Technisch Ontwerp

iConcepts

“EenmaalAndermaal”

|  |  |
| --- | --- |
| *Auteurs* | Groep 14  Mehmet Batal: ID 600720,  Danny Hageman 608227,  Hendrik Otten 619129,  Bram Soutendam 599164,  Elviana Cornelissen 598830 |
| *Cursus* | *I-Project* |
| *Opdrachtgever* | *iConcepts* |
| *Begeleiders* | *Rein Harlé – Product Owner*  *Ben van Hoof – Scrum coach*  *Rody Aldenhoven – Expert Professional Skills* |
| *Onderwijsinstelling* | *Hogeschool van Arnhem en Nijmegen* |
| *Plaats & datum* | *Arnhem, 13 juni 2019* |
| *Versie nummer*  *Status* | *1.0*  *In Bewerking* |



Inhoudsopgave

[1. Inleiding 3](#_Toc11321221)

[2. Netwerkarchitectuur 4](#_Toc11321222)

[3. Database 5](#_Toc11321223)

[3.1. Datatypen & defaults 5](#_Toc11321224)

[3.2. Check-Contraints 6](#_Toc11321225)

[3.3. Conversie 7](#_Toc11321226)

[4. Front-end-framework 8](#_Toc11321227)

[5. Responsiveness 8](#_Toc11321228)

[6. Javascript, AJAX en jQuery 10](#_Toc11321229)

[6.1. Waarom is er javascript gebruikt? 10](#_Toc11321230)

[6.2. Timer 10](#_Toc11321231)

[6.3. Rubrieken 11](#_Toc11321232)

[7. Bijlage 13](#_Toc11321233)

[7.1. Database wijzigingen 13](#_Toc11321234)

