

Guía Para Crear Un Sistema De Inicio De Sesión Y Registro Usando PHP Y MySQL

Un sistema seguro de inicio de sesión y registro es uno de los requerimientos principales al crear un sistema. Es por esto, que en esta oportunidad daremos una breve guía de como crear un **sistema de inicio de sesión y registro**, usando MySQL y PHP. No haremos uso de algún **Framework** ([6 Frameworks PHP para el desarrollo ágil](#)), solamente usaremos PHP desde cero para lograrlo ;) No es tan difícil de hacerlo, solo tienes que tener **conocimientos básicos de programación**, HTML, y algo de CSS (por si deseas modificar la apariencia). Esperamos que nuestro anterior artículo, sobre [10 consejos para ser un mejor programador en PHP](#), sea de gran utilidad.

Lo Que Se Necesita

- [Xampp](#) (PHP versión 5.3 ó posterior y la versión de MySQL 4.1.3 ó posterior)
- Conocimientos básicos sobre [PHP](#), [HTML](#) y [CSS](#)
- Algo de tiempo y paciencia ;)

Resultado Final

En las siguientes imágenes podrás ver el resultado final del sistema de **inicio de sesión y registro** al usar PHP y MySQL.



Te gusta como se ve el formulario? Puedes [descargar el código de formulario de u sistema de inicio de sesión y registro](#).

Configuración Del Servidor

Puedes usar cualquier [servicio de alojamiento web](#) que tenga PHP y MySQL ya instalados (sólo asegúrese de tener la versión PHP 5.3 ó superior y MySQL 4.1.3 ó superior). Para esta guía, se usó Xampp como servidor. Te gustaría tener tu propio **servidor online**, te invitamos a conocer nuestros planes en [servidores compartidos](#), [servidores VPS](#) y [servidores Dedicados](#) ;)

Paso 1 – MySQL

En primer lugar, es necesario que cree la base de datos y la tabla que almacenara todos los registros. Puedes hacer esto de forma manual utilizando la interfaz gráfica de usuario de [PhpMyAdmin](#) ó usar SQL para insertar esta información. Si deseas usar SQL, las siguientes líneas te crearan tanto la base de datos como la tabla:

```
[bash]
```

```
/* Create Database */
```

```
CREATE DATABASE userlitdb;
```

```
/* Create Table */
```

```
CREATE TABLE `usertbl` (
```

```
`id` int(11) NOT NULL auto_increment,
```

```
`full_name` varchar(32) collate utf8_unicode_ci NOT NULL  
default »,
```

```
`email` varchar(32) collate utf8_unicode_ci NOT NULL default  
»,
```

```
`username` varchar(20) collate utf8_unicode_ci NOT NULL  
default »,
```

```
`password` varchar(32) collate utf8_unicode_ci NOT NULL  
default »,
```

```
PRIMARY KEY (`id`),
```

```
UNIQUE KEY `username` (`username`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
[/bash]
```

Observe que todos ellos son de tipo varchar, incluso la contraseña que se convertirá más tarde en md5 para asegurar la contraseña del usuario. Además, use un AUTO_INCREMENT para asignar automáticamente un ID ó un número de usuarios que se registrarían más tarde.

Ahora que la tabla se ha creado, lo que tienes que hacer es

implementar la estructura con HTML y diseño con CSS, luego vendrá el PHP.

Paso 2 – El HTML

Ya que el proceso lo haremos del lado del servidor (por usar solo PHP), usaremos tres archivos PHP en cual se cuenta con las funciones y la estructura con el HTML. En primer lugar, se crea el archivo login.php. Sientete libre de copiar el siguiente código:

```
[php]
```

```
<?php
session_start();
?>
```

```
<?php require_once("includes/connection.php"); ?>
<?php include("includes/header.php"); ?>
```

```
<?php
```

```
if(isset($_SESSION["session_username"])){
// echo "Session is set"; // for testing purposes
header("Location: intropage.php");
}
```

```
if(isset($_POST["login"])){
```

```
if(!empty($_POST['username']) && !empty($_POST['password'])) {
$username=$_POST['username'];
$password=$_POST['password'];
```

```
$query =mysql_query("SELECT * FROM usertbl WHERE
username='".$username."' AND password='".$password.'");
```

```
$numrows=mysql_num_rows($query);
if($numrows!=0)
```

```

{
while($row=mysql_fetch_assoc($query))
{
$dbusername=$row['username'];
$dbpassword=$row['password'];
}

if($username == $dbusername && $password == $dbpassword)

{

$_SESSION['session_username']=$username;

/* Redirect browser */
header("Location: intropage.php");
}
} else {

$message = "Nombre de usuario ó contraseña invalida!";
}

} else {
$message = "Todos los campos son requeridos!";
}
}
?>

<div class="container mlogin">
<div id="login">
<h1>Logueo</h1>
<form name="loginform" id="loginform" action="" method="POST">
<p>
<label for="user_login">Nombre De Usuario<br />
<input type="text" name="username" id="username" class="input"
value="" size="20" /></label>
</p>
<p>
<label for="user_pass">Contraseña<br />
<input type="password" name="password" id="password"

