6, De tweede web app

In deze les gaan we oefenen met alles wat we tot nu hebben geleerd. We zullen een database met een N:M relatie gebruiken en we zullen zien dat dat niet echt anders is als een 1:N relatie.

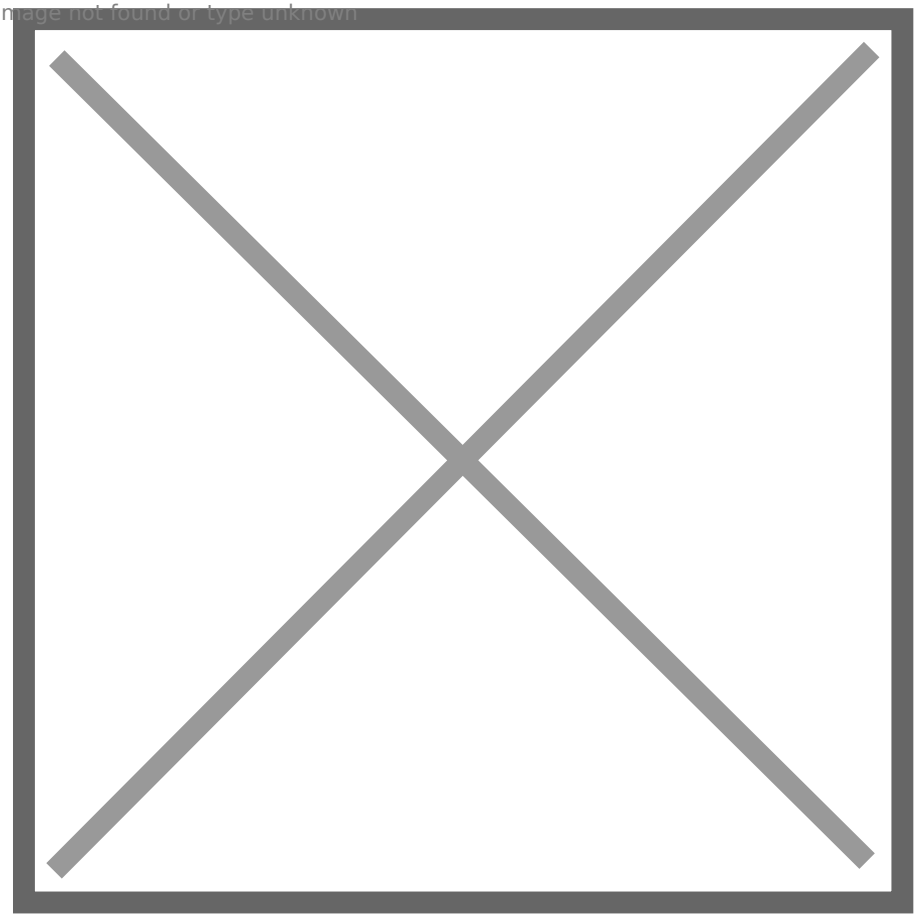
Stap 1, Database

In deze stap gaan we een nieuwe database maken.

Maak een nieuwe database; ga naar deze pagina en installeer de student database.

Je hebt nu een database student met drie tabellen; *student*, *vak* en een koppeltabel *cijfer.* Een student heeft meerdere vakken een één vak kan door meerdere studenten worden gevolgd.

Image not found or type unknown



Stap 2, Nieuw Yii project

In deze stap gaan we een nieuw Yii project maken.

```
composer create-project --prefer-dist yiisoft/yii2-app-basic student
```

Pas de **config/web.php** en de **config/db.php** aan (kijk naar naar de eerste les als je niet meer weet hoe dit precies moet).

Tip: vergeet niet de localhost in de d.php te veranderen in 127.0.01

Start de Yii PHP server.

Stap 2, Models

In deze stap gan we de models maken.

Maak met behulp van Gii zoals we dat in de eerste les hebben gedaan een model van alle *drie* de tabellen. Maak daarna met Gii een CRUD van de student tabel. Kijk nog een keer naar de eerste les als je niet meer weet hoe dit precies moet.

Stap 3, CRUDs

In deze stap gaan we de CRUDs generen

Maak met Gii een CRUD van de student-tabel, de cijfer-tabel en de vak-tabel. Kijk nog een keer naar de eerste les als je niet meer weet hoe dit precies moet.

