CRUD - Challenge

8 CRUD Challenge – Te laat meldingen

(AI-editie)

? Leerdoelen

- Je kunt zelfstandig een volledige CRUD-toepassing bouwen met PHP en PDO.
- Je kunt gebruikersinvoer controleren en foutmeldingen afhandelen.
- Je begrijpt waarom veilige databasequery's belangrijk zijn.
- Je kunt je eigen ontwerpkeuzes uitleggen en verdedigen.
- Je kunt AI gebruiken als hulpmiddel zonder afhankelijk te worden van AI.

? Uitleg

Je bouwt een webapplicatie waarin te laat meldingen van studenten worden geregistreerd. De applicatie moet gegevens kunnen toevoegen, tonen, wijzigen en verwijderen.

Je mag AI gebruiken tijdens deze opdracht. De beoordeling gaat echter niet alleen over het eindresultaat, maar ook over jouw begrip van de code.

Tijdens de beoordeling kan een docent willekeurige vragen stellen over jouw code. Kun je niet uitleggen hoe jouw oplossing werkt, dan kunnen punten vervallen.

Wat is een CRUD?

CRUD staat voor:

- **Create** → gegevens toevoegen
- **Read** → gegevens bekijken
- **Update** → gegevens wijzigen
- **Delete** → gegevens verwijderen

Deze vier onderdelen vormen samen de basis van veel webapplicaties.

?? Opdracht

Gegevens die je moet registreren

- Naam van de student
- Klas
- Aantal minuten te laat
- Reden van te laat komen

Verplicht extra veld

Bedenk zelf minimaal **één extra veld** dat volgens jou nuttig is.

Voorbeelden:

- Datum
- Docent
- Lokaal
- Type melding
- Herhaald te laat

Je moet tijdens de beoordeling kunnen uitleggen waarom je dit veld hebt toegevoegd.

Stap 1 – Database maken

- Maak een database en tabel `meldingen`.
- Gebruik de verplichte velden.
- Voeg je eigen extra veld toe.
- Voeg minimaal 5 testrecords toe.

Ontwerpvraag

Leg in een tekstbestand van minimaal 100 woorden uit:

- Waarom heb je voor dit extra veld gekozen?
- Welke datatype heb je gekozen?
- Waarom past dit datatype bij jouw veld?

Stap 2 – Read (Overzicht)

- Maak een overzichtspagina.
- Toon alle velden uit de database.
- Toon het totaal aantal meldingen boven de tabel.
- Voeg knoppen toe voor toevoegen, wijzigen en verwijderen.

Extra uitdaging

Bedenk zelf één extra functie voor het overzicht.

Voorbeelden:

- Sorteren op naam
- Sorteren op minuten te laat
- Kleurregels voor ernstige meldingen
- Totaal aantal minuten te laat tonen
- Laatste meldingen bovenaan

Je kiest zelf welke functie je bouwt.

Stap 3 – Create (Toevoegen)

- Maak een formulier.
- Sla gegevens op met een prepared statement.
- Gebruik `prepare()` en `execute()`.

Verplichte invoercontrole

- Geen lege velden.

- Minuten moet groter zijn dan 0.
- Minuten mag niet groter zijn dan 240.
- Studentnaam moet minimaal 2 tekens bevatten.

Extra uitdaging (verplicht)

Bedenk zelf nog één extra validatie.

Voorbeeld:

- Klas moet beginnen met "SD"
- Reden moet minimaal 10 tekens bevatten
- Datum mag niet in de toekomst liggen

Stap 4 – Delete

- Voeg een verwijderknop toe.
- Vraag eerst om bevestiging.
- Toon de gegevens van de melding voordat deze verwijderd wordt.

Extra uitdaging (verplicht)

Laat op de bevestigingspagina duidelijk zien welke student verwijderd gaat worden.

Voorkom dat een gebruiker zomaar een niet-bestaand ID kan verwijderen.

Stap 5 – Update

- Maak een wijzigpagina.
- Controleer of het ID bestaat.
- Toon foutmeldingen als het ID ontbreekt.
- Vul bestaande gegevens automatisch in.
- Werk de gegevens bij met een UPDATE-query.

Extra uitdaging (verplicht)

Toon bovenaan de pagina welke melding wordt aangepast.

Bijvoorbeeld:

```
Melding #17 wijzigen
Student: Max Janssen
```

? Veiligheid

Je moet kunnen uitleggen:

- Waarom `prepare()` veiliger is dan een gewone query.
- Wat SQL-injection is.
- Waarom invoercontrole belangrijk is.

? AI-verantwoording

Maak een bestand `ai-gebruik.txt`.

Beschrijf hierin:

- Welke AI-tool(s) je hebt gebruikt.
- Waarvoor je AI hebt gebruikt.
- Welke code je zelf hebt aangepast.
- Welke fout je bent tegengekomen en hoe je die hebt opgelost.

Minimaal 150 woorden.

? Mondelinge controle

Tijdens de beoordeling kiest de docent willekeurig drie onderdelen uit jouw code.

Je moet kunnen uitleggen:

- Wat de code doet.
- Waarom je deze oplossing hebt gekozen.

- Hoe de database erbij betrokken is.

? Inleveren

1. Screenshot overzichtspagina.
2. Screenshot invoerpagina.
3. Screenshot wijzigpagina.
4. SQL-export.
5. Alle PHP-bestanden.
6. Reflectieverslag.
7. ai-gebruik.txt.

? Puntentelling

Punten	Onderdeel
5	Databaseontwerp inclusief eigen extra veld.
5	Read-overzicht werkt correct.
10	Nette vormgeving en gebruiksvriendelijkheid.
10	Create inclusief verplichte validaties.
5	Eigen extra validatie.
5	Delete inclusief bevestiging en foutafhandeling.
10	Update inclusief foutafhandeling.
10	Zelfgekozen extra functie.
10	Codekwaliteit, PDO, prepare(), execute().
30	Uitleg code

Eisen

1. Minimaal 75 totaalpunten, EN
2. Minimaal 20 van de 30 punten voor Uitleg code.

Revision #21
Created 2025-06-13 20:59:46 UTC by Max

Updated 2026-07-10 09:49:16 UTC by Max