

## Tabelle mit „Ändern“ und „Löschen“ pro Zeile versehen – Teil 3: PHP, DB inkl. Fehler- bzw. Erfolgsmeldung mit GET-Methode

Öffne die Datei „schueler.php“.

Ziel:

| Vorname | Nachname | Geschlecht | Klasse | Geb.Datum  | Bild  | Aktion   |
|---------|----------|------------|--------|------------|---|--|
| Jakob   | Studeney | m          | 3CK    | 2003-02-17 |  | <input type="button" value="Ändern"/> <input type="button" value="Löschen"/> |

### 1)Bootstrap einbinden

Die Tabelle sollte mit Bootstrap etwas verschönert werden.

Dafür muss man das Bootstrap Starter template auch für diese Datei verwenden. Das einfachste ist es, dies aus der „index.php“ zu kopieren. Im Prinzip geht es um die CSS und JavaScript Dateien, die das „Bootstrap-Feeling“ erst möglich machen.

Kopiere daher aus der index.php in die „schueler.php“ die Zeilen von Beginn an bis inkl. Start des <body>: Füge diese ganz am Anfang ein.

```
1 <!doctype html>
2 <html lang="de">
3 <head>
4   <!-- Required meta tags -->
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1,
7     shrink-to-fit=no">
8
9   <!-- Bootstrap CSS -->
10  <link rel="stylesheet"
11    href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.m
12    in.css" integrity="sha384-
13    Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
14    crossorigin="anonymous">
15  <link rel="stylesheet" href="my.css">
16
17  <title>Schüler-Daten</title>
18 </head>
19 <body>
```

Danach auch den unteren Teil mit den Scripts zu den JS-Dateien, ab dem Kommentar in Zeile 34. Füge diese Kopie ganz am Ende ein.

```
33
34 <!-- Optional JavaScript -->
35 <!-- jQuery first, then Popper.js, then Bootstrap JS -->
36 <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
37   integrity="sha384-
38   J6qa4849bLE2+poT4WnyKhv5vZF5SrPo0iEjwBvK7imGFAV0wwj1yYfoRSJoZ+n"
39   crossorigin="anonymous"></script>
```

```

37 <script
   src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-
   Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
   crossorigin="anonymous"></script>
38 <script
   src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-
   wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl30g8ifwB6"
   crossorigin="anonymous"></script>
39 </body>
40 </html>

```

Nun hat auch die „schueler.php“ Bootstrap „an Bord“.

## 2) Neue Spalte einfügen

Erweitere daher den Tabellen-Kopf (th = table head) um eine Zelle, die ein „colspan“ hat.

```

34 echo "<table border='1'>";
35 echo "<tr><th>Vorname</th><th>Nachname</th><th>Geschlecht</th><th>
   <th>Geb.Datum</th><th>Bild</th><th colspan='2'>Aktion</th></tr>";

```

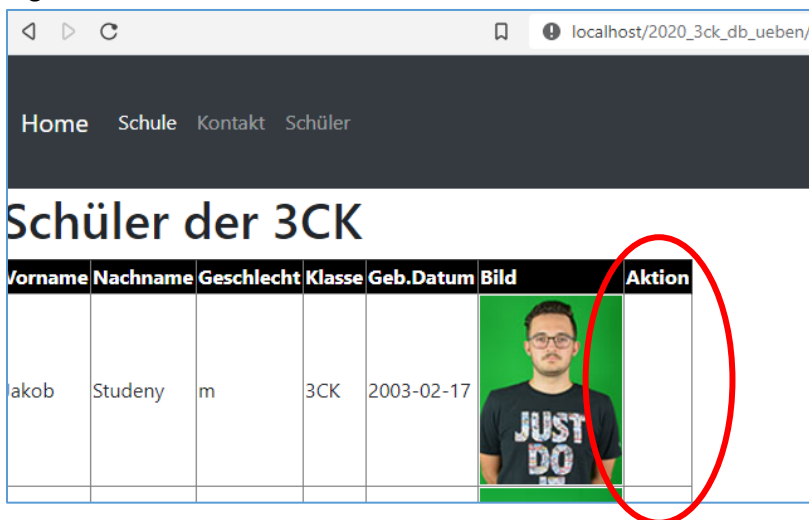
Setze nach dem Bild auch eine leere Zelle, wo später die beiden Buttons eingefügt werden.

```

43 <td><img height='150' wid
   src='data:image/jpeg;base
   </td>
44 <td></td>
45 </tr>";

