

Favorite Code Snippets

Header.php

Dit is de code voor het navigatiemenu. Ik gebruik hiervoor de vaktabel in mijn database. Zo hoef ik niet elke pagina apart te programmeren. In plaats daarvan wordt de vak.php of sprint.php ingeladen (afhankelijk van de button waarop geklikt wordt), met de aan die button gekoppelde id. Met die id wordt de bijbehorende informatie opgehaald uit de database.

```
<title>Portfolio</title>
<link rel="stylesheet" type="text/css" href="style.css">

<script src="https://cdn.ckeditor.com/ckeditor5/11.1.1/classic/ckeditor.js"></script>
</head>

<body>
  <div class="header">
    <nav>
      <ul>
        <li><a <?php if (basename($_SERVER['PHP_SELF']) == 'index.php') echo "class='active'"; ?> href="index.php">HOME</a></li>

        <?php foreach ($menudata as $menurow) {?>
          <li>
            <a <?php if (isset($vakdata)){ if ($menurow['naam'] == $vakdata['naam']) echo "class='active'"; } ?>
              href="vak.php?id=<?=$menurow['vakid']>?><?=$menurow['naam']>?></a>
            <ul class="dropdown">
              <li><a <?php if (isset($sprint)){if (($menurow['naam'] == $vakdata['naam']) && ($sprint == 1)){ echo "class='active'"; }} ?>
                href="sprint.php?id=<?=$menurow['vakid']>?>&sprint=1">Sprint 1</a></li>

              <li><a <?php if (isset($sprint)){if (($menurow['naam'] == $vakdata['naam']) && ($sprint == 2)){ echo "class='active'"; }} ?>
                href="sprint.php?id=<?=$menurow['vakid']>?>&sprint=2">Sprint 2</a></li>

              <li><a <?php if (isset($sprint)){if (($menurow['naam'] == $vakdata['naam']) && ($sprint == 3)){ echo "class='active'"; }} ?>
                href="sprint.php?id=<?=$menurow['vakid']>?>&sprint=3">Sprint 3</a></li>

              <li><a <?php if (isset($sprint)){if (($menurow['naam'] == $vakdata['naam']) && ($sprint == 4)){ echo "class='active'"; }} ?>
                href="sprint.php?id=<?=$menurow['vakid']>?>&sprint=4">Sprint 4</a></li>

              <li><a <?php if (isset($sprint)){if (($menurow['naam'] == $vakdata['naam']) && ($sprint == 5)){ echo "class='active'"; }} ?>
                href="sprint.php?id=<?=$menurow['vakid']>?>&sprint=5">Sprint 5</a></li>
            </ul>
          </li>
        <?php } ?>
      </ul>
    </nav>
  </div>
```

Vak.php

Dit is het standaard format voor alle pagina's op mijn website, behalve de homepage. De informatie wordt opgehaald uit de database door de sql code. Ik heb hierbij ORDER BY gebruikt, zodat de informatie op volgorde van sprint gesorteerd wordt, van laag naar hoog.

```
// show records
$opdrachtsql = "SELECT * FROM opdracht WHERE vakid = '$id' ORDER BY sprint ASC";
$opdrachtquery = $conn->query($opdrachtsql);
$opdrachtdata = $opdrachtquery->fetchall();

include('header.php');
<?>
<div class="opdracht">
  <h1 class="head"><?=$vakdata['naam']?></h1>
  <div class="headtext"><?=$vakdata['toelichting']?></div>

  <?php foreach ($opdrachtdata as $filtervakdata){?>
    <div class="opdrachtcontent">
      <h3 class="subhead">Sprint: <?=$filtervakdata['sprint']?></h3>
      <h3 class="subsubhead"><?=stripslashes($filtervakdata['titel'])?></h3>
      <div><?=stripslashes($filtervakdata['tekst'])?></div>

      <div class="link" >
        <?php switch (mime_content_type( filename: "uploads/".$filtervakdata['bestand'])) {

          case 'image/jpeg': ?> 
          <?php break; ?>

          <?php case 'application/pdf': ?> <a href="uploads/<?=$filtervakdata['bestand']?>" target="_blank">Bekijk hier het eindverslag</a>
          <?php break; ?>

          <?php case 'image/png': ?> 
          <?php break; ?>

          <?php case 'image/gif': ?> 
          <?php break; ?>

          <?php case 'image/jpg': ?> 
          <?php break; ?>

          <?php } ?>

        </div>
      </div>
    <?php } ?>
  </div>
</div>
```

List.php

Dit is de code voor de CMS-pagina op de website. Ik gebruik hier een innerjoin om de twee tabellen uit mijn database aan elkaar te koppelen. Deze heb ik gebruikt zodat ik in de opdracht tabel de vaknaam kan weergeven. In de opdrachttabel heeft elk vak namelijk alleen een id.

