

# Challenges

- [Takenlijst uit database met volgorde](#)

# Takenlijst uit database met volgorde

## Wat moet je weten en kunnen?

Deze Challenge gaat over Front End en Back End en hoe je dat goed kan laten samenwerken.

Je moet HTML, CSS, PHP, PDO, Databases, SQL, JavaScript en DOM(-manipulatie) beheersen. Al deze modules heb je voldaan!

## Eisen

1. Je maakt **wel** gebruik van HTML, PHP, PDO en JS.
2. Je mag **wel** gebruik maken van Bootstrap.
3. Je maakt **geen** gebruik van een PHP-framework zoals Yii of Laravel.

## Wat ga je maken?

Je krijgt een prototype die er als volgt, uit ziet:

### Takenlijst / Projectplanning

|   | Taak                          | Uren | Deadline   | Prioriteit | Opmerking  |
|---|-------------------------------|------|------------|------------|--|
| 1 | Project plannen               | 5    | 2024-05-10 | Hoog       | Projectplan finaliseren                          |
| 2 | Onderzoek                     | 8    | 2024-05-15 | Gemiddeld  | Onderzoek code debuggen en proberen te begrijpen |
| 3 | Database maken en vullen      | 10   | 2024-05-20 | Hoog       | Een prototype ontwerpen                          |
| 4 | Aanpassen prototype naar CRUD | 20   | 2024-05-25 | Hoog       | HTML deel van de tabel omzetten naar PHP         |
| 5 | Testen                        | 15   | 2024-05-30 | Gemiddeld  | Aangepast prototype testen                       |
| 6 | Documentatie                  | 10   | 2024-06-05 | Laag       | Commentaar in code plaatsen                      |
| 7 | Inleveren                     | 5    | 2024-06-10 | Hoog       | Het project inleveren en demonstreren aan docent |
| 8 | Opslaan nieuwe volgorde       | 20   |            | Gemiddeld  | Onderzoeken hoe en uitvoeren (lastig)            |

Dit is een tabel met een takenlijst voor een project.

De laatste taak (nummer 8) staat niet op de juiste plaats staat, die moet vlak achter taak 4 staan.

Met het proto-type [takenlijst.html](#) kun je de code proberen. Je kunt de rijen van de tabel verslepen, probeer maar!

In dit proto-type staat statische data (komt niet uit de database).

## Taak 1

Verander het prototype zodat de data uit een **MySQL database** komen.

## Taak 2

Verander het prototype zodat een **aangepaste volgorde** wordt opgeslagen in de **database**.

Deze 2de taak is lastiger en kost waarschijnlijk veel meer werk!

Beide taken worden hieronder verder toegelicht.

# Vorbereidingen

1. Zorg ervoor dat het prototype werkt. Probeer de code zo goed mogelijk te begrijpen. Bestudeer de code en plaats (zelf) commentaar in de code.
2. Maak je ontwikkelomgeving klaar. Hernoem het prototype van .html naar .php. Maak ene projectfolder aan, plaats het php bestand daarin en open de project-folder met VCS.
3. Als je iets begrijpt vraag ChatGPT dan om uitleg.

## Taak 1, database

1. Maak een database aan.
2. Maak in deze database een tabel aan waarin de taken komen te staan.
3. Zet alle kolommen die je in de tabel ziet in de database.
4. Voeg een primary key toe aan de database en vergeet deze niet op "Auto Increment" te zetten.  
ChatGPT prompt:  
*Leg uit wat "auto increment" is bij de definitie van row in een database tabel in MySQL.*
5. Plaats de testdata zoals die in het prototype staat in de database.
6. Maak een *database.php* bestand aan waarin je het user-id en wachtwoord voor de database plaatst. Dit bestand gebruik je om te *includen* in het prototype.
7. Pas je prototype stap-voor-stap aan:
  1. Maak een connectie en test deze.
  2. Maak een SQL-statement, voer die uit en test wat daar uitkomt.

3. Voeg één regel toe aan de tabel met de eerste regel uit de database.
4. Maak nu een tweede tabel onder de eerst en plaats daar alle resultaten die uit de database komen in toe.
8. Test alles en pas alles in de tweede tabel goed werkt dan haal je de eerste tabel weg.