```

Ergebnis:



Da die Tabelle so weit links klebt, könnte man hier einen Bootstrap-Container einfügen, der automatisch Außen- und Innenabstand bereithält:

```

28
29 <div class="container">
30 <h1>Schüler der 3CK</h1>
31
32 <?php

```

Das Ende zwischen den beiden PHP-Elementen: hier mit Kommentar in Zeile 52

```
51 ?>
52 </div> <!--ende container-->|
53 <?php
54     require "footer.php";
55 ?>
```

### 3) Einfügen der Buttons „Ändern“ und „Löschen“ und die ID der gewählten Reihe als GET Variable in die URL übergeben

In der geschaffenen Zelle erstelle

- eine Ende der PHP, davor aber mit Anführungszeichen das echo nach der <td> beenden
- eine Weiterleitung in HTML und er Übergabe mit dem Fragezeichen einleiten, dann
- die ID-Übergabe wieder in PHP (eckigen Klammern) und
- den entsprechenden Button als Bootstrapklasse.
- Dann wieder Start von PHP und Weiterführung vom „echo“

Beachte die Starts und Ende von PHP ☺ etwas kompliziert ☺

Die beiden Buttons sollen unterschiedlich zugreifen:

- Die Änderung wird in einer neuen Datei „schueler-update.php“ liegen
- Das Löschen wird in einer eigenen Datei, die in den „includes“ liegt bearbeitet. Der Name ist „prozess.php“

```
45         <td>"
46         ?>
47         <a href="schueler-update.php?aendern=<?php echo $zeile['s_id']; ?>"
48           class='btn btn-info'>Ändern</a>
49         <?php
50         echo "
51     </td>
52 </tr>"
```

<a href="schueler-update.php?aendern=<?php echo \$zeile['s\_id']; ?>" class='btn btn-info'>Ändern</a>

Ergebnis:

| Vorname | Nachname | Geschlecht | Klasse | Geb.Datum  | Bild  | Aktion |
|---------|----------|------------|--------|------------|---|--------|
| Jakob   | Studený  | m          | 3CK    | 2003-02-17 |  | Ändern |

Nun folgt noch der zweite Button in einer anderen Farbe. Kopiere den gerade erstellten Code und ändere die wesentlichen Teile ab:

```

45         <td>"
46         ?>
47         <a href="schueler-update.php?aendern=?php echo $zeile['s_id']; ?>"
           class='btn btn-info'>Ändern</a>
48         <a href="includes/prozess.php?loeschen=?php echo $zeile['s_id']; ?
           >" class='btn btn-danger'>Löschen</a>
49         <?php
50         echo "
51         </td>
52         </tr>";
53     }
54     echo "</table>";

```

| Vorname | Nachname | Geschlecht | Klasse | Geb.Datum  | Bild  | Aktion   |
|---------|----------|------------|--------|------------|---|--|
| Jakob   | Studeney | m          | 3CK    | 2003-02-17 |  | <input type="button" value="Ändern"/> <input type="button" value="Löschen"/> |

#### 4) Löschen eines Datensatzes

Dabei muss garantiert sein, dass

- überprüft wird, dass der DELETE-Button geklickt wurde und
- eine neue Variable „\$id“ erstellen, die aus der Übergabe per GET geholt wird,
- genau die übergebene ID aus der GET['loeschen']-Variable gelöscht wird
- Dafür ist ein Datenbankzugang nötig.

Erstelle eine neue Datei namens „prozess.php“ im Ordner „includes“.

