Assignment RegisterAndLogin

Date : January 2019

By : Benjamin Porobic ([b.porobic@roc-teraa.nl](mailto:b.porobic@roc-teraa.nl)) & Peter Nocker ([p.nocker@roc-teraa.nl](mailto:p.nocker@roc-teraa.nl))

Hours : ca. 9 hours (3 weeks, 1.5 hours a week @ school, rest @ home)

Assessment : Individual

Read the assignment carefully and ask questions if anything is not clear.

# Introduction

Nowadays there are a lot of platforms on social media. One thing all these platforms have in common is that users on the platform has to register and login first to be get access to the platform.

**Register and login on a website means you’ll need:**

* Database to store user info, like first and last name, but also a username and password
* PHP script to create a database connection
* PHP script to handle posted data sent by the user
* HTML script that provides a form to register
* HTML form that provides a form to login
* SQL code to store and read data from the database

**To receive a passing grade the project needs to support at least the following:**

* HTML page with a form to register
* HTML page has a neat layout
* HTML register form has at least:
  + First name
  + Last name
  + Email-address
  + Username
  + Password
* PHP script that stores the posted data into the database
* PHP form-validation
  + All form fields has to be filled in by the user to register
  + If one of the fields is left empty, PHP will generate and show a message that something isn’t correct

**To get a higher grade, between a 5.5 and 8 you should add the following features:**

* The posted e-mail-address is tested (validated by some regular expression)
* The password that is sent will be hashed by PHP before storing it into the database-table

**To get an even higher grade (above 8) you could do the following:**

The project consists of another HTML-page with a login form that has:

* Username
* Password
* PHP script that checks posted data from the login-form with info in the database
* PHP script uses the $\_SESSION array to store user-info from the database.

*Note:* The web development lessons in this period provide two different projects. During front-end lessons you will work on a different project (The Vault)!

# Prior knowledge & examples

To successfully complete this assignment it is necessary to have a basic understanding of HTML, CSS and PHP.

**You should have knowledge of the basic HTML page concepts:**

* Head section with meta info (title etc…)
* Header
* Footer
* Divisions
* Forms
* Form Fields
  + Input type = text
  + Input type = submit
  + Input type = password

Also, you should have knowledge of basic CSS concepts

* Default selectors
* Classes
* Ids
* CSS files

**Besides HTML, you should have knowledge of basic PHP concepts:**

* Variables
* Arrays
* Looping
* Printing text

**Furthermore, you should have knowledge of basic SQL:**

* Select \* from
* Creating a database using phpMyadmin
  + Tables
  + Columns
  + Column-type
    - Int
    - Varchar
    - Primary key
    - Auto increment

# Requirements (MoSCoW)

## Must 5,5

* HTML page with a form to register
* Form has:
  + First name
  + Last name
  + E-mail-address
  + Username
  + Password
* HTML page and form has a ‘pretty’ layout
* All form elements are properly lined and styled using CSS
* CSS is in a separate file
* **PHP script will check all posted data and generates a message when fields are left empty**
  + <https://www.w3schools.com/php/showphp.asp?filename=demo_form_validation_escapechar>
  + <https://www.sitepoint.com/form-validation-with-php/>
* All posted data is stored in a database table
* The database table has following columns:
  + userId : int (3), auto increment, primary key
  + userFirstName: varchar (25)
  + userLastName: varchar (50)
  + userEmailAddress: varchar (100)
  + userUserName: varchar (25)
  + userPassword: varchar (100)
* ***Note: database, table and / or column names can or may differ!***
* Once the posted data is successfully stored into the database, a message **must** be shown on the page that the user is successfully registered.

## Should 5,5 – 8

* PHP scripts for the database connection is in a separate file (use an include or require function for this)
* regular expression to check the posted e-mail-address
* the password stored in the database will be hashed
* extend the database table with other user info:
  + date of birth
  + date of register (so in future, some admin could see when a user has registered)
  + address
    - street name
    - house number
  + city
  + zip code
  + country
  + phone-number

## Could 8 - 10

* Use PHP functions for:
  + Handle register
  + Handle login
  + Handle logout
* User will be able to login once he is registered
* Use some Javascript form-validation when the user wants to register.
  + See: <https://www.w3schools.com/js/js_validation.asp>
* Additional admin page that shows all registered users
  + This page / script is only accessible for the admin
  + So, a separate user should be added in the database table especially for an admin
  + Show an HTML table (or alike) that presents all registered users
  + Use PHP to loop through the table
    - See previous projects like **RadioGaGa** or **Gameworld**

Tips & resources

* Use classroom examples and / or PowerPoint sheets from N@tschool.
* Make backups: external hard drive or cloud storage!
* Upload your finished assignment in N@tschool. Use Winrar or 7-zip to pack your files
  + Only .7zip or .rar file extension allowed
  + Filename: ***firstname\_lastname\_registerandlogin.rar***
  + Filename example: john\_doe\_registerandlogin.rar

**Reference**

* HTML & CSS
  + <http://htmlreference.io/forms/>
  + <http://htmlreference.io/element/div/>
  + <http://cssreference.io/backgrounds/#background-color>
  + <http://htmlreference.io/element/li/>
  + <http://cssreference.io/property/font-family/>
  + <http://cssreference.io/property/font-size/>
  + <http://cssreference.io/property/border/>
* PHP
  + <https://www.w3schools.com/php/php_operators.asp>
  + <https://www.w3schools.com/php/php_variables.asp>
  + <https://www.w3schools.com/php/php_forms.asp>
  + <https://www.w3schools.com/php/php_if_else.asp>

**Courses**

* <www.udacity.com>
* <www.codecademy.com>
* [www.freecodecamp.com](http://www.freecodecamp.com)
* Work together if necessary, but **do not copy**. **In case of fraud, you have to create your own assignment with the same techniques as this one. You have to discuss this with your teacher!!**

# Assessment

Assessment overview will follow soon…

# Examples

In this project you can use PHP functions to create better (maintainable, readable) code and to get a higher grade. Below you see some examples of how to declare functions and how to use them…







