Opdrachten PHP/MySql

Inhoud

[PM01 PHPMyAdmin 2](#_Toc476570459)

[PM02 Tabel ontwerpen 3](#_Toc476570460)

[PM03 Modulaire opbouw 4](#_Toc476570461)

[PM04 Verwacht in de bioscoop 5](#_Toc476570462)

[PM05 Reserveren 6](#_Toc476570463)

[PM06 Beveiliging inlogsysteem 7](#_Toc476570464)

[PM07 Inlogsysteem 8](#_Toc476570465)

[PM08 Registreren 11](#_Toc476570466)

[PM09 Menu op basis van gebruikers levels 13](#_Toc476570467)

[PM10 Mijn profiel 14](#_Toc476570468)

[PM11 Film toevoegen 15](#_Toc476570469)

[PM12 Film wijzigen 16](#_Toc476570470)

[PM13 Film verwijderen 17](#_Toc476570471)

[PM14 Film data/tijden 18](#_Toc476570472)

[PM15 Besteloverzicht (deel 1) 20](#_Toc476570473)

[PM16 Besteloverzicht (deel 2) 21](#_Toc476570474)

[PMDELUXE 23](#_Toc476570475)

[Full site 24](#_Toc476570476)

**Versie dd. 6-3-2017**

# PM01 PHPMyAdmin

Deze opdracht wordt nog ingevuld.

# PM02 Tabel ontwerpen

Deze opdracht wordt nog ingevuld.

# PM03 Modulaire opbouw

In de map "modules " vinden we een groot aantal PHP bestanden terug welke allemaal de functionaliteit van een pagina gaan bevatten. In het hoofdbestand index.php dienen deze bestanden op de juiste plek te worden toegevoegd met de functies require of include zodat de website een nette modulaire opbouw krijgt.  
  
**TIP:** lees **eerst** het hoofdstuk Require & Include uit het aanvullende PHP dictaat.  
  
**Werking programma**

Om de modulaire website werkend te maken zullen we de volgende stappen moeten uitvoeren:

1. bestand menu.php: voeg bij de menu items een link toe naar index.php en geef daaraan een GET variabele "PaginaNr " mee met als waarde een uniek nummer voor iedere pagina.
2. Bestand index.php: Lees bovenaan het script de variabele PaginaNr in middels de GET methode.
3. Voeg vervolgens op de plek waar de content dient te komen een switch-statement toe welke aan de hand van het ingelezen PaginaNr de juiste module toevoegt aan de website.

**Inleveren:**

- Screenshot(s) van de werking van je programma.

- Gemaakte code als tekst.

- Lever dit in als Word bestand (.doc of .docx).

# PM04 Verwacht in de bioscoop

Op de pagina "verwacht " dienen alle films te worden weergegeven die binnenkort in de bioscoop worden vertoond.

**Werking programma**

 Om dit te bewerkstelligen zullen we de volgende stappen moeten uitvoeren:

1. Maak een nieuwe database connectie gebruikmakend van PDO in het bestand Functies.php (zie ook voorbeeld in het aanvullend dictaat)
2. Roep in het bestand index.php de functie ConnectDB()  aan en sla de return value op in de variabele $pdo
3. Ga verder in het bestand verwacht.php. Maak een nieuwe SQL-query aan middels het prepared statement principe. In de tabel "Films " zit bij iedere film de status "inBios " of "verwacht ". Maak gebruik van deze status in je query om er voor te zorgen dat je alleen de films krijgt welke nog niet in de bioscoop draaien. Lees ook het aanvullend dictaat voor meer informatie.
4. Wanneer je een resultaat uit de database hebt verkregen dien je deze gegevens op het scherm weer te geven door de gegevens te converteren naar een leesbaar formaat ($sth->fetch()). Je zult zelf moeten beslissen hoe je dit wil weergeven, bijvoorbeeld in een tabel of div, horizontaal of verticaal, et cetera. Bedenk een manier die ook werkt wanneer er meer films aan de database worden toegevoegd.

**Inleveren:**

- Screenshot(s) van de werking van je programma.

- Gemaakte code als tekst.

- Lever dit in als Word bestand (.doc of .docx).

# PM05 Reserveren

Op de pagina "reserveren " dient de bezoeker een overzicht te krijgen van alle films die op dit moment in de bioscoop draaien. Wanneer hij of zij deze film wil reserveren dan kan er op een reserveren knopje worden geduwd waarna je naar het reserveringssysteem wordt doorverwezen.

**Werking programma**

In deze opdracht beperken we ons tot het maken van het overzicht van alle films met daarbij een reserveren knop. Het reserveringssysteem zelf maken we pas later in een van de andere opdrachten.