Am Beginn muss man eine Verbindung zur Datenbank herstellen, mittels „require“:

```

Offene Dateien
• prozess.php
Erste Schritte
> screenshots
index.html
main.css
1  <?php
2  require 'verbindung.php';
3
4  if (isset($_GET['loeschen'])) {
5      $id = $_GET['loeschen'];
6
7      $sql = "DELETE FROM schueler WHERE s_id=$id";
8      $erg = $con->query ($sql)
9          or die("Fehler bei der Speicherung.");
10

```

Es würde funktionieren, aber der User sollte wieder zurückkommen zur Übersicht der Schüler, und sogar mit einer positiven Meldung, dass alles funktioniert hat. Natürlich sieht der Benutzer auch, dass der Datensatz nun nicht mehr vorhanden ist, aber eine zusätzliche Meldung ist Standard.

Daher erstelle eine Weiterleitung mit einer „header“-Funktion:

```

12
13     header("Location: ../schueler.php");
14 }
15 ?>

```

## Erfolgs- oder Fehlermeldung anzeigen lassen:

Typische Fehler werden angezeigt, damit der Benutzer sieht, dass etwas nicht funktioniert hat, bzw. auch dass es funktioniert hat.

Beispiel: willhaben.at



Direkt an die URL kann man Information mitgeben, wenn man ein ? (Fragezeichen) anhängt. Jedoch wird diese Information oben in der URL sichtbar. Daher wird man keinesfalls das Passwort mit übergeben. Das würde man dann ja auch im Klartext lesen können.

Daher füge

- nach dem Fragezeichen ein beliebig passendes Wort ein, hier „geloescht“. Dieses Wort benötigt man dann später nochmals in der Zielformat für die Ausgabe.
- dann ein Istgleichzeichen und
- dann ein beliebiger Text inklusive Leerzeichen zwischen den Wörtern, wenn benötigt.

```
13     header("Location: ../schueler.php?geloesch=Der Schüler wurde erfolgreich  
14     gelöscht");  
15     ?>
```

Da wir wissen, dass in der Zielformat, hier „schueler.php“ das Framework Bootstrap verwendet wird, kann man hier gleich ein schönes Design mitgeben. Dafür nutzt man eine typische Bootstrap-Klasse.

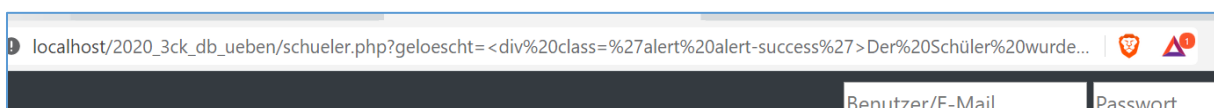
Diese wird um den gewünschten Text gelegt, also gleich nach dem Istgleichzeichen begonnen.

Als Alert-Farbe kann man aus den verschiedenen Bootstrapfarben wählen, meistens „success“ für grün und „danger“ für rot:

```
header("Location: ../schueler.php?geloesch=<div class='alert alert-  
success'>Der Schüler wurde erfolgreich gelöscht</div>");
```

Beachte die Abwechslung der Anführungszeichen, hier beim alert die einfachen Anführungsstriche, da vorher und nachher die doppelten schon vorhanden sind.

Leider sieht man in der Zielformat „schueler.php“ zurzeit nur in der URL diese Angaben, weil sie ja als GET-Variable von Datei zu Datei weitergegebene wurden.



Es hat funktioniert, wir sind in der Zielformat „schueler.php“ gelandet und haben die Info geliefert bekommen.

### Ausgabe der Erfolgsmeldung:

Diese Info kann man aber mittels PHP herausholen und in die Zielfeile ausgeben lassen, das relativ einfach sogar:

- Nun muss man in dem Formular, indem die Fehlermeldung erfolgen soll, diese „message“, die bei uns den Namen „geloescht“ hat, als Code hineinpflanzen.
- Öffne daher wieder die Datei „schueler.php“.
- Eine gute Position ist nach der Überschrift und vor dem ersten Input-Feld.

Füge passende <br> ein, damit genug Platz ist. Danach folgt folgender PHP-Code, der mittels \$\_GET die Information aus der URL ausliest und hier mit „echo“ angezeigt wird.

Mit \$\_GET, so ähnlich wie bei \$\_POST, kann man aus einem anderen Dokument Daten empfangen und auswerten. Das ist eigentlich eine sehr gute Sache. Wir werden das noch öfter brauchen!

