

CRUD - Challenge

Deze module evt. nog aanpassen

8 CRUD Challenge – Te laat meldingen

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 and or type unknown

Wat is een CRUD?

CRUD staat voor Create, Read, Update en Delete.

image.png and or type unknown

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 met PHPMyAdmin alvast een paar testregels toe.

Stap 2 - Read: overzicht maken

- Maak `read.php` waarin je alle meldingen toont in een HTML-tabel.

- Gebruik `require 'connection.php'` om je databaseverbinding te maken.
- Gebruik `query()` en `fetchAll()` om de gegevens op te halen.
- Voeg bovenaan de pagina een navigatie toe met knoppen voor “toevoegen” en “zoeken”.

Stap 3 – Create: melding toevoegen

- Maak een bestand `create.php` met een formulier voor een nieuwe melding.
- Voeg de invoervelden toe voor student, klas, minuten en reden.
- Voer de gegevens in de database in met `prepare()` en `execute()`.
- **Laat zelf de `execute()`-regel open** en vul deze correct aan.
- Controleer of de invoer geldig is: geen negatieve getallen, geen lege velden.
- Laat een foutmelding zien als de invoer niet klopt, en een succesmelding als het is gelukt.

Stap 4 – Delete: melding verwijderen

- Voeg op `read.php` bij elke rij een knop “verwijder” toe die leidt naar `delete.php?id=...`
- Toon in `delete.php` eerst een bevestiging, inclusief de naam van de student.
- Voer pas na bevestiging de `DELETE`-query uit.
- **Laat de student zelf de naam ophalen met een `SELECT`-query.**

Stap 5 – Update

- Voeg op `read.php` ook een knop “wijzig” toe bij elke melding, die leidt naar `update.php?id=...`.
- Haal in `update.php` de bestaande gegevens op met een `SELECT`-query.
- Toon een formulier met de bestaande gegevens ingevuld.
- **Laat de student zelf de `value`-attributen van de formulier-velden vullen.**
- Laat bij opslaan de gewijzigde gegevens bijwerken 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.
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 <code>prepare()</code> , <code>execute()</code> , nette bestandsstructuur.

Revision #7

Created 13 June 2025 20:59:46 by Max

Updated 26 June 2025 17:16:48 by Max