

# LEDENADMINISTRATIE

PHP en MySQL



**Judith van der Velden**

Studentnummer: 310791081

Datum: 12-03-2024

## Introductie

Dit verslag presenteert het ontwerp, de ontwikkeling en implementatie van een ledenadministratiesysteem, ontwikkeld met behulp van PHP en SQL. Het systeem is gebaseerd op de principes van Create, Read, Update en Delete (CRUD) en is gestructureerd volgens het Model-View-Controller (MVC) architectuurpatroon.

Dit verslag begint met een beschrijving van de ontwikkelomgeving, gevolgd door een uitleg van de database en de werking van de applicatie. Ten slotte wordt een kort testverslag weergegeven.

## Ontwikkelomgeving

Voor het ontwikkelen van de ledenadministratie heb ik gebruik gemaakt van Visual Studio code. AMPPS heb ik geïnstalleerd zoals aangegeven werd in het boek. Dit programma heeft toen Apache, MySQL en PHP 7.4 geïnstalleerd. Ik werk op een macbook.

Het testen van de applicatie kon, door het installeren en aanzetten van AMPPS, via de localhost in mijn browser. De bestanden voor het testen zette ik dan in de map 'www' binnen de map 'AMPPS' in 'Applications'. Deze bestanden kon ik dan aanroepen via localhost.

Voor het bekijken en eventueel aanpassen van de database gebruikte ik PHPmyAdmin waar ik ingelogd was met de juiste inloggegevens.

## Database maken

In de map 'SQL' vind je het SQL-creatiescript en database export. Wanneer dit bestand wordt uitgevoerd gebeurt het volgende:

1. Het script begint met het instellen van enkele SQL-modi en transactie-instellingen, gevolgd door de creatie van de database 'ledenadministratie' als deze nog niet bestaat.
2. Alle tabellen worden gemaakt (Boekjaar, Contributie, Familie, Familielid, Login en Soort\_lid) met hun respectieve kolommen en constraints.
3. Na de creatie van de tabellen worden de gegevens in de tabellen ingevoegd met behulp van de 'INSERT INTO'-opdrachten.
4. Tot slot worden alle externe sleutelconstraints toegevoegd aan de tabellen, die de relaties tussen de tabellen definiëren.

## Inloggegevens

De inloggegevens voor de database zijn terug te vinden en eventueel aan te passen in model/pdoconnectie.php. Als je dit wil aanpassen moet er uiteraard in de database een nieuwe gebruiker gemaakt worden die de juiste rechten heeft. De volgende gebruiker is al aangemaakt en daarmee kun je inloggen:

**Gebruikersnaam: admin**

**Wachtwoord: admin**

In de applicatie zelf moet ook ingelogd worden om toegang te krijgen. De wachtwoorden zijn hier vanwege veiligheidsoverwegingen gehasht. Er zijn al 2 gebruikers aangemaakt, dus je kunt kiezen welke je wil gebruiken. De functionaliteiten zijn hetzelfde voor beide gebruikers.

Wanneer je de volgende gegevens invoert kunt je inloggen en alle functionaliteiten zien:

**Gebruikersnaam: admin**

**Wachtwoord: admin**

of

**Gebruikersnaam: LOI**

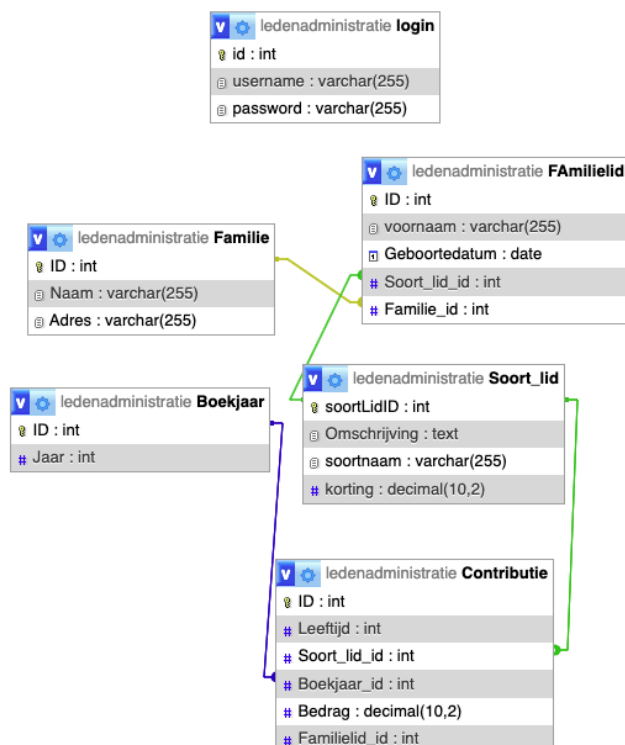
**Wachtwoord: LOI123**

## Database

Hier ga ik uitleggen hoe mijn database is opgebouwd en hoe de tabellen daarin zijn opgebouwd.