Om alle films weer te geven die je in eerste instantie vrijwel hetzelfde te doen als in de vorige opdracht, namelijk:

1. In het bestand Reserveren.php: Een SQL-query bouwen middels de prepared statement methode. Let op dat je alleen de films uitleest die op het moment in de bioscoop draaien.
2. Het verkregen resultaat omzetten in een bruikbaar formaat en dit vervolgens op de pagina weergeven.Bij het weergeven van de films op de website moeten we ervoor zorgen dat iedere film "gekoppeld " is aan een reserveren knopje. Je kunt dit doen door iedere film in een FORM te zetten of door bij iedere film een LINK te gebruiken met daarin een aantal GET variabelen. Overweeg goed de voor en nadelen van beide opties en lees ook nog eens het hoofdstuk POST & GET uit het aanvullend PHP dictaat.

**LET OP:** Het reserveren knopje moet doorlinken naar het bestand Data.Tijden.php. Data.Tijden.php maken we in een toekomstige opdracht verder af.

**Inleveren:**

- Screenshot(s) van de werking van je programma.

- Gemaakte code als tekst.

- Lever dit in als Word bestand (.doc of .docx).

# PM06 Beveiliging inlogsysteem

In deze opdracht is het de bedoeling dat je op internet en bij de docent zoveel mogelijk informatie vergaart over het maken van een veilig inlogsysteem. Hoe kun je er bijvoorbeeld voor zorgen dat de wachtwoorden van een gebruiker veilig worden opgeslagen in de database? Hiervoor zijn verschillende encryptie methoden beschikbaar. Beveiliging is echter altijd onderhevig aan veranderingen doordat naarmate de tijd verstrijkt bepaalde methoden onveilig kunnen raken.  
  
**Opdracht**

Maak een PowerPoint of een Prezi presentatie waarin jij vertelt hoe je een veilig inlogsysteem kan maken en welke encryptie methode je daarbij gaat gebruiken. Een aantal van jullie zal deze presentatie ook daadwerkelijk voor de klas geven.  
  
**Bespreken:**

- Powerpoint of Prezi met de vakdocent bespreken.

# PM07 Inlogsysteem

In deze opdracht gaan we het inlogsysteem van de website bouwen. Wanneer er straks klanten een film willen reserveren dienen ze in te loggen met hun account. Wanneer een klant ingelogd is kan hij tevens zijn of haar gegevens zien en wijzigen op de "Mijn Profiel " pagina en kan hij/zij zijn huidige bestelling terugvinden onder de knop "Besteloverzicht ". Deze pagina's bouwen we in een van de volgende opdrachten. Daarnaast wordt het inlogsysteem gebruikt door de Admin, zodat hij gegevens op de website kan toevoegen, aanpassen en verwijderen. Ook dit gaan we maken in een van de latere opdrachten.

In de vorige opdracht hebben jullie onderzocht welke methoden er zijn om een veilig inlogsysteem te bouwen. Bij Cinema7 is er gekozen voor de encryptie methode sha512 in combinatie met een random salt welke ook gecodeerd in de database wordt opgeslagen. Kijk maar eens in de database na de tabel **Klanten** en je zult zien dat er zowel een gecodeerd paswoord veld is alsmede een gecodeerd salt veld. Het is van belang voor het vervolg van alle opdrachten dat je conform deze methode het inlogsysteem bouwt. Wanneer je dit niet doet, dan krijg je problemen in de vervolgopdrachten. Overigens ben je vrij om het inlogsysteem aan te passen, echter dien je dan ook in de volgende opdrachten alles om te schrijven naar je eigen beveiligingssysteem.

**Werking programma**

Tijdens deze opdracht zal het je opvallen dat een groot gedeelte van de code al is gerealiseerd. Dit is gedaan om het niet onnodig gecompliceerd voor jullie te maken. Om het inlogsysteem zo dynamisch mogelijk te maken is er gekozen voor een allesomvattende functie "Inloggen " in het bestand inloggen.php. Op deze manier kun je op iedere gewenste plek in je website een inlogsysteem plaatsen, door het aanroepen van die functie "Inloggen ". In het bestand inloggen.php staat al het een en ander aan uitleg en een stukje code welke je dient te gebruiken. Neem dit bestand dus voordat je de opdracht begint even goed door.

**LET OP:** In de map "Forms” vind je alle in html gecodeerde formulieren voor de website terug. Dit is bewust gedaan ten behoeve van onderhoud en overzichtelijkheid. Je kunt deze formulieren namelijk op iedere gewenste plaats in je code aanroepen zodat er bij een wijziging in het formulier maar op één punt iets hoeft te worden aangepast en dit direct op alle pagina's die het formulier aanroepen is gewijzigd.

