



C1 - PHP pool

C-DEV-111

Pool PHP 10

Web again

Pool PHP 10

repository name: pool_php_10

repository rights: ramassage-tek



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).



CONTENTS

- Introduction
- Exercise 01 (1pt)
- Exercise 02 (3pts)
- Exercise 03 (2pts)
- Exercise 04 (3pts)
- Exercise 05 (4pts)
- Exercise 06 (4pts)
- Exercise 07 (3pts)



INTRODUCTION

If you have done correctly and completely exercise 9 of the preceding day, you may go directly to exercise 2. Otherwise, start with exercise 1.

EXERCISE 01 (1PT)

Turn in: pool_php_10/ex_01/inscription.php

Restrictions: None.

Create a registration form that saves a new user in a database.

You will create a page inscription.php that displays a registration form with the following input fields:

- A field of type "text" with attribute "name" with value "name"
 - A field of type "email" with attribute "email" with value "email"
 - A field of type "password" with attribute "password" with value "password"
 - A field of type "password" with attribute "password_confirmation" with value "password_confirmation"
- You can also use placeholders in place of values. (better practice).

The name must contain between 3 and 10 characters. It must be saved in the database in the field "name". In case of error, you will need to display "Invalid name" on top of the form.

The email must be valid.

It must be saved in the database in the field "email". To check that the email is valid, learn about **Regular Expressions**.

In case of error, you will display "Invalid email" on top of the form.

The password must match its confirmation, must contain 3 to 10 characters, and must be hashed with the PHP function password_hash.

This hash must be saved in the database in the field "password". In case of error, you will display on top of the form "Invalid password or password confirmation".

The creation date of the new user must be saved in the database in the field "created_at". If the user creation succeeds, you will display the form again preceded by the message "User created".



EXERCISE 02 (3PTS)

Turn in: pool_php_10/ex_02/index.php

pool_php_10/ex_02/login.php

Restrictions: None.

You will create a page login.php that will let a registered user authenticate himself, and a page index.php that will be the home for the site.

You will create a form on the connection page with the following fields:

A field of type "email", with name attributes set to "Email".

A field of type "password" with name attributes set to "Password".

When the user validates the form with the "Submit" button:

- If the email and password are incorrect, you will display the form again with the message on top "Incorrect email/password"
- If the email and password are correct, you redirect the user to the page index.php

You will create the page index.php that will have the following behavior:

- If the user is not authenticated, he must be redirected to the page login.php
- If the user is authenticated, this page must greet him with his user name, for example "Hello Bob"

This authentication must persist as long as the user has not closed the browser.



EXERCISE 03 (2PTS)

Turn in: pool_php_10/ex_03/index.php

pool_php_10/ex_03/login.php

pool_php_10/ex_03/logout.php

Restrictions: None.

Start with the code of the preceding exercise.

You must now integrate a link “Logout” in the file index.php if the user is authenticated (below the greeting message “Hello Bob”).

The link will let the user disconnect and not be authenticated anymore.

After clicking the button, the user must be redirected to the login page.

EXERCISE 04 (3PTS)

Turn in: pool_php_10/ex_04/index.php

pool_php_10/ex_04/login.php

pool_php_10/ex_04/logout.php

Restrictions: None.

Start with the code of the preceding exercise.

You will now add a form field of type “checkbox” with attribute “name” of value “remember_me” (unchecked by default). If the user checks the box and his identification has succeeded, he will remain connected even if he closes the browser between two visits.

Of course, if the user logs out, he must not be connected on his new visit.



EXERCISE 05 (4PTS)

Turn in: pool_php_10/ex_05/index.php
pool_php_10/ex_05/login.php
pool_php_10/ex_05/logout.php
pool_php_10/ex_05/modify_account.php
Restrictions: None

You will create a page `modify_account.php` that will let users modify their account.
You can use the registration form of exercise 1.
The fields will be filled by default with the actual data of the connected account.
When the users modify the fields and validate the form, your database will be updated.
The password field and its confirmation should be empty.
You must also add a link "Settings" in the file `index.php` if the user is authenticated (underneath the link "Logout").
This link will let the user modify the data of his account by sending him to the page `modify_account.php`.
If a non-authenticated user arrives on the page `modify_account.php`, he will be redirected to `index.php`



EXERCISE 06 (4PTS)

Turn in: pool_php_10/ex_06/index.php

pool_php_10/ex_06/admin.php

Restrictions: None

You will now create an administration space to delete users.
Add manually the following account:

`"login => admin" | "password => admin" | "email => admin@admin.com" | "is_admin => TRUE";`

On the page index.php, create underneath the "Settings" link an "Admin settings" link that **only** appears if the connected user is an admin (`is_admin = TRUE`).

This link redirects the admin to the page admin.php.

The page admin.php will contain a bulleted list, each line being a link displaying the email address of a user.

Use of this link will delete the corresponding user in the database, if that user is not an administrator. All users must be listed and sorted alphabetically by email.

If you try to delete an administrator user, the same page will be displayed and the list will be preceded by the message "You can't delete an administrator".

Any individual not considered as administrator will be redirected to index.php in case of access to the page admin.php.



EXERCISE 07 (3PTS)

Turn in: pool_php_10/ex_07/index.php

pool_php_10/ex_07/newsletter.php

Restrictions: None

In this part, you will create a section that will allow administrators to create news.

The Page newsletter.php contains a form with two fields:

- A field of type "text" with attribute "name" set to "title", representing the title of the news. This title must be between 2 and 50 characters.
- A field of type "textarea" with attribute "name" set to "content", representing the content of the news. The content cannot be empty.

You must also add a validate button for the form containing the text "Submit".

When the form is validated, you must validate and save the newsletter in the database and fill the associated fields ("title", "content", "author_id"). Learn about SQL foreign keys to fill in the field "author_id".

In case of error, you will display again the form preceded by the error messages, such as "Invalid title" or "Invalid content". When the error is displayed, the two fields must be filled with the values entered previously by the user.

In case of success, you will redirect the user to page index.php

The page index.php must create a link "Create newsletter". The page index.php must also list the various news, sorted by descending id. The display of a news must have the following format:

> \$content, created by \$author.

\$author represents the field "name" of the user who created the news.