



cvo panta rhei

talentontwikkeling
voor volwassenen

Academiejaar 2014-2015

Graduaat Informatica – Netwerkbeheer

Projectwerk

Eindproject voorgedragen tot het behalen van het diploma Graduaat Informatica

Jeroen DE MEERLEER

Promotor: Wim De Bruyn

Voorwoord

In het middelbaar noemden ze mij de computercrasher. Ik had die bijnaam dan ook een beetje verdiend: iedere computer die ik durfde aanraken kreeg spontaan een blauw scherm. Gelukkig was dat maar het middelbaar en maak ik nu een eindwerk met computers zonder ze te laten crashen.

Als eindwerk had ik oorspronkelijk gekozen voor een project in Madagascar. Men is daar bezig met een kliniek op te bouwen en ik zou een webshop maken zodat de medicatie via het net te bestellen is. Echter, door wat gebrek aan informatie ben ik tot de conclusie gekomen dat als ik niet direct alle informatie kon krijgen die ik moest hebben ik niet zou slagen dit jaar. Bijgevolg heb ik aan mijn promotor Wim De Bruyn laten weten dat ik heb gezocht naar een ander project. Dit project lag wat in dezelfde lijn maar was iets waar ik direct alle informatie over had. Het project is namelijk mijn persoonlijke website.

De website staat reeds online op jeroened.be en wil daarvoor ook mijn leerkrachten die ik in mijn jaren in de Avondschool heb gehad bedanken. In het bijzonder wil ik Bert Van Vreckem bedanken. Hij was leerkracht Netwerkbeheer Linux en heeft mij een hele hoop nieuwe dingen geleerd die ik in de toekomst zeker zal gebruiken.

Een speciale vermelding gaat naar Sirrah Baldeh. Zij is medestudente bij mij in de Avondschool. Op het moment dat ik haar leerde kennen zat zij nog maar in het eerste jaar waarbij ik in het tweede zat. We zijn de beste vrienden geworden en spreken regelmatig af. Vooral op de momenten dat ik het moeilijk had was zij daar.

Sirrah, thank you for always being there. Whenever I needed it.

Jeroen De Meerleer
Gent, juni 2015

Inhoudsopgave

Voorwoord.....	i
Inhoudsopgave	ii
Inleiding	1
1 Het plan.....	2
2 Het ontwerp.....	3
2.1 Inleiding	3
2.2 Publiek gedeelte.....	3
2.2.1 De basis.....	3
2.2.2 Pagina's.....	4
2.2.3 Technologieën	4
2.3 Het administratiegedeelte.....	4
2.3.1 Verklaring van de naam.....	5
2.3.2 Het logo.....	5
2.3.3 De header.....	5
2.3.4 Het menu.....	6
2.3.5 De content.....	6
2.3.6 Knoppen, keuzerondjes en invoervelden	6
2.3.7 Dialoogvensters.....	7
2.3.8 Technologieën	7
3 De uitvoering in Symfony.....	8
3.1 De eerste kennismaking	8
3.2 De database.....	8
3.3 Het publieke gedeelte	10
3.3.1 De portfoliocontroller	10
3.3.2 De api-controller	10
3.4 Het administratiepaneel	10
3.4.1 Het aanmelden	11
3.4.2 De usercontroller.....	11

3.4.3	De menu- en page-controller.....	12
3.4.4	De portfolio-controller	12
4	De test-procedure.....	13
4.1	Inleiding	13
4.2	De functionele test.....	13
4.2.1	De resultaten	14
4.3	De usability test	15
4.3.1	De resultaten	16
5	Presto	17
5.1	De naam	17
5.2	Het design	18
5.3	Het programmeren.....	18
	Nawoord.....	20
	Noten, bronnen en referenties	21

Inleiding

In een wereld waar internet en informatie steeds belangrijker worden is de nood aan het snel beschikbaar maken ervan de hoogste prioriteit. Jammer genoeg is een internetpagina nog steeds gecodeerd volgens een bepaald stramien. Om zo'n pagina eenvoudig aan te maken zijn er door de jaren heen verschillende tools verschenen waarin een website-eigenaar gemakkelijk zo'n website kan aanmaken. Als eindwerk heb ik het framework Symfony2 bestudeerd. Dit framework is gebaseerd op PHP en stelt de programmeur in staat om op eenvoudige wijze gemakkelijk te bewerken websites te maken.

Met het Symfony2 framework kun je op een eenvoudige manier jouw database aanmaken, sjablonen voor jouw website maken en veel meer. Symfony is vandaag een van de meest gebruikte PHP-frameworks en het staat dan ook heel regelmatig vermeld op vacatures als verplichte kennis.

Tijdens mijn ervaring met het maken van een website in Symfony waren al snel de voordelen ervan te zien. Zo is het niet moeilijk om zonder veel kennis te beginnen met Symfony. Ik raad zelf aan direct met Symfony te beginnen zonder diep in te gaan op PHP. Een ander voordeel is dat het een heel snel framework is.

Jammer genoeg zijn er ook wat zaken waar je moet mee rekening houden als je een site in Symfony wilt maken. Zo ben je quasi verplicht ssh-toegang te hebben tot de server om jouw website op een deftige manier te kunnen lanceren. Zonder ssh-toegang kun je bijvoorbeeld jouw tijdelijke bestanden niet opruimen of kun je geen demo-data in jouw website steken. Daarnaast is het ook aangeraden om jouw website te draaien in de submap web. Want anders is kan iedereen die jouw website opent door kleine aanpassingen in de URL-balk jouw website naar de maan helpen.

Mijn eindproject was mijn persoonlijke website. Deze website was eigenlijk aan vernieuwing toe en dat heb ik bij deze gedaan. Wil je de code van mijn website zien kan je dit doen via mijn git-server. Deze is te bereiken via <https://git.jeroened.be/jeroened/website.git>.

1 Het plan

Toen ik in 2012 begon met de opleiding Informatica had ik niet verwacht dat ik 3 jaar later zonder veel moeilijkheden mijn eindwerk zou doen. Het kiezen van een eindwerk is echter andere koek. Wat moet je kiezen van eindwerk? Mijn oorspronkelijk idee was om voor een webdesignbedrijf servers aan te bieden waar zij 'in-house' hosting konden aanbieden. Ik had naar verschillende bedrijven mails gestuurd maar stootte altijd op een "njet". Het laatste bedrijf dat ik contacteerde bleef twijfelen tot het uiteindelijk ook "njet" werd op de dag dat ik mijn eerste afspraak had met mijn promotor Wim De Bruyn. Hij kwam af met om een webshop te maken voor een kliniek in Madagascar. Ik had echter een probleem: ik had geen informatie en mijn promotor kon geen goede informatie bezorgen.