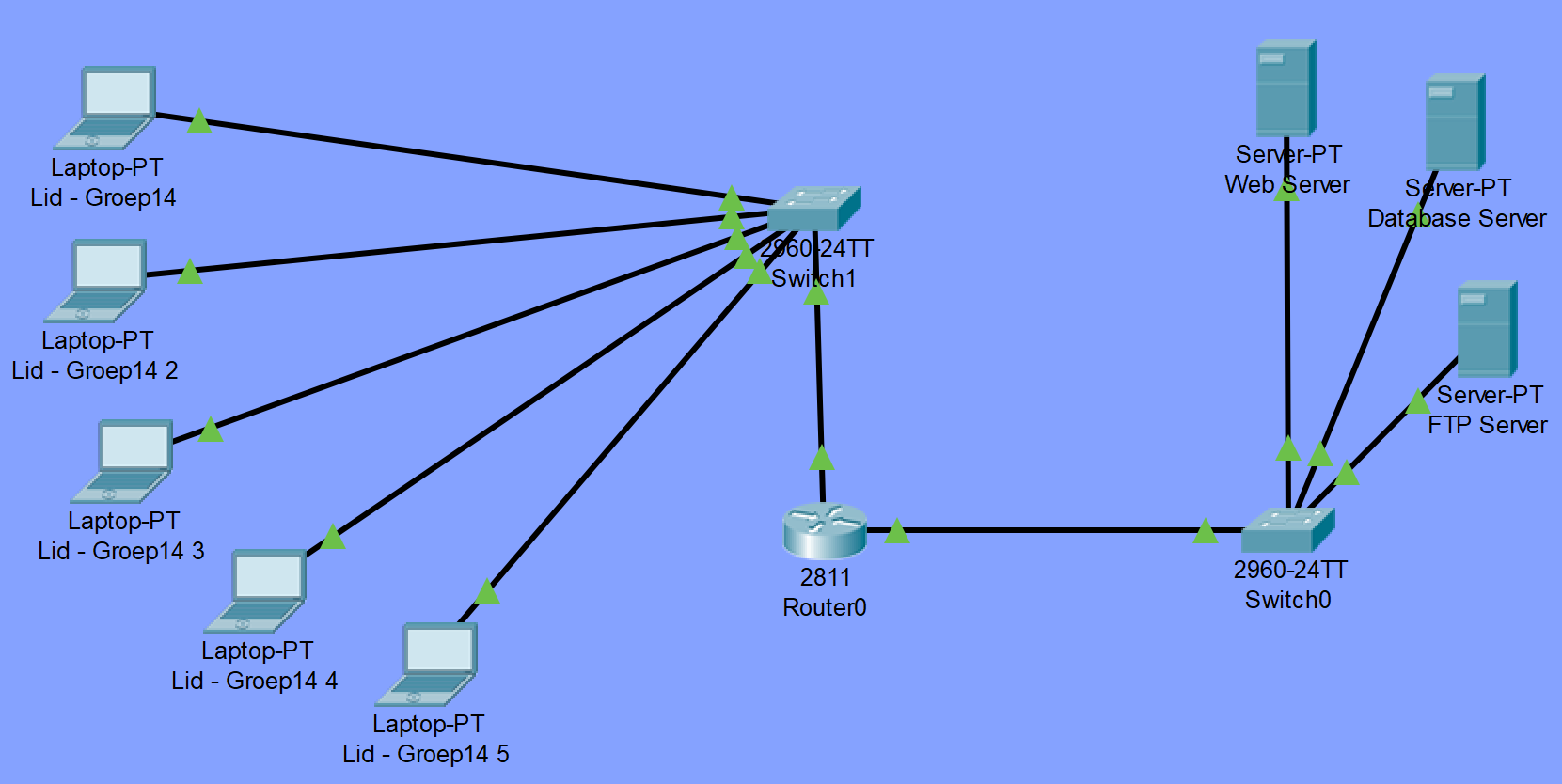
# Inleiding

In dit document worden de technische specificaties van de website ‘’EenmaalAndermaal” voor iConcepts beschreven. Onder andere staan hierin welke functies en variabelen er zijn gebruikt voor het realiseren van de website. Ook worden de gebruikte servers getoond en hoe ze in verbinding staan met de website. In dit document wordt dus vooral verteld ‘hoe’ verschillende componenten werken op/met de site.

# Netwerkarchitectuur

Om de website daadwerkelijk voor iedereen beschikbaar te stellen, is er een serveromgeving beschikbaar gesteld. Het ontwikkelen gebeurd per groepslid op een lokale server, zodat toevallige voorbijgangers niet die veranderingen zien op de server, zodat de server nog gewoon goed werkt. De server is dan ook bedoeld voor producten die af zijn en goed werken en om deze te presenteren aan de product-owner. Dit betekend dus dat er gewerkt wordt volgens de [OTAP](https://nl.wikipedia.org/wiki/OTAP)werkwijze.

* Aangereikte servers iConcepts
  + Microsoft SQL Server
  + Internet Information Server (IIS) Webserver
  + PHP 7.2
  + FTP Server
  + PHP mail functie
* Lokale omgevingen
  + Microsoft SQL Server Management Studio 17
  + PHP 7.2
  + FileZilla
  + XAMPP Versie: 7.2.12

