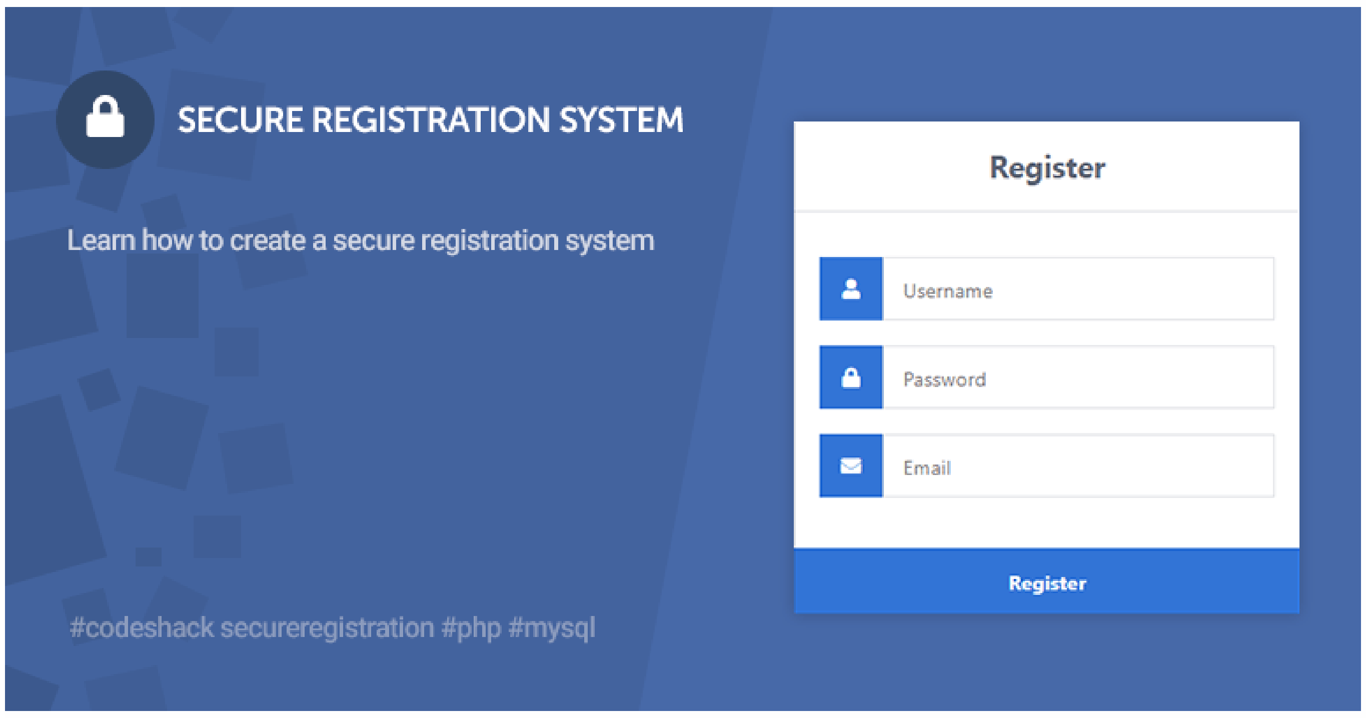
Secure Registration System with PHP and MySQL

Updated on **September 9, 2022** by **David Adams**


SECURE REGISTRATION SYSTEM
Learn how to create a secure registration system
#codeshack secureregistration #php #mysql
Register
Username
Password
Email
Register

This tutorial is a follow up to our previous tutorial Secure Login System with PHP and MySQL. In this tutorial, we'll be creating a secure registration form and implementing basic validation.

A registration form is what your website's visitors can use to register their details, which will subsequently be stored in a MySQL database.

The Advanced package includes additional features and a download link to the source code.

Contents

1. Getting Started

There are a few steps we need to take before we create our secure registration system. We need to set-up our web server environment and make sure we have the required extensions enabled *(skip if you followed the secure login system tutorial)*.

1.1. Requirements

- If you haven't got a local web server set-up, you will need to download and install XAMPP. XAMPP is a server-side web development environment that includes the essentials for back-end web developers.