Zuerst wird mit einer

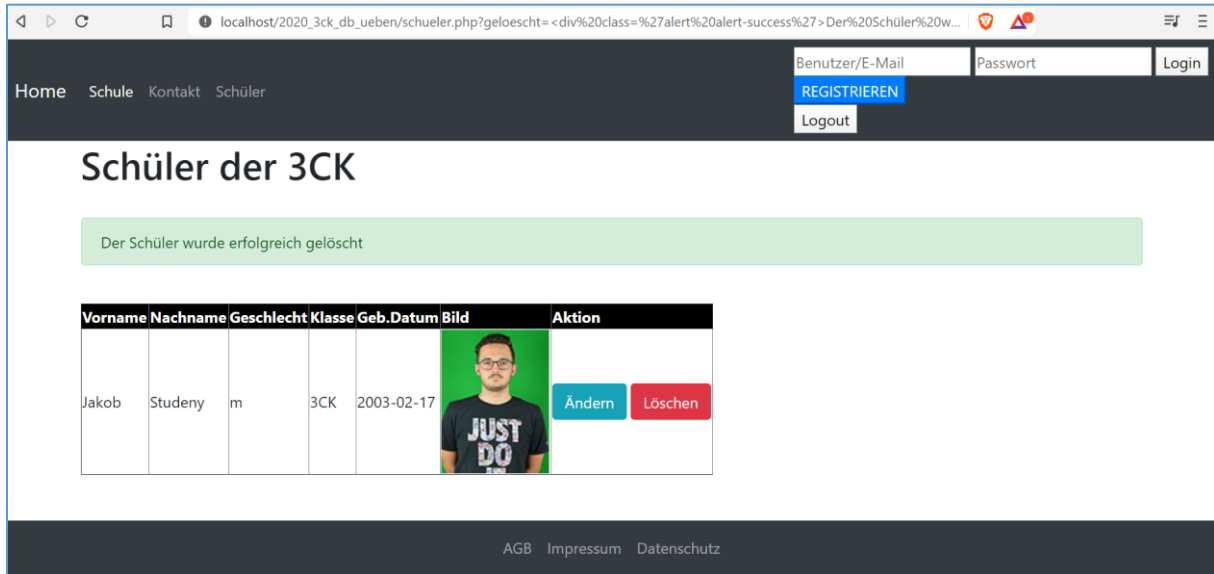
- IF-Abfrage die Bedingung gestellt, ob überhaupt eine GET-Information vorhanden ist. Dazu verwendet man die Funktion „isset()“.
- Dann wird der Inhalt der GET in eine neu Variable, hier \$geloescht gespeichert
- Danach wird diese Variable einfach nur ausgegeben. Das Design wurde bereits, inkl. Farbe „danger“ in der Datei festgelegt, welche die Daten hierher senden. Das ist ebenfalls die „schueler.php“ beim Button „Löschen“.

```
30 <h1>Schüler der 3CK</h1>
31 <br>
32 <?php
33     if(isset($_GET['geloescht'])) {
34         $geloescht = $_GET['geloescht'];
35         echo $geloescht;
36     }
37 >
38 <br>
39 <?php
40 $erg = $con->query("SELECT * FROM schueler");
```

Code:

```
<br>
<?php
    if(isset($_GET['geloescht'])) {
        $geloescht = $_GET['geloescht'];
        echo $geloescht;
    }
?>
<br>
```

Ergebnis:




## 5)Ändern eines Datensatzes

Dazu muss man

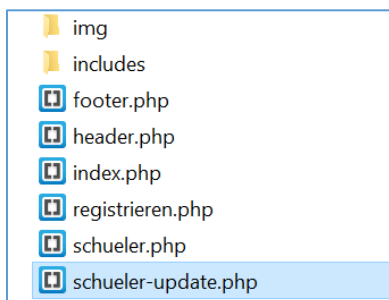
- Wenn der Button „Ändern“ geklickt wurde
- die passende ID auswählen und den passenden Datensatz auswählen
- den Datensatz in ein Formular geben, um es ändern zu können
- mit dem in dem Formular vorhandenen „Ändern“-Button das Update in der Datenbank durchführen

