

**Keuzeminor: Verdieping Programmeren**

**Periode 4: My First Agenda**

Leerjaar: 1

Crebonummer: 25604

Sector: Software developer­

# Introductie

In de vorige periode heb je kennisgemaakt met de minors die in de opleiding Software development worden aangeboden, namelijk:

* Verdieping programmeren
* Digital Signage
* IOT
* XR

Jij hebt een keuze gemaakt voor Verdieping programmeren.

In deze periode gaan jullie individueel aan de slag en maken een agenda-applicatie. In deze applicatie moet je afspraken en taken kunnen toevoegen, maar ook aanpassen en verwijderen. En natuurlijk moet je de afspraken overzichtelijk op het scherm zien.

Een applicatie die bovenstaande taken kan uitvoeren, noemen we een CRUD-applicatie (Create, Read, Update, Delete). De volledige applicatie wordt gemaakt in PHP.

De basisuitleg voor deze module is terug te vinden op LinkedIn:

[**https://www.linkedin.com/learning/php-with-mysql-essential-training-1-the-basics**](https://www.linkedin.com/learning/php-with-mysql-essential-training-1-the-basics)

Er worden hier nog andere tutorials genoemd die ook uitleg geven over de lesstof.

En natuurlijk kan je ook zelf op zoek gaan naar uitleg over de diverse onderdelen.

Aan het einde van deze module heb je voor jezelf een eenvoudige, overzichtelijke agenda-applicatie gemaakt.

# Tijdsplanning

|  |  |  |
| --- | --- | --- |
| **Les 1** |  |  |
| De Database | 2 uur | OPDRACHT 1.1: Database maken  OPDRACHT 1.2: Tabel maken en vullen  OPDRACHT 1.3: Queries schrijven |
| **Les 2** |  |  |
| PHP en de database | 2 uur | OPDRACHT 2.1a: Het connection script  OPDRACHT 2.1b: Errors tonen via het connection script  OPDRACHT 2.2: De uitleespagina |
| **Les 3** |  |  |
| Overzicht en detail | 2uur | OPDRACHT 3.1: De uitleespagina verbeteren  OPDRACHT 3.2: Detailpagina maken |
| **Les 4** |  |  |
| Gegevens toevoegen | 2 uur | OPDRACHT 4.1: Een statische toevoegpagina maken  OPDRACHT 4.2: Het toevoegformulier  OPDRACHT 4.3: De verwerkpagina |
| **Les 5** |  |  |
| Gegevens verwijderen | 2 uur | OPDRACHT 5.1: Link maken naar verwijderpagina  OPDRACHT 5.2: De verwijderpagina maken  OPDRACHT 5.3: De verwijder-verwerkpagina maken  OPDRACHT 5.4: Extra aanpassingen |
| **Les 6** |  |  |
| Gegevens aanpassen | 2 uur | OPDRACHT 6.1: Link naar aanpaspagina  OPDRACHT 6.2: Gegevens tonen op aanpaspagina  OPDRACHT 6.3: Gegevens tonen in een formulier  OPDRACHT 6.4: Aanpasformulier verwerken |
| **Les 7** |  |  |
| Verdieping | 2 uur |  |

# LES 1 – DE database

In deze module gaan jullie gebruik maken van gegevens die opgeslagen worden in een **database**.

## **Wat is een database?**

Een database is een verzameling van gegevens die digitaal en gestructureerd worden bewaard:   
in **tabellen**.

Wat een database is, kan je lezen op de volgende website:

* <https://informaticalessen.be/gegevensbeheer/wat-is-een-database/>

**Oefening 1\_A:**

Lees de informatie op bovenstaande website door.  
Maak de bijbehorende puzzel:

* <https://informaticalessen.be/wp-content/oefeningen/gegevensbeheer/kruis.htm>

Zoek zelf nog een andere tutorial waar je kan leren wat een database is en hoe die is opgebouwd.

## **SQL**

Om gegevens in een database te kunnen benaderen (toevoegen, uitlezen, aanpassen of verwijderen), gebruiken we een speciale taal: SQL (Structured Query Language). Met deze taal kan je via queries (vragen/opdrachten) de database benaderen.

Deze taal heeft verschillende ‘dialecten’, oftewel versies. Zoals PostgreSQL, SQLite en MySQL. En dan heb je ook nog NoSQL-databases….

PHP gebruikt MySQL om databasegegevens te benaderen.

Neem de volgende tutorials door om meer te leren over SQL en MySQL:

* <https://icttrainingen.nl/blog/wat-is-sql>
* <https://programmeerplaats.nl/wat-is-mysql/>

Een database die via MySQL te benaderen is, noemen we een MySQL-database.

Jullie hebben de mogelijkheid om via PLESK een MySQL-database te maken. Daarvoor gebruiken we een database management system (DBMS): PHPMyAdmin.

## **PHPMyAdmin**

Om gebruik te kunnen maken van een database moeten we er eerst eentje maken.

Het maken van een database doe je in een speciale (online) omgeving of programma.

Wij gebruiken PHPMyAdmin:

* <https://www.phpmyadmin.net/>
* <https://www.phpmyadmin.nl/wat-is-phpmyadmin/wat-is-phpmyadmin>

Op LinkedIn is een volledige PHPMyAdmin tutorial te vinden:

* <https://www.linkedin.com/learning/learning-phpmyadmin/welcome?u=36995692>

Neem minimaal de volgende lessen door om meer over PHPMyAdmin te leren:

* Introduction – Welcome
* Hoofdstuk 1: What phpMyAdmin Does

Via PLESK (<https://web02.ict-lab.nl:8443/>) kan je naar jouw eigen PHPMyAdmin navigeren en zelf een database maken.

Omdat PLESK haar eigen manier voor het maken van een database heeft, kan je hiervoor gebruik maken van een stappenplan.

# OPdracht 1.1 – Een database maken

Open het word document: “PHPMyAdmin stappenplan.docx”.

Lees het hoofdstuk: “PLESK: een database maken”.

* Maak een database op PLESK – PHPMyAdmin met de volgende gegevens:
  + Database name: **54321\_database**
  + Database username: **db54321**
  + New password: **xxxxxxxxxx** (Kies deze zelf. Tip: PLESK-wachtwoord)
* Ga nu naar <https://pma.ict-lab.nl/>

Log in met jouw database-user en wachtwoord.

Als het goed is, opent PHPMyAdmin met jouw database in het linker venster.

## **Tabellen**

Zoals je al hebt kunnen lezen op <https://informaticalessen.be/gegevensbeheer/wat-is-een-database/> bestaat een tabel uit records (of rijen) en kolommen (of velden).