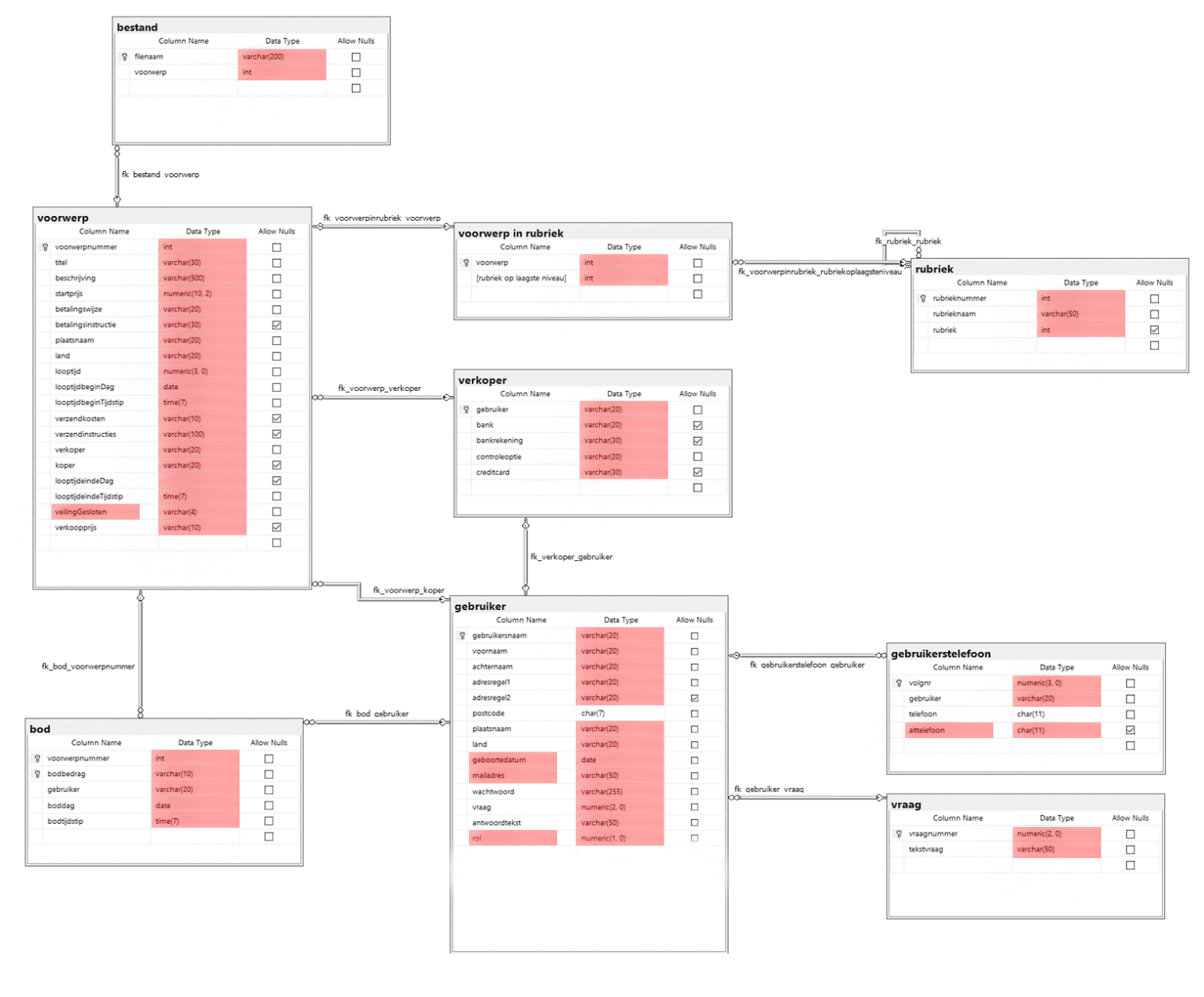
Hieronder staat een afbeelding van het netwerk waarin gewerkt is. Je ziet dat er drie servers aanwezig zijn. Dat zijn de servers die hierboven genoemd zijn. Ook zijn onze laptops aangegeven in het ontwerp. Zoals je ziet op de afbeelding communiceren onze laptops met de servers en andersom.

Figuur 2: Netwerkarchitectuur

# Database

## Datatypen & defaults

Hieronder staat het logisch relationeel schema van onze database. In het rood staan datatypen en namen die wij gewijzigd hebben. Dit wordt in de bijlagen gedocumenteerd (zie bijlage 7.1).



Figuur 3: Logisch Relationeel Schema

## Check-Contraints

In onze database zitten een aantal check-constraints. Deze check constraints zitten in onze database om vuile data te omzeilen, zodat de database niet vol raakt met onzinnige data. Hieronder staan al onze check-constraints per tabel beschreven.

**Tabel ‘gebruiker’**

* Op de kolom ‘geboortedatum’ zit een check constraint die een datum dan later dan vandaag tegenhoud.
* Op de kolom ‘wachtwoord’ zitten twee check constraints. De eerste is dat een wachtwoord minimaal 7 karakters moet bevatten. De tweede is dat een wachtwoord minimaal één letter en één cijfer moet bevatten.

