

Módulo 2 - PHP | Exercício 3 – AVILA Crew Website

Formulários e sessões em PHP

Temas abordados

- Construção de formulários de login
- Validação de dados
- Criação de sessões

Versão resolvida em: http://labmm.clients.ua.pt/deca_20L4/deca_20L4_p/avila03/

Neste exercício pretende-se criar um formulário de login para os mentores. Antes de começar, consulte o repositório no GitHub e obtenha o seguinte ficheiro:

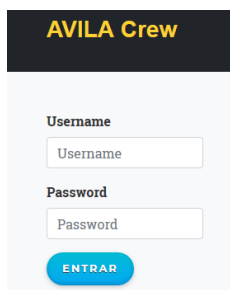
- Nova versão do ficheiro “data_team_members.php”

NOTA: Quem completou o exercício anterior com arrays pode fazer uma adaptação deste enunciado e continuar essa solução mais avançada.

Parte 1 – “cp_login_user.php”

Altere a página “login.php” para incluir o componente “cp_login_user.php”, descrito de seguida. O componente “login_user.php” consiste no formulário onde o mentor vai introduzir as suas credenciais. Para tal:

1. Crie o “components/cp_login_user.php”;
2. Nesse ficheiro, adicione um formulário de login que inclua dois campos (username e password).



Veja exemplos de formulário com Bootstrap em <http://getbootstrap.com/css/#forms>

3. O formulário deve utilizar o método “POST” para garantir que as credenciais do utilizador não são passadas no URL;
4. O atributo “action” do formulário deve apontar para o script “sc_login_control.php”, que será criado na parte seguinte do enunciado;
5. Altere a página “login.php” para que carregue o componente que acabou de criar.

Parte 2 – Script “sc_login_control.php”

Nesta parte será criada a lógica de verificação das credenciais introduzidas no formulário de login. A este bloco de código vamos designar por **Script** por se tratar de um ficheiro que se pretende executar diretamente, sem estar integrado numa página. Para tal:

6. Crie a pasta “scripts” na raiz do website “Avila”;
7. Crie um novo script “sc_login_control.php” na pasta criada no ponto anterior;
8. Neste script, adicione o código necessário para validar o username e palavra passe. Utilize os valores “username” e “pass” recebidos do formulário, através do método POST. Deve validar esses dados com as credenciais dos mentores, existentes no ficheiro “data_team_members.php”;
9. Se o login for válido deverá reencaminhar para a página “index.php”, caso contrário deverá voltar a abrir a página de login. Dicas:
 - a. Comece por validar apenas para o primeiro mentor. Depois de garantir que funciona para o primeiro, altere a condição para funcionar com os 3 mentores disponíveis no ficheiro de dados;
 - b. Para reencaminhar para outra página em PHP deve utilizar o seguinte método:
 - i. `header1('Location: destino.php');`
 - ii. Esta chamada tem de acontecer antes de qualquer output de HTML para a página,
10. Publique e verifique se tudo funciona como previsto!

Parte 3 – Sessões

Nesta parte do exercício vamos introduzir as alterações necessárias para que a navegação fique completa e permita saber, caso exista, o nome do utilizador que fez login com sucesso.

No script “sc_login_control.php” adicione o seguinte comportamento:

11. Caso as credenciais introduzidas no formulário sejam válidas então deve ser criada uma variável de sessão com o “username” do utilizador.
 - a. Não esquecer que é necessário iniciar a sessão!
 - b. As instruções para manipular variáveis de sessão têm de ser introduzidas antes de redirecionar para outra página.

No componente de navegação “cp_navigation.php” inclua a ligação para as páginas de login e logout. Efetue as seguintes alterações:

12. Caso não exista uma sessão de utilizador com login efetuado deve mostrar na barra de topo uma ligação para a página “login.php”;
13. Caso já exista uma sessão de utilizador com login efetuado deve:
 - a. Mostrar o username do utilizador na barra de topo;

¹ <http://php.net/manual/en/function.header.php>

- b. Nesse novo elemento, incluir um menu com uma opção que liga ao script “logout.php”. Esse script deve ser criado e tem como objetivo a destruição da sessão e redirecionar o utilizador para a página de login.

Dicas: Use o *Alert* disponível em:

<https://getbootstrap.com/docs/4.3/components/alerts/>

Parte 4 – Feedback ao utilizador no login

Sempre que o utilizador introduz credenciais erradas é redirecionado para a página de login mas não lhe é apresentada informação de feedback. Para apresentar a mensagem de erro adicione os seguintes comportamentos:

14. Se as credenciais são inválidas, crie uma variável de sessão “msg” com o valor “1” para sinalizar que é necessário dar feedback ao utilizador na página do formulário;
15. Na página de login (componente “cp_login_user.php”), caso necessário, mostre a seguinte mensagem de feedback:

```
<div class="alert alert-danger">  
  <strong>Erro!</strong> As credenciais não são válidas!  
</div>
```
16. Garanta que a mensagem de aviso não é repetida no próximo acesso à página de login.

AVANÇADO

1. Estructure a informação de login e password, do ficheiro “data_team_members.php”, em dois **arrays**.
2. Efetue as alterações necessárias no script “sc_login_control.php” para efetuar o login com base nos dados armazenados nos arrays.

NOTA: Caso já tenha uma solução totalmente baseada em arrays procure adicionar a informação de login e password na estrutura de dados anteriormente criada.

Disclaimer: Num sistema de informação real é inadmissível armazenar as passwords em texto sem qualquer tipo de encriptação. Neste exercício, tal é aceitável porque não é solicitada a introdução de qualquer password por parte dos utilizadores.

Como desafio, visite a página <http://php.net/manual/en/book.password.php> e explore o conceito de **hash** através dos métodos nativos do PHP: `password_hash()` e `password_verify()`.

Implemente estes conceitos no sistema de login deste exercício.