Als je in PHPMyAdmin een nieuwe tabel maakt, dan moet je van tevoren weten uit hoeveel en welke kolommen de tabel bestaat.

Het maken van een tabel in PHPMyAdmin kan je op één van de volgende manieren leren:

* Tutorial:  
  <https://www.homeandlearn.co.uk/php/php12p2.html>
* LinkedIn:

<https://www.linkedin.com/learning/learning-phpmyadmin/choosing-the-right-column-types?u=36995692> en de rest van hoofdstuk 3: ‘Creating Tables

* ‘PHPMyAdmin stappenplan.docx’: Hoofdstuk ‘PHPMyAdmin’.
* Of zoek zelf een goede tutorial over het gebruik van PHPMyAdmin.

# OPdracht 1.2 – Een tabel maken en vullen

Maak de tabel **verprog1\_agenda** met daarin de volgende 7 kolommen:

|  |  |  |  |
| --- | --- | --- | --- |
| **Naam** | **Type** |  | **Opmerking:** |
| **ID** | INT | Index: PRIMARY, Auto Increment (A\_I) | De key, auto nummering |
| **Onderwerp** | VARCHAR(30) |  | Titel |
| **Inhoud** | TEXT |  | Wat moet er allemaal gebeuren? |
| **Begindatum** | DATE |  | Standaard de dag van vandaag |
| **Einddatum** | DATE |  | Alleen datum, geen tijd |
| **Prioriteit** | TINYINT |  | van 1 tot 5 |
| **Status** | ENUM | 'n', 'b', 'a' | 'niet begonnen', 'bezig', 'afgerond' |

LET OP: bij VARCHAR moet je aangeven hoeveel karakters dit veld maximaal mag hebben. Dit kan maximaal 255 zijn. Vul onder ‘Length/Values’ dit getal in.

Voeg 5 taken toe die jij nog moet doen. Dat kan huiswerk zijn, maar ook andere taken, zoals een tandartsafspraak.

## **Queries**

Op LinkedIn wordt de basis van MySQL en queries uitgelegd:

<https://www.linkedin.com/learning/php-with-mysql-essential-training-1-the-basics/mysql-introduction?u=36995692>

In een tabel kunnen meerdere acties uitgevoerd worden. We gebruiken een **query** om aan te geven wat er moet gebeuren.

De acties kunnen we verdelen in vier handelingen:

|  |  |  |  |
| --- | --- | --- | --- |
| **Handeling** |  | **Omschrijving** | **Query** |
| Toevoegen | **C**reate | Maken van een tabel  Toevoegen van gegevens aan een tabel | “CREATE TABLE ..”  “INSERT INTO tabelnaam …” |
| Lezen | **R**ead | Lezen van gegevens uit 1 of meerdere tabellen | “SELECT \* FROM tabelnaam…” |
| Aanpassen | **U**pdate | Gegevens in een tabel aanpassen | “UPDATE tabelnaam SET …” |
| Verwijderen | **D**elete | Verwijderen van gegevens uit een tabel | “DELETE FROM tabelnaam…” |

De 4 handelingen zijn dus, in het Engels: **C**reate, **R**ead, **U**pdate en **D**elete.

Afgekort: **CRUD**.

Als we in PHP een applicatie bouwen die al deze handelingen kan uitvoeren, noemen we dit een **CRUD-applicatie**.

Hoe zien die queries er nu precies uit?

Dat gaan we eerst bekijken in PHPMyAdmin.

## **CREATE**

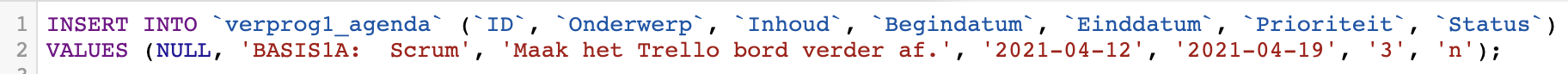
Bij het maken van de tabel ‘verprog1\_agenda’ heeft PHPMyAdmin meerdere queries uitgevoerd. Helaas heb je die niet kunnen zien, maar dat zijn de volgende queries geweest:

Text

Description automatically generated

Deze queries zal je zelf voorlopig nog niet gaan maken, maar het is goed om deze een keer gezien te hebben.

Waarschijnlijk heb je er niet op gelet, maar bij het toevoegen van elk agenda-item is er wél een query getoond. Dat zag er zo uit:



De values zullen bij jullie anders zijn.

**Oefening 1\_B**

Neem de bovenstaande queries door en probeer voor jezelf te begrijpen wat er gebeurt.

## **READ**

A picture containing logo

Description automatically generatedAls je de tabel ‘verdprog1\_agenda’ selecteert en je klikt op het tabblad ‘Browse’, dan verschijnen alle records uit de tabel op het scherm.

Bovenaan het scherm staat een query:

**Oefening 1\_C**

* Klik op het tabblad ‘SQL’.  
  Waarschijnlijk zie je daar bovenstaande query staan, met daarachter ‘WHERE 1’
* Haal alle tekst in het SQL-venster weg en typ in:  
   *SELECT \* FROM verprog1\_agenda.*
* Klik rechts onderin op “Go”  
  Je ziet nu weer de tabel met alle gegevens op het scherm.
* Klik bovenin op “Show query box”
* Vul de query aan, zodat er staat:

*SELECT \* FROM verprog1\_agenda WHERE ID = 2*  
Je ziet nu alleen de gegevens van het record met ID ‘2’.

* Probeer nu zelf alle agenda-items te tonen met prioriteit ‘3’

## **UPDATE**

Het aanpassen van een record in een tabel gaat via een UPDATE-query.

**Oefening 1\_D**

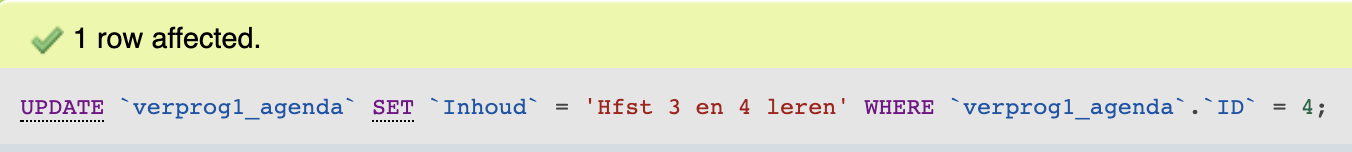
* Zorg dat in PHPMyAdmin de tabel verprog1\_agenda zichtbaar is.
* Klik voor één van de records op ‘Edit’.



* Pas één of meerdere gegevens aan en klik op ‘Go’.

De tabel verschijnt weer met de aangepaste gegevens.