## Structuur

Mijn database bestaat uit 6 tabellen. Dit is hoe het eruit ziet:



## Tabellen

Zoals ik al zei bestaat de database uit 6 tabellen. Deze hebben allemaal weer kolommen. Ik zal hier uitleggen hoe de tabellen zijn opgebouwd.

### Familie

In de familietabel komen de families die ingevoerd worden in de applicatie te staan. Deze tabel bestaat uit de volgende kolommen:

- ID: Dit is de primary key van deze tabel. Met AUTO INCREMENT wordt deze telkens toegevoegd wanneer er een nieuwe familie wordt toegevoegd. De familie id wordt doorgegeven aan de tabel Familielid aan Familie\_id, omdat meerdere leden in dezelfde familie kunnen behoren.
- Naam: Dit is de achternaam van de familie, waar in de tabel Familielid meerdere leden aan toegevoegd kunnen worden. Deze kolom is VARCHAR omdat hij voornamelijk zal bestaan uit letters en je niet weet uit hoeveel letters, maar ook weer niet zo veel dat je er text van zou maken.
- Adres: Hierin wordt het adres waar de familie woont opgeslagen. Dit bestaat uit letters en cijfers en is daarom ook VARCHAR.

### Familielid

In de tabel Familielid worden alle familieleden opgeslagen. Deze moeten altijd gekoppeld worden aan een familie uit de tabel Familie. De tabel Familielid bestaat uit de volgende kolommen:

- ID: Dit is de primary key van deze tabel. Telkens als een nieuw lid wordt toegevoegd, wordt met AUTO INCREMENT een nieuw id aangemaakt. Deze ID wordt doorgegeven aan de Contributietabel, zodat de contributies juist gekoppeld worden.
- Voornaam: Hierin wordt uiteraard de voornaam van het lid toegevoegd. Dit is een VARCHAR, omdat het uit letters zal bestaan en je weet nooit hoeveel letters.
- Geboortedatum: Hier wordt de geboortedatum ingevoerd op basis van date. Deze is nodig om te berekenen hoe oud een lid is en in welke kortingsgroep hij of zij zal vallen.
- Soort\_lid\_id: Deze wordt overgenomen van de tabel Soort\_lid en laat zien in welke kortingsgroep het lid valt. Een lid kan maar tot 1 groep behoren.
- Familie\_id: Deze wordt overgenomen van de tabel Familie, wanneer het lid gekoppeld wordt aan die familie.

### Contributie

De contributietabel bestaat voornamelijk uit kolommen waarvan de waarden worden overgenomen uit andere tabellen.

- ID: Dit is de primary key van deze tabel en wordt automatisch met AUTO INCREMENT aangemaakt wanneer een lid wordt toegevoegd aan een contributie.

- Leeftijd: Deze wordt berekend in de code via de geboortedatum die wordt doorgegeven in de tabel Familielid en het Jaar dat wordt doorgegeven in de tabel Boekjaar. Dit zijn alleen maar cijfers, dus is deze kolom INT.
- Soort\_lid\_id: Deze zagen we in de tabel Familielid ook al voorbij komen. Hierin wordt de kortingsgroep waar het lid bij hoort bepaald. Dit is nodig zodat het kortingspercentage bepaald kan worden. Dit zijn alleen cijfers, dus INT.
- Boekjaar\_id: Deze is gekoppeld aan de tabel Boekjaar en is nodig om te bepalen voor welk jaar de contributie wordt aangemaakt. Dit is dan weer nodig om de leeftijd van het lid in dat jaar te bepalen. Ook dit is een kolom met alleen cijfers, dus INT.
- Bedrag: Dit is een decimaal getal dat wordt berekend op basis van het soort lid, de geboortedatum en het boekjaar. Zo weet de beheerder wat het lid moet betalen voor dat jaar.
- Familielid\_id: Hiermee wordt het juiste lid gekoppeld aan het juiste bedrag.

## Boekjaar

De tabel boekjaar is om te kunnen kiezen voor welk jaar een contributie wordt aangemaakt en bestaat uit de volgende kolommen:

- ID: Dit is de primary key en een AUTO INCREMENT die wordt aangemaakt wanneer een nieuw boekjaar wordt toegevoegd.
- Jaar: Dit is het jaar dat wordt toegevoegd. Dit zal dus altijd een getal zijn, dus INT.

