

Registrieren absichern, viele IF-Verzweigungen und Fehlermeldungen erstellen - Teil 4

1. registrieren insert.php absichern
2. Fehlermeldung - error handler, \$ GET mit JavaScript (zum Testen) und PHP
3. hashen des Passworts – verschlüsseltes Passwort in die Datenbank schreiben

1)registrieren insert.php im Ordner „includes“ ändern

Das ist die Datei, die im Formular „registrieren.php“ in der „action“ angesprochen wird. Sie soll die eingegebenen Daten des Besuchers in die Datenbank schreiben, damit er später mittels „Login“ sich anmelden kann.

Überprüfen, ob der „Schüler anlegen“-Button geklickt wurde.

Schüler anlegen

Wird der Submit-Button geklickt? Damit man nicht „illegaler Weise“ durch Eingabe des Seitenpfades dorthin kommt.

Daher „isset“ mit dem „name“ des Buttons aus der Datei „registrieren.php“. Der „name“ lautet „formular-submit“

```
<button type="submit" name="formular-submit" class="block">Schüler anlegen</button>
```

Daher wird dieser mit der Funktion „isset()“ abgefragt:

```
Offene Dateien
registrieren.php
• registrieren_insert.php
Erste Schritte
screenshots
quick-edit.png

1 <?php
2
3 //wenn Button geklickt
4 if (isset($_POST['formular-submit'])) {}
5
6 //hier werden Variablen erstellt, die aus d
7 $vorname = $_POST['vorname'];
```

Wichtig: da in der IF nur im Fall „true“ die Verbindung hergestellt wird und die Daten in die Datenbank gelangen sollen, muss man die schließende geschwungene Klammer aus der IF ganz nach unten schieben, damit alles eingeschlossen ist:

```
24
25 echo "Danke für die Anmeldung. I
26 }
27 ?>
```

2) Einfache Fehlerbearbeitung bzw. positive Bestätigung anzeigen

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

Beispiel: willhaben.at



Dann wird eine Fehlerabfrage gestellt, ob überhaupt die Felder im Formular ausgefüllt wurden. Das passiert mit der Funktion „empty“.

Beachte:

- statt „oder“ = „or“ kann man auch die || verwenden
- die IF wird beim letzten geschlossen, hier nach \$pass

```
15 $pass = $_POST["pass#"];
16
17 require 'verbindung.php';
18
19 if(empty($benutzername) || empty($email) || empty($pass) )
20 {
```

Das heißt: ist eines der Felder leer

- also leer ist „true“ dann soll der Benutzer auf die Registrieren-Seite nochmals zurückgesendet werden, am besten mit einer Fehlermeldung „bitte mach das gescheit“.
- Ansonsten ist leer „false“, somit ist alles ausgefüllt und die korrekte Weiterleitung der Daten in die Datenbank kann erfolgen

Dann Weiterleitung (a) und Error-Behandlung (b):

2a) Eine Weiterleitung in PHP

auf eine andere Seite erfolgt in PHP mittels

```
header("Location: name.php");
```

geht auch: `header("location:name.php");` kleines L und kein Abstand

Hier somit:

```
header("Location: registrieren.php");
```

Das heißt, der Benutzer wird zurück auf die Eingabemöglichkeit im Formular geleitet.

2b) Fehler Behandlung

Da besonders die drei unteren Felder sehr wichtig sind, für eine spätere LOGIN-Verwaltung, sollen hier nur diese drei diese Fehlerbehandlung durchlaufen. Natürlich kann man das für alle <input>-Felder eines Formulars erstellen.

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.

Man spricht auch von der \$_GET-Methode, die eben, anders als die sichere \$_POST-Methode, über die URL sichtbar ist.

Beachte:

- hänge gleich nach dem Dateinamen das „?“ an – das steht für „Hinzufügen von Daten“
- danach ohne Leerzeichen „error=Bitte alle Felder ausfüllen“
- error kann auch ein anderes Wort sein, hauptsache, wenn es in einer anderen Zieldatei wieder aufrufbar sein soll, muss es gleich lautend sein.

