2)Login und Registrieren (Shop) -PDO, Datenbank

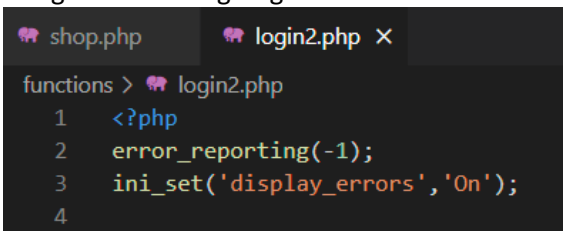
Inhalt:

1. Login: E-Mail und Passwort vergleichen, SESSION erzeugen
2. Navbar ändern, SESSION nutzen, Logout-Button
3. Registrieren erstellen
    a. Frontend-Site in „templates“
    b. Backend für die Übergabe an die Datenbank (SQL - INSERT INTO) mit prepared statements und verschlüsseltem Passwort

1)Login der Eingabe mit Login in der Datenbank vergleichen

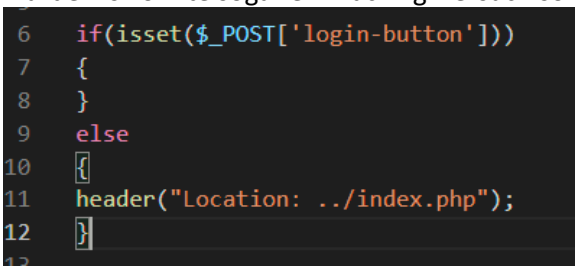
Nun soll in der „login2.php“ der Kontakt zur Datenbank hergestellt werden. Die „login2.php“ wird nach dem Klick auf den Button in der „login“ mittels „action="functions/login2.php" angesprochen.

Erstelle im Ordner „functions“ diese „login2.php“.
Diese soll, wie viele andere PHP-Dateien auch die erleichterte Fehler-Info ausgeben können. Daher soll ganz oben eingefügt stehen:




```
<?php
error_reporting(-1);
ini_set('display_errors','On');
```

Darunter wird nun **mit einer IF geprüft**, ob der Button aus der Login (login.php) überhaupt gedrückt wurde. Wenn nicht, dann soll er zur „index.php“ leiten, weil dann kein normaler Zugang gewählt wurde. Es könnte sogar ein Hacking-Versuch somit ausgebremst werden.



```
6   if(isset($_POST['login-button']))
7   {
8   }
9   else
10  {
11  header("Location: ../index.php");
12  }
```

```
if(isset($_POST['login-button']))
{}
```

Damit die „isset“ auch genutzt werden kann, muss vorne in der „login.php“ der Button auch genau diesen Namen erhalten. Füge daher in der „login.php“ im Ordner „templates“ das genau so ein:

```
36      <div class="card-footer">
37      <button type="submit" name="login-button" class="btn btn-primary">Login</button>
38    </div>
```

INFO: Weiterleitung in PHP: mittels header("Location: ");

Beispiel:
```
header("Location: ../index.php");
```

Wenn der Button korrekt gedrückt ist, dann soll die Datenbank benutzt werden und die Abfrage durchgeführt werden. Daher ist alles nun in der IF.

```
6   if(isset($_POST['login-button']))
7   {
8       require "database.php";
9   }
10  else
```

Vor dem Datenbankaufruf holt man noch die beiden Parameter aus der Login-Eingabe des Formulars. Diese werden hier in zwei Variablen gespeichert, die man dann noch benötigt.

```
5   if(isset($_POST['login-button']))
6   {
7       $email = $_POST['email'];
8       $userpwd =$_POST['password'];
```

Dann folgt die SQL-Abfrage, um die E-Mail aus der Datenbank mit dem der Eingabe des Kunden zu vergleichen.
Die Abfrage ermöglicht die sichere Variante mittels „**preparded statements**“:
Das funktioniert hier bei der Methode „PDO“ mit Hilfe von Doppelpunkt – siehe Zeile 13 und die Übergabe in Zeile 17. (Zur Erinnerung: bei MySQLi waren das Fragezeichen).

```
9
10      require "database.php";
11
12      $sql = "SELECT u_id,email,passwort FROM users WHERE email = :EMail";
13
14      $statement = $dbh->prepare($sql);
15      $statement->execute([
16          ':EMail'=>$email
17          ]);
18      $row = $statement->fetch();
```