**Tabel ‘voorwerp’**

* Op de kolom ‘betalingswijze’ zit een check constraint die kijkt of de betalingswijze in een van de vier opgegeven opties zit.
* Op de kolom ‘startprijs’ zit een check constraint die kijkt of de opgegeven startprijs groter is dan nul.
* Op de kolom ‘veilinggesloten’ zit een check constraint die kijkt of de data ‘wel’ of ‘niet’ is. Hiermee wordt gekeken of de veiling is gesloten.
* Met de check-constraint ‘ck\_voorwerp\_verkoopprijs\_negatief’ wordt gekeken of de verkooprijs gelijk of groter is dan de startprijs.
* Met de check-constraint ‘ck\_voorwerp\_verkoper\_geen\_koper’ wordt gekeken of de koper van een product niet ook de verkoper was van dat product.
* Met de check-constraint ‘ck\_looptijd’ wordt gekeken of de data ‘1-3-5-7-10’ gebruikt wordt, dit is de looptijd in dagen. Er kan dus bijvoorbeeld geen 12 dagen gekozen worden want dit wordt tegen gehouden door de check-constraints.

**Tabel ‘bod’**

* Met de check constraint ‘ck\_NietBiedenEigenProduct’ wordt gekeken of de koper niet op zijn eigen product biedt.
* Met de check constraint ‘CHK\_VeilingOpen’ wordt gekeken of een veiling nog open staat of al gesloten is.

## Conversie

In dit deel wordt er beschreven hoe wij de conversie hebben aangepakt. Het zal per tabel worden aangegeven hoe dit is gedaan, en waarom wij bepaalde keuzes hebben gemaakt.

Van tabel ‘Users’ naar tabel ‘Gebruikers’

Veel gebruikers uit het conversiescript hadden geen: voornaam, achternaam, adres, geboortedatum en e-mail. Deze items mogen in onze database geen NULL zijn er moest dus data worden toegevoegd. Dus hebben wij ervoor gekozen om overal ‘Onbekend’ van te maken behalve bij de geboortedatum, dat moest een datum zijn dus is er gekozen voor 01/01/1970. Er is ook gebruik gemaakt van de LEFT functie, dit is gedaan omdat sommige data uit de conversie meer karakters bevatte dan dat er werd toegelaten in onze database. De gebruikersnamen uit de conversie bevatten meerdere dezelfde gebruikersnamen dat is ook niet toegestaan in onze database, daarom hebben wij gebruik gemaakt van DISTINCT. DISTINCT pakt geen dubbele waardes, dus is elke gebruikersnaam uniek.

Van tabel ‘Categorieën’ naar tabel ‘Rubriek’

De categorieën uit het conversie script begonnen bij een -1 als rubrieknummer, maar onze site hanteert dit niet. Daarom hebben wij ervoor gekozen om dat met één op te hogen, zodat alles nog wel wordt overgezet. We hebben LEFT gebruikt omdat de naam van de categorieën te groot was voor onze database, onze database staat maximaal 50 karakters toe. De parent kolom is ook steeds met één opgehoogd omdat het rubrieknummer dus ook is opgehoogd.

Van tabel ‘Items’ naar tabel ‘Verkoper’

Sommige gebruikers zijn ook verkopers in het conversie script. Deze gebruikers worden in de tabel verkoper geplaatst. Sommige verkopers stonden er twee keer in, dit laat onze database niet toe en daarom is er gebruik gemaakt van de DISTINCT functie. De verkopers uit de conversie hadden geen bank en bankrekening, deze hebben wij als waarde NULL gegeven. Daarna worden de rollen van alle gebruikers die in de verkoper tabel zitten geüpdatet naar 3, ons nummer voor een verkoper.