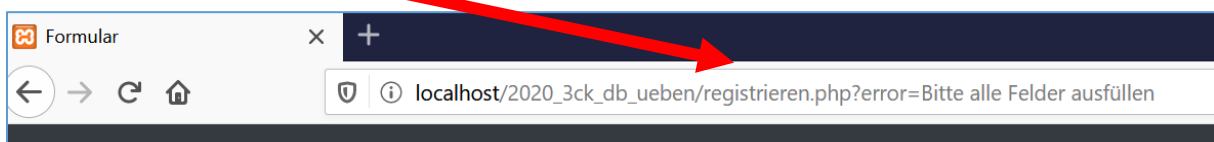
```
19 if(empty($benutzername) || empty($email) || empty($pass) )
20 {
21     //Umleitung
22     header("Location: ../registrieren.php?error=Bitte alle Felder
23         ausfüllen");
24 }
```

Dann auch die Methode „exit“. Diese beendet somit die Weiterführung des Codes, da ja zuerst die Felder „nicht empty“ sein dürfen.

```
21     //Umleitung
22     header("Location: ../registrieren.php?error=Bitte alle Felder
23         ausfüllen");
24     exit();
25 }
```

Testen:

- **GIB KEIN einziges Feld** ein und klicke auf den Button „registrieren“. Hier wird nun in der URL die Fehlermeldung angezeigt:



Diese Fehlermeldung ist gut, aber **nicht besonders SICHTBAR.**

2c)Alert-Fenster (JavaScript)

Kommentiere die Zeilen aus: verwende in jeder der beiden Zeilen das doppelte //

```
14 ▼ if(empty($username) || empty($email) || empty($password) ||  
    empty($passwordWieder)) {  
15     //header("Location: ../registrieren.php?  
    error=Felder_sind_leer&uid=".$username."&email=".$email);  
16     //|exit();  
17 }
```

Daher soll lieber ein Alert-Fenster aufgehen, das diese Fehlermeldung anzeigen soll. Das ist möglich, wenn man den JavaScript-Code mittels „echo“ einbindet.

Erklärung:

Echo ermöglicht den Betrieb nicht nur von HTML, sondern auch von JavaScript.

Der Code für die Umleitung auf eine Datei lautet hier „window.location“. Beachte dabei den Wechsel von einfachen und doppelten Anführungszeichen.

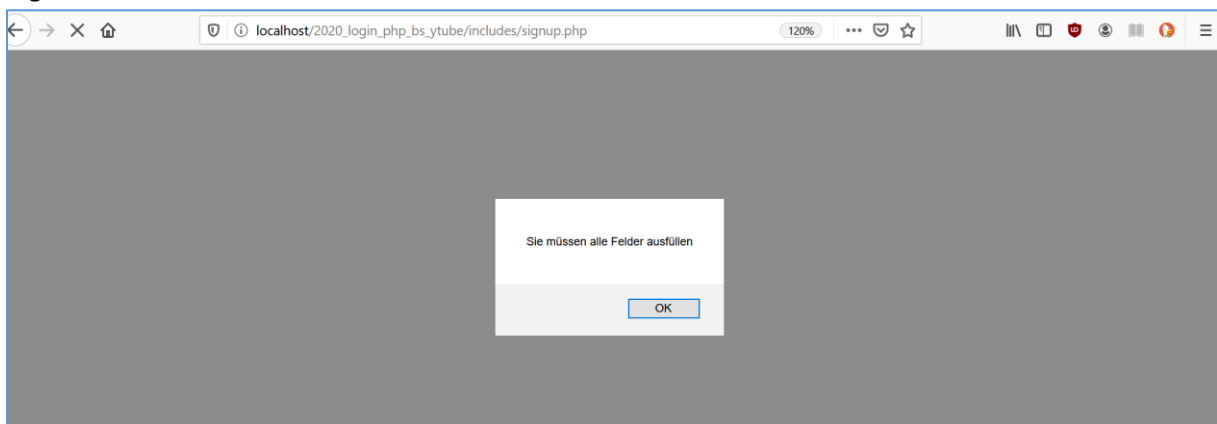
Der direkte Code, wie wir es bis jetzt in PHP durchgeführt haben, nämlich „Location:“ würde hier, direkt nach JavaScript nicht durchkommen und somit nicht ausgeführt. Daher muss man nicht nur das „alert“ hier ausführen, sondern eben auch die Umleitung.

