|  |
| --- |
| **puc obtenir en php un mysql\_native\_password?** |

Sí, en **PHP** es pot generar un **hash de tipus mysql\_native\_password**, però cal entendre què és exactament.

**🔐 Què és mysql\_native\_password?**

És l'algoritme antic de MySQL/MariaDB per guardar contrasenyes. No fa servir password\_hash() com PHP modern, sinó un **SHA1 doble**.

El hash que genera comença amb un asterisc i té 41 caràcters, per exemple:

\*94BDCEBE19083CE2A1F959FD02F964C7AF4CFC29

**🛠️ Com generar-lo en PHP (compatible amb mysql\_native\_password):**

<?php

function mysql\_native\_password($password) {

$stage1 = sha1($password, true); // binari

$stage2 = sha1($stage1); // hexadecimal

return '\*' . strtoupper($stage2);

}

**🔍 Exemple:**

$pwd = "hola123";

$hash = mysql\_native\_password($pwd);

echo $hash;

// → \*94BDCEBE19083CE2A1F959FD02F964C7AF4CFC29

**⚠️ Importància**

* Aquest hash **no és recomanat per a sistemes nous** perquè és **vulnerable** comparat amb password\_hash().
* Només té sentit si esteu autenticant usuaris contra una base de dades **MariaDB/MySQL que encara usa mysql\_native\_password** com a plugin d’autenticació.

**Vols que t’ajudi a usar aquest hash per comparar-lo amb el que tens a la taula personal?**

|  |
| --- |
| **els inputs no surten aliniats:** |

Els **inputs no surten alineats** perquè estàs posant els elements <label> i <input> sense un contenidor clar que els alinee. A més, HTML5 **no permet posar <tr> ni <td> fora d’una taula**, i això també pot fer que el navegador es comporti de manera inesperada.**✅ Solució simple amb HTML i CSS**

Et recomano que **usis un contenidor div per a cada línia** i una mica de CSS bàsic per alinear els label i els input.

**🔧 HTML corregit:**

<!DOCTYPE html>

<html>

<head>

<style>

.form-group {

margin-bottom: 10px;

}

label {

display: inline-block;

width: 100px; /\* ajusta segons calgui \*/

text-align: right;

margin-right: 10px;

}

input[type="text"],

input[type="password"] {

width: 200px;

}

.centrat {

width: 400px;

margin: 50px auto;

}

</style>

</head>

<body>

<div class="centrat">

<form action="/1-Restaurant/login-app/login.php" method="POST">

<div class="form-group">

<label for="username">Username</label>

<input type="text" name="username" placeholder="username" required>

</div>

<div class="form-group">

<label for="password">Password</label>

<input type="text" name="password" placeholder="password" required>

</div>

<div class="form-group">

<label for="host">Host</label>

<input type="text" name="host" placeholder="host" required>

</div>

<div class="form-group">

<input type="submit" value="Entrar">

</div>

</form>

</div>

</body>

</html>

**✅ Resultat:**

* Els label apareixen alineats a l'esquerra.
* Els input apareixen alineats perfectament a la dreta.
* El formulari està centrat a la pàgina.

|  |
| --- |
| **no funciona be** |

El teu HTML i JavaScript tenen diversos problemes que impedeixen que funcioni correctament. Et detallo **els errors principals** i et dono una **versió corregida i funcional** al final.**❌ Problemes detectats:**

1. **document.write("elMeuError")** escriu literalment "elMeuError", no el contingut de cap variable.
2. Intentes accedir a un element amb id="error" però **no existeix cap element HTML amb aquest id**.
3. Estàs fent servir document.write() **després del renderitzat inicial**, la qual cosa pot **esborrar tot el contingut anterior**.
4. Tens comentaris HTML mal tancats que poden confondre la lectura del codi.
5. Estàs intentant manipular l'element elMeuError abans que existeixi.

**✅ Solució recomanada**

Aquí tens una versió neta i funcional del teu fitxer error.html que mostra el missatge d'error segons els paràmetres error i lloc de la URL:

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title>403 - Acceso Denegado</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #fafafa;

text-align: center;

padding: 50px;

}

.error-box {

display: inline-block;

padding: 30px;

border: 1px solid #ccc;

background-color: white;

border-radius: 10px;

box-shadow: 0 0 10px rgba(0,0,0,0.1);

}

h1 {

font-size: 60px;

color: #e74c3c;

}

p {

font-size: 18px;

color: #555;

}

.error {

color: red;

margin-top: 20px;

}

a {

text-decoration: none;

color: #3498db;

}

a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="error-box">

<h1>403 - Accés Denegat</h1>

<p>No tens permís per veure aquesta pàgina.</p>

<p class="error" id="missatge-error"></p>

<p><a href="index.html">Tornar a l'inici</a></p>

</div>

<script>

const params = new URLSearchParams(window.location.search);

const error = params.get('error');

const lloc = params.get('lloc');

const missatge = document.getElementById("missatge-error");

if (lloc) {

missatge.innerHTML += lloc + "<br>";

}

if (error === 'camps\_buits') {

missatge.innerHTML += "Cal completar tots els camps.";