Bovenin het venster zie je de UPDATE-query:



## **DELETE**

Voor het verwijderen van een record wordt de DELETE-query gebruikt.

**Oefening 1\_E**

* Zorg dat in PHPMyAdmin de tabel verprog1\_agenda zichtbaar is.
* Klik voor één van de records op ‘Delete.



* In een pop-up venster wordt een bevestiging gevraagd.  
  Je ziet ook de DELETE-query die uitgevoerd gaat worden.  
  Graphical user interface, text, application

  Description automatically generated  
  Klik op ‘OK’.

De tabel verschijnt weer met de aangepaste gegevens.

## **EXTRA UITLEG**

Bekijk de volgende tutorials voor meer uitleg over SQL-queries:

* <https://www.youtube.com/watch?v=AY-7mVO4jSU>
* <https://gkoetsier.nl/tut_mysql.html>   
  (uitgebreid, maar zoek naar de juiste queries)

**Oefening 1\_F**

De volgende tutorial heeft de mogelijkheid om queries te oefenen op de website:

<https://www.mysqltutorial.org/basic-mysql-tutorial.aspx>

Neem in ieder geval de volgende hoofdstukken door:

* Section 2. Querying data
  + <https://www.mysqltutorial.org/mysql-select-statement-query-data.aspx>
* Section4. Filtering data
  + <https://www.mysqltutorial.org/mysql-where/>
* Section 10. Modifying data in MySQL
  + <https://www.mysqltutorial.org/mysql-where/>
  + <https://www.mysqltutorial.org/mysql-update-data.aspx>
  + <https://www.mysqltutorial.org/mysql-delete-statement.aspx>

Graphical user interface, application

Description automatically generated

Klik op de knop om te oefenen met de queries.

# OPdracht 1.3 – Queries schrijven

LET OP: kopieer de queries die je in deze opdracht maakt naar een word document: ‘12345\_verprog1.docx’.

Open in PHPMyAdmin jouw tabel ‘verprog1\_agenda’.

Open het tabblad “SQL” en maak de volgende queries:

1. Schrijf een query om een item (record) aan jouw agenda toe te voegen.
2. Schrijf nóg een query om een item toe te voegen.
3. Schrijf een query om alle agenda-items te laten zien waarvan de status op ‘bezig’ (‘b’) staat.
4. Schrijf een query om de prioriteit van één van de agenda-items aan te passen.
5. Schrijf een query om een agenda-item (record) te verwijderen.

Voer alle queries uit. Kopieer de queries naar jouw word document.  
Maak een screenshot van jouw uiteindelijke tabel met daarin alle gegevens en zet deze ook in je word document.

# LES 2 – PHP en dE database

Nu gaan we de eerste stappen zetten om een CRUD-applicatie te bouwen.

Het is uiteindelijk de bedoeling dat we alle handelingen om de tabel aan te passen via PHP gaan uitvoeren.

De theorie uit dit hoofdstuk, en de volgende hoofdstukken, is ook terug te vinden in de volgende tutorials:

* <https://www.linkedin.com/learning/php-with-mysql-essential-training-1-the-basics>  
  (Hoofdstuk 5, 6 en 7)
* <https://www.w3schools.com/php/php_mysql_intro.asp>
* <https://www.tutorialrepublic.com/php-tutorial/php-mysql-introduction.php>

## **Connection script**

Het eerste dat we nodig hebben is een verbinding vanuit PHP naar de database.

Neem de volgende tutorial door:

<https://www.linkedin.com/learning/php-with-mysql-essential-training-1-the-basics/connect-to-mysql-with-php>

Je ziet dat er een stukje PHP-code nodig is om een verbinding te maken met de database.

Daarin worden 4 eigenschappen voor de verbinding vastgelegd:

* $db\_hostname = De naam van de server waar de database staat.
* $db\_username = De naam van de eigenaar van de database (jouw gebruikersnaam).
* $db\_password = Het wachtwoord van de eigenaar.
* $db\_database = De naam van de database.

Daarna wordt er een mysqli-functie uitgevoerd om de verbinding te maken.

Deze verbinding wordt ‘bewaard’ in de variabele $mysqli. (Dit is eigenlijk een *object*.)

# OPdracht 2.1a – Het connection script

Allereerst:

1. Ga naar PLESK: <https://web02.ict-lab.nl:8443/>.
2. Maak in de rootmap (‘Files -> httpdocs’) de map **verprog1** aan.

Daarna:

1. Maak de PHP-pagina ‘config.php’ aan.
2. Zet daarin onderstaande code.  
   **LET OP**: vul **jouw** gegevens in!
3. Sla de pagina op en upload deze naar de map verprog1.

Graphical user interface, text, application, email

Description automatically generated

Als laatste:

1. Test de pagina door deze te openen in de browser.
   1. Als het goed is, krijg je de melding dat de verbinding is gemaakt.

**LET OP**: als het connection script goed werkt, dan **verwijder** je de test!

Er hoeft niet elke keer een melding op het scherm te verschijnen dat de verbinding is gemaakt.

## **Connection script: errors tonen**

De webserver van school is zo ingesteld dat PHP-fouten niet worden getoond. Dit is gedaan vanwege de veiligheid: als een gebruiker fouten in de code kan lezen, dan zou hij/zij hier misbruik van kunnen maken.

Als developer is het juist fijn als die fouten wél getoond worden. Gelukkig kan dat door een aantal regels code toe te voegen; dit doen we in het bestand ‘config.php’. Omdat dit bestand in (bijna) alle PHP-pagina’s wordt bijgevoegd (included) kunnen we zo op al deze pagina’s de fouten tonen.

**LET OP**: als een webapplicatie klaar is en opgeleverd wordt aan de klant, dan moet je deze regels (voor de veiligheid) weer verwijderen uit de code!

Er zijn meerder opties om foutmeldingen te tonen. Zoek maar eens op Internet naar “PHP error reporting”. Wij gebruiken de volgende regels:

*ini\_set('display\_errors', 1);  
error\_reporting(E\_ALL);*

# OPdracht 2.1B – Errors tonen via het connection script

1. Open het bestand ‘config.php’
2. Text

   Description automatically generatedZet bovenstaande regels helemaal bovenaan in de pagina:

## **Gegevens uitlezen**

Neem de volgende tutorials door:

* <https://www.linkedin.com/learning/php-with-mysql-essential-training-1-the-basics/retrieve-data>  
  & <https://www.linkedin.com/learning/php-with-mysql-essential-training-1-the-basics/work-with-retrieved-data>
* <https://www.w3schools.com/php/php_mysql_select.asp>   
  (Example (MySQLi Procedural)

Zoals je hebt kunnen zien in de tutorial bestaat het gebruik van queries in PHP uit een aantal stappen:

1. Maak verbinding met de database.
2. Maak een query en voer die uit.
3. Doe iets met het resultaat van de query.
4. Sluit de verbinding.