1.2. What You Will Learn in this Tutorial

- **Form Design** — Design a registration form with HTML5 and CSS3.
- **Prepared SQL Queries** — How to prepare SQL queries to prevent SQL injection and insert new records into a MySQL database.
- **Basic Validation** — Validating form data that is sent to the server (username, password, and email).

1.3. File Structure & Setup

We now need to start our web server and create the files and directories that we're going to use for our registration system.

RECENT POSTS
File Upload Progress Bar with JS and PHP
Newsletter System with PHP and MySQL
Live Support Chat with AJAX, PHP and MySQL

TAGS
tutorials php mysql programming javascript snippets ajax css form php class html login freebies scripting pdo mysqli shopping cart registration register columns live support chat database modals snippet survey form python jquery voting system progamming mvc newsletter system table commenting system javascript class sessions express hotel reservation form poll system progress bar crud event calendar sort

- Next to the MySQL module click ***Start***
- Navigate to XAMPPs installation folder (***C:\xampp***)
- Open the ***htdocs*** folder
- Create the following folders and files:

File Structure

```
\-- phplogin
    |-- register.html
    |-- style.css
    |-- register.php
    |-- activate.php (optional)
```

Each file will contain the following:

- ***register.html*** — Registration form created with HTML5 and CSS3. As this file doesn't require us to use PHP, we'll save it as plain HTML.
- ***style.css*** — The stylesheet (CSS3) for our secure registration form.
- ***register.php*** — Validate form data and insert a new account into the MySQL database.
- ***activate.php*** — Activate the user's account with a unique code (email based activation).

2. Creating the Registration Form Design

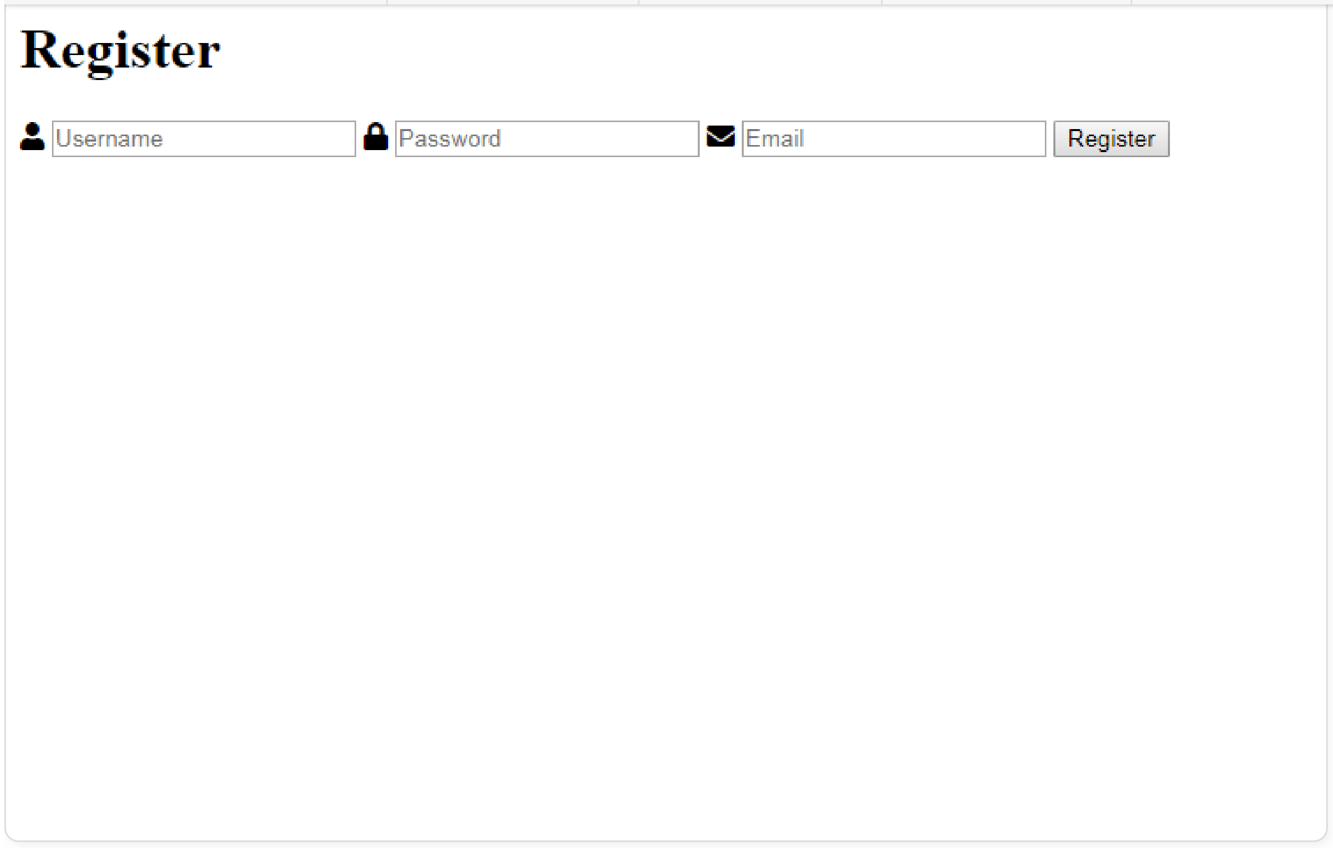
The registration form will be used by our websites visitors. They can use it to input their account information. We'll be creating the registration form with HTML and CSS.