Durch den schon erstellten „Ändern“-Button und dem vorhandenen Code dort, wird in der URL bereits die ID mitgegeben und angezeigt. Klicke auf den Button und überprüfe es:

 localhost/2020\_3ck\_db\_ueben/schueler-update.php?aendern=1

## 5a)Erstelle eine neue Datei „schueler-update.php“

Dafür erstelle eine neue Datei mit dem Namen „schueler-update.php“, das auf der Ebene von der Datei „schueler.php“ liegen soll. (Wegen dem besseren Zugriff in der „header.php“, da ein Platz in „includes“ zu Schwierigkeiten in der Navigation führen würde.)



- Beginne mit den <?php Eröffnung und darunter sofort die Verbindung zur Datei, die die Verbindung zur Datenbank beinhaltet.

- Zuerst wird überprüft, ob der Button die GET-Info abgegeben hat, und daher, ob er gedrückt wurde. Danach wird die enthaltene ID in eine Variable „\$id“ hineingegeben. Diese Variable kann gerne denselben Namen haben wie ein paar Zeilen darüber im „Löschen“-Code. Das stört nicht.
- Danach wird alles, was in der Datenbank zu dieser ID passt, abgeholt, nämlich mit SELECT.

```

Offene Dateien
schueler.php
schueler-update.php

Erste Schritte
screenshots
index.html
main.css

1 <?php
2 require "includes/verbindung.php";
3 |
4 //nun folgt der Ändern-Button
5 if (isset($_GET['aendern'])) {
6     $id = $_GET['aendern'];
7     $sql = "SELECT * FROM schueler WHERE s_id=$id";
8     $result = $con->query ($sql)
9         or die($mysqli->error()); //ist besser als Fehler bei der Speicherung

```

```

if (isset($_GET['aendern'])) {
    $id = $_GET['aendern'];
    $sql = "SELECT * FROM schueler WHERE s_id=$id";
    $result = $con->query ($sql)
        or die($mysqli->error()); //ist besser als Fehler bei der Speicherung
}

```

Nun sollen alle Daten des Datensatzes mit der Funktion „fetch\_array()“ geholt werden und in neuen Variablen gespeichert werden. Weil eine direkte Darstellung in einer Tabelle bringt hier nichts. Die Variablen sollen viel mehr hier in einem Formular landen.

Danach sollte man auf dieser Seite, die Daten aus dem geholten Datensatz in die <input>-Felder zu erstellenden Formulars übergeben. Dieser Schüler-Datensatz wird somit angezeigt und kann nun überschrieben werden. Dann wird er mit \$\_POST wieder in die Datenbank geschrieben.

Das neue Formular soll den Großteil des alten Formulars abbilden, inkl. Benutzername und E-Mail.  
**Nicht das Passwort.**

```

21     $row = $result->fetch_array();
22     $vornameNeu = $row['vorname'];
23     $nachnameNeu = $row['nachname'];
24     $geschlechtNeu = $row['geschlecht'];
25     $klasseNeu = $row['klasse'];
26     $gebNeu = $row['geb'];
27     $bildNeu = $row['bild'];
28     $benutzernameNeu = $row['benutzername'];
29     $emailNeu = $row['email'];
30 }

```

```

$row = $result->fetch_array();
$vornameNeu = $row['vorname'];
$nachnameNeu = $row['nachname'];
$geschlechtNeu = $row['geschlecht'];
$klasseNeu = $row['klasse'];
$gebNeu = $row['geb'];
$bildNeu = $row['bild'];

```

```
$benutzernameNeu = $row['benutzername'];
$emailNeu = $row['email'];
```

### **5b)Erstelle ein neues Formular darunter**

Der Benutzer soll in einem übersichtlichen Formular seine Daten betrachten können und hier dann ändern können.

