkelaars. Er werd een CMS gezocht dat bovenop het PHP framework Laravel gebouwd is.

Via literatuurstudie kwam OctoberCMS boven water als CMS gebouwd op Laravel. Dit CMS zou idealiter positief moeten beantwoorden aan een systeem gefocust op ontwikkelaars. Beide CMS'en werden onderworpen aan een casus met vooropgestelde template. De werking en bevindingen werden neergeschreven. Hierna werden testen en bijkomende resultaten neergeschreven voor punten die belangrijk zijn voor ontwikkelaars bij het kiezen van een nieuw framework of systeem.

Items die aan bod kwamen: leercurve, documentatie, community, performantie, beheersbaarheid en database structuur. Met het oog op ontwikkelaars biedt OctoberCMS een snelle start met relatief kleine leercurve. Een bepaalde kennis over het framework moet eerst opgedaan worden. Maar de werking van het systeem zit logisch in elkaar waardoor de ontwikkelaar kan verder bouwen op zijn eerder opgebouwde kennis. OctoberCMS biedt uitgebreide informatie op hun officiële site aan met voldoende code voorbeelden. Externe informatie bronnen met bijvoorbeeld tutorials en screencast zijn er nauwelijks te vinden. Een toevoeging van tutorials en/of screencasts zou de leercurve nog minder steil maken door deze praktische voorbeelden. De community van OctoberCMS is in twee jaar tijd uitgegroeid van de twee makers tot een groep van 141 personen die het systeem helpen verbeteren. Een actieve community waaraan dagelijks verbeteringen worden aangebracht en openstaande fouten worden hersteld. Er bestaat een forum waarop fouten en problemen op gepost kunnen worden. De hoeveelheid posten met bijhorende oplossingen blijft voorlopig beperkt. Dit zal groeien naarmate het gebruik van OctoberCMS toeneemt. Op gebied van performantie zal OctoberCMS nooit een uitschieter worden. Mits een goede caching kan performantie wel worden opgedreven. Voor beheersbaarheid of controle met betrekking tot coding,

HOOFDSTUK 9. CONCLUSIE

theming en foutberichten, biedt OctoberCMS een zeer gunstige uitkomst. Het systeem is ontwikkeld met de focus op ontwikkelaars en dat komt duidelijk naar voor. Er is volledige controle over het systeem en bijhorende code. De databasesstructuur is compact en bevat geen overbodige relaties. Het hergebruiken van de database voor andere doeleinden is geen enkel probleem.

OctoberCMS kan een goed alternatief CMS zijn voor ontwikkelaars die op zoek zijn naar meer controle en logische werking van een systeem. OctoberCMS is een CMS ontwikkeld voor ontwikkelaars en niet onbelangrijk, bijzonder eenvoudig te beheren door de niet technische eindgebruiker/klant. De community rondom OctoberCMS is zeer actief maar biedt voorlopig te weinig antwoord op gestelde vragen en problemen van gebruikers. Het uitzoeken van een probleem kan hierdoor zeer veel tijd in beslag nemen.

Dit stuk behandelt slechts één van de meerdere bestaande CMS'en gebouwd bovenop Laravel. Het onderzoeken van enkele andere systemen kan een volgende stap zijn voor dit onderzoek.

Bijlage A

Casus code - Github

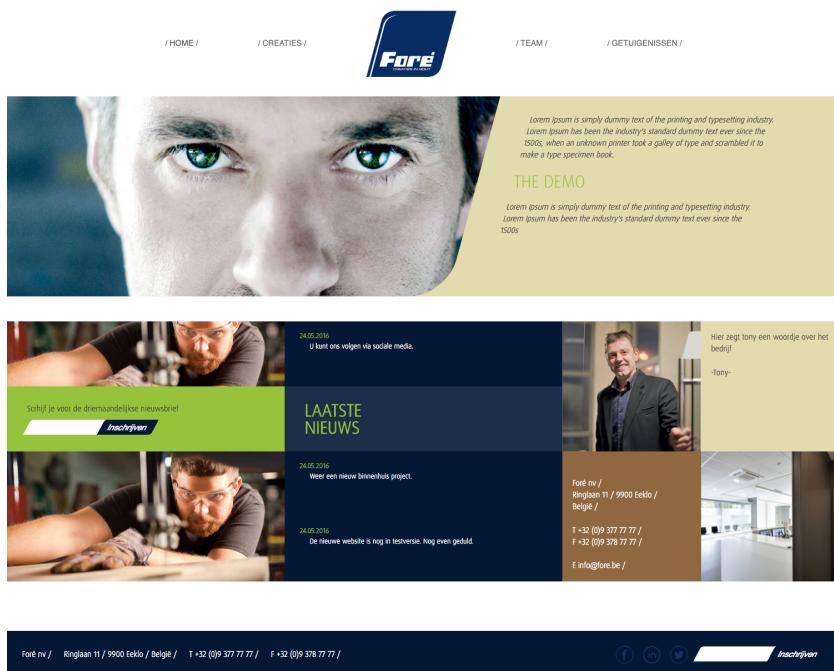
Aan de hand van de vooropgestelde casus werd een site opgesteld voor beide systemen (Drupal en OctoberCMS).