Die Umleitung erfolgt hier wieder auf diese selbe Seite zurück, damit der Benutzer gleich wieder die Chance hat, korrekte Daten einzugeben.

Code:

```
else {  
    //hier folgt JavaScript im echo-Code  
    echo '<script>';  
    echo 'alert("Sie müssen alle Felder ausfüllen");';  
    echo 'window.location = "../registrieren.php";';  
    echo '</script>';  
}
```

Ergebnis:



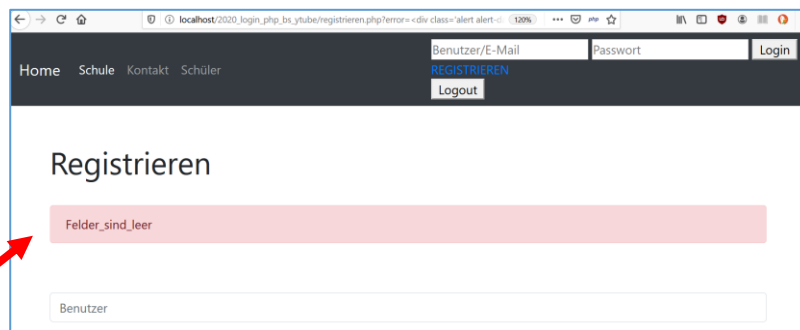
Leider sieht man den Hintergrund nicht, also die Website des Registrierens. Der Hintergrund ist hier grau. Das gefällt auch nicht wirklich, oder?

2d)mit PHP selbst lösen – und als Klasse von Bootstrap

Lösche dem Code wieder und aktiviere den ursprünglichen, auskommentierten Code:

```
19 if(empty($benutzername) || empty($email) || empty($pass) )
20 {
21     //Umleitung
22     header("Location: ../registrieren.php?error=Bitte alle Felder
           ausfüllen");
23 }
```

Schön wäre eine direkte Fehlermeldung im Formular „Registrieren“, so wie diese:



Ziel:

Dafür ist gar nicht soo viel zu tun.

error-Meldung in der registrieren.php anzeigen lassen:

Nun muss man in dem Formular, indem die Fehlermeldung erfolgen soll, diese „message“, die bei uns den Namen „error“ hat, als Code hineinpflanzen.

Öffne daher wieder die Datei „registrieren.php“.

Eine gute Position ist nach der Überschrift und vor dem ersten Input-Feld.

Füge passende
 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!

Siehe Code: Zeile 25-30:

```
21 <h1>Formular bitte ausfüllen</h1>
22 <br>
23 <?php
24     if(isset($_GET['error'])) {
25         $error = $_GET['error'];
26         echo $error;
27     }
28     ?>
29 <br><br>
30
31 <form action="includes/registrieren_insert.p
```

Code:

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

Ergebnis, wenn man nichts ausfüllt und auf den Button „Registrieren“ klickt:

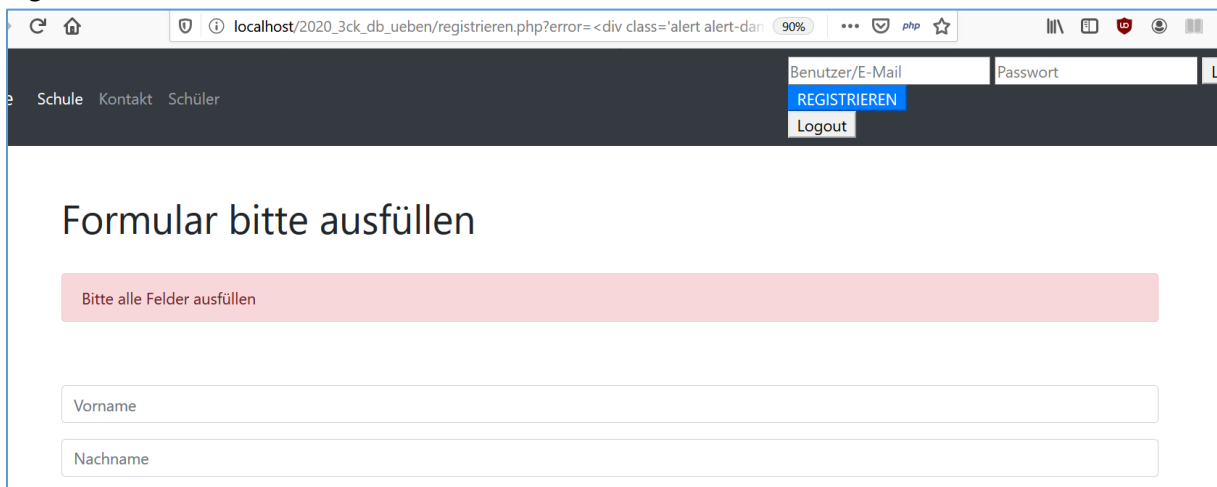


Damit man das etwas auffälliger machen kann, muss man wieder in die Datei „registrieren_insert.php“ wechseln und dort neben dem Text, der hier als Fehlermeldung ausgegeben wird folgendes einfügen.

Beachte: Direkt nach dem „=“-Zeichen und vor dem Text, sowie nach dem Text und vor dem schließendem Anführungszeichen:

```
21 //Umleitung
22 header("Location: ../registrieren.php?error=<div class='alert alert-
23 danger'>Bitte alle Felder ausfüllen</div>");
24 exit();
```

Ergebnis in rot:



3) Daten in die Datenbank schreiben

Wurden bis jetzt alle Daten eingegeben, kann man nun die Daten in die Datenbank schreiben. Erstelle somit die „else“. Das beginnt sofort nach der schließenden geschwungenen Klammer der IF-Abfrage und öffnet ebenfalls eine geschwungene Klammer vor dem vorhandenen \$sql-Befehl, der die Daten in die Datenbank schreibt. Die End-Klammer ist nach dem Strichpunkt der \$erg zu setzen. Hier Zeile 29.

```
23     exit();
24 }
25 else {
26     $sql = "INSERT INTO schueler (vorname, nachname, geschlecht, klasse,
geb, bild, benutzername, email, pass) VALUES ('$vorname', '$nachname',
'$geschlecht', '$klasse', '$geb', '$bild', '$benutzername', '$email',
'$pass')";
27
28     $erg = $con->query ($sql) or die("Fehler bei der Speicherung.");
29 }
30
31 echo "Danke für die Anmeldung. Du hast $vorname $nachname gerade angelegt";
32 }
```

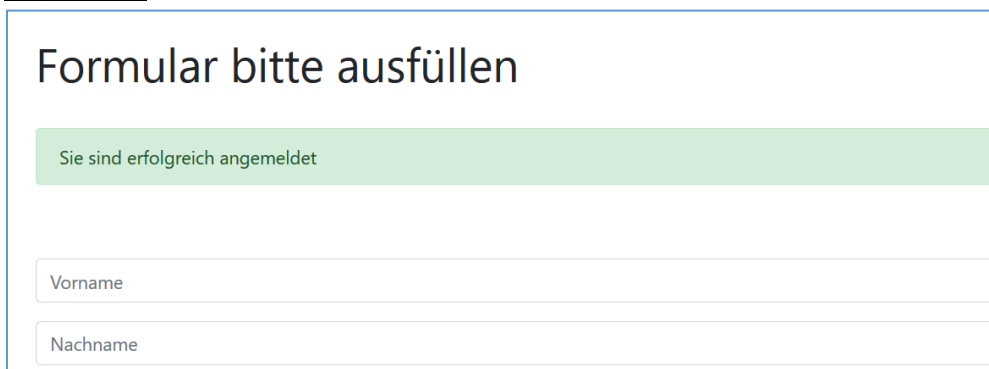
Dann sollte man nicht vor einer leeren Website stehen, sondern zu einer gewünschten Seite geleitet werden. Diese wird hier mit einer Erfolgsmeldung versehen, die hier durchaus in der URL gezeigt wird und nicht als schöne Ausgabe in einem bestimmten Teil des Formulars.

```
28     $erg = $con->query ($sql) or die("Fehler bei der Speicherung.");
29
30     header("Location: ../registrieren.php?error=<div class='alert alert-
success'>Sie sind erfolgreich angemeldet</div>");
31     exit();
32
33 }
```