Om het inlogsysteem compleet te maken zullen we de volgende stappen nog moeten uitvoeren:

1. Maak in het bestand "InloggenForm.php" een formulier (POST method) met 2 velden (gebruikersnaam en wachtwoord) en 1 knop (Inloggen). Maar daarnaast onderaan het formulier een link naar de registratiepagina met de volgende tekst: " Heeft u nog geen Account? Registreer dan hier", waarbij het woord "hier " de link is waarop geklikt kan worden.
2. Vanaf nu werken we verder met het bestand: "Inloggen.php "uit de map modules.  
   Wanneer de gebruiker zijn of haar gegevens invult en op de knop "Inloggen " drukt, dan dien je die gegevens middels de superglobal $\_POST uit te lezen en mee te geven aan de functie "login ". Deze functie kent als invoer 3 parameters, namelijk gebruikersnaam, wachtwoord en PDO. Met PDO bedoelt men de aangemaakte database connectie. De andere twee spreken voor zich.
3. Plaats nu om de aanroep van de functie een IF-statement zodat je kunt controleren of de gebruiker succesvol is ingelogd of dat een van de twee gegevens niet correct waren. Dit komt omdat de functie "login "als output true of false teruggeeft. Wanneer je false terugkrijgt, dan dient de gebruiker de volgende melding te krijgen: " De Inlognaam of het paswoord is onjuist.". Daarnaast roep je het formulier "InloggenForm.php " opnieuw aan zodat de gebruiker het nog eens kan proberen. Wanneer je true terugkrijgt, dan krijgt de gebruiker de volgende melding: " U bent succesvol ingelogd " en wordt deze doorgestuurd naar de homepage. Hiervoor kun je de functie " RedirectNaarPagina” gebruiken welke meegeleverd is in het bestand "Functies.php”.
4. Om bovenstaande ook daadwerkelijk te laten werken zullen we de functie "inloggen "moeten aanvullen zodat deze ook daadwerkelijk iets gaat controleren. De doen we op de volgende manier:  
   Vraag de gegevens van de gebruiker op uit de database tabel "Klanten " middels een prepared statement. Gebruik hiervoor de meegegeven parameters bovenaan de functie.
5. In de code vind je de volgende regels terug:  
   - if ($sth->rowCount() == 1)  
   - $password = hash('sha512', $password . $row['Salt']);  
   - $user\_browser = $\_SERVER['HTTP\_USER\_AGENT'];  
   - if ($row['Paswoord'] == $password)  
   Zet middels commentaar in je eigen woorden achter de regels wat de code precies doet.
6. Het enige wat nu nog rest is het aanvullen van de SESSION gegevens. Dit voorkomt dat we continu opnieuw de gegevens uit de database moeten halen.

**LET OP:** om te testen of je inlogsysteem werkt kun je proberen in te loggen met een van de volgende drie gebruikers welke al in de database aanwezig zijn.

Gebruikersnaam: Admin

Paswoord: cinema7admin

Gebruikersnaam: Ilse

Paswoord: cinema7Ilse

Gebruikersnaam: Nicolette1983

Paswoord: customer1

**Inleveren:**

- Screenshot(s) van de werking van je programma.

- Gemaakte code als tekst.

- Lever dit in als Word bestand (.doc of .docx).

# PM08 Registreren

In deze opdracht gaan we een registratiepagina maken waardoor een bezoeker van de website zich kan registreren om vervolgens klant te worden en te kunnen inloggen.

**Werking programma**

1. In het bestand "RegistrerenForm.php " vind je een compleet html formulier welke je voor deze opdracht dient te gebruiken. In het formulier vind je ook hier en daar wat regels PHP terug. Geef met commentaar aan wat deze regels doen.
2. We gaan nu verder met het bestand Registreren.php. Wanneer de gebruiker op de knop registreren heeft gedrukt, dan dient in eerste instantie het complete formulier te worden ingelezen middels de POST methode.
3. Daarna gaan we de ingelezen waardes valideren. In het bestand "Functies.php " vind je een aantal functies die je hierbij kunnen helpen (zie ook opdracht 9.05 van de cursus PHP). Hierbij hebben de velden de volgende eisen:
   * de volgende velden zijn verplicht: voornaam, achternaam, telefoonnummer, e-mail,gebruikersnaam, paswoord en retype paswoord.
   * De volgende velden mogen alleen letters bevatten: voornaam, achternaam en plaats.
   * De volgende velden moeten minimaal 2 karakters lang zijn: voornaam en achternaam.
   * De volgende velden moeten minimaal 6 karakters lang zijn: paswoord en retype paswoord.
   * De postcode moet een geldige NL postcode zijn.
   * Het telefoonnummer moet een geldig NL telefoonnummer zijn.
   * Het e-mailadres moet aan het formaat van een e-mailadres voldoen.
   * Wachtwoord en retype wachtwoord moeten identiek zijn.
   * De gebruikersnaam mag niet reeds in de database voorkomen.
4. Na het valideren kunnen er twee situaties ontstaan, namelijk: er is een validatie fout ontstaan of het formulier is compleet correct gevalideerd. Wanneer er een validatie fout is ontstaan dan dient de gebruiker opnieuw het formulier te zien met daarin aangegeven welke validatie fout(en) er zijn aangetroffen en hoe hij deze kan verbeteren.
5. Wanneer het formulier correct is gevalideerd, dan kunnen we gaan kijken hoe we de gegevens in de database kunnen opslaan. Maak hiervoor een nieuwe INSERT query aan middels het prepared statement principe. Via deze query zorg je ervoor dat de gegevens van de bezoeker in de tabel "gebruikers " wordt gezet. Het paswoord en de salt moeten op dit moment ook worden aangemaakt omdat deze gecodeerd in de database moeten worden ingevoerd. Het stukje code wat dit verzorgt is al in het bestand aanwezig maar gecommentariseerd. Je moet dit stukje dus wel gebruiken. Daarnaast dien je het gebruikers level op 1 te zetten!
6. Wanneer de gegevens succesvol aan de database zijn toegevoegd, dan krijgt de bezoeker de volgende melding op zijn scherm: "U heeft zich succesvol geregistreerd en kunt vanaf nu inloggen op de website ". Daaronder komt te staan: "U wordt binnen 5 seconden doorgestuurd naar de hoofdpagina ".