Edit the ***register.html*** file and add the following code:

HTML Copy

```html
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Register</title>
        <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
    </head>
    <body>
        <div class="register">
            <h1>Register</h1>
            <form action="register.php" method="post" autocomplete="off">
                <label for="username">
                    <i class="fas fa-user"></i>
                </label>
                <input type="text" name="username" placeholder="Username" id="username" required>
                <label for="password">
                    <i class="fas fa-lock"></i>
                </label>
                <input type="password" name="password" placeholder="Password" id="password" requir
                <label for="email">
                    <i class="fas fa-envelope"></i>
                </label>
                <input type="email" name="email" placeholder="Email" id="email" required>
                <input type="submit" value="Register">
            </form>
        </div>
    </body>
</html>
```

Navigate to ***http://localhost/phplogin/register.html***, our registration form will look like the following:



Register

Username Password Email Register

Pretty basic for a registration form, now let's add some CSS, edit the ***style.css*** file and add the following:

CSS Copy

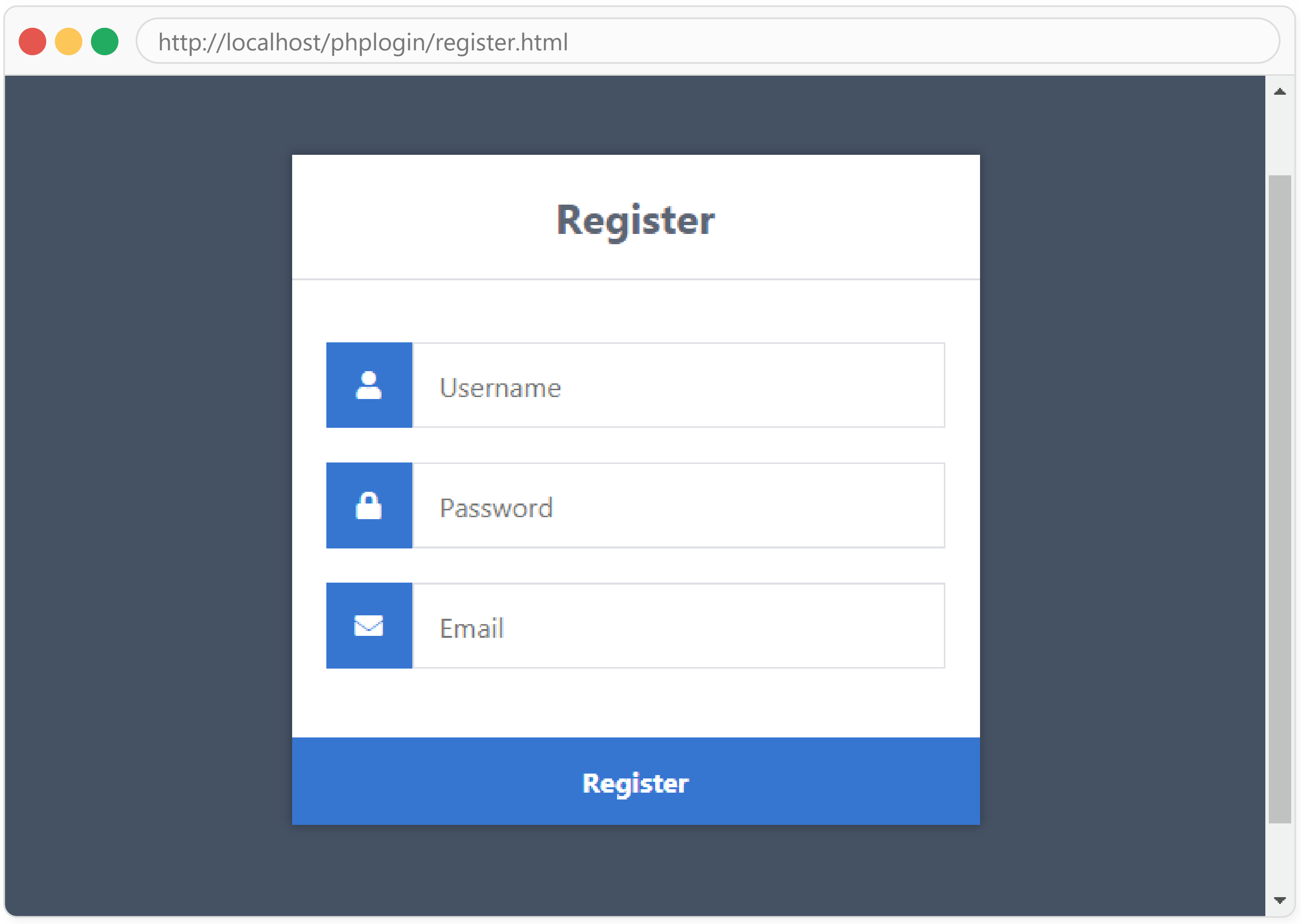
```css
* {
    box-sizing: border-box;
    font-family: -apple-system, BlinkMacSystemFont, "segoe ui", roboto, oxygen, ubuntu, cantarell,
    font-size: 16px;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
}
body {
    background-color: #435165;
    margin: 0;
}
.register {
    width: 400px;
    background-color: #ffffff;
    box-shadow: 0 0 9px 0 rgba(0, 0, 0, 0.3);
    margin: 100px auto;
}
.register h1 {
    text-align: center;
    color: #5b6574;
    font-size: 24px;
    padding: 20px 0 20px 0;
    border-bottom: 1px solid #dee0e4;
}
.register form {
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    padding-top: 20px;
}
.register form label {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 50px;
    height: 50px;
    background-color: #3274d6;