Kopiere alles aus der „registrieren.php“, auch die Bootstrap-Codes und speichere es unterhalb des gerade erstellten PHP-Codes. Ändere dann darin einiges.

```
18     $benutzernameNeu = $row['benutzername'];
19     $emailNeu = $row['email'];
20 }
21 ?>
22
23 <!doctype html>
24 <html lang="de">
25 <head>
26 <!-- Required meta tags -->
27 <meta charset="utf-8">
28 <meta name="viewport" content="width=device-width, initial-scale=1, sh
to-fit=no">
29
30 <!-- Bootstrap CSS -->
```

Ändere folgende Codeelemente:

- die <h1> auf „Daten ändern“
- die action, wohin die Form die Daten übermitteln soll auf „includes/aendern.php“
- man benötigt keine „placeholder“. Lösche alle.

```
43 <h1>Daten ändern</h1>
44 <br>
45 <form action="includes/aendern_insert.php" method="post"
enctype="multipart/form-data">
46 <div class="form-group">
47 <input type="text" name="vorname" class="form-control">
48 </div>
```

- die Überschrift im Mittelteil kann entfernt werden

```
63 </div>
64 <br>
65
66 <div class="form-group">
67 <input type="text" name="benutzername" class="form-control">
68 </div>
```

- lösche den <div>-Bereich für das Passwort. Das soll hier nicht angezeigt werden.
- Ändere den Namen des Buttons auf „aendern-submit“ und auch den Inhalt des Buttons auf „Daten ändern“.

```

72     <br>
73     <div class="form-group">
74         <button type="submit" name="aendern-submit" class="btn btn-success btn-
            block">Daten ändern</button>
75     </div>
76 </form>

```

Da aber noch keine Daten der betreffenden ID vorhanden sind, folgen diese jetzt.

Dazu muss man in jede `<input>` einen „value“ einfügen. Dieser hat mit PHP-Code „echo“ die Ausgabe der neuen Variablen zu erledigen. Somit sind die Daten des betreffenden Datensatzes in dem Formular vorhanden und können überschrieben und abgeändert werden.

Beispiel bei Vorname:

```

<input type="text" name="vorname" value="<?php echo $vornameNeu; ?>"
class="form-control">

```

Eine erste Überprüfung ist soweit erfolgreich, jedoch mit einer Fehlermeldung, da ja kein „Ändern“-Button geklickt wurde und somit keine ID ausgewählt ist. Also kein Problem soweit.

Kopiere diesen „Value“-Code und füge ihn in den anderen `<input>` ein und ändere ihn jeweils passend um.

value="<?php echo \$vornameNeu; ?>"

```

45 <form action="includes/aendern_insert.php" method="post"
    enctype="multipart/form-data">
46     <div class="form-group">
47         <input type="text" name="vorname" value="<?php echo $vornameNeu; ?>"
            class="form-control">
48     </div>
49     <div class="form-group">
50         <input type="text" name="nachname" value="<?php echo $nachnameNeu; ?>"
            class="form-control">
51     </div>
52     <div class="form-group">
53         <input type="text" name="geschlecht" value="<?php echo $geschlechtNeu; ?>"
            class="form-control">
54     </div>

```

Ergebnis:

Hat geklappt. Jedoch das Bild funktioniert nicht.

Änderung:

Das Bild aus der Datenbank soll eigenständig angezeigt werden. Danach soll ein Hinweis stehen, dass, auch wenn das Bild stimmt, auf jeden Fall ein weiteres Mal ein Bild ausgewählt werden muss. Sonst gibt es nämlich Probleme beim Speichern in die Datenbank.