```

```

class="input" value="" size="20" /></label>
</p>
<p class="submit">
<input type="submit" name="login" class="button"
value="Entrar" />
</p>
<p class="regtext">No estas registrado? <a href="register.php"
>Registrate Aquí</a>!</p>
</form>

</div>

</div>

```

```

<?php include("includes/footer.php"); ?>

<?php if (!empty($message)) {echo "<p class=\"error\">" .
"MESSAGE: ". $message . "</p>";} ?>

[/php]

```

A continuación, creamos el archivo **register.php**. Copie y pegue el siguiente código:

```

[php]

<?php require_once("includes/connection.php"); ?>
<?php include("includes/header.php"); ?>

<?php

if(isset($_POST["register"])){

if(!empty($_POST['full_name']) && !empty($_POST['email']) &&
!empty($_POST['username']) && !empty($_POST['password'])) {
$full_name=$_POST['full_name'];
$email=$_POST['email'];
$username=$_POST['username'];
$password=$_POST['password'];

$query=mysql_query("SELECT * FROM usertbl WHERE

```

```

username="'. $username. "'");
$numrows=mysql_num_rows($query);

if($numrows==0)
{
$sql="INSERT INTO usertbl
(full_name, email, username,password)
VALUES('$full_name','$email', '$username', '$password')";

$result=mysql_query($sql);

if($result){
$message = "Cuenta Correctamente Creada";
} else {
$message = "Error al ingresar datos de la informacion!";
}

} else {
$message = "El nombre de usuario ya existe! Por favor, intenta
con otro!";
}

} else {
$message = "Todos los campos no deben de estar vacios!";
}
}
?>

<?php if (!empty($message)) {echo "<p class=\"error\">" .
"Mensaje: ". $message . "</p>";} ?>

<div class="container mregister">
<div id="login">
<h1>Registrar</h1>
<form      name="registerform"      id="registerform"
action="register.php" method="post">
<p>
<label for="user_login">Nombre Completo<br />
<input  type="text"   name="full_name"   id="full_name"

```

```

class="input" size="32" value="" /></label>
</p>

<p>
<label for="user_pass">E-mail<br />
<input type="email" name="email" id="email" class="input"
value="" size="32" /></label>
</p>

<p>
<label for="user_pass">Nombre De Usuario<br />
<input type="text" name="username" id="username" class="input"
value="" size="20" /></label>
</p>

<p>
<label for="user_pass">Contraseña<br />
<input type="password" name="password" id="password"
class="input" value="" size="32" /></label>
</p>

<p class="submit">
<input type="submit" name="register" id="register"
class="button" value="Registrar" />
</p>

<p class="regtext">Ya tienes una cuenta? <a href="login.php"
>Entra Aquí!</a>!</p>
</form>

</div>
</div>

<?php include("includes/footer.php"); ?>

[/php]

```

A continuación, se crea la pagina de bienvenida en el archivo intropage.php. Esto servirá como la página principal una vez que un usuario inicia sesión exitosamente.

[php]

```
<?php
session_start();
if(!isset($_SESSION["session_username"])) {
header("location:login.php");
} else {
?>

<?php include("includes/header.php"); ?>
<div id="welcome">
<h2>Bienvenido,          <span><?php          echo
$_SESSION['session_username'];?>! </span></h2>
<p><a href="logout.php">Finalice</a> sesión aquí!</p>
</div>

<?php include("includes/footer.php"); ?>

<?php
}
?>
```

[/php]

Si la estructura se maneja con HTML, ¿por qué **usar PHP con HTML**? Simple, PHP es tan flexible que nos permite usar HTML incrustado, si prefieres y quieres probar a tu modo, puedes separar la estructura del HTML del código PHP ;) Pero bueno, que sería del HTML sin el CSS en estos días?