$sql = "SELECT u_id,email,passwort FROM users WHERE email = :EMail";

```
$statement = $dbh->prepare($sql);
$statement->execute([
  ':EMail'=>$email
  ]);
$row = $statement->fetch();
```

Bis jetzt wurde das email des Users abgefragt.

Nun folgt die Abfrage des Passworts:

Da das Passwort nur verglichen wird, wenn das email vorhanden ist, kann man das Passwort nun vergleichen. Dies passiert in einer IF.

Weil wird das Passwort in der Datenbank verschlüsselt abgelegt haben, müssen wir nun auch das vom Kunden eingegebene Passwort mit der gleichen Verschlüsselungssoftware nutzen.

Dazu muss die erstellte Variable ebenfalls mit MD5 verschlüsselt werden.

Füge daher unter der Variable eine neue ein, die die alte verschlüsselt:

```
8       $userpwd =$_POST['password'];
9       $hashpwd = MD5($userpwd);  //md5 hashen weil in Datenbank auch MD5
10
11      require "database.php";
```

Dieses neue verschlüsselte Passwort kann nun in der IF verwendet werden.

Die Ergebnisse der IF sind im positiven Fall die Weiterleitung an die Seite „shop.php“ und im negativen Fall an die Seite „index.php“.

```
21      //Passwort checken direkt, weil password_verify nicht funktioniert
22      if ($hashpwd == $row['passwort'])
23          {
24              // erfolgreich eingeloggt
25              header("Location: ../shop.php");
26              exit();
27          }else
28          {
29              header("Location: ../index.php");
30              exit();
31          }
32  }
33  else
34  {
35  header("Location: ../index.php");
36  }
37
38  ?>
```

Wenn man nun erfolgreich eingeloggt ist, soll eine SESSION erstellt werden, in die die ID und die E-Mail gespeichert werden soll. Dadurch sind nun im Hintergrund beide Werte eine gewisse Zeit vorhanden und man kann auf diese zugreifen. Das werden wir später in der „navbar.php“ tun, wenn wir anzeigen lassen: „Du bist eingeloggt als:“.

Daher muss eine SESSION gestartet werden und danach werden

- Zwei SESSIONs mit Namen versehen und darin aus der Datenbank das entsprechende Element übergeben

```php
if($hashpwd === $row['password']){

    // erfolgreich eingeloggt
    session_start();
    $_SESSION['userId'] = $row['u_id'];
    $_SESSION['email'] = $row['email'];
    header("Location: ../shop.php");
    exit;
```

// erfolgreich eingeloggt
session_start();
$_SESSION['userId'] = $row['u_id'];
$_SESSION['email'] = $row['email'];

Damit die Session auch genutzt werden kann, muss es auch gestartet werden:

Damit man mit Sessions hier nun arbeiten kann, muss man in alle Seiten der Website diese ganz am Beginn der Seiten starten lassen.

Ohne diesem Starten der Session würde man nicht auf die durch das LOGIN erstellten und nun vorhandenen Session-Variablen zugreifen können, um zu sehen, ob sie verfügbar sind.

Füge in „shop.php“ im oberen PHP-Bereich den Start der Session ein, siehe Zeile 7.

navbar.php | shop.php

templates > shop.php

```php
<?php
error_reporting(-1);
ini_set('display_errors', 'On');

require "../functions/database.php";

session_start();

?>
```

2)Änderungen in der „navbar.php“

Öffne die „navbar.php“.