**Inleveren:**

- Screenshot(s) van de werking van je programma.

- Gemaakte code als tekst.

- Lever dit in als Word bestand (.doc of .docx).

# PM09 Menu op basis van gebruikers levels

Binnen een website zal je vaak gebruikers levels tegenkomen. Dit houdt in dat de ene gebruiker meer rechten heeft dan de andere gebruiker. Een Admin mag bijvoorbeeld informatie op de website aanpassen terwijl een normale gebruiker dit natuurlijk niet mag.

**Werking programma**

In deze opdracht is het de bedoeling dat we het menu dynamisch gaan maken in plaats van het statische menu wat er nu is. In de database vind je een tabel "menu ", waar alle menu items in staan. Wanneer je deze tabel opent zie je dat ieder menu item een eigen level heeft. Een gebruiker dient minimaal dit level te hebben alvorens het menu item wordt weergegeven op de website. Binnen de website van Cinema 7 betekent dit:

- Bezoeker: deze kan alleen standaard pagina's zien (level 0).

- Klant: deze kan de standaard pagina's zien en de pagina "Mijn Profiel " en "Besteloverzicht " (level 1).

- Admin: kan alles zien wat de klant kan zien + er is een beheer gedeelte (level 3).

We zullen de volgende stappen moeten uitvoeren om het menu op basis van levels te realiseren:

1. Standaard gaan we er van uit dat iemand nog niet is ingelogd (hij of zij is dus level 0). Standaard is dus ook de knop inloggen in het menu te zien. Dit dien je hard-coded te regelen. Dit doe je door de variabelen $Level en $MenuInUitloggen te initializeren op hun standaard waarden.
2. Wanneer de gebruiker is ingelogd dient het level te worden veranderd in het level wat de gebruiker heeft in de database en dient de knop inloggen te worden veranderd in een knop uitloggen. Dit doen we middels de functie LoginCheck uit het bestand "Functies.php". Met deze functie kun je controleren of de gebruiker is ingelogd. De functie geeft true terug wanneer de gebruiker is ingelogd en false wanneer dit niet het geval is. Bij true dien je de standaard waarden van $Level en $MenuInUitloggen te overschrijven.
3. Zorg er nu vervolgens voor dat je de dynamische menu items via een SQL-query uit de database haalt. Let hierbij goed op het level van de gebruiker, deze dien je in je Query te verwerken.
4. Geef de menu items op de juiste manier weer zodat er weer een goed uitziend menu ontstaat. De werking van het menu kun je testen door met de verschillende gebruikers in te loggen.

Inleveren (enz.)

# PM10 Mijn profiel

Zoals we in de vorige opdracht gezien hebben krijgen gebruikers die inloggen in het menu een extra optie "Mijn Profiel ". De bedoeling van deze pagina is dat de gebruiker zijn of haar gegevens kan inzien of wijzigen.

**Werking programma**

Om dit te bewerkstelligen zullen we de volgende handelingen moeten uitvoeren:

1. In het bestand MijnProfiel.php: Haal het gebruikersID uit de SESSION op en maak een SELECT query waarbij je aan de hand van dat gebruikersID de gegevens van de gebruiker opvraagt.
2. De opgehaalde gegevens dienen te worden weergegeven in een formulier zodat de gebruiker ze eventueel kan aanpassen. Maak hiervoor een formulier in het bestand: "MijnProfielForm.php ".  
   **TIP:** dit lijkt erg op het formulier dat je hebt gebruikt in de opdracht "registreren “. De gebruiker mag alles aanpassen behalve zijn gebruikersnaam en paswoord. Dit laten we even achterwegen.
3. Wanneer de gebruiker wijzigingen maakt in zijn of haar gegevens dienen we deze in eerste instantie de gegevens weer in te lezen.
4. We valideren deze waardes weer met dezelfde eisen als bij de pagina "registreren ".
5. Wanneer het formulier succesvol is gevalideerd dienen we de gewijzigde gegevens door te voeren in de database middels een UPDATE query. Maak ook hierbij gebruik van de prepared statement methode. Wanneer er iets tijdens de validatie is misgegaan dan dient de gebruiker, net als op de pagina registreren, hiervan op de hoogte te worden gesteld.
6. Wanneer de gegevens succesvol in de database zijn veranderd, dan krijgt de gebruiker de volgende melding: "Uw gegevens zijn succesvol aangepast ".