De query die gebruikt wordt om gegevens uit een tabel te lezen, “SELECT …”, heb je ook al in hoofdstuk 1 gebruikt.

Bij het uitlezen van gegevens uit een tabel worden de volgende mysqli-queries gebruikt:

* **mysqli\_query()** : voert de query uit op de database; dit levert een resultaat op.  
  <https://www.w3schools.com/php/func_mysqli_query.asp>
* **mysqli\_fetch\_assoc()** : leest één rij uit van het resultaat en zet de gegevens in een array.   
  <https://www.w3schools.com/php/func_mysqli_fetch_assoc.asp>Vaak wordt een andere functie gebruikt: mysqli\_fetch\_array(). Deze geeft hetzelfde resultaat.
* **mysqli\_num\_rows()** : telt het aantal records in het resultaat.

# OPdracht 2.2 – De uitleespagina

In deze opdracht ga je de gegevens van de tabel **verprog1\_agenda** uitlezen en deze tonen in een PHP-pagina.

1. Maak een nieuwe PHP-pagina aan: **toonagenda.php**.
2. Graphical user interface, text

   Description automatically generatedNeem onderstaande code over in ‘toonagenda.php’.  
   Sla de pagina op en upload deze in de map ‘verprog1’ op PLESK.
3. Test de pagina.  
   Als het goed is, zie je van alle agenda-items het onderwerp en de inhoud op het scherm.

Als je een fout maakt in de query dan verschijnt er niks op het scherm. Er zal ook geen foutmelding verschijnen, ondanks de error-reporting in jouw config.php.  
Daarom voegen we een extra stukje code toe om de SQL-fouten te tonen:

1. Graphical user interface, text

   Description automatically generatedVoeg onderstaande code toe aan jouw pagina ‘toonagenda.php’  
   Doe dit op de juiste plaats.

**LET OP**: Ook hiervoor geldt: deze code is eigenlijk alleen voor de developer; een bezoeker zou een getoonde foutmelding kunnen misbruiken.

# LES 3 – Overzichts- en detailpagina

Tijdens de vorige les heb je een overzichtspagina gemaakt.

De gegevens worden echter nog niet netjes op het scherm getoond.

# OPdracht 3.1: De overzichtspagina verbeteren

Zorg ervoor dat alle gegevens overzichtelijk op het scherm getoond worden. Dit kan met bijvoorbeeld divjes (en een css grid) of een tabel.

1. Open ‘toonagenda.php’.
2. Graphical user interface, text, application

   Description automatically generatedVerbeter de weergave van de gegevens.  
   Je mag zelf een structuur bedenken. Hieronder vind je een voorbeeld:
3. Sla de pagina op, upload deze naar jouw webmap ‘verprog1’ en test de pagina.

## **Detailpagina**

Niet alle gegevens van een agenda-item zijn zichtbaar. De rest van de gegevens zullen worden getoond op een detailpagina.

Neem nu eerst de volgende tutorials door:

* <http://html.net/tutorials/php/lesson10.php> (“Passing variables in a URL”)
* <https://www.linkedin.com/learning/php-with-mysql-essential-training-1-the-basics/find-a-single-record?u=36995692> (“Find a single record”)

Om de juiste gegevens van elk agenda-item op de detailpagina te tonen, moet het volgende gebeuren:

* Op de overzichtspagina krijgt elk item een link naar de detailpagina.
* In de link wordt de ID van het item meegestuurd.  
  (Deze staat in de key-kolom van de tabel).
* Op de detailpagina wordt de ID uit de url gehaald met **$\_GET[..]**.
* Met de juiste query worden de details van het agenda-item uit de database gehaald.
* De gegevens worden getoond.

# OPdracht 3.2: De detailpagina maken

In deze opdracht ga je de detailpagina maken en een link maken op de overzichtspagina.

1. Open 'toonagenda.php’
2. Voeg voor elk item de volgende link toe:
3. Maak de pagina ‘**detail.php**’
4. Graphical user interface, text, application

   Description automatically generatedZet de volgende code in deze pagina:
5. Sla de pagina op
6. Logo

   Description automatically generated with low confidenceUpload beide pagina’s naar de webserver en test de link van een item uit de overzichtspagina.  
   Als het goed is, verschijnt de detailpagina met de volgende tekst:  
   En in de adresbalk van de browser staat:  
   

Het maakt nu niet uit op welk item je klikt; altijd verschijnt het getal 2.  
Nu gaan we ervoor zorgen dat de juiste ID wordt meegestuurd.

1. Pas de detail-link op de overzichtspagina aan. De ‘2’ wordt vervangen door de ID van het item. Let goed op de aanhalingstekens!!
2. Test opnieuw en controleer of de juiste ID wordt getoond op de detailpagina.

Op de detailpagina moet nu een query geschreven worden, die de gegevens van alleen dit agenda-item toont.

1. Text

   Description automatically generatedPas de detailpagina aan en zorg dat de bijgaande code erin komt te staan:
2. Upload en test de pagina’s.
3. Verbeter de opmaak van de detailpagina.

# LES 4 – Gegevens toevoegen

In het vorige hoofdstuk zijn we begonnen met het Read-gedeelte uit de CRUD-applicatie.

In dit hoofdstuk gaan we het Create-gedeelte bouwen.

Het is mogelijk om vanuit jouw PHP-pagina volledige databasetabellen te maken, maar dat doen we nu niet. We gaan alleen een bestaande tabel vullen met gegevens.

Of beter nog: we laten een gebruiker de gegevens invoeren. Hiervoor gebruiken we een formulier. Het formulier wordt verwerkt in PHP door een aparte verwerkpagina.

De INSERT-queries die in dit hoofdstuk gebruikt worden, zijn al behandeld in Les 1.

Neem nu eerst de volgende tutorials door:

* <https://www.linkedin.com/learning/php-with-mysql-essential-training-1-the-basics/use-form-data-to-create-records?u=36995692>
* <https://www.tutorialrepublic.com/php-tutorial/php-mysql-insert-query.php>

## **Toevoegpagina maken met PHP**

De query die gebruikt wordt om een record met gegevens toe te voegen aan een tabel heeft de volgende structuur:

**INSERT INTO** *tabel* **(***kolom1***,** *kolom2***, …) VALUES (***waarde1***,** *waarde2***, …)**

Het spreekt voor zich dat waarde1 in kolom1 terecht komt, waarde2 in kolom 2, enzovoort.

LET OP: elke waarde **moet** tussen aanhalingstekens staan als de kolom een tekst of datum bevat (VARCHAR, TEXT, DATE, ENUM, …).