} else if (error === 'no\_existeix\_usuari\_al\_host') {

missatge.innerHTML += "No existeix usuari al host.";

} else if (error) {

missatge.innerHTML += "Error no controlat: " + error;

}

</script>

<br><br>

<form action="./index.html">

<input type="submit" value="Tornar a l'INICI">

</form>

</body>

</html>

**🔄 Exemple de crida funcional:**

error.html?error=no\_existeix\_usuari\_al\_host&lloc=welcome.php

Vols que et mostri també com redirigir des de PHP amb aquests paràmetres?

|  |
| --- |
| **què és això: function manejador\_excepciones($excepción) {**  **echo "<h3>Error: " , $excepción->getMessage(), "<h3>";**  **echo "<h3><a href='../index.php'>Anar a l'index</a><h3>\n";**  **}**  **set\_exception\_handler('manejador\_excepciones');** |

Això que has mostrat és una **funció personalitzada de maneig d'excepcions en PHP**, juntament amb la crida a set\_exception\_handler() per registrar-la.**🔍 Què fa exactament aquest codi?**

function manejador\_excepciones($excepción) {

echo "<h3>Error: " , $excepción->getMessage(), "<h3>";

echo "<h3><a href='../index.php'>Anar a l'index</a><h3>\n";

}

set\_exception\_handler('manejador\_excepciones');

**1. manejador\_excepciones($excepción)**

És una funció que s'executarà automàticament **quan es llanci una excepció (throw) i no es capturi amb try/catch**. El paràmetre $excepción és un objecte de tipus Exception.

**2. set\_exception\_handler('manejador\_excepciones')**

Registra la funció "manejador\_excepciones" com a **gestor global d’excepcions no controlades**. Això vol dir que si el teu programa fa:

throw new Exception("Això ha fallat");

i no hi ha cap try { ... } catch { ... }, PHP cridarà automàticament manejador\_excepciones().**🔧 Què fa en aquest cas concret?**

Si salta una excepció no controlada:

* Mostra el missatge d’error amb <h3>Error: ...</h3>
* Afegeix un enllaç per tornar a la pàgina principal (../index.php)

**🧪 Exemple pràctic:**

function manejador\_excepciones($excepción) {

echo "<h3>Error: " , $excepción->getMessage(), "<h3>";

echo "<h3><a href='../index.php'>Anar a l'index</a><h3>\n";

}

set\_exception\_handler('manejador\_excepciones');

// Simulem un error

throw new Exception("La connexió ha fallat");

**Sortida:**

Error: La connexió ha fallat

[Anar a l'index]

**✅ Avantatges**

* Et permet mostrar missatges d’error més **elegants i controlats**.
* Pots **redirigir l’usuari o registrar** errors crítics.
* Evites que l’usuari vegi missatges d’error del sistema.

Vols que et mostri com combinar aquest gestor d'excepcions amb try/catch per cobrir tant errors controlats com no controlats?

|  |
| --- |
| **si** |

Perfecte! Ara t’explico com combinar el **gestor global d’excepcions** amb el bloc try/catch per controlar millor els errors tant **controlats** com **no controlats**.

**🔧 1. Definim el gestor d’excepcions global**

Aquest codi s'executarà quan una excepció **no sigui capturada** amb try/catch.

function manejador\_excepciones($excepcio) {

echo "<h3>Error global: " . $excepcio->getMessage() . "</h3>";

echo "<a href='index.php'>Tornar a l'inici</a>";

}

set\_exception\_handler('manejador\_excepciones');

**🧪 2. Exemple amb try/catch**

try {

// Suposem que fem una connexió a la base de dades o altra acció crítica

$connexio = new mysqli("localhost", "usuari", "contrasenya", "basedades");

if ($connexio->connect\_error) {

// Aquí generem una excepció manual si falla

throw new Exception("No s'ha pogut connectar: " . $connexio->connect\_error);

}

echo "Connexió correcta!";

// Simulem un altre error

// throw new Exception("Un altre error inesperat");

} catch (Exception $e) {

// Aquí controlem l’error dins del bloc try

echo "<h3>Error capturat amb try/catch: " . $e->getMessage() . "</h3>";

}

**🎯 3. Com funciona tot junt**

* Si **hi ha un throw dins un try/catch**, s’executa **el catch**.
* Si llences una excepció **fora d’un try/catch**, entra al **gestor global** (manejador\_excepciones()).

**💡 Recomanació pràctica**

Pots usar el try/catch per controlar errors que esperes (connexió, consulta SQL, etc.) i deixar el set\_exception\_handler() per **errors inesperats o globals**.

Vols que t’adapti aquest exemple per a un fitxer funcions.php amb la connexió dins d’una funció?

|  |
| --- |
| **i on es defineix el set\\_exception\\_handler(), a la configuració de php?** |

No — set\_exception\_handler() **no es defineix a la configuració de PHP** (com el php.ini), sinó que **l’has de definir tu mateix en el teu codi PHP**.

**📌 On col·locar set\_exception\_handler()?**

Normalment, ho poses **al principi del teu script principal**, o millor encara, en un fitxer comú de configuració (per exemple, init.php, funcions.php, o config.php), que després inclous a tots els teus scripts.