Stap 4, View

In deze stap gaan we onze eigen view maken waarin we per student alle (behaalde) cijfers laten zien.

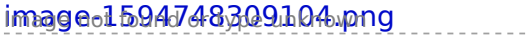
In deze stap zien we een relatie, namelijk een 1:N relatie. Elke student heeft immers 0,1, of meer vakken.

In de vorige lessen hebben we hier mee geoefend, weet je nog? In Eén land werden er 1 of meer talen gesproken. In dat overzicht lieten we alle talen per land zien en nu laten we alle vakken per student zien. Technisch werkt dit op dezelfde manier.

Stap 5, Update Cijfer

In deze stap gaan we een het automatisch gegenereerde form waarmee we een cijfer kunnen aanpassen veranderen.

In de standaard CRUD van de tabel cijfer kun je cijfers updaten. Ga naar http://localhost:8080/cijfer en druk op edit (rechts). Je komt nu in het update-scherm van de tabel cijfer.



Nu zie je alleen de gegevens uit de tabel *cijfer*: je zou liever de naam van de student en de naam van het vak zien. Dat gaan we fixen!

Allereerst kijken we naar de URL: http://localhost:8080/cijfer/update?id=0

Om te bepalen welke view we moeten aanpassen volgend de routeringsregels. We gaan naar de *CijferController.php* file en zoeken naar de *actionUpdate*. We zien dat aan het einde van deze function (method) de view update wordt aangeroepen:

```
return $this->render('update', [ 'model' => $model, ]); }
```

De update view staat in *views/cijfer/update.php*, maar als je die opent, dan zie je heel weinig code. Dat komt, omdat de *update* en *insert* heel veel op elkaar lijken. Het invulscherm (form) is namelijk voor beide hetzelfde. En beide views (*update.php* en *insert.php*) roepen dus beiden *_form.php* aan. Als we in *_form.php* wijzigingen maken, dan gelden die dus voor update en voor insert.

We willen dat het scherm er ongeveer zo gaat uitzien:

image-1594750980851.png
Image not found or type unknown

We kunnen in dit form alleen het cijfer aanpassen; de naam van de student en de naam van het vak kunnen hier niet worden aangepast. Het cijfer staat overigens als 100-tal in de database. In dit voorbeeld staat 85 en dat staat voor een 8.5.

<video uitleg - met gebruik van dd>

Stap 6, De link tussen overzicht en update

Nu moeten we een link maken tussen ons overzichtsscherm en de juisten update pagina. Het moet je moet op het vak/cijfer van een student kunnen klikken en dan moet jet juiste update scherm worden getoond.

Als we naar de link kijken die wordt gebruikt om een cijfer te updaten dan zien we dat daar een id wordt meegegeven. Dit is het ID van de tabel cijfer, de primary key. Om dus naar het juiste update-scherm te kunnen springen hebben we het ID van de *cijfer* tabel dus nodig.

Ons overzicht *overzicht.php* staat in de student directory.

Om zeker te weten dat we het juiste ID gebruiken, maken we een tijdelijke aanpassing in het overzicht en drukken het ID van de cijfer tabel af.

Opdracht (1): druk na het *cijfer* in student/overzicht.php het id af van de koppeltabel waarin het cijfer voor het vak staat. Zie voorbeeld hieronder.

image-1594752066080.png
Image not found or type unknown

We hebben nu alle informatie om van het cijfer een hyperlink te maken naar het juiste update scherm.

Opdracht (2): maak van het cijfer dat wordt afgedrukt in student/overzicht.php een hyperlink (<a href=....), zodat als je op het cijfer klikt, het juiste en zonet gemaakte update scherm verschijnt.

Het skelet van de code om de hyperlink af te drukken:

```
<?php $url='/cijfer/update?id='.XXXXXX; ?>
<a href=<?=$url?> >
  <?= number_format(($cijfervak->cijfer)/10,1) ?>
</a>
```

Om het overzichtelijk te houden maak je eerst de variabele $url met de juisten URL en maak je daarna de hyperlink. De XXXXXX moet nog worden vervangen door het juiste id (van de tabel vak). Deze heb je bij de vorige opdracht al laten afdrukken.

--

Revision #18
Created 11 July 2020 23:21:44 by Max
Updated 23 March 2021 13:48:54 by Admin