De rest van de code gebruik ik om een lijst van de opdrachten te weergeven. Elke opdracht heeft een knop die naar de "Bewerken"-pagina navigeert en een "delete"-knop die naar de (voor de gebruiker onzichtbare) delete-pagina navigeert.

```
<?php
//Stap 1: Verbinding maken
include('connect.php');

//Stap 2: Query maken
$query = $conn->query( statement: "SELECT * FROM opdracht INNER JOIN vak ON opdracht.vakid = vak.vakid");

//Stap 3: Data ophalen
$data = $query->fetchAll();

include('header.php');
?>

<div class="content" style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;">
    <h1 class="head">CMS</h1>
    <table>
        <tr>
            <th class="tableitem">Titel</th>
            <th class="tableitem">Sprint</th>
            <th class="tableitem">Vak</th>
            <th class="tableitem">Bestand</th>
            <th class="tableitem"></th>
            <th class="tableitem"></th>
            <th class="tableitem"></th>
        </tr>
        <?php foreach ($data as $row) {?>
            <tr>
                <td class="tableitem"><?=$row['titel'] ?></td>
                <td class="tableitem"><?=$row['sprint'] ?></td>
                <td class="tableitem"><?=$row['naam'] ?></td>
                <td class="tableitem"><?=$row['bestand'] ?></td>
                <td class="tablebutton"><a class="submit" href="update-form.php?id=<?=$row['id']?>">Bewerken</a></td>
                <td class="tablebutton"><a class="submit" href="delete.php?id=<?=$row['id']?>">delete</a></td>
                <td class="tableitem"></td>
            </tr>
        <?php }?>
    </table>
    <br>
    <a href="add-form.php" class="submit">Toevoegen</a>
</div>
```

Delete.php

Elke opdracht heeft een eigen deleteknop. Als de id van een opdracht correspondeert met de knop waarop geklikt wordt, dan wordt het bestand uit de database verwijderd. Daarna wordt de gebruiker teruggestuurd naar de List.php.

```
<?php
    include('connect.php');

    $id = $_GET['id'];
    // sql to delete a record, use exec() because no results are returned.
    $conn->exec($sql = "DELETE FROM opdracht WHERE id='$id'");
    echo "Record deleted successfully";
    //redirect back to list
    header( string: "Location: list.php");
    exit;
```

Add.php

De informatie uit het add-form.php bestand wordt door deze code toegevoegd aan mijn database. Daarna wordt de gebruiker teruggestuurd naar de List.php.

```
<?php
include("connect.php");
include("upload.php");

// Evaluates as true because $var is set
if (isset($_POST["titel"])) {

    $titel = addslashes($_POST["titel"]);
    $tekst = addslashes($_POST["tekst"]);
    $bestand = $_FILES["bestand"]["name"];
    $sprint = $_POST["sprint"];
    $vakid = $_POST["vakid"];

    // sql to insert a record
    $sql = "INSERT INTO opdracht (titel, tekst, bestand, sprint, vakid) VALUES ('$titel', '$tekst', '$bestand', '$sprint', '$vakid')";

    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Bestand uploaden is geslaagd.";
    header( string: "Location: list.php");
}
else {
    echo "Er is iets mis gegaan. Probeer het opnieuw.";
    exit;
}
```

Update-form.php

Dit is het PHP gedeelte van de Update-form. Hier haal ik de geuploade opdracht op uit de database aan de hand van de id van de button die correspondeert met de opdracht waarop geklikt is. De opgeslagen informatie wordt weergegeven op het scherm.

```
<?php
//Stap 1: Verbinding maken
include('connect.php');

//Stap 2: Query maken
$selectquery = $conn->query( statement: "SELECT vakid, naam FROM vak");

//Stap 3: Data ophalen
$selectdata = $selectquery->fetchAll();

$id = $_GET['id'];
$updatequery = $conn->query( statement: "SELECT * FROM opdracht INNER JOIN vak ON opdracht.vakid = vak.vakid WHERE id=$id");
$updatedata = $updatequery->fetchAll();

foreach ($updatedata as $updaterow) {
    $updateitel = $updaterow["itel"];
    $updatetekst = $updaterow["tekst"];
    $updatesprint = $updaterow["sprint"];
    $updatevaknaam = $updaterow["naam"];
    $updatebestand = $updaterow["bestand"];
}

include('header.php');
?>
```

