

Assignment/exercise 1: PHP Login System

Michał Pawlicki

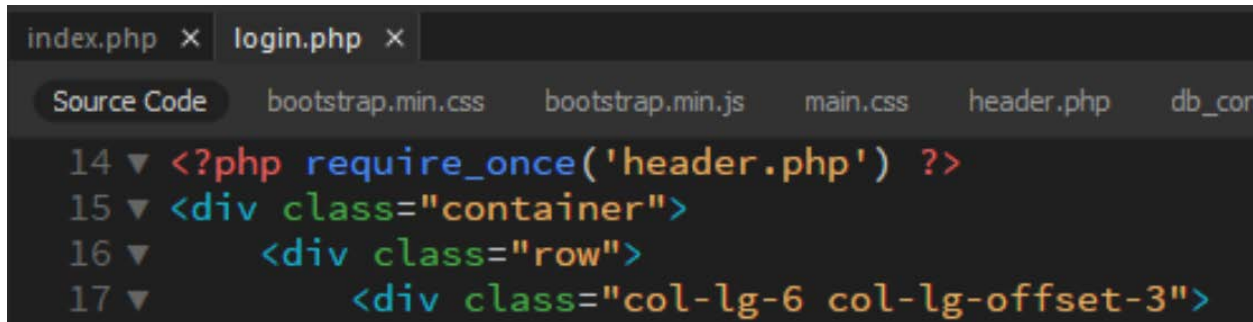
Cph-mp394@cphbusiness.dk

Solution: <http://projects.michaeldesign.pl/loginsystem/>

Source on GitHub: <https://github.com/magicmychal/loginsystem>

Project description

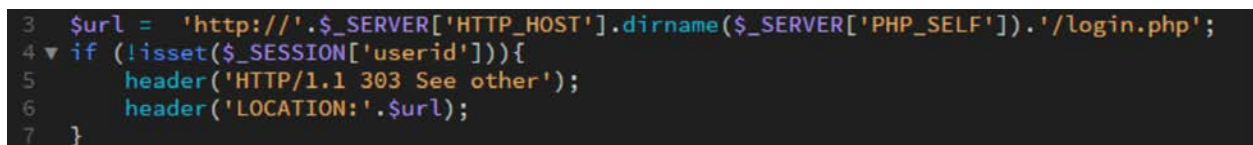
Website is divided into 3 pages with an actual content: index.php (the secret page), login.php and register.php. There is also a header.php that, as the name suggests, contains the header on a page and it shows up in every other page thanks to the require_once() function in the code and it is used right after <body>.



```
index.php x login.php x
Source Code bootstrap.min.css bootstrap.min.js main.css header.php db_con
14 <?php require_once('header.php') ?>
15 <div class="container">
16     <div class="row">
17         <div class="col-lg-6 col-lg-offset-3">
```

And also logout.php, which will be described soon.

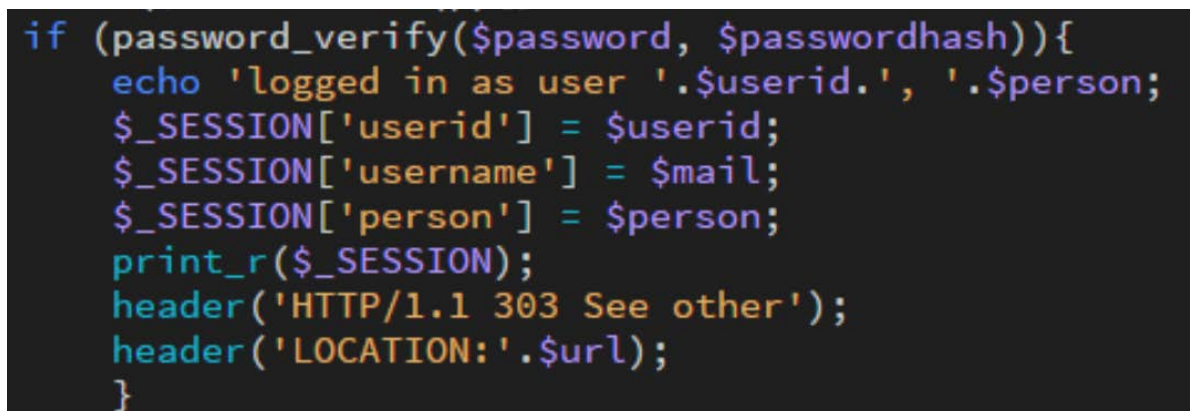
When user tries to access the website via URL given as a solution, server looks for the index.* file. When it finds it, the first few lines of it asks to check if there's an active session in the browser. If so, then user sees the 'secret content', if not, then the web page goes to the /login.php where they can find simple login form. It is worth notice, that the URL in this if statement is independent and it is going to work on with every domain that the files are on.



```
3 $url = 'http://'.$_SERVER['HTTP_HOST'].dirname($_SERVER['PHP_SELF']).'/login.php';
4 if (!isset($_SESSION['userid'])){
5     header('HTTP/1.1 303 See other');
6     header('LOCATION:'.$url);
7 }
```

The login page is basically the first thing that a new user sees. It has a simple structure based on a Bootstrap 4 framework. It allowed to use a grid, set fixed navbar and style the inputs.

