*Atividades login.php*

1. **Explique para que serve o arquivo login.php no contexto de um sistema administrativo.**

Serve para maior controle de usuários, contando que tem como objetivo processar dados das pessoas que utilizarão o site, portanto cada usuário deve ter uma conta, na qual vai conter suas credenciais, como nome, CPF, e outros dados pessoais. Este arquivo terá a função inicial do site, a primeira página que aparecerá, sendo a página de Login, para acessar a conta de algum usuário, já existente.

1. **Como o arquivo login.php utiliza o Bootstrap, e qual é a vantagem dessa integração?**

O formulário, onde contém as caixas para resposta das informações do usuário, como e-mail e senha, foi desenvolvido pelo framework, portanto não precisamos gastar tempo para fazê-lo, apenas mudar a aparência com o CSS. Utilizar o framework resulta em um maior proveito do horário, deixando quem programa esse sistema com um maior foco em outras coisas além disso.

1. **Qual a importância de definir os atributos required nos campos de entrada do formulário?**

Seu funcionamento é para que nenhum campo do formulário não seja preenchido, ou seja, esteja em branco.

1. **Discorra sobre o papel do método POST na submissão do formulário em login.php.**

É a forma de interação do usuário com o banco de dados, sendo duas formas ‘GET’, ‘POST’. Ambos são a forma que o usuário interage, porém o GET, passa as informações digitadas para a URL do site também, tornando suas declarações totalmente públicas. Contudo o POST serve para armazenar essas informações de forma mais segura, na requisição da URL, não deixando as credenciais cadastradas visíveis.

*Atividades processa\_login\_admin.php*

1. **Explique como a validação de senha é realizada no script processa\_login\_admin.php.**

a) Primeiro o num\_rows verifica se há resultados existentes no banco de dados com o que o usuário escreveu. - Abre o primeiro IF

b) O fetch\_assoc torna o resultado um array

c) Já o password\_verify verifica se a senha dada pelo usuário é a mesma senha, criptografada (hash), que está no banco de dados. - Abre o segundo IF

d) Se a senha estiver correta ela inicia duas variáveis de sessão:

I - A primeira é para afirmar se o usuário está logado(true) ou não(false).

II - A outra serve para armazenar o nome do usuário

e) E depois redireciona o usuário para a página list\_products.php

f) Se não (do segundo IF) vai aparecer a mensagem: Nome de usuário ou senha inválidos.

g) Se não (do primeiro IF) vai aparecer a mensagem: Nome de usuário ou senha inválidos.

h) O close() fecha a declaração SQL($stmt) e a conexão com o banco de dados($conn) para liberar recursos.

1. **Como o script processa\_login\_admin.php gerencia a sessão do usuário após uma autenticação bem-sucedida?**

A primeira é para afirmar se o usuário está logado(true) ou não(false). A outra serve para armazenar o nome do usuário. Logo após abrirá a página list\_products.php

1. **Qual é a função do arquivo processa\_login\_admin.php e como ele se relaciona com o login.php?**

De duas formas principais, a primeiraaction="processa\_login\_admin.php", presente no login.php, que serve para relacionar a duas páginas, uma com a outra. A outra é name=”” que serve para quando você quiser se referir à resposta dada pelo usuário pelo $\_POST['password'] .

1. **Quais medidas de segurança são implementadas no processa\_login\_admin.php para proteger contra injeção de SQL?**

Prepared statements (declarações preparadas) são uma técnica de segurança que permite que você defina uma consulta SQL com placeholders (marcadores de parâmetros) em vez de incluir diretamente os valores dos parâmetros na consulta.

Ao usar prepared statements, o MySQL trata os valores dos parâmetros como dados e não como parte do código SQL. Isso impede que qualquer valor fornecido pelo usuário seja interpretado como código SQL, prevenindo a injeção de SQL.

* $stmt = $conn->prepare("SELECT \* FROM administradores WHERE username = ?");
* $stmt->bind\_param("s", $username);
* $stmt->execute();