## Soort\_lid

In de tabel Soort\_lid worden de kortingsgroepen bepaald. Deze tabel bestaat uit de volgende kolommen:

- soortLidID: Dit is de primary key en een AUTO INCREMENT die wordt aangemaakt wanneer een nieuw soort groep wordt aangemaakt.
- Omschrijving: Hierin wordt beschreven wie er bij de groep horen, dus wie er gebruik mogen maken van de korting. Omdat dit een lang stuk tekst zou kunnen zijn is het een text.
- soortnaam: Hierin wordt een naam gegeven die de nieuwe groep omschrijft. Dit bestaat uit letters en is daarom VARCHAR.
- korting: Hierin wordt weergegeven hoeveel korting je krijgt als je volgens je leeftijd bij deze groep hoort. Dit is altijd een decimaal getal, dus daarom Decimal(10,2).

## Login

Deze tabel heeft niet zo veel te maken met de rest van de tabellen, maar wordt gebruikt om in te kunnen loggen in de ledenadministratie.

- ID: Dit is de primary key en een AUTO INCREMENT die wordt aangemaakt als een nieuwe gebruiker wordt toegevoegd.

- username: De gebruikersnaam van de user wordt hier toegevoegd en is VARCHAR, omdat het verschillende soorten tekens kan hebben.
- password: Het wachtwoord wordt hier toegevoegd en is ook VARCHAR omdat het verschillende soorten tekens kan hebben. Het wachtwoord dat wordt toegevoegd zul je niet kunnen zien in de database, omdat het gehasht is.

## Werking applicatie

### Opzet applicatie

De applicatie is MVC georiënteerd. Dat wil zeggen dat het is opgebouwd volgens de model-view-controller architectuur.

Mijn index.php staat buiten de mappenstructuur van de mvc. Dit is de startpagina en moet daarom buiten de mappenstructuur staan.

In de index.php wordt een sessie gestart, de controller geladen en de databaseverbinding opgezet. Vervolgens wordt gecontroleerd of een loginformulier is verzonden. Als dat zo is, wordt geprobeerd in te loggen. Ook wordt gecontroleerd of een logout-verzoek is ingediend. Als er een foutmelding is, wordt deze getoond.

Daarna wordt de HTML-pagina gerenderd, inclusief een header, navigatiemenu, en de hoofdinhoud. Als de gebruiker is ingelogd, wordt de hoofdinhoud (main.php) geladen. Als de gebruiker niet is ingelogd, wordt een inlogtekst getoond. Als er een foutmelding is, wordt deze ook weergegeven. De footer wordt aan het einde van de pagina toegevoegd. Kortom, deze pagina controleert de gebruikerssessie, toont inhoud op basis van de sessiestatus en verwerkt inlog- en uitlogacties.

Welke pagina wordt weergegeven in de main van het scherm wordt bepaald door de acties in main.php.

### Model

Het model staat voor de gegevens en de logica van de applicatie. Het bevat alle informatie en regels die nodig zijn om de applicatie te laten werken.

Ik heb ervoor gekozen om mijn model op te breken per pagina. Zo is alles net wat overzichtelijker.

In al mijn modelbestanden definieer ik eerst een klasse die fungeert als het model. Daarvoor include ik de pdoconnectie, zodat verbinding met de database wordt gemaakt.

De klasse is verantwoordelijk voor het beheren van de gegevens in het systeem. Het model maakt verbinding met de database en biedt methoden om verschillende bewerkingen uit te voeren, zoals het ophalen van alle families, het toevoegen van een nieuw familielid, het aanpassen van een contributie of het verwijderen van een boekjaar.

De klasse bereid SQL-query's voor met behulp van voorbereide statements om SQL Injection te voorkomen. Het maakt ook gebruik van binding van parameters om de veiligheid van de database bewerkingen te waarborgen.

Kortom, het model fungeert als interface tussen de applicatie en de database voor het beheren van gegevens, waardoor het mogelijk is om gegevens op een gestructureerde en veilige manier te manipuleren.

## View

De view is verantwoordelijk voor het presenteren van informatie aan de gebruiker. Het neemt gegevens uit het model en toont deze op een visueel aantrekkelijke manier aan de gebruiker.

Ook de view heb ik helemaal uit elkaar gehaald. Voor elke pagina bestaat een andere view en ook voor de header, footer en nav bestaat een andere view. Ik denk dat dat overzichtelijker is en dat je beter weet waar je moet zijn als je iets aan moet passen.

De views bestaan voornamelijk uit html code, aangevuld met php voor bijvoorbeeld het ophalen van gegevens voor de dropdownmenu's.