- Die bereits vorhandenen 2 Links zu „Login“ und „Registrieren“ erhalten davor eine IF-Abfrage, ob eine SESSION vorhanden ist, oder nicht. Diese wurde bei erfolgreichem Login erzeugt.
    - Ist sie vorhanden (abgefragt durch „isset“) wird der Inhalt der SESSION übernommen und ausgegeben
    - Zusätzlich wird der Button „Logout“ angezeigt
    - Login und Registrieren wird hier nicht benötigt, da der User ja schon eingeloggt ist
- Hat das Login nicht funktioniert, wurde auch keine SESSION erstellt und die beiden Buttons werden wieder angezeigt.

```
19
20          <ul class="navbar nav">
21            <li>
22            <?php
23            if(isset($_SESSION['email'])):?>
24                <a class="nav-link">Du bist eingeloggt als:
25                      <?php echo $_SESSION['email'] ?>  
26                </a>
27            </li>
28            <li>
29                <a href="templates/logout.php">
30                  <button class="btn btn-danger">Logout</button>
31                </a>
32            <?php endif;?>
33            </li>
34          </ul>
```

```
<ul class="navbar nav">
 <li>
 <?php
 if(isset($_SESSION['email'])):?>
    <a class="nav-link">Du bist eingeloggt als:
         <?php echo $_SESSION['email'] ?>  
    </a>
 </li>
 <li>
    <a href="templates/logout.php">
      <button class="btn btn-danger">Logout</button>
    </a>
 <?php endif;?>
 </li>
</ul>
```

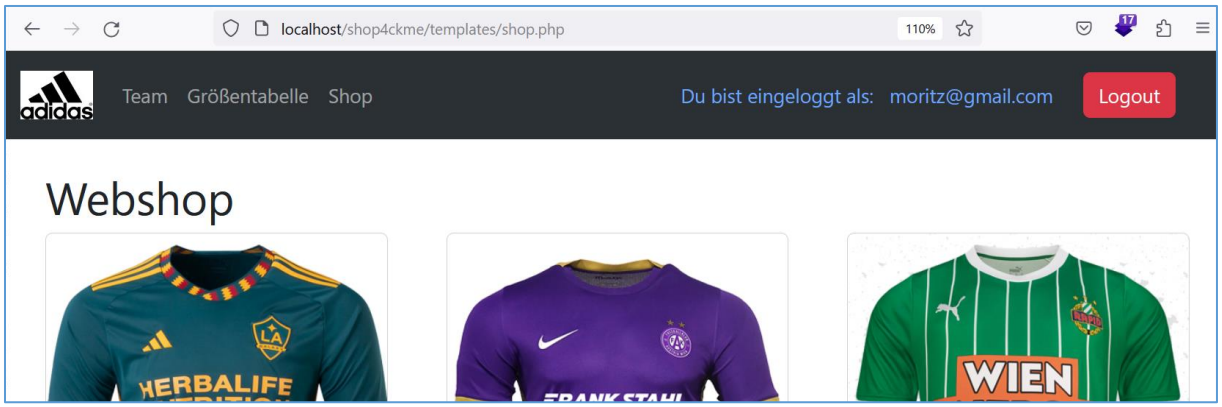
Die beiden Links „Login“ und „Registrieren“ werden verschönert und als Button angezeigt.

```
33              </li>
34
35              <li class="nav-link">
36              <?php if(!isset($_SESSION['email'])):?>
37                  <a button class="btn btn-outline-success"
38                  href="templates/login.php">Login</a>
39                  <a button class="btn btn-outline-primary"
40                  href="templates/registrieren.php">Registrieren</a>
41              <?php endif;?>
42              </li>
43              </ul>
44          </div>
45      </div>
46    </nav>
```

```
<li class="nav-link">
<?php if(!isset($_SESSION['email'])):?>
   <a button class="btn btn-outline-success"
   href="templates/login.php">Login</a>
   <a button class="btn btn-outline-primary"
```

```
   href="templates/registrieren.php">Registrieren</a>
<?php endif;?>
</li>
</ul>
```

Ergebnis:



3)Registrieren

3a)HTML-Formular

Öffne die „navbar.php“ und füge auch die Verlinkung zum Registrieren ein:

```
            </form>
            <form class="d-flex">
              <a class="nav-link" href="templates/registrieren.php">Registrieren</a>
            </form>
          </div>
```

registrieren.php

Erstelle im Ordner „templates“ die Datei „registrieren.php“.

