

Creating a Login Form

This standalone login page validates credentials with PDO, starts a secure session and redirects authenticated users. It uses one generic failure message so it does not reveal whether a username exists.

Login page

```
<?php
session_start();
require __DIR__ . "/includes/db.php";

$error = "";

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $username = trim($_POST["username"] ?? "");
    $password = $_POST["password"] ?? "";

    $stmt = $pdo->prepare(
        "SELECT user_id, username, password, role
        FROM users
        WHERE username = :username"
    );
    $stmt->execute(["username" => $username]);
    $user = $stmt->fetch();

    if ($user && password_verify($password, $user["password"])) {
        session_regenerate_id(true);
        $_SESSION["user_id"] = (int) $user["user_id"];
        $_SESSION["username"] = $user["username"];
        $_SESSION["role"] = $user["role"];

        header("Location: dashboard.php");
        exit;
    }
}
```

```
        $error = "Username or password was not accepted.";
    }
    ?>
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Log in</title>
</head>
<body>
<h1>Log in</h1>

<?php if ($error): ?>
    <p role="alert"><?= htmlspecialchars($error) ?></p>
<?php endif; ?>

<form method="post">
    <label for="username">Username</label>
    <input id="username" name="username" autocomplete="username" required>

    <label for="password">Password</label>
    <input id="password" name="password" type="password"
        autocomplete="current-password" required>

    <button type="submit">Log in</button>
</form>
</body>
</html>
```

Why the security steps matter

- prepared statements keep input separate from SQL
- `password_verify()` checks the stored hash
- session ID regeneration reduces session fixation risk
- role comes from the database, not the form
- a generic error reduces account discovery

Test

Test a valid standard user, a valid administrator, a wrong password, an unknown username and empty fields. Confirm that failed attempts do not create session identity values.

Check

- ☐ Session starts before output.
- ☐ Query uses a prepared statement.
- ☐ Password hash is verified correctly.
- ☐ Session ID changes after login.
- ☐ User ID, username and role are stored.
- ☐ Failure message is generic.

Revision #2

Created 2026-06-08 04:50:20 UTC by Mr Napper

Updated 2026-08-18 23:49:13 UTC by Mr Napper