Uiteindelijk heb ik beslist om mijn persoonlijke website te vernieuwen. Deze was aan vernieuwing toe aangezien ik een paar maanden voordien een indringer op mijn server had. Hij was niet akkoord met bepaalde zaken die ik op mijn blog schreef en had mijn private sleutel kunnen krijgen en heeft mijn server laten crashen. Ik begon dan met een nieuwe website te schrijven en deze is uiteindelijk mijn eindproject geworden.

Het starten met symfony ging redelijk vlot. Als je wat gewoon bent om te programmeren ben je hier vlot mee weg. OP 27 april kwam tot grote vreugde een eerste dynamische versie online. Dynamisch was in dit geval misschien wel wat overdreven aangezien er rechtsreeks in de database moest gewerkt worden om iets aan te passen. Deze versie kreeg zoals gebruikelijk versie 0.5 mee.

Vele commits en vele frustraties later kwam er ook een versie 1.0. Dit houdt in dat de website kan worden bewerkt zonder gebruik te maken extra technologieën als FTP of PhpMyAdmin. De website kan worden bewerkt via een administratiegedeelte. En dat is een grote vooruitgang ten opzichte van waar ik stond op 10 april.

2 Het ontwerp

2.1 Inleiding

Het kunnen maken van een mooi ontwerp is iets dat niet iedereen gegeven is. Als je het kan, is de magie niet ver af. Ikzelf ben echter grafisch niet de sterkste en veel van mijn ontwerpen zijn compleet fout en gaan velen afbreken als nineties-design. Om er toch wat deftig van te maken had ik veel hulp aan mijn goede vriendin Sirrah Baldeh. Zij heeft een goed oog voor mooi ontwerp en het volledige administratiepaneel is samen met haar ontworpen. Ik wil dan ook mijn oprechte dank betuigen voor samen met mij deze ontwerpen te maken.

2.2 Publiek gedeelte

Het publiek gedeelte van de website is ontworpen op 23 januari als voorbereiding om reeds een idee te hebben hoe ik mijn website er zou laten uitzien. Na het maken van het ontwerp vroeg ik via de sociale media wat men ervan vond.

Op 24 januari werd een eerste slicing van de website gemaakt waarbij rekening gehouden werd met de opmerkingen die ik via facebook had ontvangen.

2.2.1 De basis

Mijn uitgangspunt was dat mijn website een presentatie van mezelf moest worden. Daarom had ik ervoor gekozen om een foto van mezelf op de eerste pagina te plaatsen. Deze foto was volgens sommigen echter niet de goede foto en heb dan gekozen om op de voorpagina mijn naam met een tag-line te plaatsen.

De link met powerpoint kwam er ook redelijk snel. En de benodigde technologieën waren ook snel beslist. Als standaard-achtergrond voor de website had gekozen om een zwarte achtergrond met wat willekeurige strepen. Zo'n afbeelding vinden was ook niet moeilijk en had ik gevonden op de website subtlepatterns.com. Het lettertype is het vrij beschikbare Open Sans.

2.2.2 Pagina's

Buiten het portfolio zijn er ook pagina's als mijn Curriculum Vitae te bekijken. Deze pagina's worden weergegeven in een venster dat boven het portfolio zweeft. Dit venster heeft afgeronde randen en is licht doorzichtig.

Om de pagina's op een deftige manier weer te geven op een mobiel apparaat is gekozen om gebruik te maken van media-queries. Hierdoor is het mogelijk om bij een bepaalde maximale breedte de weergave aan te passen. Hierdoor kun je bijvoorbeeld tabellen anders weergeven dan hoe deze eruit zien op een tablet.

Sommige zaken waren echter niet af te handelen met media-queries. Zo was het niet mogelijk om mijn computerkennis op mijn CV van 2 kolommen naar 1 kolom over te zetten. Dit heb ik dan ook gedaan met jQuery.

2.2.3 Technologieën

Het ontwerp van het publiek gedeelte is gemaakt in Photoshop. De slicing werd gemaakt met HTML5. Om van de ene dia naar de andere te gaan is gebruik gemaakt van reveal.js. Dit presentatie-framework is gemaakt door Hakim El Hattab. Dit framework word door velen gebruikt als alternatief voor Powerpoint.

Waarom kies ik voor Reveal.js? Omdat het eigenlijk het enige dergelijk framework is dat ik ken. Het word door heel veel mensen gebruikt en heb ik leren kennen via Bert Van Vreckem die het gebruikte bij een presentatie over vagrant.

Om pagina's weer te geven wordt gebruik gemaakt van jQuery. Dit javascript-framework word de laatste jaren steeds meer gebruikt als vervanging voor gewone Javascript. Dit komt omdat jQuery browser-specifieke keuzes zoals Ajax-requests zelf afhandelt waardoor de programmeur hier zelf geen zorgen over hoeft te maken.

2.3 Het administratiegedeelte

Het administratiegedeelte is ontworpen door mezelf in samenwerking met Sirrah Baldeh. Om de technologie achter deze website later te herbruiken in andere projecten hebben we ervoor gekozen om er een andere naam op te plakken. De keuze voor de naam is gevallen op CmsED. Deze naam ligt in lijn met andere namen die ik reeds gebruik waarbij er op het einde ED aan toegevoegd word.

2.3.1 Verklaring van de naam

Als naar voor mijn Content Management Systeem heb ik gekozen voor CmsED. Het eerste deel (Cms) staat eigenlijk voor wat het systeem is nl. een CMS.

Een CMS of Content Management System is een systeem waarbij een website kan worden bewerkt via een beheersgedeelte van deze website. Bekende voorbeelden zijn Drupal en Wordpress. Deze voorbeelden zijn gratis te downloaden, maar vele ontwikkelaars kiezen ervoor om zelf een systeem uit te programmeren of een bestaand systeem uit te breiden.

Met het deel ED wou ik oorspronkelijk van een woord de verleden tijd maken. Ik gebruik dit stramen sinds een paar jaar en hiermee kan ook verwezen worden naar “De Meerleer” waarbij het eerste deel wordt omgedraaid.

2.3.2 Het logo

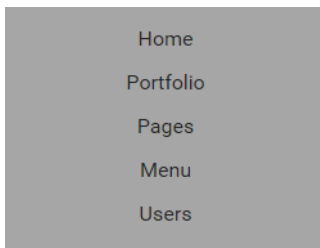


Het logo voor CmsED bestaat uit de naam CmsED met eronder een pijl die naar de toekomst wijst. De reden achter deze pijl is omdat “ed” achteraan een werkwoord in het engels meestal verwijst naar de verleden tijd. Door een pijl richting de toekomst erbij te zetten word er een tegenstelling gecreëerd zoals dit ook in de wereld meer dan genoeg het geval is.

2.3.3 De header

De header bestaat uit een grote reeks woorden die allemaal te maken hebben het webdesign of webdevelopment. Hiermee proberen we te bewijzen dat we echt iets van webontwikkeling kennen. Het lettertype dat werd gebruikt in de header is Lato. Dat eveneens een veel gebruikt gratis lettertype is.