**🧱 Exemple recomanat:**

**fitxer config.php o funcions.php:**

<?php

function manejador\_excepciones($ex) {

echo "<h3>Error global: " . $ex->getMessage() . "</h3>";

echo "<a href='index.php'>Tornar a l'inici</a>";

}

set\_exception\_handler('manejador\_excepciones');

?>

**En qualsevol script PHP important:**

<?php

require\_once("config.php"); // Aquí registres el manejador

// El teu codi aquí...

**🔒 Notes importants**

* Si no poses set\_exception\_handler(), PHP mostrarà el típic error tècnic (pot ser perillós en producció).
* Pots posar aquest manejador només un cop per cada execució del codi. Si ho tornes a cridar, sobreescrius l’anterior.

Vols que et prepari un exemple amb un fitxer init.php que puguis incloure a totes les pàgines per inicialitzar sessió, connexió i gestió d’errors?

|  |
| --- |
| **si** |

Perfecte! Aquí tens un exemple complet d’un fitxer init.php que pots incloure a totes les teves pàgines PHP per:

✅ Iniciar la sessió  
✅ Connectar amb la base de dades  
✅ Configurar el gestor d’excepcions**📄 Fitxer init.php**

<?php

session\_start(); // Sempre primer!

// 📌 Gestor d'excepcions global

function manejador\_excepciones($ex) {

echo "<h3>Error global: " . $ex->getMessage() . "</h3>";

echo "<a href='/1-Restaurant/index.php'>Tornar a l'inici</a>";

}

set\_exception\_handler('manejador\_excepciones');

// 📌 Funció per connectar a la base de dades

function getConnexio() {

// Si vols permetre connexions per múltiples usuaris

$usuari = $\_SESSION['username'] ?? 'usuari\_per\_defecte';

$contrasenya = $\_SESSION['password'] ?? '';

$host = $\_SESSION['host'] ?? 'localhost';

$bd = 'restaurant'; // Canvia-ho pel nom real de la base de dades

$conn = new mysqli($host, $usuari, $contrasenya, $bd);

if ($conn->connect\_error) {

throw new Exception("Connexió fallida: " . $conn->connect\_error);

}

return $conn;

}

?>

**🧾 Com fer-lo servir**

A qualsevol pàgina PHP on vulguis fer ús d'això, **inclou-lo al principi**:

<?php

require\_once("init.php");

$conn = getConnexio(); // Ara tens connexió a la base de dades

// El teu codi continua...

**✅ Avantatges**

* Centralitzes la connexió i la gestió d'errors.
* Pots reutilitzar-lo a totes les teves pàgines.
* Si canvies lògica, només ho fas en un lloc.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

|  |
| --- |
| **en php com obtinc informaciò (ip, nom...) del localhost ?** |

En PHP pots obtenir **informació del servidor local (localhost)**, com la IP, el nom del servidor, etc., a través de **variables superglobals** com $\_SERVER.

Aquí tens les més útils:**🌐 Informació del servidor i client**

<?php

echo "IP del client: " . $\_SERVER['REMOTE\_ADDR'] . "<br>";

echo "Nom del servidor: " . $\_SERVER['SERVER\_NAME'] . "<br>";

echo "IP del servidor: " . $\_SERVER['SERVER\_ADDR'] . "<br>";

echo "Nom de l'host local (hostname): " . gethostname() . "<br>";

echo "Software del servidor: " . $\_SERVER['SERVER\_SOFTWARE'] . "<br>";

echo "Port del servidor: " . $\_SERVER['SERVER\_PORT'] . "<br>";

echo "Agent del navegador: " . $\_SERVER['HTTP\_USER\_AGENT'] . "<br>";

?>

**🧠 Explicació ràpida**

| **Variable** | **Què retorna?** |
| --- | --- |
| $\_SERVER['REMOTE\_ADDR'] | IP del client que accedeix a la pàgina |
| $\_SERVER['SERVER\_ADDR'] | IP del servidor (localhost en local) |
| $\_SERVER['SERVER\_NAME'] | Nom del servidor (ex: localhost) |
| gethostname() | Nom de màquina del servidor (host local) |
| $\_SERVER['SERVER\_SOFTWARE'] | Software del servidor (Apache, Nginx...) |
| $\_SERVER['HTTP\_USER\_AGENT'] | Informació del navegador de l’usuari |

**⚠️ Nota**

* En local (localhost), REMOTE\_ADDR sovint serà 127.0.0.1.
* Si treballes darrere d’un proxy o reverse proxy, pot ser necessari consultar també $\_SERVER['HTTP\_X\_FORWARDED\_FOR'].

**Vols mostrar aquestes dades només si estàs en entorn local (localhost) i ocultar-les en producció?**

|  |
| --- |
| **és possible en php permetre l'accés a un namespace (a un lloc o pagina web del localhost ) en concret?** |

En PHP, **no pots restringir directament l'accés a un "namespace"** (en el sentit de codi namespace de PHP), però **sí que pots controlar l'accés a una pàgina o ruta concreta del teu servidor** — per exemple, una pàgina que fa ús d’un namespace.