Paso 3 – El CSS

Para darle un aspecto bien interesante al ejercicio, le daremos un pequeño retoque con CSS, es por esto que se ve bastante llamativo el formulario ;)

[css]

```
*= GENERAL STYLES
```

```
_____*/  
body {  
background: #efefef;  
Open Sans', sans-serif;  
color: #777;  
}  
  
a {  
color: #f58220;  
font-weight: 400;  
}  
  
span {  
font-weight: 300;  
color: #f58220;  
}  
  
.mlogin {  
margin: 170px auto 0;  
}  
  
.mregister {  
margin: 80px auto 0;  
}  
  
.error {  
margin: 40px auto 0;  
border: 1px solid #777;  
padding: 3px;  
color: #fff;  
text-align: center;  
width: 650px;  
background: #f58220;  
}  
  
.regtext {  
font-size: 13px;  
margin-top: 26px;  
color: #777;
```

```
}
```

```
/*= CONTAINERS
```

```
_____*/
```

```
.container {  
padding: 25px 16px 25px 10px;  
font-weight: 400;  
overflow: hidden;  
width: 350px;  
height: auto;  
background: #fff;  
-webkit-box-shadow: 0 1px 3px rgba(0,0,0,.13);  
-moz-box-shadow: 0 1px 3px rgba(0,0,0,.13);  
box-shadow: 0 1px 3px rgba(0,0,0,.13);  
}
```

```
#welcome {  
width: 500px;  
padding: 30px;  
background: #fff;  
margin: 160px auto 0;  
-webkit-box-shadow: 0 1px 3px rgba(0,0,0,.13);  
-moz-box-shadow: 0 1px 3px rgba(0,0,0,.13);  
box-shadow: 0 1px 3px rgba(0,0,0,.13);  
}
```

```
.container h1 {  
color: #777;  
text-align: center;  
font-weight: 300;  
border: 1px dashed #777;  
margin-top: 13px;  
}
```

```
.container label {  
color: #777;  
font-size: 14px;  
}
```

```
#login {
width: 320px;
margin: auto;
padding-bottom: 15px;
}

.container form .input,.container input[type=text],.container
input[type=password],.container input[type=e] {
background: #fbfbfb;
font-size: 24px;
line-height: 1;
width: 100%;
padding: 3px;
margin: 0 6px 5px 0;
outline: none;
border: 1px solid #d9d9d9;
}

.container form .input:focus {
border: 1px solid #f58220;
-webkit-box-shadow: 0 0 3px 0 rgba(245,130,32,0.75);
-moz-box-shadow: 0 0 3px 0 rgba(245,130,32,0.75);
box-shadow: 0 0 3px 0 rgba(245,130,32,0.75);
}

/*= BUTTONS
_____*/

.button{
border: solid 1px #da7c0c;
background: #f78d1d;
background: -webkit-gradient(linear, left top, left bottom,
from(#faa51a), to(#f47a20));
background: -moz-linear-gradient(top, #faa51a, #f47a20);
filter:
progid:DXImageTransform.Microsoft.gradient(startColorstr='#faa
51a', endColorstr='#f47a20');
color: #fff;
```

```
padding: 7px 12px;
-webkit-border-radius:4px;
-moz-border-radius:4px;
border-radius:4px;
float: right;
cursor: pointer;
}

.button:hover{
background: #f47c20;
background: -webkit-gradient(linear, left top, left bottom,
from(#f88e11), to(#f06015));
background: -moz-linear-gradient(top, #f88e11, #f06015);
filter:
progid:DXImageTransform.Microsoft.gradient(startColorstr='#f88
e11', endColorstr='#f06015');
}

/*= FOOTER
_____*/
footer {
color: #777;
font-size: 12px;
text-align: center;
margin-top: 20px;
}
[/css]
```

Puntos A Tener

En Cuenta

- Se uso la re utilización de código haciendo uso del `include`, esto lo encontramos bastante en el header y el footer.
- Variables constantes para la conexión a la base de datos, esto lo encontramos en el archivo *connection.php*.
- La sesión de maneja con [`session_destroy`](#), cuando el usuario desea finalizar la sesión, haremos uso de esta función.
- Si eres curioso, y miras el código fuente dentro del navegador, no se vera el código PHP ;)

Finalmente

Hoy has aprendido a crear un sencillo sistema de inicio de sesión utilizando PHP y MySQL. Aunque hay muchas cosas a considerar cuando se trata de seguridad, esto es un buen comienzo para aquellos principiantes. Eres libre de modificar los anteriores códigos para tu proyecto. Si tiene alguna idea para mejorar los códigos, por favor, deja un buen comentario con el código y explicando el por que ;)