De code kan terug gevonden worden op GitHub,
via deze link: <https://github.com/DieterVyncke/BP>

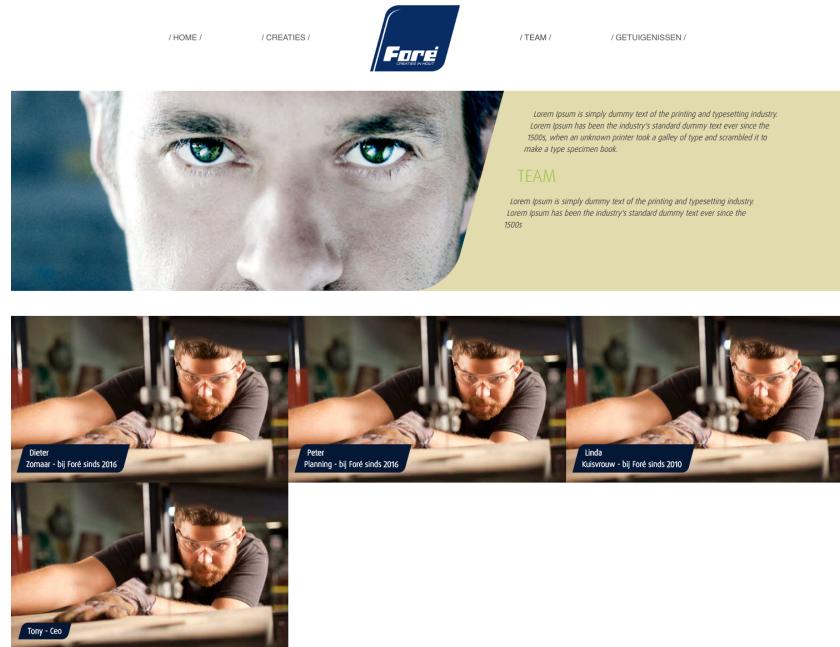
Bijlage B

Casus - Designs

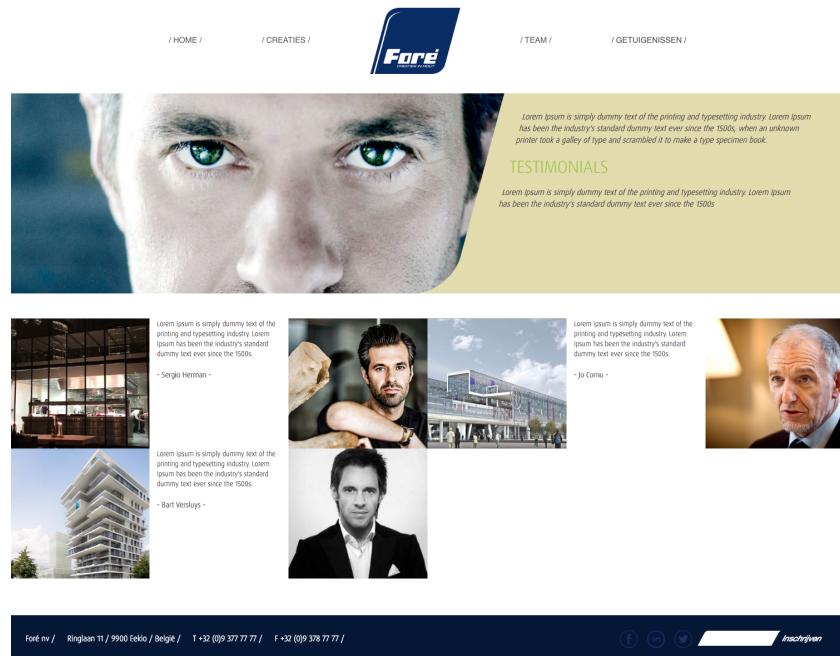
De casus bevat een voorgedefinieerde template die exact zal uitgewerkt worden in twee systemen. Beide systemen worden uitgetest op een bestaand design voor een bedrijfswebsite. Deze website bevat de meest typerende en meest belangrijke elementen als voorbeeld voor een CMS.