Van tabel ‘Items’ en ‘illustraties’ naar tabel ‘Voorwerp’

De kolom ID heeft de functie RIGHT gekregen. Hier worden de rechter negen nummers gepakt van het ID, omdat een bigint groter is dan een int en dit word niet toegelaten in onze database. Er is geen gebruik gemaakt van left, omdat er dan problemen waren met dubbele waardes, ook al wordt de DISTINCT functie gebruikt. Voor een aantal kolommen is de LEFT functie toegepast. Ook dit doen we weer omdat onze database minder karakters toe laat dan er zijn in de conversie. De betalingswijze mag in onze database geen NULL zijn, maar toch waren er voorwerpen uit de conversie die geen betalingswijze hadden. Daarom hebben wij gekozen om elk product ‘Paypal’ als betalingsmethode te geven. De voorwerpen uit de conversie bevatten geen looptijd, daarom hebben wij een standaard looptijd van 7 dagen meegegeven.

Van tabel ‘illustraties’ naar tabel ‘Bestand’

De kolom illustratiefile bevatte langere bestandsnamen dan onze database toeliet, daarom is ook hier weer gebruik gemaakt van een LEFT, maar dan t/m 200 karakters. Omdat eerder het ID ook al is aangepast zodat hij in onze database past, passen wij hem nu weer op dezelfde manier aan. Door het gebruik van RANK hebben wij ervoor gezorgd dat hij maximaal 4 afbeeldingen per voorwerpnummer pakt.

Van tabel ‘Items’ en ‘illustraties’ naar tabel ‘[Voorwerp in rubriek]’

Het ID word weer aangepast met de functie RIGHT, zoals eerder is beschreven. Ook het rubrieknummer wordt hier weer opgehoogd met 1, zoals eerder is beschreven.

# Front-end

## Framework

Wij hebben gekozen voor ‘Foundation’ als front-end-framework. Wij zijn op deze keuze gekomen, nadat er een onderzoeksrapport is gemaakt over verschillende front-end-frameworks. In dat onderzoek is als eerste een longlist gemaakt van vijf verschillende frameworks. Op die vijf frameworks hebben wij een aantal criteria toegepast, waardoor er drie overbleven. Deze drie zijn allemaal getest, en wij hebben de beste gekozen. Foundation werkte voor ons het makkelijkst en het best. Verder waren er veel voorbeelden van websiteonderdelen te vinden op hun website. Doordat er zo veel voorbeelden te vinden waren is onze keuze uiteindelijk gevallen voor Foundation.



Figuur 1: Foundation

## Responsiveness

Wij hebben foundation gebruikt om de website responsive te maken. Responsiveness betekent dat de website zich aanpast aan andere devices, dus niet alleen voor een laptop of computer maar ook voor een telefoon of tablet.

Foundation stelde een grid beschikbaar dat is opgedeeld in 4 vakken: een header, een aside, een main en een footer. De hele website zit verwerkt in dat grid. Hierdoor is alles responsive. Voor de telefoon hebben wij zelf een aantal aanpassing gemaakt, zoals het verplaatsen van de zoekbalk en een kleiner logo gebruiken zodat alles netjes blijft.

# Back-end

## Javascript, AJAX en jQuery

In het product is gebruik gemaakt van javascript. In dit hoofdstuk worden de meest belangrijke onderdelen van de website die gebruikt hebben van javascript beschreven. De timer, het navigatie menu, rubrieken kiezen op laagste niveau en het toevoegen van meerdere bestanden.

## Waarom is er javascript gebruikt?

De grootste reden waarom hiervoor gekozen is, is zodat het mogelijk is om met javascript een pagina te updaten zonder dat de gebruiker de pagina zelf moet herladen. Met PHP is dat wat lastiger, bijvoorbeeld als je een timer wil plaatsen die blijft lopen. Omdat je niet live ziet wanneer het precies afloopt op seconden, kan het zijn dat de gebruiker te laat is om te bieden zonder dat hij het zelf merkt. Dus door gebruikt te maken van javascript, wordt deze situatie voorkomen.

Er zijn verschillende manieren gebruikt in javascript om een onderdeel werkend te krijgen. Hieronder zal uitgelegd worden welke delen van het product javascript hebben gebruikt, en hoe dat is gebeurt.