2.3.4 Het menu

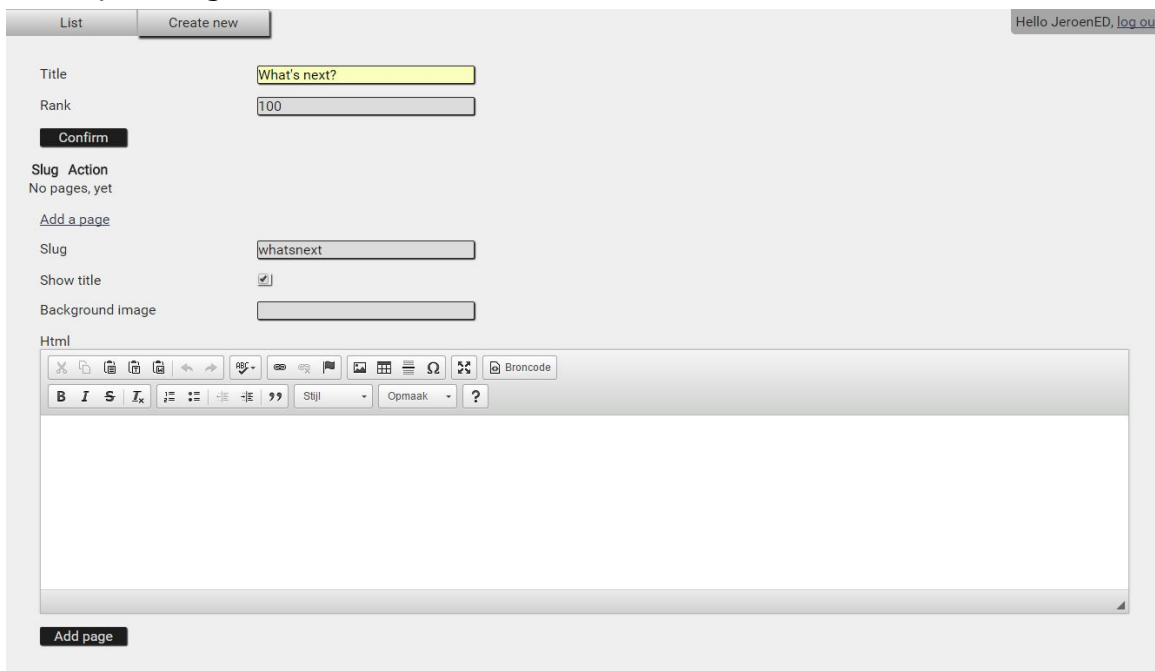


Aan de linkerkant van het scherm worden de configureerbare onderdelen weergegeven. Op mijn website worden hier Portfolio, Pages, Menu en Users weergegeven. Alle onderdelen duiden duidelijk aan wat je kan aanpassen in dit item. Het lettertype is het vrij beschikbare Roboto

2.3.5 De content

In het hoofdgedeelte worden de aanpassingen gedaan. Bovenaan is er de keuze tussen verschillende onderdelen of acties die kunnen worden gedaan zonder een specifiek item te kiezen. Daaronder word de huidige actie weergegeven. Dit kan een formulier zijn om een pagina te bewerken of dit kan ook de lijst van huidige items zijn.

In de lijst van huidige items worden de mogelijke acties weergegeven. Via deze acties kun je een item verwijderen of bewerken. Het is ook mogelijk details van het item op te vragen.

A screenshot of a content management system's editing interface. At the top, there are buttons for 'List' and 'Create new', and a user greeting 'Hello JeroenED, log out'. The main form has a 'Title' field with 'What's next?' and a 'Rank' field with '100'. Below these is a 'Confirm' button. The 'Slug' field contains 'whatsnext' and the 'Show title' checkbox is checked. There is also a 'Background image' field. At the bottom, there is an 'Add page' button. The interface includes a rich text editor toolbar with various icons for text formatting and a 'Broncode' button.

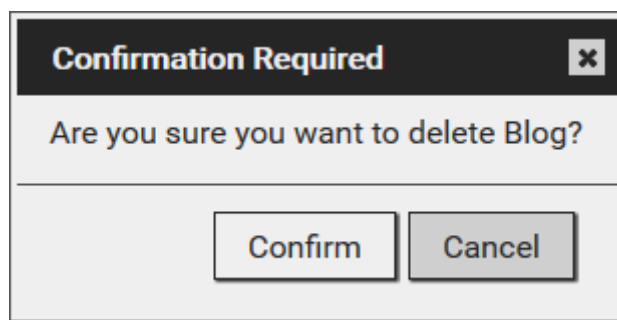
2.3.6 Knoppen, keuzerondjes en invoervelden

Dankzij css3 is het mogelijk de weergave van de invoervelden aan te passen. De stijl voor die invoervelden was allemaal ongeveer dezelfde. Voor de knoppen koos ik voor een radicaal ander ontwerp. De reden hiervoor is omdat het duidelijk moet zijn dat een knop een andere bedoeling heeft dan een invoerveld. Voor de knoppen is er

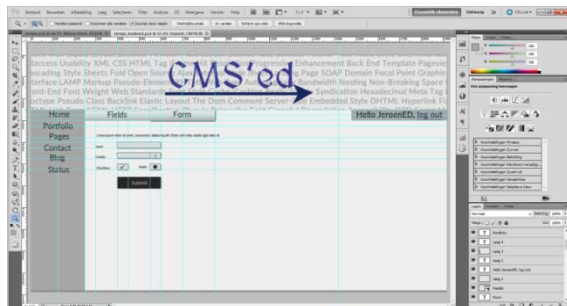
bijgevolg voor een zwarte achtergrond met witte letters gekozen en voor de invoervelden witte achtergrond met zwarte letters.

2.3.7 Dialoogvensters

Als er belangrijke zaken zouden gebeuren of zaken die niet omkeerbaar zijn wordt een bevestiging gevraagd. Pas nadat er is bevestigd dat de verwijdering mag gebeuren wordt de verwijdering uitgevoerd. De dialoogvensters waren een iets andere stijl aangezien ik deze zou maken met jquery UI. Dit is een uitbreiding op jQuery bedoeld om moeilijk te implementeren zaken, gemakkelijk te maken. Ik heb bij de slicing een thema aangemaakt in de configurator van jQuery UI.



2.3.8 Technologieën



Het ontwerp werd oorspronkelijk gemaakt in photoshop. Met het ontwerp van photoshop werd met HTML en CSS een slicing gemaakt. De lettertypes werden geladen vanuit Google Webfonts waardoor deze geen plaats innemen op onze servers.

De html-tag textarea word op de gepaste plaatsen vervangen door een wysiwyg editor. wysiwyg staat voor “What you see is what you get” en is bedoeld om op de plaatsen waar html moet worden gezet, een grafische editor te voorzien zodat html kennis niet nodig is. Deze editor, alsook jQuery, worden geladen vanuit een andere locatie op internet.