**Inleveren:**

- Screenshot(s) van de werking van je programma.

- Gemaakte code als tekst.

- Lever dit in als Word bestand (.doc of .docx).

# PM11 Film toevoegen

Wanneer je inlogt als Admin dan krijg je 2 extra menu items (film toevoegen & film aanpassen/verwijderen) waarmee je de films kan beheren. Middels het menu item "film toevoegen" kun je een nieuwe film aan de database toevoegen via de website.

**Werking programma**

Om deze pagina te bewerkstelligen, dienen we de volgende stappen uit te voeren:

1. Maak in het bestand FilmToevoegenForm.php een form met de benodigde aantal velden. Kijk in de database tabel "films " om te zien welke informatie er over een film wordt opgeslagen en welke velden je dus nodig hebt. Kies bij ieder veld voor het juiste bijpassende type input field (bijvoorbeeld: textfield, select, radio, et cetera).
2. Wanneer de gegevens zijn ingevuld en men op de knop "toevoegen " heeft gedrukt, dan dienen deze gegevens opgeslagen te worden in de database. Controleer eerst in het bestand FilmToevoegen.php of het formulier is verzonden.
3. Voordat we de gegevens gaan opslaan moeten deze eerst nog worden gevalideerd. Ook hiervoor kun je de functies uit het bestand Functies.php gebruiken. Hierbij hebben de velden de volgende eisen:
   * De volgende velden zijn verplicht: titel, beschrijving, duur en prijs.
   * Het veld prijs moet een geldig getal zijn. Dit mag zowel een geheel getal als een komma getal zijn.
   * Het veld duur mag alleen een geheel getal bevatten. Dit kun je controleren met de volgende PHP functie: ctype\_digit()
4. Nadat je alles hebt gevallideerd, voeg je alles toe aan de database, wederom met een INSERT query, nadat je de connectie met de database hebt gelegd. Wanneer de gegevens succesvol aan de database zijn toegevoegd, dan komt er de volgende melding op het scherm te staan: "De film is toegevoegd aan de database ". Hierna zijn alle velden weer leeg zodat je gemakkelijk verder kan gaan met het toevoegen van de volgende film.

**Inleveren:**

- Screenshot(s) van de werking van je programma.

- Gemaakte code als tekst.

- Lever dit in als Word bestand (.doc of .docx).

# PM12 Film wijzigen

In de komende 2 opdrachten gaan werken in 1 en hetzelfde bestand, namelijk FilmAanpassenVerwijderen.php. In eerste instantie gaan we ervoor zorgen dat we de gegevens van een film kunnen wijzigen.

**Werking programma**

Om dit te bewerkstelligen, dien je de volgende stappen uit te voeren:

1. Maak eerst in het bestand FilmAanpassenForm.php het formulier wat nodig is om een film te kunnen aanpassen. TIP: dit is vrijwel identiek aan die van film toevoegen.
2. Zorg er daarna voor dat je een overzicht krijgt van alle films die in de database staan. Hiervoor kun je van het stukje code gebruik maken wat wij hebben gemaakt in opdracht "reserveren ". **LET OP:** alle films dienen zichtbaar te zijn, ongeacht hun status.  
   Vervang vervolgens in de code het knopje "reserveren " door een knopje "wijzigen " en een knopje "verwijderen". De knoppen dienen een GET variabele "Action" mee te krijgen met als waarde "Edit" bij de knop wijzigen en "Del" bij de knop verwijderen. Daarnaast is het handig om het FilmID ook middels een GET variabele onder de knop te hangen.
3. Wanneer er door de Admin op het knopje "wijzigen " wordt gedrukt dan krijgt deze een soortgelijk formulier te zien als bij de opdracht "film toevoegen ". Het formulier is nu echter gevuld met de gegevens van de film die is aangeklikt. Dit doe je door in eerste instantie een SELECT query uit te voeren op de database waarbij je gebruik maakt van de in de GET aanwezige variabele FilmID.
4. Vervolgens zet je de waardes uit de database over naar de juiste variabelen en roep je het formulier aan middels een require().
5. De Admin kan wijzigingen in de gegevens van de film doorvoeren en vervolgens op de knop "wijzigen " drukken. Wanneer dit gebeurd zullen de gegevens in eerste instantie weer moeten worden gevallideerd. Hierbij gelden dezelfde regels als bij Film Toevoegen, d.w.z.:
   * De volgende velden zijn verplicht: titel, beschrijving, duur en prijs.
   * Het veld prijs moet een geldig getal zijn. Dit mag zowel een geheel getal als een komma getal zijn.
   * Het veld duur mag alleen een geheel getal bevatten. Dit kun je controleren met de volgende PHP functie: ctype\_digit()
6. Tot slot dienen we de database van deze nieuwe gegevens te voorzien door gebruik te maken van een UPDATE query. Wanneer de gegevens succesvol in de database zijn veranderd, dan krijgt de Admin de volgende melding: "De gegevens van deze film zijn aangepast ". Daarna wordt de beheerder binnen 5 seconden teruggestuurd naar de beheerpagina.