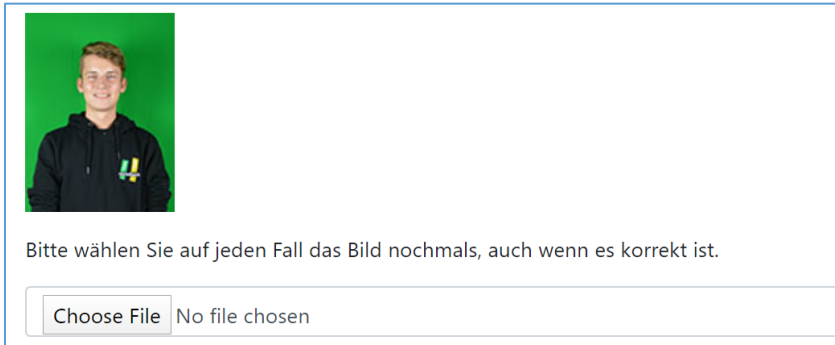
Darunter wird die Prozedur der Bildauswahl einfach aus dem Formular „registrieren.php“ übernommen und hier hinein kopiert (Zeile 68-70).

```
60         </div>
61         <!--Achtung wegen Bild-->
62         <div class="form-group">
63             <?php echo "<img height='150' width='auto'
64                 src='data:image/jpeg;base64,'" . base64_encode($bildNeu) . "' />" ?>
65         </div>
66         <div class="form-group">
67             <p>Bitte wählen Sie auf jeden Fall das Bild nochmals, auch wenn es
68                 korrekt ist.</p>
69         </div>
70         <div class="form-group">
71             <input type="file" name="bild" class="form-control"
72                 enctype="multipart/form-data">
73         </div>
74     </div>
```

**Code:**

```
<!--Achtung wegen Bild-->
    <div class="form-group">
        <?php echo "<img height='150' width='auto'
        src='data:image/jpeg;base64,'" . base64_encode($bildNeu) . "' />" ?>
    </div>
    <div class="form-group">
        <p>Bitte wählen Sie auf jeden Fall das Bild nochmals, auch wenn es korrekt ist.</p>
    </div>
    <div class="form-group">
```

```
<input type="file" name="bild" class="form-control" enctype="multipart/form-data">
</div>
```



### Daten ändern


Kevin

Eckelhart

m

3CK

11/02/2002



Bitte wählen Sie auf jeden Fall das Bild nochmals, auch wenn es korrekt ist.

Choose File No file chosen

Kevin200

kevin-e@gmail.com

### 5c) Änderungen absenden

Nun muss man die Daten aus dem Formular, die abgesendet werden, in der Datei „aendern\_insert.php“ entgegen nehmen und in die Datenbank updaten.

```
45 ▼ <form action="includes/aendern_insert.php" method="post"
      enctype="multipart/form-data">
```

Der Datenbank muss man aber mitteilen, dass das jetzt kein neuer Datensatz ist, sondern einen alten, mit einer bestimmten ID überschreiben soll (update).

Daher muss man die ID mitnehmen. Diese kann man im Formular mit einem „versteckten“ Inputfeld einbauen. (**type="hidden"**)

Ganz am Beginn des Formulars schreibe (dazu benötigt man kein >form-group>) - Zeile 46

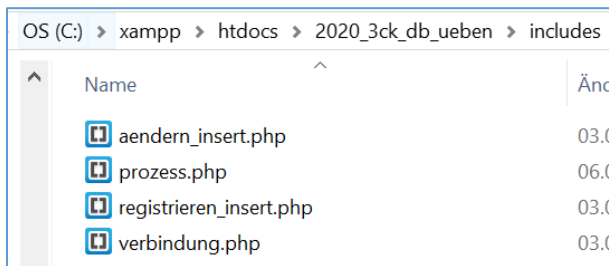
```
45 ▼ <form action="includes/aendern_insert.php" method="post"
      enctype="multipart/form-data">
46     <input type="hidden" name="id" value="<?php echo $id; ?>" |
47 ▼     <div class="form-group">
```

Den Wert für diese Variable \$id hat man ja aus dem Beginn der Datei, nämlich aus Zeile 6, die wir schon geschrieben haben:

```
4 //nun folgt der Ändern-Button
5 ▼ if (isset($_GET['aendern'])) {
6     $id = $_GET['aendern'];
```

Nun muss man die Datei „aendern\_insert.php“ im Ordner „includes“ erstellen.

Kopiere dafür die Datei „registrieren\_insert.php“ und ändere diese.



Öffne nun die neue Datei.