De reden waarom ik kies om jQuery en de lettertypes vanuit een content delivery netwerk te laden is omdat via een CDN wordt gekozen voor de snelste locatie. Mijn website word gehost in Roubaix. Maar voor iemand die in Amerika of China zit worden die delen geladen vanuit Silicon Valley en niet vanuit Europa. Hierdoor gaat de snelheid van mijn website de hoogte in.

3 De uitvoering in Symfony

3.1 De eerste kennismaking

Ik had in het verleden al kennis willen maken met Symfony. Het probleem was echter dat je op bepaalde momenten echt nood hebt aan ssh-toegang. Dat was op dat moment een probleem. Ik draaide op dat moment mijn website op een shared hosting en het was dus niet mogelijk om een ssh-verbinding op te zetten. Shared hosting houdt in dat uw website word gehost op een server waar tegelijk meerdere andere websites op draaien. Hierdoor kun je bij een verkeerd lopende opdracht duizenden andere websites tegelijk laten fout lopen. Sommige hosting providers lossen dit op door quasi geen rechten te geven op de ssh-verbinding, maar dit is helemaal niet ideaal. Hierdoor had ik voor het toenmalig project ervoor gekozen om geen Symfony te gebruiken.

Toen ik uiteindelijk voor dit project begon met gebruik te maken van Symfony, was dit een wereld die voor mij openging. Het maken van mijn website ging heel vlot en ik vroeg mij af waarom ik dit niet eerder had gedaan. Ik had overigens een getting-started tutorial gedownload van het internet en na een paar pagina's was ik reeds vertrokken.

3.2 De database

Het eerste dat ik maakte was de database. De database creëerde ik met Doctrine. Dit pakket maakt deel uit van symfony en is bedoeld om de database aan te maken en te besturen. Om een database te maken in Doctrine moet je jouw velden maken met PHP. Met annotaties maak je jouw specifieke configuratie voor dat veld en daarna kun je met het commando `doctrine:generate:entities` de benodigde getters en setters aanmaken. Nadat de getters en setters voor alle klassen zijn aangemaakt kun verdergaan met het aanmaken van de database.

```
1. /**
2.  * @ORM\Entity
3.  * @ORM\Table(name="portfolio")
4.  */
5.
6. class PortfolioItem {
```

```

7.
8.     /**
9.      * @ORM\Column(type="integer")
10.     * @ORM\Id
11.     * @ORM\GeneratedValue(strategy="AUTO")
12.     */
13.     protected $id;
14.
15.     /**
16.     *
17.     * @ORM\Column(type="string", length=50)
18.     */
19.     protected $title;
20.
21.     /**
22.     *
23.     * @ORM\Column(type="text")
24.     */
25.     protected $pages;
26.
27.     /**
28.     *
29.     * @ORM\Column(type="integer")
30.     */
31.     protected $rank;
32.
33. }

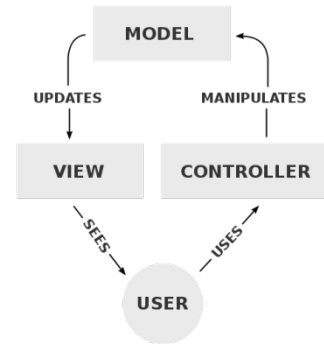
```

Nadat alle getters en setters zijn aangemaakt kun je jouw database aanmaken. Hiervoor moet alle configuratie naar de database zijn aangemaakt en moet de database ook effectief bestaan. De configuratie kan worden ingesteld in het bestand `app/config/config.yml`. Deze configuratie word ook gevraagd op het moment dat je symfony installeert via Composer. De database kan je via `doctrine:database:create` aanmaken. Nadat alles is geconfigureerd kun je met het commando `doctrine:schema:update` alle tabellen en velden aanmaken.

Je kan in Doctrine ook voorbeelddata aanmaken. Hiervoor moet echter wel een nieuwe bundle worden geïnstalleerd. Deze bundle heet `doctrine/doctrine-fixtures-bundle` en kun je via composer installeren. Nadien hoef je een klasse te maken die de interface `FixtureInterface` implementeerd waarna je in de `load`-method de database kan vullen. Het vullen gebeurt via de `$manager->persist()` method waarna een `$manager->flush()` word aangeroepen om de wijzigingen effectief weg te schrijven. Het uitvoeren van deze code gebeurt in het commando `doctrine:fixtures:load`

3.3 Het publieke gedeelte

Een website wordt vandaag de dag bijna altijd met het MVC-pattern gemaakt. MVC staat voor Model-View-Controller. In dit paradigma stuur je alle aanvragen voor de website naar de controller. De controller vraagt aan de database (het model) om de data die hij nodig heeft. De controller stuurt alle gegevens die nodig zijn voor een goede weergave naar de view die alles mooi in elkaar steekt.



3.3.1 De portfoliocontroller

Om mijn portfolio weer te geven had ik nood aan een controller die op voorhand wat code kon uitvoeren. Dit had ik nodig omdat bij alle actions in de controller mijn portfolio in de achtergrond aanwezig blijft. Bij het initialiseren van deze controller moest dan ook eerst het portfolio worden geladene. Via wat zoeken op internet heb ik code hiervoor gevonden. De code die ik moest uitvoeren was eigenlijk alles ophalen van de database want alles dat ik nodig had uit de database had ik in alle acties van de controller nodig. Het enige dat anders was in de acties was een mogelijke api-call om een pagina op te halen.