## Controller

De controller fungeert als de tussenpersoon tussen het model en de view. Het ontvangt invoer van de gebruiker, zoals klikken op knoppen of het invullen van formulieren en voert vervolgens acties uit op basis van die invoer. Deze acties kunnen bijvoorbeeld het bijwerken van gegevens in het model, het ophalen van nieuwe gegevens, of het doorsturen van de gegevens naar de view om weer te geven, omvatten.

Mijn controller maakt gebruik van de modellen om gegevens uit de database te halen en te bewerken. Ook maakt de code gebruik van de verschillende views om gegevens weer te geven en gebruikersinteractie mogelijk te maken.

Ik heb mijn controller opgedeeld in verschillende klassen om het overzicht te bewaren:

- De FamilieController behandelt functionaliteiten zoals inloggen, uitloggen, het weergeven van de hoofdpagina, ledenpagina, het toevoegen, bewerken en verwijderen van families en familieleden. Het berekent ook de contributie voor elke familie op basis van leeftijd en soort lid.
- De ContributieController beheert de contributies, inclusief het bekijken, toevoegen, bewerken en verwijderen van contributies voor familieleden.
- De GroupController biedt functionaliteit voor het beheren van groepen, waaronder het bekijken, toevoegen en verwijderen van groepen.
- De BoekjaarController regelt het beheer van boekjaren, inclusief het toevoegen, bewerken en verwijderen van boekjaren.

## Funcities

Deze ledenadministratie is opgezet volgens de CRUD methode. Wanneer je op het beginscherm komt zie je dat het is opgebouwd uit 4 vlakken. Bovenin zie je de header, waarin de titel en het inlogveld staat. Links zie je de navigatie en daarnaast staat de hoofdinhoud. Voordat je bent ingelogd staat daar nog 'Je moet eerst inloggen om toegang te krijgen tot de hoofdinhoud.', zodat niet iedereen de inhoud zomaar kan bekijken. Onderin zie je de footer waarin op dit moment het jaartal en de titel zijn weergegeven. Hier zouden eventueel nog adresgegevens of linkjes naar websites kunnen staan.

## Inloggen

Als je de hoofdinhoud wil zien moet je eerst inloggen. Dit doe je in de header in de invulvelden met de opgegeven gegevens. Wanneer je bent ingelogd wordt er op index.php een session gestart en is de hoofdinhoud te zien in de main van de applicatie. Ook wordt nu een uitlogknop zichtbaar in de header.

## Menu

Op het scherm zie je links een navigatiemenu. Hierin kun je navigeren naar de tabellen met daarin de aangemaakte families, de familieleden, de contributies, het jaaroverzicht en de kortingsgroepen.

## Families

Op de pagina 'Families' is de tabel met alle aangemaakte families te zien. Hierin zie je de Familie ID, de familienaam, de familieleden, het adres van de familie en de contributie van de totale familie dat betaald moet worden in 2024.

Boven deze tabel zie je 3 knoppen volgens CRUD:

- Toevoegen: Hier kun je een nieuwe familie toevoegen. Je ziet hier 2 invulvelden waarin een naam en een adres toegevoegd moeten worden.
- Aanpassen: Hierin kun je via een dropdownmenu een beschikbare familie kiezen en daar een nieuwe familienaam en een nieuw adres aan toevoegen. Dit wordt dan geüpdatet in de database. De Familie ID van de gekozen naam krijgt dan de nieuwe naam en het nieuwe adres.
- Verwijderen: Hier zie je een dropdownmenu met daarin de familie ID en de familienaam die je kunt verwijderen. Als je op verwijderen klikt zul je eerst een melding krijgen of je zeker weet dat je de familie wil verwijderen. Als je op OK klikt zullen ook de familieleden verwijderd worden.



## Leden

Op de pagina 'Leden' zie je een tabel met alle leden. Deze staat gesorteerd op Familienaam. Alle leden van de families zullen dus altijd bij elkaar staan op volgorde van wanneer de familie is toegevoegd. Verder zie je de ID van het familielid, de voornaam, de geboortedatum en de groep waar het lid bij hoort.

Ook hier zie je de 3 knoppen volgens CRUD:

- Toevoegen: Als je een lid wil toevoegen, kies je eerst in het dropdownmenu een familie waaraan je het lid wil toevoegen. Dan geef je de voornaam op en vervolgens de geboortedatum.
- Aanpassen: Als je een lid wil aanpassen geef je eerst het lidnummer op, dan geef je een nieuwe naam en selecteer je de familie waaraan je het lid wil toevoegen. Dan geef je ook nog een nieuwe geboortedatum.
- Verwijderen: Hier zie je een dropdownmenu met de ID en de voor- en achternaam van alle leden. Hier kies je er eentje uit en als je dan op verwijderen klikt krijg je weer een melding waar je moet bevestigen dat je het lid wil verwijderen.