- Nach dem DOCTYPE die HTML-Sprache wieder auf „de“ umstellen – auf Deutsch.
- Erstellen eines Formulars <form>
- In diesem <form> wird durch die „action“ festgelegt, wohin der Inhalt nach dem Klicken auf den Absende-Button, zur Weiterverwendung gesendet wird.
- Darin wird auch die Methode der Versendung mit „POST“ festgelegt.
- Einzelne labels und Input-Felder. Zur bessern Bedienung auf einem Smartphone soll auch folgendes berücksichtigt werden, was jedoch sonst KEINEN anderen Grund hat:
  - for – im Label
  - id – im Input-Feld -> beides muss den selben Begriff aufweisen, der sonst total irrelevant ist, daher kann es auch sein: vn oder nn. Eine Beziehung zu „vorname“ und „nachname“ im daneben stehenden „name“ ist nicht gegeben.
- „name“ im Input-Feld ist nötig, damit mit der Methode „post“ dieser Inhalt des Inputfeldes, das der Kunde ja händisch eingibt, an eine nachfolgende „registrieren2.php“, übergeben werden kann.
- Bevor die <form> endet muss der Absende-Button mit dem type=“submit“ vorhanden sein. Dieser erhält ebenfalls einen beliebigen, aber passenden Namen, wie z.B. name="reg-button". Dieser wird ebenfalls in der „registrieren2.php“ in einer IF-Abfrage verwendet.

```
<form method="post" action="functions/registrieren2.php">
```

templates > registrieren.php

```
<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pullishop</title>
    <base href="/4ckpulli_test/">
;    <link rel="stylesheet" href="assets/css/bootstrap.css">
    <link rel="stylesheet" href="assets/css/styles.css">
</head>
```

```
</head>
<body>
  <?php
  require "navbar.php";
  ?>

<br><br><br><br>
```

```
<?php
 require __DIR__.'/../includes.php';
?>

<!DOCTYPE html>
<html lang="de">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Pullishop</title>
   <base href="/4ckpulli_test/">
;   <link rel="stylesheet" href="assets/css/bootstrap.css">
   <link rel="stylesheet" href="assets/css/styles.css">
</head>
<body>
 <?php
 require "navbar.php";
 ?>
```

Darunter nun das Formular, das ganz oben die E-Mail und das Passwort verlangt. Danach ist ein horizontaler Strich und der Rest zum Eingeben

```
19      ?>
20
21    <br><br><br><br>
22
23    <div class="container">
24    <form method="post" action="functions/registrieren2.php">
25    <div class="card">
26      <div class="card-header">
27       REGISTRIEREN
28      </div>
```

```
<br><br><br><br>

<div class="container">
<form method="post" action="functions/registrieren2.php">
<div class="card">
 <div class="card-header">
  REGISTRIEREN
 </div>
```

```
28      </div>
29      <div class="card-body">
30        <div class="row g-3">
31            <div class="col">
32            <label for="em" class="form-label text-uppercase">E-Mail</label>
33            <input type="email" class="form-control" id="em" name="email">
34            </div>
35            <div class="col">
36            <label for="pw" class="form-label text-uppercase"> Passwort</label>
37            <input type="password" class="form-control" id="pw" name="password">
38            </div>
39        </div>
40        <br>
41        <hr>
```

    <label for="em" class="form-label text-uppercase">E-Mail</label>
    <input type="email" class="form-control" id="em" name="email">

    <label for="pw" class="form-label text-uppercase"> Passwort</label>
    <input type="password" class="form-control" id="pw" name="password">

  <hr>

```
41        <hr>
42        <br>
43        <div class="row g-3">
44            <div class="col">
45                <label for="vn" class="form-label">Vorname</label>
46                <input type="text" class="form-control" id="vn" name="vorname">
47            </div>
48            <div class="col">
49                <label for="nn" class="form-label">Nachname</label>
50                <input type="text" class="form-control" id="nn" name="nachname">
51            </div>
52        </div> <!--ende row-->
```