3.3.2 De api-controller

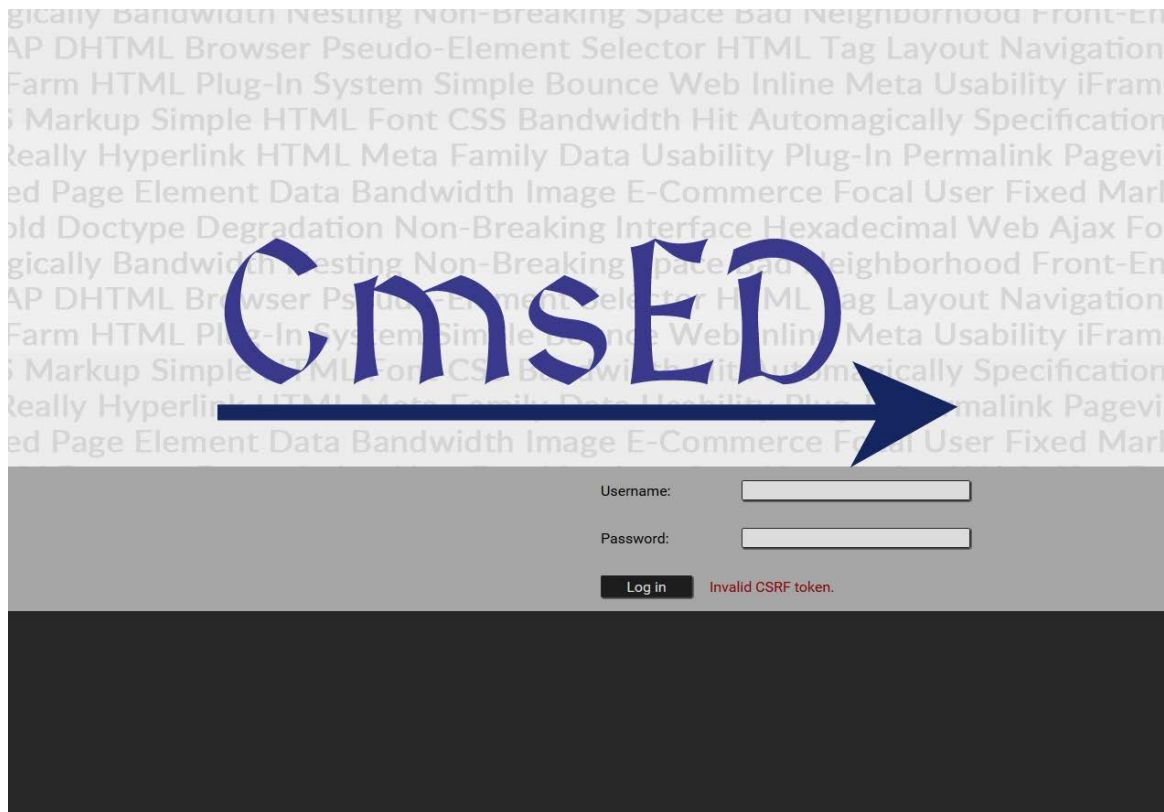
Om pagina's op te halen wordt gebruik gemaakt van een zelfontworpen api. De api word aangeroepen door een request naar de website met daarna /api/ te surfen. Door hierachter de juiste method toe te voegen met de juiste parameters wordt de juiste pagina opgehaald en kan deze in de website worden weergegeven.

3.4 Het administratiepaneel

Het administratiepaneel werd ontworpen met zoveel mogelijk flexibiliteit in het achterhoofd. Zo was het de bedoeling om ditzelfde adminstratiepaneel later te herbruiken bij een andere website waarbij bijvoorbeeld een fotoalbum zou moeten worden aan toegevoegd. Ik kwam uiteindelijk op een idee om de aanpasbare onderdelen in de linkerbalk te zetten waar er genoeg plaats was om andere zaken toe te voegen.

3.4.1 Het aanmelden

Om aan te melden moet eerst een gebruiker worden aangemaakt. Deze kan worden aangemaakt via een commando dat ik gecreëerd heb. Dit commando kan via rechte reeks een gebruikersnaam en e-mailadres krijgen door deze achteraan het commando toe te voegen. Dit is echter niet verplicht en kan daarom zonder deze gegevens worden aangeroepen. Het wachtwoord moet echter altijd via het commando worden ingegeven. Dit kan niet in een argument.



Als

een gebruiker is aangemaakt, kan worden aangemeld door een beveiligde pagina te bezoeken. Deze stuurt via de firewall automatisch door naar de login pagina waarna deze, na te zijn ingelogd, terug doorgestuurd wordt naar de pagina die de persoon wilde bezoeken.

3.4.2 De usercontroller

Om nieuwe gebruikers aan te maken kan de usercontroller worden gebruikt. In deze controller kunnen gebruikers worden bewerkt en aangemaakt. Het is ook mogelijk om gebruikers te verwijderen. Indien een nieuwe gebruiker wordt aangemaakt zal het wachtwoord van deze gebruiker automatisch worden beveiligd via het bcrypt algoritme. Dit algoritme zorgt zelf voor de extra beveiligingslaag die normaal door de programmeur moet worden geïmplementeerd. Deze beveiligingslaag heet

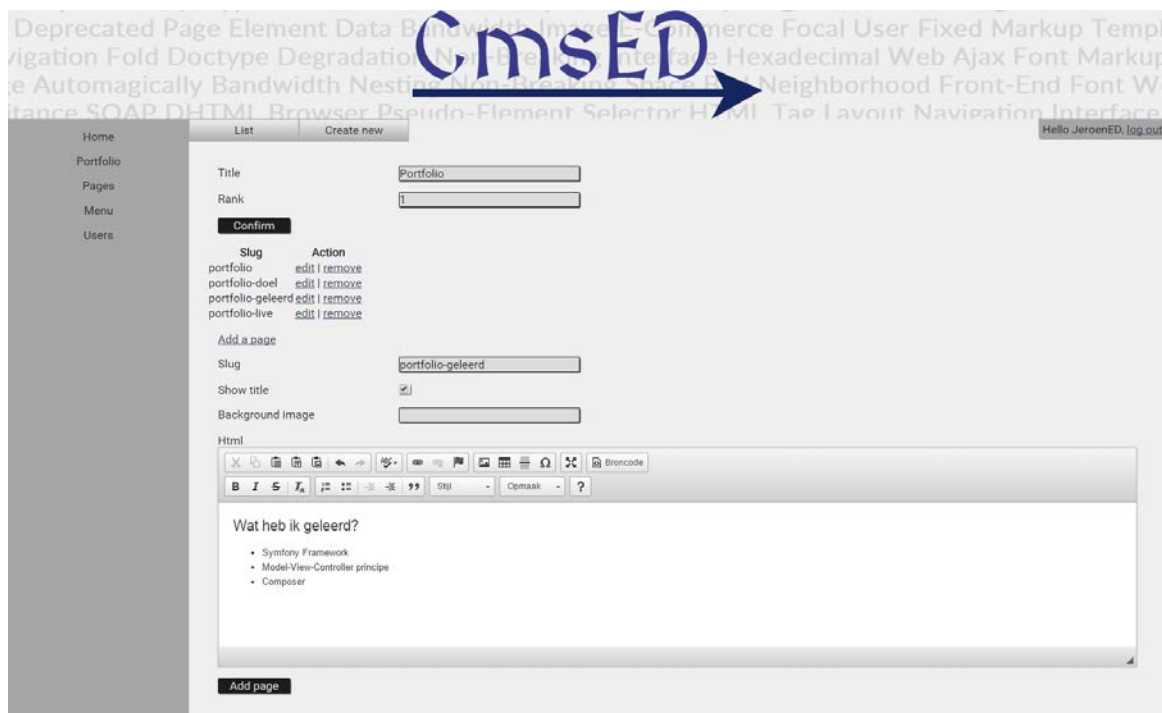
salting. Salting is het toevoegen van een willekeurige waarde aan het wachtwoord zodat bij een hack het wachtwoord niet kan achterhaald worden zonder de salt-waarde te weten.

3.4.3 De menu- en page-controller

De menu en page-controllers werden aangemaakt door middel van scaffolding. Scaffolding is in MVC-frameworks een manier om snel views te kunnen genereren. Bij scaffolding word een basis-template gebruikt waarna via een algoritme in het framework de pagina's worden gegenereerd.