## Contributies

Op de pagina met contributies zie je een tabel met daarin de ID van de contributie, de voornaam van het lid waar de contributie aan toegevoegd kan worden, de achternaam, het lidnummer, het bedrag dat betaald moest worden voor het lid in dat jaar en het boekjaar. Ook hier zie je uiteraard weer de 3 knoppen:

- Toevoegen: Hier kies je op basis van het lidnummer en de voor- en achternaam in een dropdownmenu een lid waarvoor je een contributie wil toevoegen. En vervolgens kies je een boekjaar waarin die contributie betaald moet worden.
- Aanpassen: Hier zie je een dropdownmenu waarin je de ID van de contributie kunt kiezen, dan nog een dropdownmenu waarin je het lidnummer kunt kiezen die je aan die contributie wil koppelen en een boekjaar dat je daaraan wil koppelen.

## Jaaroverzicht

Op de pagina jaaroverzicht zie je een tabel met 3 kolommen. Links staat het Boekjaar ID, in het midden staat het jaar dat is gekoppeld aan die ID en rechts zie je hoeveel leden er in dat boekjaar staan ingeschreven. Deze wordt automatisch geüpdatet als er een contributie wordt toegevoegd. Ook hier zie je de 3 bekende knoppen:

- Toevoegen: Hier kun je een jaartal invoeren. Als dit jaartal al bestaat, krijg je een foutmelding.
- Aanpassen: Hier selecteer je een boekjaar ID en kies je welk jaartal je daaraan wil koppelen. Als het jaartal al bestaat, krijg je een foutmelding.
- Verwijderen: Hier selecteer in een dropdownmenu je een boekjaar dat je wil verwijderen.

## Groepen

Op de pagina groepen zie je welke soort leden er zijn. In deze tabel staat het ID dat gekoppeld is aan een groep, de naam van de groep, de omschrijving van wie er bij de groep horen en de korting die je krijgt als je bij deze groep hoort. Ook hier staan natuurlijk de 3 knoppen:

- Toevoegen: Hier vul je een naam, omschrijving en korting in om een nieuw soort groep toe te voegen.
- Aanpassen: Hier selecteer je in het dropdownmenu een bestaande groep en kun je die een nieuwe naam, nieuwe omschrijving en nieuwe korting geven.

## Uitloggen

Bovenin de header kun je ook weer uitloggen. Wanneer je op de uitlogknop klikt wordt via de controller de sessie 'gedestroyt'. Je ziet dan weer de inlogvelden en kunt de hoofdinhoud niet meer zien.

## Testverslag

Alles in de applicatie werkt naar behoren. Je kunt in alle tabellen gegevens toevoegen, aanpassen en verwijderen. Ik heb veel problemen gehad met het inloggen, wat ik niet werkend kreeg. Uiteindelijk bleek dit te komen omdat ik gebruik maakte van de `verify_password` functie met de `$hash` variabele, maar toen nog geen gehashte wachtwoorden had.

Verder heb ik natuurlijk af en toe wat kleine foutjes moeten oplossen. Daarvoor maakte ik dan gebruik van het volgende:

```
error_reporting(E_ALL);  
ini_set('display_errors',1);
```

Dit zorgde ervoor dat ik de errors die de applicatie meegaf kon zien in mijn scherm en meteen kon aanpakken.

## Verbeterpunten

- Een verbeterpunt zou wellicht kunnen zijn om een registratiepagina te maken. Omdat ik nu erg blij was dat het inloggen lukte, heb ik ervoor gekozen om dat niet te doen. Ik was bang dat ik het dan weer verpestte.
- Een ander verbeterpunt zit in de contributietabel. Je kunt nu hetzelfde lid meerdere keren aan hetzelfde jaartal toevoegen. Dat zou niet moeten kunnen.

## Mappenstructuur

In mijn ledenadministratie staan een aantal mappen. Hier leg ik uit waar je wat kunt vinden.

- `Index.php`: Deze staat in de hoofdmap en is de startpagina. Alles in de applicatie gaat via `index.php`.
- `Bijlagen`: Hier staat dit verslag en de video van de applicatie.

- controller: Hier vind je de controller van de applicatie.
- css: Hier staat alle styling van de applicatie.
- model: Hier vind je de verschillende modellen.
- SQL: Hier vind je het SQL-creatiescript en de export van de database.
- view: Hierin staan alle view-bestanden.