Bij INT (of DOUBLE) hoeven er géén aanhalingstekens om de waarde gezet te worden.

# OPdracht 4.1: Een statische toevoegpagina maken

Stap 1 is: maak eerst een toevoegpagina zonder formulier. We bepalen zelf welke gegevens we willen toevoegen aan de database en zetten die rechtstreeks in de PHP-code.

1. Maak een nieuwe PHP-pagina: **toevoeg.php**.
2. Zet daarin de volgende code (met comments!):  
   Bepaal zelf de waarden (‘VALUES’) die je in de tabel wilt zette.n

**Graphical user interface, text, application, email

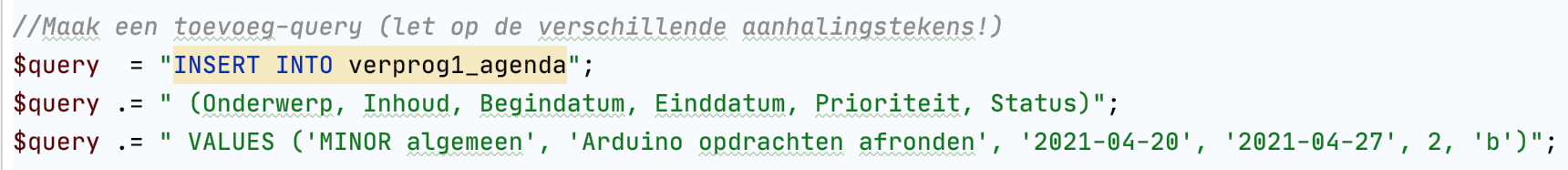
Description automatically generated**

1. Sla de pagina op en test deze op de server.
2. Controleer of het record is toegevoegd.

Opmerkingen:

* De ID wordt niet genoemd.  
  Dat hoeft ook niet, want die is ingesteld op autonummering (auto increment).
* Zoals je ziet, staan alle VALUES tussen aanhalingstekens.   
  Behalve Prioriteit, want dat is in de tabel een INT.
* Het is een toevoegquery, dus je leest geen gegevens uit.  
  Toch krijg je een resultaat: *$result*.   
  Dit resultaat geeft aan of het toevoegen is gelukt (‘true’) of niet (‘false’).  
  In de if()-statement controleer je of *$result* gelijk is aan ‘true’.
* Als het toevoegen niet is gelukt, zorgen we ervoor dat er een foutmelding op het scherm komt: *mysqli\_error($mysqli)*.
* Natuurlijk willen we controleren of het toevoegen gelukt is, vandaar een link terug naar de overzichtspagina.

Je ziet dat de toevoegquery zo lang is dat deze langer is dan één regel.

Om de code overzichtelijk te houden kan je de query in stukken knippen en stuk voor stuk toevoegen aan de query:

LET OP: Op de eerste regel begin je de query: *‘$query* ***=*** *…*’;

Op de volgende regel voeg je iets toe aan de query: *‘$query* ***.=*** *…* ‘; (let op de punt!)

EN: Let op de spaties aan het begin van elke volgende regel.

Anders wordt de code zonder spaties aan elkaar geplakt.

****Je kan de query schrijven zonder de kolomnamen te noemen:

Let dan wel goed op de volgorde van de values; deze moet precies overeenkomen met die in de tabel.

Én: je moet aangeven dat de eerste kolom niet door jou wordt ingevuld, dus daar plaats je een NULL.

## **Toevoegformulier**

Met bovenstaande methode kan de gebruiker nog steeds niet zelf gegevens toevoegen; de applicatie is daardoor nog niet erg dynamisch.

De volgende stap is daarom het maken van een toevoegformulier.

De gegevens die de gebruiker invult worden in de database gezet via een verwerkpagina.

Je moet dus 2 pagina’s maken: toevoegForm.html en toevoegVerwerk.php.

# OPdracht 4.2: Het toevoegformulier

Maak een html-pagina: toevoegForm.html.

Plaats hierin een formulier waarmee de gebruiker de gegevens van een agenda-item kan invullen.

Bedenk zelf wat voor inputvelden je het beste kan gebruiken; geef ze de volgende namen:

* onderwerpVeld
* inhoudVeld
* begindatumVeld
* einddatumVeld
* prioriteitVeld
* statusVeld

Het is wel logisch dat je bijvoorbeeld voor de datum een datumveld (‘date’) gebruikt.

Zorg ervoor dat de gebruiker alle velden moet invullen.

Het formulier wordt verstuurd naar **toevoegVerwerk.php**.

LET OP: alle velden **moeten** een naam (‘name=’) hebben. Deze namen worden door PHP gebruikt op de verwerkpagina.

Het formulier zou er bijvoorbeeld zo uit kunnen zien:

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generatedHierbij een stukje van de code van bovenstaand formulier:

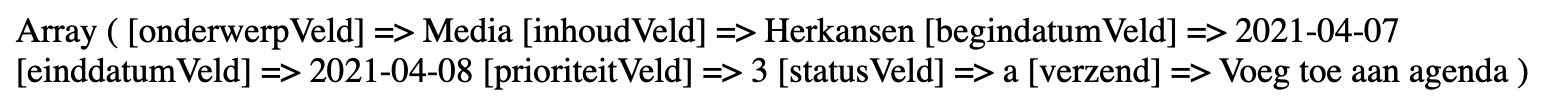
## **Verwerkingspagina**

Het formulier wordt verstuurd naar de verwerkpagina:

Deze verwerkpagina gaan we nu maken. Op de verwerkpagina worden alle formuliervelden uitgelezen. Dat gebeurt met de volgende code: ***$\_POST[‘naamVeld’]***

# OPdracht 4.3: De toevoeg-verwerkpagina

1. Maak een nieuwe PHP-pagina: **toevoegVerwerk.php**.
2. Graphical user interface, text, application, email

   Description automatically generatedZet de volgende code in de pagina:  
   Om te testen of alle velden + inhoud worden verstuurd, gebruiken we de testcode ‘print\_r($\_POST)’. Hiermee wordt in één keer alle inhoud van de $\_POST-array (!) getoond.
3. Sla de pagina op en upload het formulier en de verwerkpagina.
4. Test de verwerkpagina door het formulier in te vullen en op de submitknop te drukken. Als het goed is, opent de verwerkpagina en zie je de namen van de velden en de ingevoerde waarden (“[naamVeld]=>inhoud”). Bijvoorbeeld:

Als dit werkt dan kunnen we de volgende stap maken: alle velden apart uitlezen en een (dynamische) toevoegquery schrijven.

