

Creating a User Registration Form

This standalone registration page accepts a fictional username and password, validates both on the server, hashes the password and inserts a standard `user` account with PDO.

Database connection

The example expects `includes/db.php` to create a PDO object named `$pdo`.

Registration page

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

$username = "";
$errors = [];
$created = false;

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

    if (!preg_match('/^[A-Za-z0-9_]{3,50}$/', $username)) {
        $errors[] = "Username must be 3-50 letters, numbers or underscores.";
    }
    if (strlen($password) < 10) {
        $errors[] = "Password must contain at least 10 characters.";
    }
    if ($password !== $confirm) {
        $errors[] = "Passwords do not match.";
    }
}
```

```

}

if (!$errors) {
    $check = $pdo->prepare(
        "SELECT user_id FROM users WHERE username = :username"
    );
    $check->execute(["username" => $username]);

    if ($check->fetch()) {
        $errors[] = "That username is unavailable.";
    } else {
        $insert = $pdo->prepare(
            "INSERT INTO users (username, password, role)
            VALUES (:username, :password, 'user')"
        );
        $insert->execute([
            "username" => $username,
            "password" => password_hash($password, PASSWORD_DEFAULT)
        ]);
        $created = true;
        $username = "";
    }
}

?>
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Create account</title>
</head>
<body>
<h1>Create account</h1>

<?php if ($created): ?>
    <p role="status">Account created. You can now log in.</p>
<?php endif; ?>

<?php if ($errors): ?>

```

```

<div role="alert">
  <p>Please correct the following:</p>
  <ul>
    <?php foreach ($errors as $error): ?>
      <li><?= htmlspecialchars($error) ?></li>
    <?php endforeach; ?>
  </ul>
</div>
<?php endif; ?>

<form method="post">
  <label for="username">Username</label>
  <input id="username" name="username" maxlength="50"
    autocomplete="username" required
    value="<?= htmlspecialchars($username) ?>">

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

  <label for="confirm_password">Confirm password</label>
  <input id="confirm_password" name="confirm_password" type="password"
    autocomplete="new-password" required>

  <button type="submit">Create account</button>
</form>
</body>
</html>

```

The form never accepts a role. Every public registration receives the server-controlled `user` role.

Test

Test valid input, short passwords, mismatched passwords, invalid usernames, duplicates and missing fields. Confirm that unsuccessful attempts create no database row.

Check

- ☐ Labels are associated with inputs.
 - ☐ PHP validates all inputs.
 - ☐ Username duplicates are handled.
 - ☐ Password is hashed before insertion.
 - ☐ Prepared statements are used.
 - ☐ Registration cannot create an administrator.
-

Revision #2

Created 2026-06-08 04:42:22 UTC by Mr Napper

Updated 2026-08-18 23:49:09 UTC by Mr Napper