3.4.4 De portfolio-controller

De portfoliocontroller had wat meer voeten in de aarde dan gewoon wat velden plaatsen in een pagina. In deze controller is er voor het aanmaken van nieuwe items een samengesteld veld. In de database zelf bestaat dit veld uit een JSON-string die bij het bekijken van de website word uitgepakt in verschillende elementen (een array). Om dit veld te maken had ik een apart formulier nodig die deze velden apart kon genereren. Dit formulier werd aangestuurd via Javascript die op zijn beurt in een verborgen veld de JSON-string maakte. Hier werd met een zelfde soort tabel gewerkt als bij de andere onderdelen.



4 De test-procedure

4.1 Inleiding

Het testen van een programma is heel belangrijk. Een programma mag niet zomaar plots een foutmelding genereren omdat een bezoeker per ongeluk een verkeerd adres heeft ingegeven. Ook mag het niet zijn dat plots alles op zwart gaat omdat er te snel naar een andere pagina is gesurft. Ook de gebruiksvriendelijkheid moet worden getest en dat doet de programmeur best niet zelf. Hij heeft namelijk al een perfect idee hoe iets moet gebeuren. Hiervoor had ik de hulp van Sirrah Baldeh. Zij kende mijn website enkel van wat ik erover had gezegd en was het perfecte test-publiek voor mijn website.

4.2 De functionele test

Bij de functionele test was het de bedoeling om te kijken of alles naar behoren werkte. Kwam ik op de pagina die ik moest krijgen of kreeg ik een error voorgeschoteld. Het testen gebeurde door het proberen te laten crashen van het systeem. Zo werden waarden aangepast zodat een pagina niet kon worden geladen. Ook werd er geprobeerd data te verkrijgen die men niet mocht verkrijgen.

Indien er een fout werd gevonden werd dit in de issues van de GIT-repository gezet. Er werd een template gemaakt waar deze fouten werden beschreven. Dit template zag er als volgt uit. Het was zoals de git-repository dit voorschrijft geschreven in Markdown formaat. De variabele zaken in dit template zijn geplaatst in dubbele accolades `{{ }}` zoals dit ook in symfony gebeurt.

```
Title: {{ Korte omschrijving van het probleem }}

{{ Uitgebreide omschrijving van het probleem }}

## Prerequisites

* {{ lijst van alles wat benodigd is om de bug te laten werken
}}
```

Steps to reproduce

1. {{ Stappen die moeten worden ondernomen om de fout in werking te laten treden }}

Expected

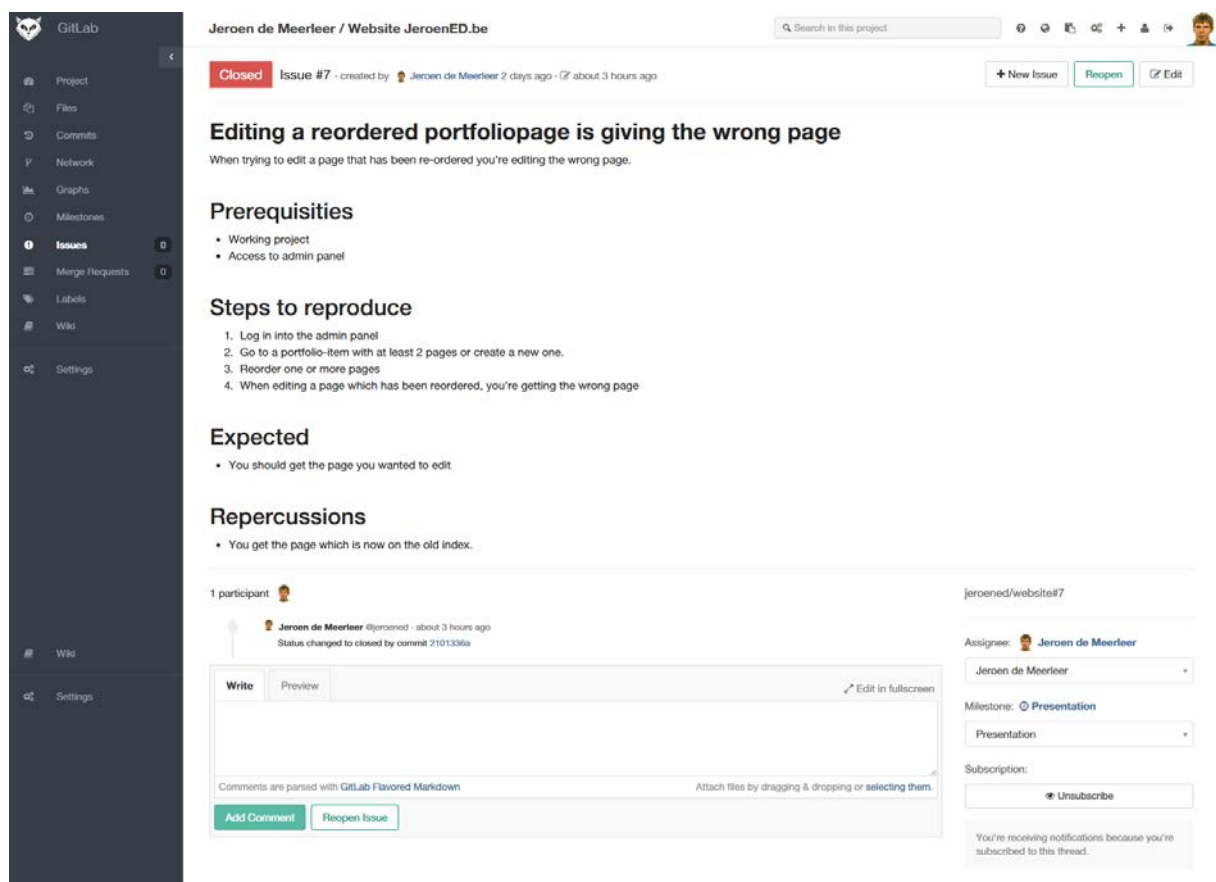
* {{ Lijst van wat verwacht werd }}

Repercussions

* {{ Gevolgen van de bug }}

4.2.1 De resultaten

Veel fouten werden er eigenlijk niet gevonden. De eerste gevonden fout was een fout waarbij een verkeerde pagina werd aangevraagd. In plaats van een foutmelding “page not found” weer te geven bleef je gewoon wachten terwijl er niet kwam. Dit heb ik opgelost door indien een niet bestaande pagina werd aangevraagd een response te geven met “Page not found”.