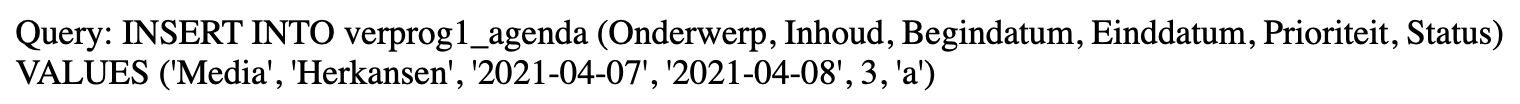
1. Graphical user interface, text, application

   Description automatically generatedHaal de TEST uit het if-statement en zet de volgende code daarvoor in de plaats:

Je ziet dat eerst alle formuliervelden worden uitgelezen. De waarden van de velden worden in variabelen gestopt. Die variabelen (dus: de waarden uit het formulier) worden gebruikt in de query.

Zoals je ziet worden de variabelen op de plaats van de values geplaatst. Omdat deze variabelen in de query-string worden gezet (en niet erbuiten), worden er om elke variabele accolades (‘curly braces’) geplaatst, zodat het voor PHP duidelijk is dat het hier om variabelen gaat.

Je kan de variabelen wel buiten de string zetten en netjes concentreren, maar daar wordt het niet overzichtelijker van:

1. Sla de nieuwe verwerkpagina op en upload deze.
2. Test opnieuw vanaf het toevoegformulier.  
   Als het goed is, krijg je een query te zien. Bijvoorbeeld:

Nu ga je deze query rechtstreeks toevoegen in PHPMyAdmin

1. Kopieer de query (zonder het woord ‘Query:’)
2. Open jouw database in PHPMyAdmin
3. Klik op het SQL-tabblad
4. Plak jouw query in het SQL-venster en klik op “Go”
5. Controleer of jouw query is toegevoegd aan de tabel.  
   Zo niet: lees de foutmelding en pas jouw PHP-code aan, zodat je een juiste query krijgt.

De laatste stap is: laat PHP de query uitvoeren.  
De code die daarvoor nodig is, hebben we al een keer gebruikt in opdracht 4.1.

1. Graphical user interface, text, application, chat or text message

   Description automatically generatedPas jouw code aan zodat PHP de query uitvoert.  
   Haal daarvoor de TEST weg (‘echo $query’) en typ daarvoor in de plaats:

Als het toevoegen fout gaat toon je niet alleen de melding maar ook de query. Zo kan je bekijken of je een fout hebt gemaakt in het construeren van de query.

1. Voer een laatste test uit en controleer of de gegevens uit het formulier worden toegevoegd aan de database.

!! Waarschijnlijk werkt de code nog niet omdat je bovenaan de pagina nog 1 belangrijke regel bent vergeten toe te voegen…  
Welke?

# LES 5 – Gegevens verwijderen

We hebben al twee letters van CRUD behandeld: ‘Create’ en ‘Read’.

In deze les behandelen we de ‘D’: Delete. We gaan niet een hele tabel verwijderen, hoewel dit ook kan via PHP en de juiste queries.

In deze les leer je hoe je een record kan verwijderen uit een tabel.

## **Gegevens verwijderen**

Graphical user interface, text, application

Description automatically generatedIn les 1 heb je via PHPMyAdmin records verwijderd. Elke keer als je een record verwijdert, zie je een pop-up-venster met daarin de volgende melding:

In deze melding zie je een query:

**DELETE FROM** verprog1\_agenda **WHERE** ID = 2

De query is iets aangepast:

* De aanhalingstekens zijn weggehaald.  
  PHPMyAdmin zet deze er automatisch omheen, maar in PHP/MySQLi heb je deze niet nodig.
* Achter ‘WHERE’ staat alleen de ID.  
  Je hoeft niet de tabel (‘verprog1’) te noemen, omdat bij ‘FROM’ al staat uit welke tabel je een record wilt verwijderen.

LET OP: er staat geen asterisk (‘\*’) tussen ‘DELETE’ en ‘FROM’. Het hele record wordt verwijderd, dus je hoeft geen kolommen aan te geven, ook niet dat alle kolommen verwijderd moeten worden.

Neem de volgende tutorials door:

* <https://www.linkedin.com/learning/php-with-mysql-essential-training-1-the-basics/delete-a-record?u=36995692>
* <https://www.tutorialrepublic.com/php-tutorial/php-mysql-delete-query.php>

We gaan nu vanuit de overzichtspagina voor elk agenda-item een link maken naar de verwijderpagina. In deze link wordt de ID van het item (= de key van het record) meegestuurd, zodat de verwijderpagina weet welk item verwijderd moet worden.

# OPdracht 5.1: De link maken naar de verwijderpagina

1. Open de overzichtspagina (‘toonagenda.php’) in jouw editor.
2. Zorg dat er bij elk item een link komt te staan naar de verwijderpagina:  
   **verwijder.php**

Table

Description automatically generatedDe overzichtspagina zou er dan zo uit kunnen zien:

LET OP: zorg dat de ID wordt meegestuurd naar de verwijderpagina.  
Je kan hiervoor de code gebruiken van de link naar de detailpagina.

# OPdracht 5.2: De verwijderpagina maken

1. Maak een nieuwe PHP-pagina: ‘verwijder.php’.
2. Graphical user interface, text, application, chat or text message

   Description automatically generatedLees de ID uit de url (net als bij de detailpagina) en toon deze op het scherm:
3. Test of het werkt:   
   Als je op de overzichtspagina een item aanklikt, verschijnt dan op de verwijderpagina het bijbehorende ID?
4. Maak onderaan de pagina een link terug naar de overzichtspagina.

Nu gaan we eerst een extra controle inbouwen. Als iemand per ongeluk op de verwijderlink heeft geklikt, dan mag het item niet direct worden verwijderd.

1. Zet op de verwijderpagina de vraag: “weet je het zeker?”.  
   Als de gebruiker op “JA” klikt, dan moet het item verwijderd worden op een nieuwe pagina: ‘verwijder\_verwerk.php’.   
   LET OP: de ID moet weer worden meegestuurd naar de verwerkpagina.  
     
   Graphical user interface, text

   Description automatically generatedDe code zou er zo uit kunnen zien:

# OPdracht 5.3: De verwijder-verwerkpagina maken

Als de gebruiker op ‘JA’ heeft geklikt, dan mag het agenda-item echt verwijderd worden.

Dit doen we op de verwerkpagina van de verwijderpagina.  
Ook hier lees je eerst de ID uit de url. Daarna maak je een query waarmee je het item met dat ID verwijdert.