Això s’aconsegueix controlant l’accés mitjançant:**✅ 1. Control per IP (per exemple, només localhost)**

<?php

if ($\_SERVER['REMOTE\_ADDR'] !== '127.0.0.1') {

header('HTTP/1.1 403 Forbidden');

exit('Accés denegat. Només disponible des de localhost.');

}

?>

**✅ 2. Control per sessió/login/rol**

<?php

session\_start();

if (!isset($\_SESSION['rol']) || $\_SESSION['rol'] !== 'admin') {

header('Location: /403.html');

exit();

}

?>

**✅ 3. Combinat amb namespaces (organització del codi)**

Pots tenir codi organitzat amb namespace, però això **no controla l'accés**. L'accés s'ha de controlar en el punt d’entrada (scripts .php) que usen aquests namespace.

Exemple:

<?php

// arxiu a /admin/pagina\_secreta.php

require\_once('../init.php');

use Admin\Controlador\Secrets;

if ($\_SERVER['REMOTE\_ADDR'] !== '127.0.0.1') {

header('HTTP/1.1 403 Forbidden');

exit("Només per localhost.");

}

Secrets::mostrar();

**🔐 Conclusió**

* **Els namespaces només organitzen codi, no gestionen accés.**
* **El control d’accés s’ha de fer a nivell d’arxiu o URL**, via IP, sessió, o algun sistema de permisos.
* També pots afegir regles a .htaccess o a la configuració d’Apache/Nginx si cal fer-ho des del servidor.

Vols que et prepari un exemple real amb una carpeta /admin protegida només per accés local i sessió iniciada?

|  |
| --- |
| **si** |

Perfecte! Aquí tens un **exemple complet** d’una estructura on una carpeta /admin només és accessible:

* si l’usuari està **loguejat**
* i accedeix **des de localhost (127.0.0.1)**

**📁 Estructura de fitxers**

/1-Restaurant/

│

├── index.php ← login

├── init.php ← configuració comú (connexió, sessió)

├── login.php ← gestiona login

├── logout.php ← tanca sessió

├── /admin/

│ └── secrets.php ← protegit!

**🧠 init.php (inicialització comuna)**

<?php

session\_start();

function manejador\_excepciones($ex) {

echo "<h3>Error: " . $ex->getMessage() . "</h3>";

echo "<a href='/1-Restaurant/index.php'>Tornar</a>";

}

set\_exception\_handler('manejador\_excepciones');

function getConnexio() {

$host = $\_SESSION['host'] ?? 'localhost';

$usuari = $\_SESSION['username'] ?? 'usuari';

$pwd = $\_SESSION['password'] ?? '';

$bd = 'restaurant';

$conn = new mysqli($host, $usuari, $pwd, $bd);

if ($conn->connect\_error) throw new Exception("Error connexió DB");

return $conn;

}

**🔐 /admin/secrets.php**

<?php

require\_once('../init.php');

// Protecció IP: només localhost

if ($\_SERVER['REMOTE\_ADDR'] !== '127.0.0.1') {

header('HTTP/1.1 403 Forbidden');

exit('Accés només permès des de localhost.');

}

// Protecció per sessió

if (!isset($\_SESSION['username'])) {

header('Location: /1-Restaurant/index.php?error=acces\_denegat');

exit();

}

echo "<h2>Benvingut a la zona secreta!</h2>";

echo "<p>Usuari autenticat: " . htmlspecialchars($\_SESSION['username']) . "</p>";

**🔑 login.php (simplificat)**

<?php

require\_once('init.php');

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$\_SESSION['username'] = $\_POST['username'];

$\_SESSION['password'] = $\_POST['password'];

$\_SESSION['host'] = $\_POST['host'];

header("Location: /1-Restaurant/admin/secrets.php");

exit();

}

**🧪 Prova-ho**

1. Ves a index.php i inicia sessió.
2. Accedeix a /admin/secrets.php
3. Si **no estàs en localhost**, tindràs un error 403.
4. Si **no tens sessió**, et redirigirà al login.

Vols que afegim també protecció per rol (com per exemple només si ets "admin")?

|  |
| --- |
| **si** |

Perfecte! Afegirem **una protecció per rol** de manera que només els usuaris amb rol "admin" puguin accedir a /admin/secrets.php.**🛠️ Supòsits**

Suposem que la taula personal té una columna rol amb valors com "admin" o "empleat".

**🔄 1. Modifiquem el login.php per guardar el rol a la sessió:**

<?php

require\_once('init.php');

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$username = $\_POST['username'];

$password = $\_POST['password'];

$host = $\_POST['host'];

$\_SESSION['username'] = $username;

$\_SESSION['password'] = $password;

$\_SESSION['host'] = $host;

$conn = getConnexio();

$stmt = $conn->prepare("SELECT rol FROM personal WHERE username = ?");

$stmt->bind\_param("s", $username);

$stmt->execute();

$result = $stmt->get\_result();

if ($row = $result->fetch\_assoc()) {

$\_SESSION['rol'] = $row['rol']; // <- Guardem el rol a la sessió

header("Location: /1-Restaurant/admin/secrets.php");

exit();

} else {

echo "Usuari no trobat.";

