CRUD - Challenge

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 de rol van `prepare()`, `execute()`, en 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 geleerd hebt over forms, PDO, databaseverwerking en CRUD-operaties.

Het eindresultaat is een overzichtspagina met alle meldingen, waarin je nieuwe meldingen kunt toevoegen, bestaande meldingen kunt wijzigen en meldingen kunt verwijderen (met bevestiging).

Voorbeeld

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

De bedoeling is dat je een CRUD gaat maken. Wat is een CRUD? Dat leggen we zo uit. Eerste het voorbeeld, dat kan je vinden op:

https://stampwerk.nl

image.png

Wat is een CRUD?

CRUD staat voor Create, Read, Update en Delete.

image.png

Deze vier functies zijn de basisfuncties die je op een tabel uit de database kan uitvoeren. Stel je hebt een tabel student, je kunt een student toevoegen (Create), je kunt een overzicht krijgen van studenten (Read), je kunt de gegevens van een student aanpassen (Update) en als laatste kun je een student ook weer verwijderen.

De challenge is dat jij een CRUD gaat maken voor te laat meldingen. Maak daarvoor eerst een tabel in de database waarin je te laat meldingen kan registreren.

Als je naar het voorbeeld kijkt dan zie je dat je van een te laat melding de volgende gegevens wilt vastleggen:

- naam van de student
- klas
- aantal minuten te laat
- de 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 (minimaal 3) 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”.

- "zoeken" staat er als knop maar hoef je niet uit te werken.

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.
- *Front-end controle mag, en **back-end** controle moet en je moet kunnen laten zien dat het werkt.*

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=...`.
- Controleer op de update pagina of het id is meeggegeven em of het id bestaat.
- *Ontbreekt het id of is deze niet aanwezig dan laat je een foutmelding zien.*
- 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 er goed, en waar had je hulp bij nodig?
- Wat zou je in een volgend project anders aanpakken?
- Hoe zorg je ervoor dat je database veilig blijft bij gebruikersinvoer?

? Inleveren

- Eén screenshot van je overzichtspagina (`read.php`).
- Eén screenshot van je invoerpagina (`create.php`).
- Eén screenshot van je wijzigpagina (`update.php`).
- Alle PHP-bestanden, SQL-export van de database, en eventuele CSS.
- Een reflectieverslag in .txt of .pdf met de bovenstaande vragen beantwoord.
- Maak een afspraak met een docent die je uitlegt hoe je code werkt.

? Puntentelling

Je moet minimaal **81 van de 100 punten** halen om deze opdracht succesvol af te ronden.

Punten	Onderdeel
10	**Database en tabel**: juiste velden, datatypes en primary key zijn aangemaakt.
10	**Read**: overzichtspagina toont meldingen correct in een tabel.
10	**Layout**: nette opmaak met CSS (Bootstrap, Tailwind, ...), duidelijke structuur. Gebruik het voorbeeld als uitgangspunt.
20	**Create**: formulier voegt een nieuwe melding toe, inclusief invoercontrole.
10	**Invoercontrole**: negatieve of ongeldige invoer wordt opgevangen met een melding.
10	**Delete**: verwijderen werkt inclusief bevestiging én juiste studentnaam.
20	**Update**: bestaand record kan worden aangepast via een formulier met ingevulde velden.
10	**Codekwaliteit & veiligheid**: gebruik van `prepare()`, `execute()`, nette bestandsstructuur.

Revision #14
Created 2025-06-13 20:59:46 UTC by Max
Updated 2026-03-23 11:01:25 UTC by Max