1. Maak een nieuwe PHP-pagina: **verwijder\_verwerk.php**.
2. Lees de ID uit de link.
3. Graphical user interface, text, application, chat or text message

   Description automatically generatedAls de ID **niet leeg** is, dan moet de volgende code uitgevoerd worden:  
     
   Je ziet dat deze code heel veel lijkt op de toevoegpagina; alleen de query (en de meldingen) is anders.
4. Maak de pagina verder af:  
   - Zorg dat de query goed wordt uitgevoerd.  
   - Geef een melding als de ID leeg is (niet is gevonden)  
   - Maak een link terug naar de overzichtspagina (toonagenda.php).
5. Upload alle pagina’s en test het geheel.

De basis van het verwijderen werkt nu. Maar er kan hier en daar nog wel iets worden verbeterd.

Daarom een extra opdracht: pas onderstaande verbeteringen toe. Natuurlijk mag je de applicatie verder naar eigen inzicht aanpassen, zolang deze maar goed blijft werken en overzichtelijk blijft.

# OPdracht 5.4: Extra aanpassingen

Verbeter het verwijderen door de volgende onderdelen toe te voegen:

1. Maak een link vanaf de detailpagina naar de verwijderpagina.  
   Zorg er daarbij voor dat het juiste item kan worden verwijderd.
2. Op ‘verwijder.php’ staat alleen maar de ID van een agenda-item; het is niet duidelijk over welk item het gaat.
3. Toon op ‘verwijder.php’ ook het onderwerp en de inhoud van het agenda-item dat verwijderd moet worden.   
   TIP: gebruik de code van de detailpagina.

# LES 6 – Gegevens aanpassen

Het laatste onderdeel van de CRUD-applicatie dat nog behandeld moet worden is de Update, oftewel het aanpassen van gegevens

Ook deze query heb je al kunnen zien als je in PHPMyADdmin een record aanpast:

De Update-query bestaat uit 3 onderdelen:

* Achter ‘UPDATE’ staat de tabel die aangepast moet worden.
* Achter ‘SET’ staan de kolommen en de nieuwe waarden; er kunnen meerdere kolommen in één keer worden aangepast.
* Achter WHERE staat de selectie: welke record(s) moeten worden aangepast? Als er maar één record moet worden aangepast, dan staat hier de sleutel van de tabel (‘ID’).

De query die we in PHP gebruiken ziet er iets anders uit:

*UPDATE verprog1\_agenda SET Inhoud = 'Hoofdstuk 7 afmaken' WHERE  ID = 3;*

De meeste aanhalingstekens zijn weg, behalve (!) bij de tekst die aangepast moet worden.  
En bij WHERE is de tabelnaam verwijderd. Deze hoeft niet genoemd te worden.

## **Gegevens aanpassen**

Neem de volgende tutorials door:

* <https://www.linkedin.com/learning/php-with-mysql-essential-training-1-the-basics/use-form-data-to-update-records?u=36995692>
* <https://www.tutorialrepublic.com/php-tutorial/php-mysql-update-query.php>

Het aanpassen van gegevens in de CRUD-applicatie bestaat uit een aantal stappen:

1. Via een **link** op de overzichtspagina kan de gebruiker een record aanpassen.
2. Alle gegevens van het aan te passen record verschijnen in een **formulier**.  
   De gebruiker bepaalt zelf welke gegevens moeten worden aangepast.
3. Als de gebruiker op de submit-knop klikt, worden alle gegevens op de aanpas-verwerk-pagina **aangepast**.  
   De gebruiker krijgt een melding als het is gelukt (of niet).

Bovenstaande stappen zijn, los van elkaar, ook al gebruikt bij andere onderdelen van de applicatie:

* Een link maken naar een nieuwe pagina is al gedaan bij de detailpagina.
* Een formulier laten verwerken is al gedaan bij de toevoegpagina.

Deze kennis gaan we gebruiken bij het maken van de aanpaspagina.

In onderstaande opdracht worden niet meer alle stappen uitgeschreven. Je kan terugkijken in de reader voor de verschillende stappen.

# OPdracht 6.1: Link naar aanpaspagina

1. Open de overzichtspagina (‘toonagenda.php’) in jouw editor.
2. Zorg dat er bij elk item een link komt te staan naar de aanpaspagina:  
   **pasaan.php**

Graphical user interface, application, table

Description automatically generated with medium confidenceDe overzichtspagina zou er dan zo uit kunnen zien:

LET OP: zorg dat de ID wordt meegestuurd naar de aanpaspagina.

# OPdracht 6.1: Gegevens tonen op aanpaspagina

1. Maak een nieuwe PHP-pagina: **pasaan.php**.
2. Zorg ervoor dat **alle gegevens** van het agenda-item onder elkaar op deze pagina getoond worden.
3. Gebruik hiervoor de code van de detailpagina.

Omdat we verderop een html-formulier gaan toevoegen, plaatsen we de html-tags boven en onder de PHP-code.

1. Graphical user interface, text, application, chat or text message

   Description automatically generatedZet boven en onder de PHP-code de volledige html-tags die nodig zijn.  
   Alle PHP-code staat in de <body> van de html-pagina.  
   LET OP: de pagina blijft opgeslagen als een PHP-pagina (‘pasaan.php’)
2. Test de pagina.

# OPdracht 6.31: Gegevens tonen in een formulier

De volgende stap is het maken van het aanpasformulier. Alle gegevens van het agenda-item moeten in de verschillende velden worden ingevuld, zodat de gebruiker alleen maar aanpassingen hoeft te maken.

Het formulier dat we hebben gemaakt in de toevoegpagina kunnen we hier gebruiken.

1. Kopieer de html-code van het toevoegformulier uit ‘toevoeg.php’.  
   LET OP: alleen het formulier! Dus: van ‘<form>’ t/m ‘</form>’.
2. A picture containing text

   Description automatically generatedZoek in ‘pasaan.php’ de regel waar het item wordt uitgelezen:
3. Sluit daaronder de PHP-code af (‘?>’)
4. Plaats de html-code van het formulier op de volgende regel.
5. Start de PHP-code opnieuw (‘<?php’)

Graphical user interface, text, application

Description automatically generatedDe code moet er nu als volgt uitzien:  
  
Nu gaan we het formulier aanpassen.

1. Pas de gegevens van het formulier aan:
2. Pas de submit-button aan:  
   

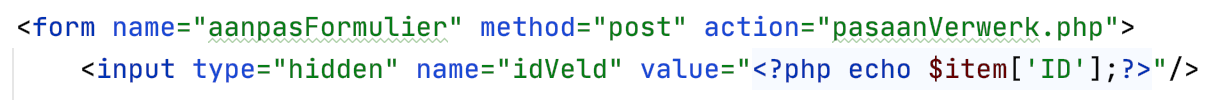
Volgende stap: alle velden vullen met de gegevens van het item.