De volgorde waarin de taken op het scherm komen is nu nog niet van belang. Het is wel belangrijk dat je de taken nog kunt verslepen en de volgorde op het scherm kan aanpassen. In de volgende stap gaan we de volgorde ook echt opslaan in de database.

## Taak 2, volgorde opslaan.

Dit is een lastige stap en we delen deze stap eerst op in een paar deelstappen.

### sort\_order

*Stap 1, we gaan de sort order in de database vastleggen.*

De volgorde waarin de taken worden getoond ligt op hert moment niet vast.

We gaan dus eerst de database aanpassen en we plaatsen een extra kolom in de tabel waarin de taken staan van het type integer. Vervolgens zetten we nummers in de database en passen we de query aan zodat de taken worden getoond op volgorde van het getal in deze extra kolom.

1. Maak een extra kolom in de tabel waarin de taken staan met de naam *sort\_order* en het type integer.
2. Zet handmatig via phpMyAdmin de volgorde in deze kolom. De eerst taak krijgt een 1, de tweede een 2, etc. etc.
3. Pas in de SQL-query aan en zorg ervoor dat de resultaten worden gesorteerd om deze kolom *sort\_order*.
4. Test of dit werkt!

## Aanpak, uitleg

*Stap 2, we gaan uitdenken wat onze strategy is.*

Om de volgorde aan te kunnen passen moeten we de database vertellen wat de (nieuwe) volgorde moet zijn.

We kunnen dit doen door de id's in de juiste (nieuwe) volgorde te geven, voorbeeld:

## Volgorde na inladen data uit database

Omdat de data gesorteerd op *sort\_order* uit de database komt, is dit de volgorde waarin de taken op het scherm komen.

| id | Taak | Uren | Deadline | sort_order |
|----|------|------|----------|------------|
| 1  | .... | .... | ....     | 1          |
| 2  | .... | .... | ....     | 2          |
| 3  | .... | .... | ....     | 3          |
| 6  | .... | .... | ....     | 4          |
| 4  | .... | .... | ....     | 5          |

## Volgorde na aanpassen volgorde via de UI

Stel we hebben taak id 4 twee plaatsen naar boven geschoven dan staat er nu dit op het scherm. De *sort\_order* klopt niet meer.

| id | Taak | Uren | Deadline | sort_order |
|----|------|------|----------|------------|
| 1  | .... | .... | ....     | 1          |
| 2  | .... | .... | ....     | 2          |
| 4  | .... | .... | ....     | 5          |
| 3  | .... | .... | ....     | 3          |
| 6  | .... | .... | ....     | 4          |

## Volgorde na inladen data uit database

We moeten iets maken wat de *sort\_order* weer corrigeert, zodat uiteindelijk dit in de database komt te staan.

| id | Taak | Uren | Deadline | sort_order |
|----|------|------|----------|------------|
|----|------|------|----------|------------|

|   |      |      |      |   |
|---|------|------|------|---|
| 1 | .... | .... | .... | 1 |
| 2 | .... | .... | .... | 2 |
| 4 | .... | .... | .... | 3 |
| 3 | .... | .... | .... | 4 |
| 6 | .... | .... | .... | 5 |

Wat gaan dit als volgt aanpakken. We sturen alle id's op de juiste (nieuwe) volgorde naar de code. De code gaat één voor één alle id's af en zet telkens de juiste `sort_order` bij het id. In dit geval zouden we dus 1,2,4,3,6 opsturen naar de code. De code maakt dan sql query's die er als volgt uitzien:

```
update <tabel> set sort_order = 1 where id = 1
update <tabel> set sort_order = 2 where id = 2
update <tabel> set sort_order = 3 where id = 4
update <tabel> set sort_order = 4 where id = 3
update <tabel> set sort_order = 5 where id = 6
```

De volgende keer als we dan weer alle regels uit de database halen, komen ze op de juiste volgorde terug omdat ze worden gesorteerd op `sort_order`.

Tot zover de aanpak, de strategie, nu de implementatie (=uitvoering).