When a user tries to log in, the script checks if this user exists in the database and if the password is correct. Since the password has been hashed and salted during registration, which will be described in a moment, the website uses password_verify() function to determine if a data is correct. Then the scripts creates new variables in a session – userid, username (which is mail) and person (it is a Full Name of a user). After that user is redirect to the index.php, where the secret content is.



```
if (password_verify($password, $passwordhash)){
    echo 'logged in as user '.$userid.', '.$person;
    $_SESSION['userid'] = $userid;
    $_SESSION['username'] = $mail;
    $_SESSION['person'] = $person;
    print_r($_SESSION);
    header('HTTP/1.1 303 See other');
    header('LOCATION:'.$url);
}
```

Also, while being on the login.php, user has possibility to click the link in order to go create a new account. Link is referring to the register.php

SC

Secret society

☰

It takes courage to make this one step that is needed to enter the secret

Login to the secret society

Connected to the database

Are you not in the Secret society? Don't worry! You can join us in couple of seconds. **Just fill the registration form here.**

There, simply by giving name, email and password, user is able to create an account.

SC

Secret society

☰

Join us simply by giving your name, e-mail and password!

We respect your privacy, thus all the passwords are encrypted and we are not able to restore them.

Please remember, once registered you cannot go back.

Sing up to the secret society

Are you already a member of the Secret society? **Log in here!**

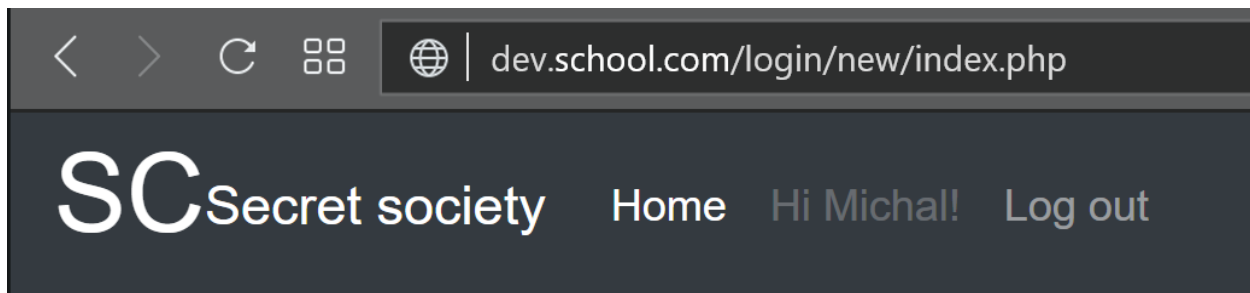
After clicking 'go!' button or hitting enter key, the script first of all checks if the email already exists in

the database. It is very important, otherwise, the account would not be created and the user would not see any kind of error.

```
58      //checking if the user exist in the database
59      $sqlcheck = 'SELECT mail FROM logintest WHERE mail=?';
60      $stmtcheck = $con->prepare($sqlcheck);
61      $stmtcheck->bind_param('s', $mail);
62      $stmtcheck->execute();
63      $stmtcheck->bind_result($mailcheck);
64      while($stmtcheck->fetch()){
65          if($mail == $mailcheck){
66              echo "<div class='alert alert-danger' role='alert'>
67                  This mail is already in our database.
68                  </div>";
69      }
```

For that, the code search through database for the email, it is login.php alike. When there is no match, the code goes to the next step, which is creating a user and displaying success message. From now on, user has to click the link that goes to the login page.



Now, the secret page is the index.php, which is technically nothing fancy. The whole page has 'glitching' effect that was sourced from codepen.io, the website with MIT licensed ideas. On this website, the header is important. Since only logged in users can see it, there is a 'log out' functionality.



After clicking that user goes to the logout.php. There is a simple PHP code that destroys a session by replacing all the variables in an array with the empty values and also it deletes cookies associated with it. Then it quickly goes back to the login.php, that is why there is nothing to be displayed for the user.

```
1  <?php session_start();
2  $url = 'http://'.$_SERVER['HTTP_HOST'].dirname($_SERVER['PHP_SELF']).'/login.php'; ?>
3  <!doctype html>
4  <html>
5  <head>
6  <meta charset="utf-8">
7  <title>Log out</title>
8  </head>
9  <?php
10     //log out
11     //unsetting all variables in a session, creating empty array
12     $_SESSION = array();
13
14     if (ini_get('session.use_cookies')){
15         $params = session_get_cookie_params();
16         setcookie(
17             session_name(), '', time() - 4200,
18             $params['path'], $params['domain'],
19             $params['secure'], $params['httponly']
20         );
21         header('HTTP/1.1 303 See other');
22         header('LOCATION:'.$url);|
23     }
24     else{
25         echo 'It looks like you are already logged out';
26     }
27     session_destroy();
28 ?>
29 </body>
30
31 </body>
32 </html>
```

The structure of the database that stores all users information is simple.

	#	Nazwa	Typ	Metoda po
<input type="checkbox"/>	1	ID 	int(11)	
<input type="checkbox"/>	2	username 	varchar(255)	utf8_bin
<input type="checkbox"/>	3	mail	varchar(255)	utf8_bin
<input type="checkbox"/>	4	passwordhash	varchar(255)	utf8_bin

There is an ID with auto incrementation, username, mail and passwordhash are all in varchar with maximum 255 characters.

This project taught me that security is important. There are plenty of websites and plenty of solutions, on sites like stackoverflow, that don't hash the password, which in the end result is a huge security flaw. And even though there is a very simple solution to that, it's still happening.

Reflection

During this project, I also learned the power of global variables and how to use them. PHP, as a lot of other languages, has a great base of functions that are there to help developers use fast, safe and reliable solutions and I believe that's the biggest value that I earned during this project.