Code:

```
header("Location: ../registrieren.php?error=<div class='alert alert-success'>Sie sind erfolgreich
angemeldet</div>");
exit();
```

Ergebnis: Inkl. Erfolgsmeldung:



Formular bitte ausfüllen

Sie sind erfolgreich angemeldet

Vorname

Nachname

Daher kann man die alte Meldung ganz zum Schluss auch löschen, da sie nicht mehr gebraucht wird. Hier Zeile 37.

```
36
37 echo "Danke für die Anmeldung. Du hast $vorname $nachname gerade angelegt";
38 }
39 ?>
```

4) Verbesserung: Verschlüsselung des Passworts

Dafür muss man eine neue Variable erstellen und diese dann in die Datenbank schreiben:

Erstelle gleich nach dem „else“ eine neue Variable, die die PHP-Funktion „password_hash()“ aufruft. Diese Funktion hat zwei Parameter, nämlich das originale Passwort (das eingegeben wird) und der Version der Verhashung. Die letzte gute Version ist „b-crypt“. Diese Version wird von PHP-Entwicklern laufend aktualisiert und als „password_default“ bezeichnet. Ältere Versionen sind MD5 oder SHA256. Diese sind nicht mehr sicher.

- Erstelle eine neue Variable, die diesen Hash-Vorgang erstellt – Zeile 26
- Danach muss die „alte“ Variable durch die neue ersetzt werden, in der Funktion, die die Elemente „bindet“, also in Zeile 28 ganz am Ende.

```
25 else {
26     $hashedPwd = password_hash($pass, PASSWORD_DEFAULT);
27
28     $sql = "INSERT INTO schueler (vorname, nachname, geschlecht, klasse,
geb, bild, benutzername, email, pass) VALUES ('$vorname', '$nachname',
'$geschlecht', '$klasse', '$geb', '$bild', '$benutzername', '$email',
'$hashedPwd|')";
```

Ergebnis: Passwort ist verschlüsselt

+ Optionen										
	s_id	vorname	nachname	geschlecht	klasse	geb	bild	benutzername	email	pass
<input type="checkbox"/>	1	Jakob	Studeny	m	3CK	2003-02-17	[BLOB - 29,1 KiB]			
<input type="checkbox"/>	13	Jakob	Österreicher			0000-00-00	[BLOB - 21,0 KiB]			
<input type="checkbox"/>	15	Melvin	Rötzer	m	3CK	2002-11-11	[BLOB - 20,5 KiB]	melvin-r	melvin@gmx.at	\$2y\$10\$JQCh0k8sKZaa1i565MPR4uDoneHhS6R.paizpvnetAg...

Detail:

benutzername	email	pass
melvin-r	melvin@gmx.at	\$2y\$10\$JQCh0k8sKZaa1i565MPR4uDoneHhS6R.paizpvnetAg...

5)Abschluss

Als letzte Sicherheit kommt noch eine „else“-Bedingung, die ausgeführt wird, wenn jemand Zutritt zu der Seite findet und keinen Button geklickt hat. Prinzipiell ist das schon soweit gesichert, weil oben das „isset“ verwendet wurde. Hier wird eine einfache Umleitung zur Registrierungsseite installiert.

Füge diese letzte „else“ nach der letzten geschwungenen Klammer und vor dem PHP-Ende-Tag ein:

```
36 } |
37 ▾ else {
38     header("Location: ../registrieren.php");
39     exit();
40 }
41 ?>
```

Code:

```
else {
    header("Location: ../registrieren.php");
    exit();
}
```

Fertig.

Quelle: <https://www.youtube.com/watch?v=LC9GaXkdxF8>

Schönere Darstellung der Fehlerausgabe: <https://www.youtube.com/watch?v=1jWl6erVmbk>