      <label for="vn" class="form-label">Vorname</label>
      <input type="text" class="form-control" id="vn" name="vorname">

      <label for="nn" class="form-label">Nachname</label>
      <input type="text" class="form-control" id="nn" name="nachname">

   <!--ende row-->

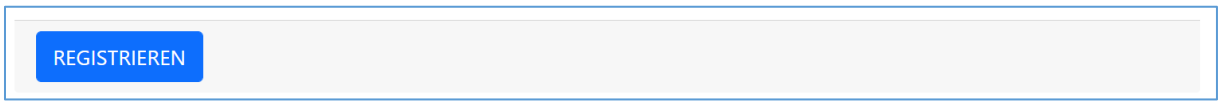
```
53      <div class="row">
54          <div class="col-3">
55              <label for="pl" class="form-label">PLZ</label>
56              <input type="num" class="form-control" id="pl" name="plz">
57          </div>
58          <div class="col-9">
59              <label for="or" class="form-label">Ort</label>
60              <input type="text" class="form-control" id="or" name="ort">
61          </div>
62      </div> <!--ende row-->
```

- Kopiere diese <row> und füge sie darunter ein, damit man die Zellen auf „PLZ“ und „Ort“ umändern kann.
- Dann in einer eigenen <row> die Strasse und die Telefonnummer.

```
63      <div class="row">
64          <div class="col-12">
65              <label for="str" class="form-label">Strasse und Nr.</label>
66              <input type="text" class="form-control" id="str" name="strasse">
67          </div>
68      </div> <!--ende row-->
69      <div class="row">
70          <div class="col-12">
71              <label for="te" class="form-label">Telefon</label>
72              <input type="num" class="form-control" id="te" name="telefon">
73          </div>
74      </div>
75      <br>
```

```
75      <br>
76      <div class="card-footer">
77       <button type="submit"  name="reg-button" class="btn btn-primary text-uppercase">
78           Registrieren</button>
79    </div>
80  </div>
81  </form>
82  </div> <!--ende container-->
83  </body>
84  </html>
```

Ergebnis:



Zum Vergleich hier nochmals die Datenbank:

#	Name	Typ	Kollation	Attribute	Nul
1	u_id	int(11)			Nei
2	email	varchar(50)	utf8mb4_general_ci		Nei
3	password	varchar(100)	utf8mb4_general_ci		Nei
4	vorname	text	utf8mb4_general_ci		Nei
5	nachname	text	utf8mb4_general_ci		Nei
6	plz	int(5)			Nei
7	ort	text	utf8mb4_general_ci		Nei
8	strasse_nr	varchar(50)	utf8mb4_general_ci		Nei
9	telefon	varchar(20)	utf8mb4_general_ci		Nei

3b)Senden in die Datenbank - registrieren2.php

Dabei wird wieder überprüft, ob der Button in dem Formular von „registrieren.php“ gedrückt wurde. Nur dann erhält man Zugang zu den nächsten Informationen.

Passwort: „passwort“

- das Passwort wird als Variable „passwort“ mit hartem „t“ gespeichert, nachdem es mittels $_POST[] übernommen wurde. In dem Formular „registrieren.php“ ist es nicht maßgeblich, wie es dort im name=“password“ geschrieben wird, Hauptsache es wird hier mit $_POST übernommen. Könnte natürlich da wir dort auch „karli“ heißen 😊
- „passwort“ mit hartem „t“ ist daher wichtig, weil das „password“ bereits in SQL reserviert sein kann und daher zu Problemen führen kann. Nämlich, dass es dann nicht in die Datenbank gespeichert wird.