}

}

**🔐 2. Afegim la validació del rol a /admin/secrets.php:**

<?php

require\_once('../init.php');

// ✅ Només localhost

if ($\_SERVER['REMOTE\_ADDR'] !== '127.0.0.1') {

header('HTTP/1.1 403 Forbidden');

exit('Accés només permès des de localhost.');

}

// ✅ Sessió iniciada

if (!isset($\_SESSION['username'])) {

header('Location: /1-Restaurant/index.php?error=no\_login');

exit();

}

// ✅ Només usuaris amb rol "admin"

if (!isset($\_SESSION['rol']) || $\_SESSION['rol'] !== 'admin') {

header('HTTP/1.1 403 Forbidden');

exit('Només administradors poden accedir aquí.');

}

echo "<h2>Zona d'administració</h2>";

echo "<p>Usuari: " . htmlspecialchars($\_SESSION['username']) . " (rol: " . $\_SESSION['rol'] . ")</p>";

**🔓 3. Per tancar sessió (logout.php):**

<?php

session\_start();

session\_unset();

session\_destroy();

header("Location: /1-Restaurant/index.php");

exit();

Ara només un usuari amb el rol "admin" i des de localhost pot accedir a /admin/secrets.php.

Vols afegir una vista diferent per a usuaris amb rol "empleat"?

|  |
| --- |
| **si** |

Perfecte! Afegirem una **vista alternativa per als usuaris amb rol "empleat"**, així com una redirecció automàtica segons el rol des del login.php.**🎯 Objectiu**

* Si és "admin": va a /admin/secrets.php
* Si és "empleat": va a /empleat/intranet.php
* Si no té cap rol reconegut: mostra error

**🧭 1. login.php amb redirecció segons rol**

<?php

require\_once('init.php');

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$username = trim($\_POST['username']);

$password = trim($\_POST['password']);

$host = trim($\_POST['host']);

$\_SESSION['username'] = $username;

$\_SESSION['password'] = $password;

$\_SESSION['host'] = $host;

$conn = getConnexio();

$stmt = $conn->prepare("SELECT rol FROM personal WHERE username = ?");

$stmt->bind\_param("s", $username);

$stmt->execute();

$result = $stmt->get\_result();

if ($row = $result->fetch\_assoc()) {

$\_SESSION['rol'] = $row['rol'];

if ($row['rol'] === 'admin') {

header("Location: /1-Restaurant/admin/secrets.php");

} elseif ($row['rol'] === 'empleat') {

header("Location: /1-Restaurant/empleat/intranet.php");

} else {

echo "Rol no reconegut.";

}

exit();

} else {

echo "Usuari no trobat.";

}

}

**🧱 2. Crea carpeta i vista d’empleat: /empleat/intranet.php**

<?php

require\_once('../init.php');

if (!isset($\_SESSION['username'])) {

header("Location: /1-Restaurant/index.php?error=no\_login");

exit();

}

// Només permet empleats

if ($\_SESSION['rol'] !== 'empleat') {

header('HTTP/1.1 403 Forbidden');

exit('Accés només permès per empleats.');

}

echo "<h2>Benvingut/da, " . htmlspecialchars($\_SESSION['username']) . "</h2>";

echo "<p>Aquesta és la intranet per empleats.</p>";

echo '<a href="/1-Restaurant/logout.php">Tancar sessió</a>';

**✅ 3. Prova-ho**

* Entra amb un usuari rol = 'empleat' → et portarà a la intranet.
* Entra amb rol = 'admin' → et portarà a la zona d’administració.
* Intenta accedir manualment a /admin/secrets.php amb un empleat → veuràs error.

Vols que els empleats també puguin veure el seu nom o funcions específiques a la intranet?

|  |
| --- |
| **si** |

Perfecte! T'ensenyaré com mostrar el **nom complet de l’usuari empleat** i afegir algunes funcionalitats específiques dins de la **intranet**.

**🎯 Objectiu**

* Mostrar: nom, rol i hora d’entrada.
* Afegir: enllaços útils per empleats (ex. veure reserves, canviar contrasenya).
* Tot protegit només per empleats.

**🧱 1. Extensió de /empleat/intranet.php:**

<?php

require\_once('../init.php');

if (!isset($\_SESSION['username'])) {

header("Location: /1-Restaurant/index.php?error=no\_login");

exit();

}

// Només permet empleats

if ($\_SESSION['rol'] !== 'empleat') {

header('HTTP/1.1 403 Forbidden');

exit('Accés només permès per empleats.');

}

// Obtenim informació de l'empleat des de la base de dades

$conn = getConnexio();

$stmt = $conn->prepare("SELECT nom, rol FROM personal WHERE username = ?");

$stmt->bind\_param("s", $\_SESSION['username']);

$stmt->execute();

$result = $stmt->get\_result();

if ($row = $result->fetch\_assoc()) {

$nom = $row['nom'];

$rol = $row['rol'];

$hora = date('H:i:s');

} else {

$nom = $\_SESSION['username'];

$rol = "desconegut";

$hora = date('H:i:s');