The screenshot shows a GitLab issue page for the project 'Jeroen de Meerleer / Website JeroenED.be'. The issue is titled 'Editing a reordered portfolio page is giving the wrong page' and is marked as 'Closed'. It was created by Jeroen de Meerleer 2 days ago and closed about 3 hours ago. The issue description includes sections for 'Prerequisites', 'Steps to reproduce', 'Expected', and 'Repercussions'. The 'Steps to reproduce' section lists four steps: 1. Log in into the admin panel, 2. Go to a portfolio-item with at least 2 pages or create a new one, 3. Reorder one or more pages, 4. When editing a page which has been reordered, you're getting the wrong page. The 'Expected' section states 'You should get the page you wanted to edit'. The 'Repercussions' section states 'You get the page which is now on the old index.' The issue has one participant, Jeroen de Meerleer, who commented that the status was changed to closed. The right sidebar shows the assignee as Jeroen de Meerleer, the milestone as 'Presentation', and an 'Unsubscribe' button. The bottom of the page shows a 'Write' comment box with 'Add Comment' and 'Reopen Issue' buttons.

De tweede fout die was gevonden was het feit dat een pagina in dezelfde sessie was geopend, niet kon opnieuw geopend worden. De reden hiervoor was omdat bij het sluiten van de pagina, de pagina niet gesloten werd maar enkel onzichtbaar werd gemaakt. Dit gaf tot gevolg dat de pagina opnieuw werd opgeroepen, maar dat hij onzichtbaar bleef omdat jQuery de eerste pagina opnieuw in het midden plaatste.

De derde fout had te maken met een nieuwe feature die ik er had ingestoken nadat Sirrah mij haar testresultaten gaf (lees kopje 4.3 voor meer info) Het probleem dat ontstond was dat als een portfoliopagina van plaats was gewisseld, en je wilt deze nadien bewerken, je de verkeerde pagina in de editor krijgt. Dit kwam omdat de index naar de pagina niet was aangepast, waardoor de verkeerde pagina werd gevonden.

4.3 De usability test

Om te testen of alles logisch in elkaar zat had ik de hulp van Sirrah Baldeh. Als programmeur heb je zelf een idee hoe iets moet worden aangemaakt maar dat is meestal niet het geval voor de persoon die het effectief moet gebruiken. Daarom had ik aan een persoon die er niets van kent gevraagd om een paar zaken op mijn website aan te passen en te kijken of dit wel duidelijk was dat het zo moest gebeuren. De resultaten werden wederom in de GIT-repository geplaatst waarna ik deze kon aanduiden of deze fout is opgelost.

Om De usability-issues in de Git-repository in te geven werd wederom een template in markdown aangemaakt. Dit keer zag dit er anders uit aangezien dit niet om echte fouten ging, maar meer om inconsistenties in de logica van de website. Dit template zag er als volgt uit.

```
Title: {{ korte uitleg van het probleem }}
```

```
{{ Uitegebreide omschrijving van het problem }}
```

```
## Reason
```

```
{{ De redden waarom dit niet goed of onduidelijk is }}
```

```
## How to fix
```

```
{{ Voorstel van een mogelijke fix }}
```

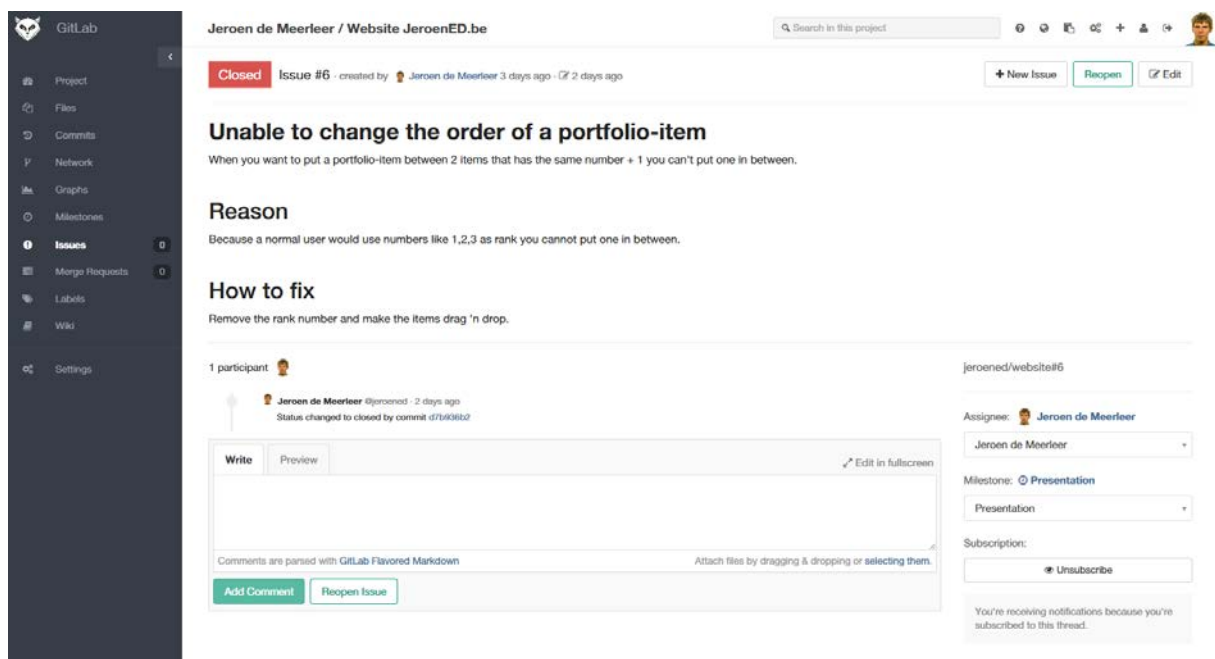
4.3.1 De resultaten

De resultaten die Sirrah gaf waren vooral gericht op design. Zo gaf ze bijvoorbeeld een opmerking op mijn scaffolding tabellen. Bij deze tabellen word enkel de ruimte gebruikt die echt nodig is. Hierdoor komen de acties te dicht bij de naam van het item waardoor dit onduidelijk is. Dit werd opgelost door een klasse aan elke cel mee te geven. Door gebruik te maken van deze klasse in de css was het mogelijk om een breedte aan de label-kolommen mee te geven.

Iets anders dat ze zei was het feit dat de log-out link te veel naar de rand was geplaatst. Hierdoor was het op een tablet mogelijk dat je naast het scherm tikte waardoor je niet uitgelogd werd.

Naast de zaken die Sirrah mij had gegeven, had ikzelf ook nog een paar zaken ontdekt. Zo was het zeer moeilijk om portfolio-items te verplaatsen op de website. Er was een property rank in de database maar dit was echter moeilijk in te schatten waar dit moest op worden ingesteld. Als je bovendien als 2 en 3 had gebruikt was het onmogelijk om hier nog een tussen te plaatsen.

Ditzelfde probleem was ook aanwezig op de portfolio-items zelf. De subpagina's verplaatsen was ook niet gemakkelijk. Bij het oplossen van deze fout werd een andere fout gecreeërd, waarbij het aanpassen van een verplaatste pagina niet de juiste pagina gaf.