Passwort hashen: $passwort_hashed()

- Damit das Passwort nicht in Reinform in der Datenbank liegt, sollte es unbedingt verschlüsselt werden.
- Damit es in der sicheren und einfachen Form verschlüsselt wird, kann man die Methode MD5 verwenden. Das wurde auch bereits im Login verwendet. Daher soll es auch so in der Datenbank landen (durch unser registrieren2.php“.
- Einfach die (eine Zeile darüber) erstellte Variable gleich darunter in die MD5 – Verschlüsselung überführen. Natürlich kann die neue Variable auch „passwort“ heißen, aber zur besseren Übersicht und hier zum Lernen nenne es „passwort_hashed“:
- $passwort_hashed = MD5($passwort);

registrieren2.php

functions > registrieren2.php

```php
<?php
error_reporting(-1);
ini_set('display_errors','On');

if(isset($_POST['reg-button']))
{

```

```php
$email = $_POST['email'];
$passwort = $_POST['password'];
$passwort_hashed = MD5($passwort); //verschlüsseln
$vorname = $_POST['vorname'];
$nachname = $_POST['nachname'];
$plz = $_POST['plz'];
$ort = $_POST['ort'];
$strasse = $_POST['strasse'];
$telefon = $_POST['telefon'];
}
```

```
<?php
error_reporting(-1);
ini_set('display_errors','On');

if(isset($_POST['reg-button']))
{
$email = $_POST['email'];
$passwort = $_POST['password'];
$passwort_hashed = MD5($passwort);
$vorname = $_POST['vorname'];
$nachname = $_POST['nachname'];
$plz = $_POST['plz'];
$ort = $_POST['ort'];
$strasse = $_POST['strasse'];
$telefon = $_POST['telefon'];
```

SQL-Befehl: INSERT INTO

Das Schreiben in die Datenbank benötig natürlich die Verbindung zur Datenbank, spätestens hier, siehe Zeile 19.

- users: lautet die Tabelle in der Datenbank. Hier müssen die Elemente den gleichen Namen und auch die gleiche Reihenfolge aufweisen
- VALUES: hier werden nicht automatisch die eben erstellten Variablen benutzt, sondern die Daten, die durch die Sicherheitsmaßnahme „prepaired statements“ erzeugt werden. Diese „Umwandlung“ zur Sicherheit erfolgt in den Zeilen darunter und es werden die Daten mit den typischen „Doppelpunkt“ geliefert.

```php
$telefon = $_POST['telefon'];
}

require "database.php";
$sql= "INSERT INTO
users(email, passwort, vorname, nachname, plz, ort, strasse_nr, telefon)
VALUES (:email,:passwort, :vorname, :nachname, :plz, :ort, :strasse_nr, :telefon)";
```

```
require "database.php";

$sql= "INSERT INTO users(email,passwort, vorname, nachname, plz, ort, strasse_nr, telefon )
```

```
VALUES(:email,:passwort, :vorname, :nachname, :plz, :ort, :strasse_nr, :telefon)";

   $statement = $dbh->prepare($sql);


$statement->execute([

      ':email'=>$email,
      ':passwort'=>$passwort_hashed,
      ':vorname'=>$vorname,
      ':nachname'=>$nachname,
      ':plz'=>$plz,
      ':ort'=>$ort,
      ':strasse_nr'=>$strasse,
      ':telefon'=>$telefon
      ]);

}
```

Nach erfolgreichem Registrieren soll automatisch die Seite „templates/login.php“ geöffnet werden. Daher füge ein:

```
32          ':strasse_nr'=>$strasse
33          ]);
34
35          header("Location: ../templates/login.php");
36          exit;
37
38
39      }else{
40          header("Location: ../index.php");
41          exit;
42      }
```

Ergebnis in der Datenbank – sollte alles „ANKOMMEN“:

	u_id	email	passwort	vorname	nachname	plz	ort	strasse_nr	telefon
Löschen	1	jo@gmx.at	81dc9bdb52d04dc20036dbd8313ed055	Jakob	Oesterreicher	2161	Poysbrunn	Jakobstrasse 1	0664123333
Löschen	14	hi@gmx.at	81dc9bdb52d04dc20036dbd8313ed055	Josef	Ebhart	2152	Miba	Berg	111