Figuur B.1: Casus design homepagina



Figuur B.2: Casus design team



Figuur B.3: Casus design testimonials

BIJLAGE B. CASUS - DESIGNS

[/ HOME /](#)
 [/ CREATIES /](#)
 
 [/ TEAM /](#)
 [/ GETUIGENISSEN /](#)

Sergio Herman

Lorum ipsum is simply dummy text of the printing and typesetting industry. Lorum ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, Lorum ipsum is simply dummy text of the printing and typesetting industry. Lorum ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries,




[Foré nv / Ringlaan 11 / 9900 Eeklo / België / T +32 \(0\)9 377 77 77 / F +32 \(0\)9 378 77 77 /](#)

[Inscríbete](#)

Figuur B.4: Casus design testimonials detail

[/ HOME /](#)
 [/ CREATIES /](#)
 
 [/ TEAM /](#)
 [/ GETUIGENISSEN /](#)

Tuinhort & Terras

Lorum ipsum is simply dummy text of the printing and typesetting industry. Lorum ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



Kasten & Dressings

Lorum ipsum is simply dummy text of the printing and typesetting industry.



Inrichtingen

Lorum ipsum is simply dummy text of the printing and typesetting industry. Lorum ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



[Foré nv / Ringlaan 11 / 9900 Eeklo / België / T +32 \(0\)9 377 77 77 / F +32 \(0\)9 378 77 77 /](#)

[Inscríbete](#)

Figuur B.5: Casus design categorie overzicht

/ HOME /
/ CREATIES /
Fore
/ TEAM /
/ GETUIGENISSEN /

Inrichtingen

Lore ipsum is simply dummy text of the printing and typesetting industry. Lore ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. Lore ipsum is simply dummy text of the printing and typesetting industry. Lore ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

[Vraag een offerte aan](#)



Realisaties



Foré nv / Ringlaan 11 / 9900 Eeklo / België / T +32 (0)9 377 77 77 / F +32 (0)9 378 77 77 / [Facebook](#) [Instagram](#) [Twitter](#) [LinkedIn](#) [#achterenvoor](#)

Figuur B.6: Casus design categorie detail

/ HOME /
/ CREATIES /
Fore
/ TEAM /
/ GETUIGENISSEN /

Huisartsen wachtpost - Hoboken

Lore ipsum is simply dummy text of the printing and typesetting industry. Lore ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularized in the 1960s with the release of Letraset sheets containing Lore ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lore ipsum.



Foré nv / Ringlaan 11 / 9900 Eeklo / België / T +32 (0)9 377 77 77 / F +32 (0)9 378 77 77 / [Facebook](#) [Instagram](#) [Twitter](#) [LinkedIn](#) [#achterenvoor](#)

Figuur B.7: Casus design realisatie

Bibliografie

- arora Varun (2013). Why we stopped using Drupal for our platform.
- Bobkov Alexey, G. S. PUTTING OCTOBERCMS INTO WORDS.
- Bobkov Alexey, G. S. (2016). OctoberCMS Documentation.
- BuildAModule (2016). BuildAModule Video Collections.
- BuiltWith (2015). CMS Usage Statistics.
- BuiltWith (2016a). Drupal 7 Usage Statistics.
- BuiltWith (2016b). Laravel Usage Statistics.
- Drupal (2015). Module developer's guide.
- Drupal (2016a). Community Documentation.
- Drupal (2016b). Community Documentation.
- Drupal (2016c). Community Forum.
- Drupal (2016d). Drupal 7 Marketplace.
- Drupal (2016e). Where is the Drupal Community?
- Drupalize-me. Drupal Tutorials.
- Drupal.org (2016). Drupal Modules.
- Falk Johan (2015). Learning Drupal.
- Gruber John (2004). Markdown.
- Kohan Bernard (2010). What is a Content Management System (CMS)?
- Learn By The Drop (2011). Installing Drupal 7.

Learn Together (2016). Install OctoberCMS on XAMPP.

MCD partners (2015). Drupal Pros and Cons: An In Depth Look.

OctoberCMS (2016a). OctoberCMS github.

OctoberCMS (2016b). OctoberCMS github documentation.

OctoberCMS (2016c). OctoberCMS USEFUL RESOURCES.

Pacurar Filip (2015). Octo-Help.

Siddharth (2009). 15 Most Important Considerations when Choosing a Web Development Framework.

SitePoint (2016). SitePoint tutorials.

Tzu-Chi Huang (2015). Stop Creating New Drupal Core Just for Fun!

Udemy (2016). Udemy online courses.

W3Techs (2016). Usage of content management systems for websites.

