

Challenge 13- PHP

Part 1:

You need to create 4 pages.

The **first page** will be a sign up/login page. It should have a background image and two buttons. (sign up and login). The forms are not on this page, only 2 links (styled as buttons) that when clicked will take the user to a page dedicated for either login or register action.

The sign up link/button should take the user to the **second page** where a form with two inputs for username and password is shown. At the top of this page the text 'Sign up form' is printed.

When the user submits the form, the data should be written into a users.txt file (format: username=password). The password should be hashed, either using md5 or password_hash (preferred option).

Every new user should be inserted in a new row in the users.txt file.

After form submission, the user should be redirected to a new page (**third page**) where the message 'Welcome \$username' will be shown.

The login link/button should take the user to a new page (**fourth page**) where a form with 2 inputs for username and password is also shown. At the top of this page the text 'Login form' is printed.

When a user submits the form – a check should be done if the data the user entered matches some of the users in the users.txt file. Remember, the passwords need to be hashed.

If the user is not found in users.txt file – show the message 'Wrong username/password combination '.

If everything is ok – Redirect the user to the third page where the same message 'Welcome \$username' is shown.

Part 2:

Add another input to the sign up page: email.

Create a new file where instead of storing only the username=password, you will save the email too, in the following format: 'email, username=password'.

When registering, add logic to check if the username already exists in the users.txt file. If it does, print a message 'Username taken'.

Also, check if the email already exists in the users.txt file. That means that when submitting the form, the user has to provide a unique email. If the email already exists in the users.txt file, print the following message: 'A user with this email already exists'. Use yellow font color for this message.

Part 3:

Add input validation. Every validation needs to be a separate function. The validations to be implemented are:

- All inputs are required (no empty fields are allowed);
- Username cannot contain empty spaces or special signs;
- Email must have at least 5 characters before @;
- Password must have at least one number, one special sign and one uppercase letter;

Then, adapt the code to print proper validation messages (in red color) for each failed validation.

Evaluation:

Part 1 of the exercise: max **2** points

Part 1 and **2** of the exercise: max **3.5** points

Part 1, 2 and **3**: max **5** points

Deadline:

Two weeks after the presentation, 23:59h end of the day.