**Inleveren:**- Screenshot(s) van de werking van je programma.  
- Gemaakte code als tekst.  
- Lever dit in als Word bestand (.doc of .docx)

# PM13 Film verwijderen

Deze opdracht gaat verder in het bestand FilmAanpassenVerwijderen.php, waar de opdracht "film wijzigen "is gestopt. Naast het toevoegen of wijzigen van films willen we ook de mogelijkheid hebben een film te verwijderen.

**Werking programma**

Vroeg in eerste instantie naast iedere "wijzigen "knop een "verwijder "knop toe, mocht je dit nog niet gedaan hebben in de vorige opdracht.

1. In principe is de uitvoering verder rechttoe rechtaan. Wanneer de Admin op de knop verwijderen drukt dan dien je middels een DELETE query de gegevens van de film uit de database te verwijderen. Maar ook hier weer gebruik van de prepared statement methode.  
   **LET OP:** zorg ervoor dat je in het WHERE gedeelte van de query het unieke nummer van de film meegeeft. Doe je dit niet, dan kan het zijn dat je meerdere films verwijderd of zelfs alle films.

**Inleveren:**

- Screenshot(s) van de werking van je programma.

- Gemaakte code als tekst.

- Lever dit in als Word bestand (.doc of .docx).

# PM14 Film data/tijden

In één van de eerste opdrachten hebben we de pagina ‘Reserveren’ gevuld met een schema waarin alle in de bioscoop draaiende films zichtbaar zijn. Elke film heeft daarnaast en knopje ‘reserveren’ gekregen die linkt naar de pagina data.tijden.php wanneer je er op drukt.

De gebruiker dient op de pagina ‘data/tijden’ een overzicht te krijgen van de data en de tijden waarop zijn/haar gekozen film draait. De database heeft hiervoor de tabel ‘vertoningen’. In deze tabel zien we welke film in welke zaal draait, op welke dag en op welke tijd. Dit noemen ze een koppeltabel tussen de tabellen ‘films’ en ‘zalen’.

**Werking programma**

1. De bedoeling is dat je een SELECT query uitvoert op de tabel vertoningen met in het WHERE gedeelte het ID van de film die de gebruiker wil reserveren. Het resultaat geef je weer op het scherm. Op deze manier krijgt de gebruiker een overzicht van de data en tijden waarop de film draait. Je dient iedere regel van een Radiobutton te voorzien. **Let op** dat deze van dezelfde groep zijn.  
   De gebruiker kan middels de radiobuttons kiezen op welke dag en tijd hij/zij de film wil reserveren. Helemaal bovenaan dient de titel en de prijs van de film aangegeven te worden. Ook hiervoor kun je natuurlijk een query uitvoeren, echter kun je ook beide tabellen in één keer aanspreken door een query met een JOIN te gebruiken:  
     
   **SELECT** f.Titel, f.Prijs, v.\* **FROM** Films f, Vertoningen v **WHERE** v.FilmId = f.FilmId **AND** v.FilmId = xxx  
     
   Bovenstaande query mag je gebruiken mits je in je code uitleg geeft, middels commentaar, hoe deze query werkt.  
     
   Onderaan de pagina moet een Select field komen waarin de gebruiker het aantal te bestellen kaarten kan kiezen. De keuze staat standaard op 1 en kan maximaal 10 zijn.  
   Verder dien je de volgende knop toe te voegen:
2. Wanneer er op bovenstaande knop gedrukt wordt dan moeten alle benodigde gegevens in een SESSION worden opgeslagen en wordt de gebruiker doorgestuurd naar het bestand  ‘besteloverzicht.php’. Het is aan te bevelen de bestelgegevens in een associatief array te zetten waarbij de volgende gegevens minimaal dienen te worden opgeslagen ( Dit Array geef je bijvoorbeeld de naam “Film” ):
   * VertoningsId
   * FilmId
   * Titel
   * Prijs
   * AantalKaartjes
3. Daarnaast dient er een tweede array te zijn met de naam "Bestelling".Hierin worden alle films die worden gereserveerd opgeslagen. Dus voor iedere film die een klant reserveert dient het array "Film" te worden toegevoegd aan het array "Bestelling " (dit kun je doen middels de array\_push() methode) en vervolgens in de sessie ($\_SESSION[“ Bestelling”]) te worden geplaatst. Je zal dus moeten controleren of de sessie ($\_SESSION[“ Bestelling”]) al bestaat. Zo ja, lees deze dan uit en voeg de nieuwe film toe, zo nee, maak een leeg array “ Bestelling” aan.