In het html gedeelte heb ik de invoervakken en knoppengeprogrammeerd, waarmee de opdracht aangepast kan worden. Voor het tekstvak heb ik een editorplugin gebruikt. De rest heb ik zelf geprogrammeerd.

```
<div class="content">
  <h1 class="head">Opdracht bewerken</h1>

  <form action="update.php" method="post" id="update-form" enctype="multipart/form-data">

    <div class="lines">
      <p>Titel: <input type="text" name="itel" value="<?php echo $updateitel; ?>"></p>
    </div>

    <div class="lines">Tekst: </div>
      <div style="..."><textarea id="editor" Name="tekst" form="update-form" class="textarea"><?php echo $updatetekst; ?></textarea></div>

    <script>
      ClassicEditor
      .create( document.querySelector( '#editor' ) )
      .then( editor => {
        console.log( editor );
      } )
      .catch( error => {
        console.error( error );
      } );
    </script>

    <div class="lines">
      <p class="string">
        Selecteer bestand:
      </p>
      <input type="file" name="bestand" id="updatebestand">
    </div>
  </form>
</div>
```

```

<div class="lines">
  <p class="string">
    Sprint:
  </p>
  <select class="sprint" name="sprint">
    <option selected="selected"><?php echo $updatesprint; ?></option>
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
  </select>
</div>

<div class="lines">Vak:
  <select name="vakid">
    <option value="false" selected="selected"><?php echo $updatevaknaam; ?></option>
    <?php foreach ($selectdata as $selectrow) { ?>
      <option value="<?php echo $selectrow["vakid"]; ?>"><?php echo $selectrow["naam"]; ?></option>
    <?php } ?>
  </select>
</div>

<input type="hidden" value="<?php echo $id; ?>" name="id">

<input class="submit" type="submit" value="Uploaden">
</form>
</div>
<?php include('footer.html'); ?>
</body>
</html>

```

Update.php

Dit is de code die de database update met de bewerkte informatie. De oude informatie wordt overschreven. Als er geen nieuw bestand geupload is, blijft het vorige bestand in tact. Daarna wordt de gebruiker teruggestuurd naar de List.php.

```
if (isset($_POST["titel"])) {

    $id = $_POST['id'];
    //echo $id;

    $titel = addslashes($_POST["titel"]);
    $tekst = addslashes($_POST["tekst"]);
    $bestand = $_FILES["bestand"]["name"];
    $sprint = $_POST["sprint"];
    $vakid = $_POST["vakid"];

    $bestandquery = "";
    if (!empty($bestand)){
        $bestandquery = "bestand='".$bestand.'";"
    }

    // sql to update a record
    if ($vakid == "false"){
        $sql = "UPDATE `opdracht` SET titel='$titel', tekst='$tekst', $bestandquery sprint='$sprint' WHERE id=$id";
    } else {
        $sql = "UPDATE `opdracht` SET titel='$titel', tekst='$tekst', $bestandquery sprint='$sprint', vakid='$vakid' WHERE id=$id";
    }

    // Prepare statement
    $stmt = $conn->prepare($sql);

    // execute the query
    $stmt->execute();

    // echo a message to say the UPDATE succeeded
    echo $stmt->rowCount() . " De opdracht is succesvol aangepast.";
    header( string: "Location: list.php");
}

else {
    echo "Er is iets mis gegaan. Probeer het opnieuw.";
    exit;
}
```


Upload.php

Dit is de het bestand dat je wilt uploaden opslaat in de "uploads/" map en omzet in een path. Vervolgens wordt het path opgeslagen in de database.

De rest van de code controleert of er wel een bestand geupload is, of het bestand wel een bestand is en of het bestand niet groter is dan 64 mb. Als het bestand aan al deze eisen voldoet, wordt hij opgeslagen en wordt de gebruiker terug gestuurd naar de List.php.

```
<?php
include("connect.php");
if ($_FILES['bestand']['name'] != null) {
    $uploadquery = $conn->query( statement: "SELECT bestand FROM opdracht");
    $uploaddata = $uploadquery->fetchAll();

    $target_dir = "uploads/";
    $target_file = $target_dir . basename($_FILES["bestand"]["name"]);
    var_dump($target_file);
    // var_dump($_FILES);
    $uploadOk = 1;
    $imageFileType = strtolower(pathinfo($target_file, options: PATHINFO_EXTENSION));

    /*$finfo = finfo_open(FILEINFO_MIME_TYPE);
    $mime = finfo_file($finfo, $_FILES['file']['tmp_name']);
    $ok = false;*/

    // Check if image file is an actual image or fake image
    if(isset($_POST["submit"])) {
        $check = getimagesize($_FILES["bestand"]["tmp_name"]);
        if($check !== false) {
            echo "Bestand is een afbeelding. - " . $check["mime"] . ".";
            $uploadOk = 1;
        }
        else {
            echo "Bestand is geen afbeelding.";
            $uploadOk = 0;
        }
    }
}
```

```
// Check file size
if ($_FILES["bestand"]["size"] > 64000000) {
    echo "Sorry, het bestand is te groot. Het bestand mag niet groter zijn dan 64 MB.";
    $uploadOk = 0;
}

// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    echo "Sorry, het bestand is niet geupload.";
}

// if everything is ok, try to upload file
else {
    if (move_uploaded_file($_FILES["bestand"]["tmp_name"], $target_file)) {
        echo "Het bestand ". basename( $_FILES["bestand"]["name"]). " is succesvol geupload.";
    }
    else {
        echo "Sorry, er ging iets fout bij het uploaden van je bestand.";
    }
}
}
```