## ID's aan de frontend toevoegen

*(begin hier pas aan als je het vorige stuk hebt gelezen en snapt)*

We moeten aan de frontend kant (in de browser) dus gaan bijhouden welke regel welk id heeft. Dat de we als volgt:

1. Pas de SQL-query aan zodat je ook de id's (primaire keys) van elke taak weet.
2. Zet in elke `<tr>` een extra attribuut *database* en zet daarin het id van het de taak, bijvoorbeeld.

```
<tr database="2" index="2" draggable="true" ondragstart="start(event).....>
```

3. Test of dit werkt door de pagina broncode te inspecteren en te zien of de id's goed zijn ingevuld.

## Form

Om de aanpassingen naar de server te sturen maken we een heel klein (htm) formulier (form).

Het form heeft twee elementen:

1. Een hidden field *sortOrder* waarin de id in de juiste volgorde komen.
2. een save knop waar we een JavaScript mee opstarten.

Om goed te kunnen testen laten we het hidden veld eerst even op niet-hidden staan.

Voer de volgende stappen uit:

1. Maak een form waarbij de *action* is *update\_order.php*. Dit wordt een nieuw bestand (komt later).
2. Maak een knop en *onclick* voer het JavaScript functie *update()* uit.  
`<button type="button" class="btn btn-primary btn-sm" onclick="update();">Save</button>`
3. Maak een functie *update()* en zet als enige in de functie;  
`console.log('Hallo!');`
4. Test of je functie wordt aangeroepen als je de knop indrukt.

## Vullen hidden field

Maak nu JS code dat van alle tr's op jouw pagina de waarde van het attribute *database* opvraagt. Dit zijn alle id's. Zet al deze id's in een rij met komma's ertussen en plaats deze in het (nog niet hidden) field *sortOrder*.

Omdat je het veld kan zien controleer je of jouw JS-code goed werkt.

Gebruik ChatGPT, maar stel gerichte vragen en omschrijf de situatie goed. Probeer geen code te geven, maar probeer de vraag heel goede te stellen. Dit geeft de meeste kans op een goed antwoord, bijvoorbeeld.

### ChatGPT prompts

Hoe kan ik in JavaScript van alle regels van een tabel (tr) in html de id's opvragen en deze in een string zetten met komma's ertussen.

Hoe kan ik met JavaScript, een string in een form veld zetten. Het form heeft het id XXX en het veld heeft een id YYY.

Test of het werkt en of de id's in het veld verschijnen. Ga niet verder als dit niet het geval is.

# Order.php

Verander het JS script en plaats aan het eind een commando zodat het form wordt ge-submit.

Vraag eventueel aan ChatGPT hoe dat moet.

1. Maak het bestand update.php aan toon op het scherm de string met alle id's. Op deze manier weet je zeker dat alle id's juist worden verzonden.
2. Je kunt het form op de andere pagina nu *hidden* maken door het enige veld op `type="hidden"` te zetten.
3. Maak een verbinding met de database en *include* daarbij het veel eerder gemaakte *database.php* bestand.
4. Maak nu een loop waarin je telkens een SQL query afdrukt naar het scherm. Deze queries zien er ongeveer zo uit (vervang <tabel> door jouw tabelnaam):

```
update <tabel> set sort_order = 1 where id = 1
update <tabel> set sort_order = 2 where id = 2
update <tabel> set sort_order = 3 where id = 4
update <tabel> set sort_order = 4 where id = 3
update <tabel> set sort_order = 5 where id = 6
```

5. Test nu eerst of het werkt als je deze queries copy and past naar phpMyAdmin en uitvoert.
6. Als dat werkt dan kan je deze query's in plaats van te laten afdrukken naar het scherm, laten uitvoeren op de database.
7. Test of dit werkt. Ga terug naar de pagina waarin je de tabel met taken ziet, verander de volgorde, bewaar dit en ga weer terug naar het lijstje met taken. Is de andere volgorde opgeslagen?
8. Als dat werkt haal dan alle echo's uit de code in order.php en aan het eind van de code maak je een re-direct naar de code waarmee je het lijstje van taken laat zien.
9. Test nog een keer of alles werkt!