**Inleveren:**

- Screenshot(s) van de werking van je programma.

- Gemaakte code als tekst.

- Lever dit in als Word bestand (.doc of .docx).

# PM15 Besteloverzicht (deel 1)

Het besteloverzicht is de laatste pagina die de gebruiker ziet voordat de reservering definitief wordt. In deel 1 van deze opdracht beperken we ons vooral tot het visualiseren van het besteloverzicht. Extra functionaliteit wordt toegevoegd in deel 2.

De pagina kun je opmaak technisch het beste verdelen in 2 blokken, bijvoorbeeld aan de bovenkant een lijst met films die gereserveerd moet worden en aan de onderkant de gegevens van de gebruiker.

**Werking programma**

De volgende stappen dienen we uit te voeren om een besteloverzicht te genereren:

1. Lees bovenaan het in de SESSION opgeslagen array terug in.
2. Geef vervolgens alle door de gebruiker gereserveerde films op het scherm weer door via een For-loop door het array te lopen. De gegevens die je op het scherm moet tonen moet je verzamelen door de gegevens die in het array zitten aan te vullen met gegevens die je uit de database haalt. Immers bevat het in de sessie aanwezige array een VertoningsId en een FilmId. Daarnaast bevat de sessie waarmee de gebruiker is ingelogd het KlantId van de gebruiker. Middels queries kun je dus de benodigde gegevens uit de database halen en op het scherm toveren.  
   Zorg er tevens voor dat het aantal bestelde kaarten in een Select field achter de film komen te staan. Op deze manier kunnen we er straks voor zorgen dat je dit aantal nog kunt wijzigen. Let erop dat je dit alles tussen de Form tags (POST method) zet en dat onderaan een knopje ‘Wijzigen’ komt te staan.  
   Daarnaast dient de regel nog te worden voorzien van een knopje ‘verwijderen’, zodat de gebruiker de film later ook kan verwijderen wanneer hij/zij de film toch niet wil reserveren. Deze verwijder knop bevat een link naar de huidige pagina met als toevoeging de variabele “Del”.  
   Maak tot slot nog een overzicht van de totaalprijs.
3. Onder het lijstje met te reserveren films willen we de gegevens van de gebruiker weergeven zodat deze kan controleren of deze kloppen. Voer hiervoor een query uit op de tabel klanten, gebruikmakend van het KlantID uit de SESSION.
4. Tot slot maken we onderaan de pagina een knop ‘ Bestellen’. Wanneer er op deze knop gedrukt wordt dan dient de reservering daadwerkelijk in de database opgeslagen te worden. Dit gaan we doen in deel 2 van deze opdracht. De code van de knop is als volg  
     
   [**Bestellen**](https://leijgraaf.onderwijsonline.nl/manage/assignments/edit/index.php?PaginaNr=10&Bestel=true)  
     
   **LET OP:** Het PaginaNr van het bestand Besteloverzicht.php moet 10 zijn (pas dit aan in je index.php wanneer dit niet zo is). Het PaginaNr voor het bestand BestellingVerwerken.php moet 11 zijn.

**Inleveren: (enz)**

# PM16 Besteloverzicht (deel 2)

In het 2e deel van deze opdracht gaan we er voor zorgen dat de gegevens van de reservering ook echt in het database worden opgeslagen. Daarnaast gaan we zorgen dat de ‘verwijder’ knoppen werken en dat je het aantal kaarten nog kan wijzigen.

**Werking programma**

Om dit te bewerkstelligen, dienen we de volgende stappen uit te voeren:

1. Knopje “verwijderen” (bestand: BestelOverzicht.php)  
     
   Wanneer er bij het besteloverzicht op het knopje ‘verwijderen’ wordt gedrukt, dan dien je de film uit het array te verwijderen en vervolgens het array weer terug te stoppen in de SESSION. Hiervoor dien je met een for-loop door het array heen te lopen en te controleren of het VertoningsId van de verwijder link overeenkomt met het VertoningsId in het array. Wanneer dit het geval is, dan moet dat specifieke array element worden verwijderd. Hiervoor kun je onderstaand stukje code gebruiken:  
     
   **unset($Bestelling[$i]);**  
     
   **$Bestelling = array\_values($Bestelling);**
2. Knopje “Wijzigen” (bestand: BestelOverzicht.php)  
     
   Wanneer er op het knopje "wijzigen " wordt gedrukt, dan dient het aantal kaartjes die opgeslagen zijn in de Session array “ bestelling” te worden gewijzigd. In principe kan de klant iedere film een ander aantal kaartjes geven en deze met één druk op de knop "wijzigen " veranderen. Zorg ervoor dat ieder select Field in je html code de volgende naam heeft: **AantalKaartjes[]**. De vierkante haakjes zorgen ervoor dat alle velden in één keer worden verstuurd als een array. Dit array kun je gewoon uitlezen middels de regel **$\_POST[‘AantalKaartjes’];** zoals je gewend bent.  
     