}

?>

<!DOCTYPE html>

<html lang="ca">

<head>

<meta charset="UTF-8">

<title>Intranet Empleat</title>

</head>

<body>

<h2>Hola, <?= htmlspecialchars($nom) ?>!</h2>

<p><strong>Rol:</strong> <?= htmlspecialchars($rol) ?></p>

<p><strong>Hora d’entrada:</strong> <?= $hora ?></p>

<h3>Funcions disponibles</h3>

<ul>

<li><a href="/1-Restaurant/empleat/veure\_reserves.php">

📅 Veure reserves del dia</a></li>

<li><a href="/1-Restaurant/empleat/canvi\_contrasenya.php">

🔑 Canviar contrasenya</a></li>

<li><a href="/1-Restaurant/logout.php">🚪 Tancar sessió</a></li>

</ul>

</body>

</html>

**📝 2. Coses que pots fer després**

* Crear veure\_reserves.php perquè l'empleat vegi les reserves d’avui.
* Crear canvi\_contrasenya.php perquè pugui actualitzar la seva contrasenya.
* Pots afegir també la seva IP o dispositiu d’accés si vols més registre.

|  |
| --- |
| **un cop fet el login a una aplicació php (amb usuari root), com un usuari amb pocs privilegis continua amb la sessió?** |

Molt bona pregunta. En una aplicació PHP que utilitza MariaDB, **la sessió d’usuari a la base de dades no es manté entre peticions HTTP**. Cada vegada que accedeixes a la base de dades amb mysqli\_connect(), es crea una **nova connexió**.

Això vol dir que:

**✅ Després del login:**

* Encara que facis login com a **root**, no es manté una connexió permanent com a root.
* El que fas realment és **emmagatzemar les credencials (usuari, host, contrasenya)** a $\_SESSION, i després utilitzar-les per obrir connexions noves en cada pàgina on cal.

**💡 Solució pràctica:**

1. **Després del login amb root**, valides un usuari "normal" de la base de dades (amb pocs privilegis).
2. Desa les seves dades a la sessió:

$\_SESSION['db\_user'] = 'nom\_usuari';

$\_SESSION['db\_pass'] = 'contrasenya';

$\_SESSION['db\_host'] = 'localhost';

1. I quan cal fer consultes, uses una funció com aquesta:

function getConnexio() {

$usuari = $\_SESSION['db\_user'];

$contrasenya = $\_SESSION['db\_pass'];

$host = $\_SESSION['db\_host'];

$database = 'restaurant';

$conn = mysqli\_connect($host, $usuari, $contrasenya, $database);

if (!$conn) {

die("Error de connexió: " . mysqli\_connect\_error());

}

return $conn;

}

**🛡️ Bona pràctica**

* **No facis servir l’usuari root** per accedir a la base de dades després del login. Usa’l només per validar i després treballa amb usuaris limitats.
* Dona permisos concrets als usuaris MariaDB segons el seu rol:
* GRANT SELECT, INSERT ON restaurant.\* TO 'empleat'@'localhost' IDENTIFIED BY '1234';

Vols que t’ajudi a crear un exemple amb un usuari MariaDB amb permisos limitats que es pugui usar després del login?

|  |
| --- |
| **si** |

Perfecte! Et mostraré com fer-ho pas a pas:**🎯 Objectiu**

Crear un sistema on:

1. Et connectes inicialment amb un **usuari validat per PHP** (ex. root).
2. Validem les credencials d’un **usuari MariaDB amb permisos limitats** (ex. empleat1).
3. Un cop validat, es treballa **només** amb aquest usuari limitat durant tota la sessió.

**🛠️ 1. Crear un usuari MariaDB limitat**

Obre el terminal de MySQL/MariaDB com root i executa:

CREATE USER 'empleat1'@'localhost' IDENTIFIED BY '1234';

GRANT SELECT, INSERT ON restaurant.\* TO 'empleat1'@'localhost';

FLUSH PRIVILEGES;

Això crea un usuari empleat1 que pot llegir i inserir dades dins la base de dades restaurant.**🧠 2. Fitxer login.php**

Aquest fitxer fa el login bàsic (ex: validant contra una taula personal) i desa les credencials del nou usuari a la sessió:

<?php

session\_start();

require\_once("../functions/funcions.php");

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$usuari = $\_POST['username'];

$password = $\_POST['password'];

$host = 'localhost';

$bd = 'restaurant';

// Validació simple (ex: comprovació amb root)

$connRoot = mysqli\_connect($host, 'root', 'CONTRASENYA\_ROOT', $bd);

$stmt = $connRoot->prepare("SELECT \* FROM personal WHERE username = ? AND password = ?");

$stmt->bind\_param("ss", $usuari, $password);

$stmt->execute();

$result = $stmt->get\_result();

if ($row = $result->fetch\_assoc()) {

// Un cop validat, ara es treballarà amb el seu usuari MariaDB:

$\_SESSION['db\_user'] = $usuari;

$\_SESSION['db\_pass'] = $password;

$\_SESSION['db\_host'] = $host;

$\_SESSION['rol'] = $row['rol'];

header("Location: /1-Restaurant/empleat/intranet.php");

exit();

} else {

header("Location: /1-Restaurant/error.html?error=usuari\_incorrecte");

exit();