The screenshot shows a GitLab issue page for the project 'Jeroen de Meerleer / Website JeroenED.be'. The issue is titled 'Unable to change the order of a portfolio-item' and is marked as 'Closed'. It was created by Jeroen de Meerleer 3 days ago. The issue description explains that a user cannot place a portfolio item between two items with consecutive rank numbers (e.g., 1, 2, 3) because a normal user would use such numbers. The 'Reason' section states that because a normal user would use numbers like 1, 2, 3 as rank, you cannot put one in between. The 'How to fix' section suggests removing the rank number and using drag-and-drop. The issue has one participant, Jeroen de Meerleer, who changed the status to closed. The right sidebar shows the issue is assigned to Jeroen de Meerleer, has a milestone of 'Presentation', and the user is subscribed to the thread.

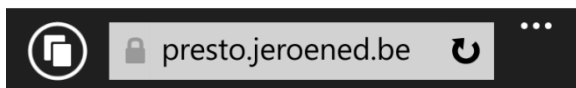
5 Presto



Please enter your presentation identifier

Sorry, presentation does not exist

Reveal the treasure



De presentatie van het eindproject was gepland voor 15 juni. Hierbij moest ik mijn project voorstellen aan een jury aan de hand van een presentatie. Omdat deze presentatie moest niet per se Powerpoint zijn had ik ervoor gekozen om deze in Reveal.js te maken. Toen ik aan mijn presentatie begon wilde ik iets extra doen. Iets waar niemand aan heeft gedacht en waarmee ik een extra indruk kon mee nalaten.

Uiteindelijk ben ik erbij gekomen om een extra zijprojectje op te zetten. Het projectje is genoemd Presto. Met Presto is het mogelijk de presentatie die je zelf gemaakt hebt ik reveal.js op het net te plaatsen, waardoor uw publiek zonder veel moeite op de achterste rij kan gaan zitten

en de presentatie gewoon kan blijven volgen via zijn smartphone.

5.1 De naam

De naam Presto is een verwijzing naar de oude renderings-engine van de browser Opera. In 2013 had Opera Software ervoor gekozen om hun eigen Presto-engine ten grave te dragen ten voordele van de nieuwe Blink-engine die door internetgrootmacht Google is ontworpen. Blink is een fork van Webkit die oorspronkelijk door apple was gemaakt. Velen, alsook mezelf, vonden dit de

slechtste keuze in de informatica wereld sinds jaren. Om de naam van de engine nog wat te laten voortleven heb ik ervoor gekozen om deze naam te gebruiken.

Een andere reden ik voor deze naam koos was om een verwijzing te maken naar Prezi. Dit is een gelijkaardig systeem dat ook presentaties in de cloud aanbied, het verschil is echter dat hij bij mij op de eigen server kan draaien.

5.2 Het design

Voor het design was er niet veel nodig. Enkel een homepagina waar een presentatie identifier kon worden ingevoerd. De rest van het design werd gedaan door reveal.js.

De inspiratie voor het logo was een presentatiescherm. Een presentatiescherm dit men meestal nodig heeft om een presentatie te geven.



Er werd daarnaast ook een favicon aangemaakt zodat de presto-site direct herkenbaar is zonder de pagina te moeten openen. Het favicon is quasi hetzelfde als het logo met als verschil dat niet de volledige naam maar enkel de eerste letter, nl. p, word getoont.

5.3 Het programmeren



Om aan een eerste dynamische versie te geraken was er niet veel nodig, de volledige codebase bestond uit slechts 14kb aan code die door mij was geschreven. En bevat

dan nog eens voornamelijk de licentie header. Zonder deze licentie headers zitten we nog maar aan 6,55kb wat 2 keer niets is.

Reveal.js word geladen vanuit een CDN zoals dit ook word gedaan bij mijn hoofdwebsite. Via API-calls naar deze CDN worden alle mogelijke versies van Reveal.js opgehaald, net zoals de mogelijke themes die standaard bij reveal.js worden geleverd.

De volledige broncode van Presto is te vinden op mijn git-repository onder <https://git.jeroened.be/jeroened/presto.git>.

Nawoord

Als ik nu terugkijk op mijn eindproject zie ik een project waar ik echt trots op ben. Toen ik een eerste volledig functionele versie had kon ik amper geloven dat ik dit voor elkaar zou kunnen krijgen. Ik kan zelf amper geloven dat ik überhaupt hier ben geraakt! Wat heb ik geleerd tijdens dit eindproject: veel! Zo leerde ik iets dat ik eigenlijk al veel eerder had moeten doen: Symfony. Daarnaast leerde ik ook werken met composer.

Maar het allerbelangrijkste vind ik wat ik over mezelf heb geleerd. Ik heb geleerd om gewoon 'into the wild' te gaan. Ik ben soms wat bang om iets nieuws te leren. Iets dat ik niet ken toch willen gebruiken en er gewoon jouw ding mee te doen.

In al deze jaren op de Avondschool heb ik fantastische mensen leren kennen. De speciale vermeldingen heb ik reeds gedaan maar zonder Sirrah was dit project echt niet mogelijk geweest. Ik hoop dat we elkaar nog veel zien en ik hoop dat we elkaar nooit laten vallen. Naast Sirrah wil toch nog een paar mensen bedanken die ik toch ergens het vermelden waard vind. Dit zijn David Duym en Gunther Bradt. Zij zijn medestudenten Informatica en zijn samen met mij gestart. David en Gunther zijn twee maten waarmee ik veel plezier mee heb gehad en zal hen niet snel vergeten.

Naast Sirrah, David en Gunther wil alle leerkrachten bedanken die mij les hebben gegeven de voorbije 3 jaar. De leerkrachten waren geen doetjes en dat was effectief nodig. Het waren 3 prachtige jaren!

Jeroen De Meerleer

Noten, bronnen en referenties

Referenties gebruikt op de website:

- [Reveal.js \(lab.hakim.se/reveal-js\)](http://lab.hakim.se/reveal-js)
- [jQuery \(jquery.com\)](http://jquery.com)
- [Subtle Patterns \(subtlepatterns.com\)](http://subtlepatterns.com)
- [IconMonstr \(iconmonstr.com\)](http://iconmonstr.com)
- [Responsive Data tables \(css-tricks.com/responsive-data-tables\)](http://css-tricks.com/responsive-data-tables)
- [Initializable Controller Interface \(matt.drollette.com/2012/06/calling-a-method-before-every-controller-action-in-symfony2\)](http://matt.drollette.com/2012/06/calling-a-method-before-every-controller-action-in-symfony2)
- [Symfony Tutorials \(symfony.com/doc/current/index.html\)](http://symfony.com/doc/current/index.html)
- [Open Sans \(google.com/fonts/specimen/Open+Sans\)](http://google.com/fonts/specimen/Open+Sans)
- [Lato \(google.com/fonts/specimen/Lato\)](http://google.com/fonts/specimen/Lato)
- [Roboto \(google.com/fonts/specimen/Roboto\)](http://google.com/fonts/specimen/Roboto)

Andere bronnen:

- [Diagram MVC-paradigma: wikipedia](#)