1. Die Variablennamen lassen wir wie sie sind. Hier kann es gerne wieder \$vorname lauten.
2. Das Passwort wurde ja nicht gezeigt und kann hier einfach weggelassen werden.
3. Bezüglich Bild kann man ebenfalls alles gleich lassen. Es musste ja sowieso ein neues vorher angeklickt werden.
4. Dann tut man sich unten beim Ändern der INSERT INTO in die neue UPDATE-Funktion leichter.
5. Füge am Beginn die if-Anweisung ein, die überprüft, ob der Button gedrückt wurde (isset)
6. Setze nach der if-Bedingung die beiden geschwungenen Klammern ein, wobei die schließende ganz nach unten geschoben werden muss, knapp vor den schließenden PHP-Tag.

```
1 <?php
2
3 ▼ if (isset($_POST['aendern-submit'])) { //Name des Buttons
4
5 //hier werden Variablen erstellt, die aus dem Formular kommen|
6 $vorname = $_POST['vorname'];
7 $nachname = $_POST["nachname"];
8 $geschlecht = $_POST["geschlecht"];
9 $klasse = $_POST["klasse"];
10 $geb = $_POST["geb"];
11 $bild = addslashes(file_get_contents($_FILES['bild']['tmp_name']));
12 $benutzername = $_POST["benutzername"];
13 $email = $_POST["email"];
14
15 require 'verbindung.php';
```

```
23
24 echo "Danke für die Anmeldung";
25 }
26 ?>
```

Die Zeile 24 kann man gleich auch löschen. Dieses echo benötigen wir nicht.

Füge danach vor der Variablen Erstellung des Vornamens, noch die Variable für die „hidden“ ID ein:

```

5 //hier werden Variablen erstellt, die aus dem F
6 $id = $_POST['id'];
7 $vorname = $_POST['vorname'];

```

Nun muss man den SQL-Code ändern. Statt einer INSERT INTO benötigt man ja eine UPDATE-Funktion.

- Der orangene vorname ist der in der Datenbank, der nun die neue Variable zugeteilt bekommt.
- Man benötigt ein Istgleichzeichen, einfache Anführungsstriche und einen Beistrich danach
- Das Passwort wird hier wieder weggelassen
- Beim WHERE ist zuerst die ID aus der Datenbank, die mit der neuen Variablen übereinstimmen muss. Dadurch wird genau dieser Datensatz geändert, und kein falscher.

```

19 $sql = "UPDATE schueler SET vorname='$vorname', nachname='$nachname',
    geschlecht='$geschlecht', klasse='$klasse', geb='$geb', bild='$bild',
    benutzername='$benutzername', email='$email' WHERE s_id=$id ";
20

```

Nun soll noch eine Umleitung zurück zur Übersicht erfolgen und dazu eine positive Erfolgsmeldung in der Farbe „grün“ angezeigt werden.

```

22     or die("Fehler bei der Speicherung.");
23
24 //zurück zur Übersicht schueler.php
25 header("Location: ../schueler.php?update=<div class='alert alert-success'>Der
    Schüler wurde erfolgreich geändert</div>");
26
27 }

```

### Code:

```

//zurück zur Übersicht schueler.php
header("Location: ../schueler.php?update=<div class='alert alert-success'>Der Schüler wurde
erfolgreich geändert</div>");

```

Natürlich muss diese neue Meldung mit dem Namen „update“ auch dort in der Datei „schueler.php“ eingebaut werden.

Am besten gleich unter der anderen Meldung. Öffne daher die „schueler.php“:

Kopiere diese Meldung und ändere sie auf „update“:

```
29 <div class="container">
30 <h1>Schüler der 3CK</h1>
31 <br>
32 <?php
33     if(isset($_GET['geloescht'])) {
34         $geloescht = $_GET['geloescht'];
35         echo $geloescht;
36     }
37     if(isset($_GET['update'])) {
38         $update = $_GET['update'];
39         echo $update;
40     }
41     ?>
42 <br>
```

Teste nun die Änderung eines Datensatzes.


Benutzer/E-Mail  Passwort  Login

REGISTRIEREN

Logout

## Schüler der 3CK

Der Schüler wurde erfolgreich geändert

| Vorname | Nachname | Geschlecht | Klasse | Geb.Datum  | Bild  | Aktion         |
|---------|----------|------------|--------|------------|---|----------------|
| Jakob   | Studeny  | m          | 3CK    | 2003-02-17 |  | Ändern Löschen |

Quelle: <https://www.youtube.com/watch?v=3xRMUDC74Cw>