}

}

?>

**🔌 3. Funció getConnexio()**

A functions/funcions.php:

<?php

function getConnexio() {

$usuari = $\_SESSION['db\_user'];

$pass = $\_SESSION['db\_pass'];

$host = $\_SESSION['db\_host'];

$database = 'restaurant';

$conn = mysqli\_connect($host, $usuari, $pass, $database);

if (!$conn) {

die("Error de connexió a la base de dades.");

}

return $conn;

}

?>

**✅ 4. Exemple d’ús (intranet)**

<?php

session\_start();

require\_once("../functions/funcions.php");

if (!isset($\_SESSION['db\_user'])) {

header("Location: ../index.php");

exit();

}

$conn = getConnexio();

$result = mysqli\_query($conn, "SELECT \* FROM reserves WHERE data = CURDATE()");

?>

<h2>Reserves d'avui:</h2>

<ul>

<?php while ($row = mysqli\_fetch\_assoc($result)) : ?>

<li><?= $row['nom\_client'] ?> - <?= $row['hora'] ?></li>

<?php endwhile; ?>

</ul>

**🔐 Notes de seguretat**

* No guardis la contrasenya en text pla si ho pots evitar.
* Pots encriptar contrasenyes (ex. amb password\_hash()) i fer que l'usuari entri com a empleat1, però validis la seva contrasenya abans de fer mysqli\_connect().

Vols que t’ajudi a fer un formulari de registre segur amb password\_hash() i que creï usuaris limitats automàticament?

|  |
| --- |
| **si** |

Perfecte! Ara et mostraré **com crear un formulari de registre segur** que:

1. Desa l’usuari en una taula personal amb password\_hash().
2. Crea automàticament un **usuari MariaDB** amb permisos limitats per accedir a la base de dades (SELECT, INSERT).
3. Aquest usuari s’utilitzarà després per establir connexions durant la sessió.

**🗃️ 1. Estructura de la taula personal**

Assegura't que tens una taula com aquesta:

CREATE TABLE personal (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) UNIQUE,

password\_hash VARCHAR(255),

rol VARCHAR(20)

);

**🧾 2. Formulari register.html**

<form action="register.php" method="POST">

<label>Nom d'usuari:</label>

<input type="text" name="username" required><br>

<label>Contrasenya:</label>

<input type="password" name="password" required><br>

<label>Rol:</label>

<select name="rol">

<option value="empleat">Empleat</option>

<option value="cuiner">Cuiner</option>

</select><br><br>

<input type="submit" value="Registrar">

</form>

**🔐 3. Script register.php**

<?php

require\_once("../functions/funcions.php"); // connecta com a root

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$username = trim($\_POST['username']);

$password = $\_POST['password'];

$rol = $\_POST['rol'];

$host = 'localhost';

$bd = 'restaurant';

// Hashejar la contrasenya

$hash = password\_hash($password, PASSWORD\_DEFAULT);

// Connexió com a root

$conn = mysqli\_connect($host, 'root', 'CONTRASENYA\_ROOT', $bd);

if (!$conn) die("Error de connexió: " . mysqli\_connect\_error());

// 1. Inserir a la taula personal

$stmt = $conn->prepare("INSERT INTO personal (username, password\_hash, rol) VALUES (?, ?, ?)");

$stmt->bind\_param("sss", $username, $hash, $rol);

$stmt->execute();

// 2. Crear usuari MariaDB amb permisos bàsics

$createUserSQL = "CREATE USER '$username'@'localhost' IDENTIFIED BY '$password'";

$grantSQL = "GRANT SELECT, INSERT ON $bd.\* TO '$username'@'localhost'";

if (mysqli\_query($conn, $createUserSQL)) {

mysqli\_query($conn, $grantSQL);

echo "Usuari registrat correctament.";

} else {

echo "Error: podria ser que l'usuari ja existeixi a MariaDB.";

}

$stmt->close();

$conn->close();

}

?>

**✅ Ara què?**

Després del registre, l’usuari pot fer login i treballar amb el seu propi accés MariaDB limitat (ja configurat).

¿Vols que afegim validació per evitar usuaris duplicats i mostrar errors més clars al registre?

|  |
| --- |
| **si** |

Perfecte! Ara millorarem el registre afegint:

1. ✅ **Validació perquè no es repeteixi un usuari** a la taula personal.
2. 🚫 Evitar error si l’usuari **ja existeix a MariaDB**.
3. 📢 Mostrar missatges clars d’error i èxit.