## Timer

Op de productpagina staat een timer die afloopt en aangeeft of een veiling gaat sluiten of niet. De timer voert een functie uit, vanaf de datum wanneer het product is aangemaakt tot de einddatum die berekend is in de database, met behulp van de looptijd. Dit wordt gedaan met PHP, maar het gedeelte dat de timer live update wordt gedaan met javascript.

## Rubrieken

Bij het navigatie menu en het kiezen van de laagste rubriek is er gebruik gemaakt van AJAX. AJAX is een manier om informatie op te vragen via de server en ook informatie te verkrijgen van de server nadat de pagina geladen is. De informatie wordt automatisch geüpdatet zodat de relevante informatie weergegeven wordt.

Links op elke pagina staan alle rubrieken weergegeven. Dit wordt het navigatie menu genoemd. Het navigatie menu werkt als volgt: er wordt op een rubriek geklikt en deze dan klapt uit, en hier worden de sub-rubrieken van die hoofdrubriek wordt weergegeven. En dat gaat zo verder totdat het laagste niveau is bereikt. De reden dat hiervoor javascript gebruikt is komt doordat er performance issues ontstonden nadat het conversiescript uitgevoerd werd. De pagina laadde te lang zonder dit stuk javascript. Alle rubrieken werden in eerste instantie tegelijk geladen en onder elkaar gezet. Met 3000+ rubrieken is dat wel te begrijpen waarom het zo lang duurt. Daarom is er gebruik gemaakt van jQuery en Ajax. jQuery is een javascript library en maakt het gebruiken van javascript makkelijker.

Met elke klik op een rubriek zal er een verzoek verstuurd worden die controleert of elk rubriek een sub-rubriek heeft. Hieronder zie je een deel van de code over hoe er een verzoek verstuurd wordt.

var ajax = new XMLHttpRequest();

ajax.open("GET", "get-rubrieken.php?rubrieknummer=" + rubrieknummer, true);

ajax.send();

*Figuur 4: Verzenden verzoek deel code*

Wanneer er een antwoordt wordt gegeven zal er een HTML-code aangemaakt worden die de sub-rubrieken laat zien. De nieuwe HTML-code wordt dan geplaatst in het element die een *id* heeft van “sub-rubriek”.

ajax.onreadystatechange = function() {

if(this.readyState == 4 && this.status == 200){

var data = JSON.parse(this.responseText);

if(data.length == 0){

var html = "";

}else{

var html = "<option disabled selected> Kies een subrubriek </option>";

}

for(var a = 0; a < data.length; a++){

html += "<option value='" + data[a].rubrieknummer + "'>"+ data[a].rubrieknaam +"</option>";

}

document.getElementById("sub-rubriek").innerHTML = html;

}

};

*Figuur 5: Het uitlezen van gevraagde data deel code*

Als er nog een keer op dezelfde rubriek gedrukt wordt zal het gedeelte met sub-rubrieken inklappen en de HTML-code verwijderen zodat het voorkomt dat de pagina eigenlijk te veel html heeft en te lang moet laden.

Bij het toevoegen van een veiling op de voorwerpplaatsenpagina kan je kiezen in welke rubriek die veiling komt, door middel van een select box. Dit kan alleen op het laagste rubriek. Als er een optie gekozen is zal er automatisch gezocht worden naar de sub rubrieken van die rubriek, en die weergeven in de volgende select box. Totdat het op het laagste niveau is. Dit wordt verstuurd met een form naar de database. Dit werkt op dezelfde manier met het navigatie menu. Als er een rubriek gekozen is zal hij vragen naar de bijbehorende sub rubrieken.  
  
**Bestanden toevoegen**

Op de voorwerp toevoegen pagina is het ook mogelijk om afbeeldingen te plaatsen. Dit wordt met PHP afgehandeld. Dus voor het plaatsen van meerdere bestanden worden de ‘bestand toevoegen’ knoppen toegevoegd. Dit maakt het overzichtelijker en makkelijker om afbeeldingen te kiezen.   
Voor het toevoegen van knoppen is er gebruik gemaakt van jQuery. Als er op de knop gedrukt wordt om meerdere bestanden toe te voegen dan zal de ‘bestand toevoegen’ knop zichtbaar worden. Hoe dit werkt gaat als volgt. Als op een knop wordt gedrukt zal deze knop een functie uitvoeren. De knop heeft een *id* die “add” heet. De javascript code die daarbij hoort staat hieronder weergegeven.

var bestand = 0;

var max = 3;

$("#add").click(function(){

bestand++;

if(bestand <= max){

$("tbody").append('<tr class="add\_row"><td>#</td><td><input class="file" name="upfile[]" type="file" required></td><td class="text-center"><button type="button" class="btn button btn-sm" id="delete" title="Verwijder bestand">Verwijder bestand</button></td><tr>');

}

if(bestand == max){

$("#add").css("visibility", "hidden");

}

});

*Figuur 6: Bestanden toevoegen deel code*

Dus als er geklikt wordt op een *“add” id* dan voert hij deze functie uit. Binnen de tbody tag zal de append() functie ervoor zorgen dat er extra HTML-code wordt toegevoegd. Er is ook een check voor de knop, als er meer dan 3 bestanden zijn toegevoegd dan zal de knop verdwijnen.

## PHP Functies

### Rubrieken Functies

|  |  |
| --- | --- |
|  |  |
| createRubriek |  |
| createSubRubriek |  |
| heeftSubriek |  |
| selectSubRubriekNaam |  |
| selectSubRubriekNummer |  |
| selectSubRubriekNummerZonderPlek |  |
| selectAantalSubRubrieken |  |
| createVoorwerpInRubriekItem | Maakt producten aan in rubriekenpagina. |
| selectAantalVoorwerpen | Telt aantal voorwerpen per rubriek. |
| heeftParentRubriek | Telt aantal sub-rubrieken onder een hoofd-rubriek |
| selectParentRubriekNaam |  |
| selectParentRubriekNummer |  |
| selectRubriekNaam |  |
| createProductRubrieken |  |

Tabel 1: Rubriek Functies

### Veiling Functies

|  |  |
| --- | --- |
|  |  |
| selectAantalVeilingen | Telt aantal veilingen van één gebruiker. |
| createZoekvoorwerpen | Toont producten op zoekresultaten pagina. |
| timeDiff | Berekent tijdsverschil tussen start en sluit tijd van een veiling. |
| createBiedingen | Laat de geplaatste veilingen zien in bied geschiedenis. |
| createProfVeilingen | Laat veilingen van verkoper zien op de profielpagina. |
| selectAantalBiedingen | Telt hoe vaak een gebruiker heeft geboden. |
| createProfBiedingen | Toont producten waarop een gebruiker heeft geboden. |

Tabel 2: Veiling Functies

### Producten Functies

|  |  |
| --- | --- |
|  |  |
| createHomepageItem | Maakt de homepagina producten aan. |
| createHomepageCard |  |
| createFotos | Maakt kleine plaatjes onder grote foto, op de product pagina. |

Tabel 3: Product Functies

### Registratie Functies

|  |  |
| --- | --- |
|  |  |
| createRandomCode | Genereert een random code die via de mail word verstuurd, om het registeren te bevestigen. |
| createQuestions | Haalt de veiligheidsvragen uit de database |

Tabel 4: Registratie Functies

### Beheerder Functies

|  |  |
| --- | --- |
|  |  |
| createGebruikers | Toont alle gebruikers op de beheerderspagina, en ze kunnen geblokkeerd worden. |
| selectAantalGebruikers | Telt aantal gebruikers, van de website. |
|  |  |
|  |  |

Tabel 5: Beheerder Functies

# Bijlage

## Database wijzigingen

In het tekst bestand waar hieronder naar wordt verwezen staan alle wijzigingen die wij hebben gemaakt op de database. Denk hierbij aan het veranderen van datatypes en namen van de kolommen en tabellen. Zie bijgestuurd document, [Groep14\_AanpassingenDatabase.txt](../../Database/Groep14_AanpassingenDatabase.txt)