1. Geef het eerste veld (‘onderwerpVeld’) een value.  
   Plaats in de value de PHP-code waarmee het onderwerp wordt ge-echoot.  
   Let goed op de aanhalingstekens.
2. Doe hetzelfde bij alle andere formuliervelden. Zet met PHP-code de juiste values in de juiste velden.  
     
   LET OP: Als je een text area gebruikt dan moet de PHP-code tussen de begin- en einde-tag van de text area staan:  
     
   EN: als je een pull down of radio buttons hebt gebruikt voor ‘Status’, dan zal je zelf moeten uitzoeken hoe je hierbij de juiste waarde kan selecteren….
3. Table

   Description automatically generatedTest de pagina door op de overzichtspagina op de aanpaslink te klikken.  
   Als het goed is, verschijnt er een formulier op het scherm met daarin de gegevens van het item dat is aangeklikt:

Het enige onderdeel dat nog niet in het formulier staat, is de ID van het agenda-item.  
Deze hebben we wel nodig op de verwerkpagina. De ID moet nog worden toegevoegd.

MAAR: de gebruiker mag de ID niet aanpassen!  
Er zijn verschillende manieren om dat op te lossen. Hier wordt gebruik gemaakt van een ‘hidden field’: een formulierveld dat niet zichtbaar is voor de gebruiker.

1. Voeg bovenaan in het formulier een **hidden field** toe.  
   Noem deze ‘idVeld’.  
   Zet in dit veld de waarde van de ID:
2. Test de aanpaspagina.   
   Kijk in de browser in de code van de pagina: wordt de hidden field gevuld met de ID van het agenda-item?  
     
   Als de items ook nog los onder het formulier worden getoond, dan moeten deze nog worden verwijderd.

# OPdracht 6.1: De aanpas-verwerkpagina maken

Nu het aanpasformulier werkt, kunnen we de verwerkpagina van dit formulier gaan maken.

Deze pagina lijkt heel veel op de verwerkpagina van het toevoegformulier (‘toevoegVerwerk.php’); we kunnen deze code gebruiken op de verwerkpagina van het aanpasformulier.

1. Maak een nieuwe PHP-pagina: **aanpasVerwerk.PHP**.
2. Text

   Description automatically generated with medium confidenceControleer of het formulier is verstuurd en lees alle velden uit:  
     
   Zorg dat ook de ID wordt uitgelezen!

Omdat we niet weten welk veld is aangepast, moeten we **alle waarden** updaten.

1. Text

   Description automatically generated with medium confidenceMaak de update-query:  
     
   Je ziet dat de ID nodig is om te bepalen welk agenda-item aangepast moet worden.  
   De query is hier in 4 stukken geknipt. Je kan ook één lange query maken, maar dat wordt onoverzichtelijk. Let op bij het knippen dat je op de juiste plekken spaties zet!
2. Graphical user interface, text, application

   Description automatically generatedVoer de query uit en controleer of het is gelukt.  
   Geef de gebruiker een melding.
3. Maak een link naar de overzichtspagina.
4. Test het geheel:  
   Klik in de overzichtspagina op een aanpaslink, verander een aantal gegevens en check of de gegevens zijn aangepast.

De CRUD-applicatie is nu af; we hebben de volgende pagina’s gemaakt:

* Create: toevoeg.php en toevoegVerwerk.php
* Read: toonagenda.php
* Update: pasaan.php en pasaanVerwerk.php
* Delete: verwijder.php en verwijderVerwerk.php

De agenda-applicatie is nog niet helemaal af.

In het volgende hoofdstuk gaan we de agenda verbeteren.

# LES 7 – Verdieping

Maak onderstaande opdrachten om jouw applicatie zoveel mogelijk te verbeteren.

Zoek zelf uit welke technieken of queries je nodig hebt om deze verbeteringen toe te passen.

## **OPMAAK**

* Zorg dat alle pagina’s een mooie opmaak krijgen.

Voeg daarvoor aan alle pagina’s html-code toe (zie opdracht 6.2) en maak een link naar een externe CSS-pagina.   
Voeg indien nodig, extra elementen (zoals divs) toe aan jouw pagina’s.

* Misschien kan je zelfs nog JavaScript toevoegen om verschillende meldingen te tonen.

## **OVERZICHTSPAGINA**

* Zorg dat de overzichtspagina meer onderdelen toont.
* Kijk hoe je kan sorteren op datum, zodat het item dat het eerst af moet zijn bovenaan staat. Of sorteer op prioriteit, of status.

Kan je ook de gebruiker zelf laten sorteren?

* Laat alleen de agenda-items zien, waarvan de prioriteit op ‘1’ staat.  
  Of toon alleen de items die nog niet zijn afgerond.

Ook hier: kan je de gebruiker laten kiezen?

* Zorg dat de gebruiker na het toevoegen van een item direct wordt teruggestuurd naar de overzichtspagina. Doe dit ook na het aanpassen en verwijderen, als dit goed is gegaan.

## **TOEVOEGPAGINA**

* Maak van het formulier op de toevoegpagina een volledig html5-formulier.  
  Gebruik ‘required’ voor verplichte velden, zorg dat bij prioriteit alleen getallen tussen 1 en 5 ingevuld kunnen worden, enzovoort.
* Check op de verwerkpagina van het toevoegformulier of elk item wel ingevuld is.  
  Bijvoorbeeld (*if ($ond == “”) { ….}*  
  Geef een foutmelding als er iets leeg is en voeg de gegevens dan NIET toe aan de databasetabel.
* Check in toevoegVerwerk.php of alle gegevens wel van het juiste type zijn.  
  Is prioriteit een getal?  
  Is status wel “n”, “b” of “a”?  
  Zijn begindatum en einddatum wel data?  
  …..

## **EIGEN VERBETERINGEN**

* Bedenk zelf hoe je de applicatie nog verder kan verbeteren.

# BEOORDELINGSFORMULIER

|  |  |  |
| --- | --- | --- |
| Module | |  |
| Naam beoordelaar | |  |
| Naam student | |  |
| Datum | |  |
| Opdracht: | Leerdoelen: | Punten: |
|  |  |  |
|  |  |  |
|  |  |  |
|  | Extra onderdelen |  |
|  | |  |

|  |  |
| --- | --- |
| Opmerkingen: | |
| *Bij te laat inleveren* | *Maximum cijfer: 6* |
| *Niets inleveren* | *Cijfer: 1* |
| Cijfer: |  |

|  |  |
| --- | --- |
| **Werkproces** | **O/V** |
| B1-K1-W3 : Realiseert (onderdelen van) software |  |
| B1-K1-W4 : Test software |  |
| Feedback: | |