**🔐 Versió millorada de register.php**

<?php

require\_once("../functions/funcions.php"); // connexió com a root

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$username = trim($\_POST['username']);

$password = $\_POST['password'];

$rol = $\_POST['rol'];

$host = 'localhost';

$bd = 'restaurant';

$hash = password\_hash($password, PASSWORD\_DEFAULT);

$conn = mysqli\_connect($host, 'root', 'CONTRASENYA\_ROOT', $bd);

if (!$conn) die("Error de connexió: " . mysqli\_connect\_error());

// 1️⃣ Comprovar si ja existeix el username a la taula personal

$stmt = $conn->prepare("SELECT id FROM personal WHERE username = ?");

$stmt->bind\_param("s", $username);

$stmt->execute();

$stmt->store\_result();

if ($stmt->num\_rows > 0) {

echo "<p style='color:red'>Aquest nom d'usuari ja existeix a l'aplicació.</p>";

} else {

// 2️⃣ Insertar a personal

$stmt->close();

$stmt = $conn->prepare("INSERT INTO personal (username, password\_hash, rol) VALUES (?, ?, ?)");

$stmt->bind\_param("sss", $username, $hash, $rol);

$stmt->execute();

// 3️⃣ Crear usuari MariaDB (si no existeix)

$createUserSQL = "CREATE USER IF NOT EXISTS '$username'@'localhost' IDENTIFIED BY '$password'";

$grantSQL = "GRANT SELECT, INSERT ON $bd.\* TO '$username'@'localhost'";

if (mysqli\_query($conn, $createUserSQL)) {

mysqli\_query($conn, $grantSQL);

echo "<p style='color:green'>Usuari registrat correctament!</p>";

} else {

echo "<p style='color:red'>Error creant usuari a MariaDB. Potser ja existeix.</p>";

}

}

$stmt->close();

$conn->close();

} else {

header("Location: register.html");

exit();

}

?>

**📝 Què hem afegit:**

| **Millora** | **Codi implicat** |
| --- | --- |
| Evitar duplicats a personal | SELECT id FROM personal WHERE username = ? |
| Evitar error MariaDB | CREATE USER IF NOT EXISTS |
| Missatges clars | echo "<p style='color:red'>...</p>" |

Et puc fer un missatge d'èxit/fracàs estil Bootstrap o amb finestres modal si ho vols. Ho vols una mica més visual?

|  |
| --- |
| **sí, amb finestres modal** |

Perfecte! Ara et mostro com fer que els **missatges d’error i d’èxit es mostrin en finestres modal (popup)**, utilitzant **Bootstrap 5**.

**✅ 1. Afegeix Bootstrap al teu register.html o register.php**

Al <head>:

<link href=<https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css>

rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js">

</script>

**🧾 2. Formulari register.html (opcional si ja el tens)**

No cal canviar-lo si ja funciona. Anem directament al register.php.

**🔐 3. Versió amb modal de register.php**

<?php

require\_once("../functions/funcions.php");

$msg = ""; $type = "";

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$username = trim($\_POST['username']);

$password = $\_POST['password'];

$rol = $\_POST['rol'];

$host = 'localhost';

$bd = 'restaurant';

$hash = password\_hash($password, PASSWORD\_DEFAULT);

$conn = mysqli\_connect($host, 'root', 'CONTRASENYA\_ROOT', $bd);

if (!$conn) die("Error de connexió: " . mysqli\_connect\_error());

$stmt = $conn->prepare("SELECT id FROM personal WHERE username = ?");

$stmt->bind\_param("s", $username);

$stmt->execute();

$stmt->store\_result();

if ($stmt->num\_rows > 0) {

$msg = "Aquest nom d'usuari ja existeix a l'aplicació.";

$type = "danger";

} else {

$stmt->close();

$stmt = $conn->prepare("INSERT INTO personal (username, password\_hash, rol) VALUES (?, ?, ?)");

$stmt->bind\_param("sss", $username, $hash, $rol);

$stmt->execute();

$createUserSQL = "CREATE USER IF NOT EXISTS '$username'@'localhost' IDENTIFIED BY '$password'";

$grantSQL = "GRANT SELECT, INSERT ON $bd.\* TO '$username'@'localhost'";

if (mysqli\_query($conn, $createUserSQL)) {

mysqli\_query($conn, $grantSQL);

$msg = "Usuari registrat correctament!";

$type = "success";

} else {

$msg = "Error creant usuari a MariaDB. Potser ja existeix.";

$type = "danger";

}

}

$stmt->close();

$conn->close();

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Registre</title>

<meta charset="UTF-8">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<?php if (!empty($msg)): ?>

<!-- Modal -->

<div class="modal fade" id="resultModal" tabindex="-1" aria-labelledby="resultModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content border-<?php echo $type; ?>">

<div class="modal-header bg-<?php echo $type; ?>">

<h5 class="modal-title text-white" id="resultModalLabel">

<?php echo ($type == 'success') ? 'Registre completat' : 'Error'; ?>

</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Tancar"></button>

</div>

<div class="modal-body">

<?php echo $msg; ?>

</div>

<div class="modal-footer">

<a href="register.html" class="btn btn-secondary">Tornar</a>

<a href="/1-Restaurant/index.php" class="btn btn-primary">Inici</a>

</div>

</div>

</div>

</div>

<script>

document.addEventListener('DOMContentLoaded', function () {

var modal = new bootstrap.Modal(document.getElementById('resultModal'));

modal.show();

});

</script>

<?php endif; ?>

</body>

</html>

**🟢 Què fa aquest codi:**

* Mostra un **modal Bootstrap automàticament** després d’intentar registrar.
* Missatge verd (success) si tot va bé.
* Missatge vermell (danger) si falla.

Vols que el formulari de registre també aparegui dins una pàgina Bootstrap bonica o ja et va bé així?

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |