New Page

8 CRUD Challenge – Te laat meldingen

datasource

? Leerdoelen

- Je kunt een volledige CRUD-toepassing bouwen met PDO en PHP.
- Je past invoercontrole en gebruikersinteractie toe in formulieren.
- Je begrijpt en gebruikt `prepare()` en `execute()` voor veilige query's.

? Uitleg

In deze challenge bouw je een complete toepassing waarin te laat meldingen van studenten worden bijgehouden. Je gebruikt alles wat je hebt geleerd over formulieren, PDO, databaseverwerking en CRUD-operaties.

Het eindresultaat is een overzichtspagina met alle meldingen. Je kunt nieuwe meldingen toevoegen, bestaande meldingen aanpassen en meldingen verwijderen (met bevestiging).

Voorbeeld

Bekijk het voorbeeld op: https://stampwerk.nl. Probeer de knop '*Weer eentje te laat*' en kijk wat er gebeurt.

Wat is een CRUD?

CRUD staat voor Create, Read, Update en Delete.

Dit zijn de vier basisbewerkingen op een database:

- Create: gegevens toevoegen
- Read: gegevens ophalen en tonen

- Update: gegevens aanpassen
- Delete: gegevens verwijderen

In deze opdracht maak je een CRUD-toepassing voor te laat meldingen. Je slaat per melding de volgende gegevens op:

- naam van de student
- klas
- aantal minuten te laat
- reden van het te laat komen

?? Stappenplan

Stap 1 – Database maken

- Maak een database en een tabel `meldingen` met de volgende velden:
  - `id` (INT, AUTO_INCREMENT, PRIMARY KEY)
  - `student` (VARCHAR)
  - `klas` (VARCHAR)
  - `minuten` (INT)
  - `reden` (TEXT)
- Voeg zelf enkele testregels toe via PHPMyAdmin.

Stap 2 – Read: overzicht maken

- Maak `read.php` waarin je alle meldingen toont in een HTML-tabel.
- Gebruik `require 'connection.php'` voor de databaseverbinding.
- Gebruik `query()` en `fetchAll()` om gegevens op te halen.
- Voeg bovenaan knoppen toe voor “toevoegen” en “zoeken” (optioneel).

Stap 3 – Create: melding toevoegen

- Maak `create.php` met een formulier.
- Voeg invoervelden toe voor student, klas, minuten en reden.

- Sla de gegevens op met een `INSERT`-query via `prepare()` en `execute()`.
- Controleer invoer:
  - geen lege velden
  - minuten is een positief getal
- Toon een foutmelding bij ongeldige invoer en een succesmelding bij correcte invoer.

Stap 4 – Delete: melding verwijderen

- Voeg op `read.php` een knop “verwijder” toe per rij.
- Laat deze verwijzen naar `delete.php?id=...`.
- Toon eerst een bevestigingspagina met de gegevens van de melding.
- Voer pas na bevestiging de `DELETE`-query uit.

Stap 5 – Update

- Voeg op `read.php` een knop “wijzig” toe per rij.
- Laat deze verwijzen naar `update.php?id=...`.
- Haal de bestaande gegevens op met een `SELECT`-query.
- Toon een formulier met ingevulde waarden.
- Werk de gegevens bij met een `UPDATE`-query.

? Reflectie

- Wat heb je geleerd over het werken met databases in PHP?
- Wat ging goed en waar liep je tegenaan?
- Wat zou je de volgende keer anders aanpakken?
- Hoe zorg je voor veilige verwerking van gebruikersinvoer?

? Inleveren

- Screenshot van je overzichtspagina (`read.php`).
- Screenshot van je invoerpagina (`create.php`).

- Screenshot van je wijzigpagina (`update.php`).
- Alle PHP-bestanden, SQL-export en eventuele CSS.
- Een reflectieverslag (.txt of .pdf).
- Wees voorbereid om je code mondeling toe te lichten.

? Puntentelling

Je moet minimaal **81 van de 100 punten** halen om te slagen.

Punten	Onderdeel
10	**Database**: juiste tabel, velden en datatypes.
10	**Read**: overzicht werkt correct.
10	**Layout**: duidelijke en verzorgde opmaak.
20	**Create**: toevoegen werkt inclusief validatie.
10	**Validatie**: ongeldige invoer wordt afgehandeld.
10	**Delete**: verwijderen werkt met bevestiging.
20	**Update**: aanpassen werkt correct.
10	**Codekwaliteit & veiligheid**: gebruik van prepared statements en nette structuur.

Revision #3
Created 2026-03-18 15:37:43 UTC by Max
Updated 2026-07-05 08:41:26 UTC by Max