   Je hebt nu dus 2 arrays, het array "Bestelling "en het array "AantalKaartjes”. Beiden arrays zijn identiek in lengte. Door gebruik te maken van een For-loop kun je dus één op één per film het aantal kaartjes wijzigen. Je hoeft niet eens de VertoningsId’s te controleren omdat zelfs de volgorde van beide arrays hetzelfde zijn.
3. Reservering opslaan in de database (bestand: BestellingVerwerken.php)
   1. Wanneer op de knop ‘Bestellen’ is gedrukt, dan gaan we daadwerkelijk de bestelling opslaan in de database. Lees hiervoor eerst de SESSION "bestelling" uit.
   2. Voor het opslaan van de bestelling maken we gebruik van 2 database tabellen, namelijk "reserveringen "en "vertoningen\_reserveringen ". Allereerst voeg je middels een INSERT query het KlantId toe aan de tabel "reserveringen ". Deze tabel heeft ook een veld ReserveringsId welke auto\_increment is. De database maakt dus zelf een nieuw uniek reserveringsnummer aan voor iedere nieuwe reservering. Nadat je de query hebt uitgevoerd dien je te achterhalen welk reserveringsnummer er is aangemaakt.
   3. Wanneer we de tabel "vertoningen\_reserveringen " gaan vullen hebben we namelijk dit reserveringsnummer nodig. De tabel "vertoningen\_reserveringen " is een koppel tabel waarin alle te reserveren films worden gekoppeld aan 1 reservering. Door gebruik te maken van een for-loop doorloop je het array "Bestelling " en vul je 1 voor 1 de tabel "vertoningen\_reserveringen ".
   4. Wanneer alles succesvol in de database staat, dan krijgt de gebruiker  de volgende melding: ‘Uw bestelling is succesvol bij ons ontvangen! Re bestelling is bij ons bekend onder bestelnr.: xxx (waarbij xxx het ReserveringsID is uit de tabel reserveringen). Wij bedanken u voor uw vertrouwen in ons en wensen u al vast veel plezier met de voorstelling.’. Daaronder komt een link naar de homepage. Gebruik hier geen header refresh.

**Inleveren:**

- Screenshot(s) van de werking van je programma.

- Gemaakte code als tekst.

- Lever dit in als Word bestand (.doc of .docx).

# PMDELUXE

Deze opdracht is bedoeld voor iedereen die klaar is met de reguliere Cinema7 opdrachten en zich verder wil verdiepen. Onderstaand zijn een aantal uitbreidingen die je zou kunnen doorvoeren binnen de website. Je kunt zelf kiezen wat je wil maken en in welke volgorde.

**Design**

* Maak een compleet nieuw design voor de website en implementeer dit.

**Content Management System (CMS)**

* Maak een pagina "Mijn Bestellingen", waarop de gebruiker zijn/haar vorige bestellingen kan inzien.
* Zorg er voor dat de Admin een overzicht kan krijgen waarop alle bestellingen te zien zijn. Je kunt hier ook nog een statistiekenpagina bij maken.
* Zorg er voor dat je bij het aanmaken en aanpassen van een film een plaatje van je computer kan selecteren en dat deze dan in de map images wordt gezet en in de database. (Form type File)
* Zorg er voor dat de Admin de tijden waarop een film draait kan toevoegen en aanpassen. Ook het ZaalNr waar de Film draait kun je op deze manier laten wijzigen.
* Zorg er voor dat je onder "Mijn Profiel" ook je wachtwoord kan wijzigen.

**User Experience (UX)**

* Maak de website Responsive of Adaptive.
* Zorg dat de gebruiker een email bevestiging krijgt wanneer hij/zij een bestelling plaatst.
* Zorg dat de gebruiker zijn/haar wachtwoord kan resetten wanneer hij/zij deze vergeten is.

**Security**

* Zorg er voor dat iemand die 3 x het verkeerde wachtwoord gebruikt bij het inloggen, een 30 minuten niet meer mag inloggen.
* Zorg er voor dat wanneer iemand zich registreert, hij/zij eerst via een email moet bevestigen dat hij/zij de gebruiker is, voordat deze gebruik kan maken van het systeem.

# Full site