Lijst van figuren

5.1	Google trends waarbij verschillende populaire PHP frameworks worden vergeleken. Laravel torent hoog boven de anderen uit.	16
5.2	Google trends vergelijking tussen Drupal en Laravel framework	17
5.3	Github trending repos	18
5.4	Google Trends voor Laravel CMS	19
5.5	Github Laravel CMS	20
7.1	Parameters zoals kleuren zijn in te stellen bij sommige thema's.	25
7.2	Template folder van een thema. Overschrijf zoveel template files als nodig.	26
7.3	Blocks bevatten verschillende bouwstenen van een pagina. Deze zijn terug te vinden onder structuur > blocks.	27
7.4	Standaard worden reeds enkele blocks voorzien, nieuwe kunnen gecreëerd worden.	27
7.5	URL aliassen kunnen automatisch of manueel ingevuld worden per node.	28
7.6	Aangemaakt menu met bijhorende menu items.	29
7.7	Wanneer een pagina wordt aangemaakt is het mogelijk direct een menu item te voorzien in een geselecteerd menu.	29
7.8	Het aanbod aan modules voor Drupal 7 is immens.	31
7.9	Definieer de gewenste velden met bijhorende data types voor iedere content type.	32
7.10	Instellingen bij het aanmaken van een nieuwe view.	33
7.11	Configuratie van een veld in views Team.	34
7.12	Overzicht van de view Team	35
7.13	Team panel pagina, samenvoegen van twee views op één pagina.	36
7.14	OctoberCMS site configuratie	37
7.15	OctoberCMS folder structuur	38
7.16	Home template pagina	39
7.17	Partial implementatie	40
7.18	URL routing	41
7.19	URL routing met parameter	41

7.20	Code voor het menu, geïmplementeerd in de layout pagina.	42
7.21	Menu met parameter	42
7.22	Overzicht geïnstalleerde plugins	43
7.23	Overzicht populaire plugins, thema's en auteurs op de MarketPlace (geraadpleegd op 05/05/2016).	44
7.24	Het aanmaken van een nieuwe plugin via de Builder plugin is eenvoudig. Via het pop-up venster geef je een unieke naam en de auteur in. Personaliseer met een icoon.	46
7.25	Maak via de tabel de nodige kolommen aan. Eenmaal opgeslagen wordt een query uitgevoerd op de database.	47
7.26	Configureer een model. Vul de nodige velden aan en link ze met tabelnaam uit de database. Configureer waar nodig.	47
7.27	Configureer en link de models aan de controller klasse.	48
7.28	Mappenstructuur plugins	49
7.29	De 'head' structuur van de layout pagina	50
7.30	Injectie van CSS via de onStart functie in de PHP sectie van de pagina.	50
7.31	Injectie van CSS door gebruik te maken van Twig markup.	51
7.32	If-statement om na te gaan of een menu item actief is.	51
7.33	De Twig tag zal vervangen worden door de inhoud van desbetreffende pagina.	51
7.34	Voeg de nodige rijen en vul de kolommen in waar nodig.	52
7.35	De configuratie van alle velden is terug te vinden in de 'fields.yaml' file.	53
7.36	Het pop-up venster voor de aanmaak van een controller.	54
7.37	Overloop de collectie via een for-lus	55
7.38	Eenvoudig tekst aanpassen via de browser, zonder enige technische kennis.	56
8.1	OctoberCMS biedt een duidelijke documentatie aan op hun officiële site.	61
8.2	Drupal performantie test: overzicht van het resultaat.	65
8.3	Drupal performantie test: overzicht van het resultaat van de back-end pagina.	66
8.4	OctoberCMS performantie test: overzicht van het resultaat.	68
8.5	Drupal's HTML code om de Header te tonen is allesbehalve proper en overzichtelijk te noemen.	70
8.6	Een foutbericht in Drupal die weinig beschrijvende info bevat.	72
8.7	De foutbericht van OctoberCMS bevat voldoende informatie om het probleem te lokaliseren en verdere stappen te ondernemen.	75
8.8	Bij de installatie van Drupal worden 73 tabellen aangemaakt.	77
8.9	Bij de installatie van OctoberCMS worden 23 tabellen aangemaakt.	79
B.1	Casus design homepagina	83
B.2	Casus design team	84

B.3 Casus design testimonials	84
B.4 Casus design testimonials detail	85
B.5 Casus design categorie overzicht	85
B.6 Casus design categorie detail	86
B.7 Casus design realisatie	86

Lijst van tabellen

8.1	Een overzicht van de laadtijden van de index pagina.	65
8.2	Een overzicht van de laadtijden van de back-end pagina.	66
8.3	Een overzicht van de laadtijden van de index pagina.	67
8.4	Een overzicht van de laadtijden van de back-end pagina's.	67