    color: #ffffff;
}
.register form input[type="password"], .register form input[type="text"], .register form input[typ
    width: 310px;
    height: 50px;
    border: 1px solid #dee0e4;
    margin-bottom: 20px;
    padding: 0 15px;
}
.register form input[type="submit"] {
    width: 100%;
    padding: 15px;
    margin-top: 20px;
    font-weight: bold;
    color: #ffffff;
    transition: background-color 0.2s;
}
.register form input[type="submit"]:hover {
    background-color: #2868c7;
    transition: background-color 0.2s;
}
```

We need to include our stylesheet in our *register.html* file, copy and paste the following code to the head section:

HTML Copy

```
<link href="style.css" rel="stylesheet" type="text/css">
```

And now our registration form will look more appealing:


http://localhost/phplogin/register.html

Let's narrow down the form so we can get a better understanding on what's going on.

- **Form** — we need to use both the `action` and `post` attributes, the `action` attribute will be set to the registration file. When the form is submitted, the form data will be sent to the registration file for processing. The `method` is to `post`, this will allow us to process the form data.
  - **Input (text/password/email)** — We need to name our form fields so the server can recognize them, so if we set the value of the attribute `name` to the `username`, we can use the post variable in our registration file to get the data, like this: `$_POST['username']`.
  - **Input (submit)** — On click the form data will be sent to our registration file.

That's basically all we need to do on the client-side, next step is to set-up the database and create the registration file with PHP.

3. Creating the Database and setting-up Tables

You can skip this step if you followed the Secure Login System Tutorial.

For this part, you will need to access your MySQL database, either using phpMyAdmin or your preferred MySQL database management application.

If you're using *phpMyAdmin* then follow these instructions:

- Navigate to: *http://localhost/phpmyadmin/*
- Click the *Databases* tab at the top

- Click ***Create***

You can use your own database name, but for this tutorial, we'll use ***phplogin***.

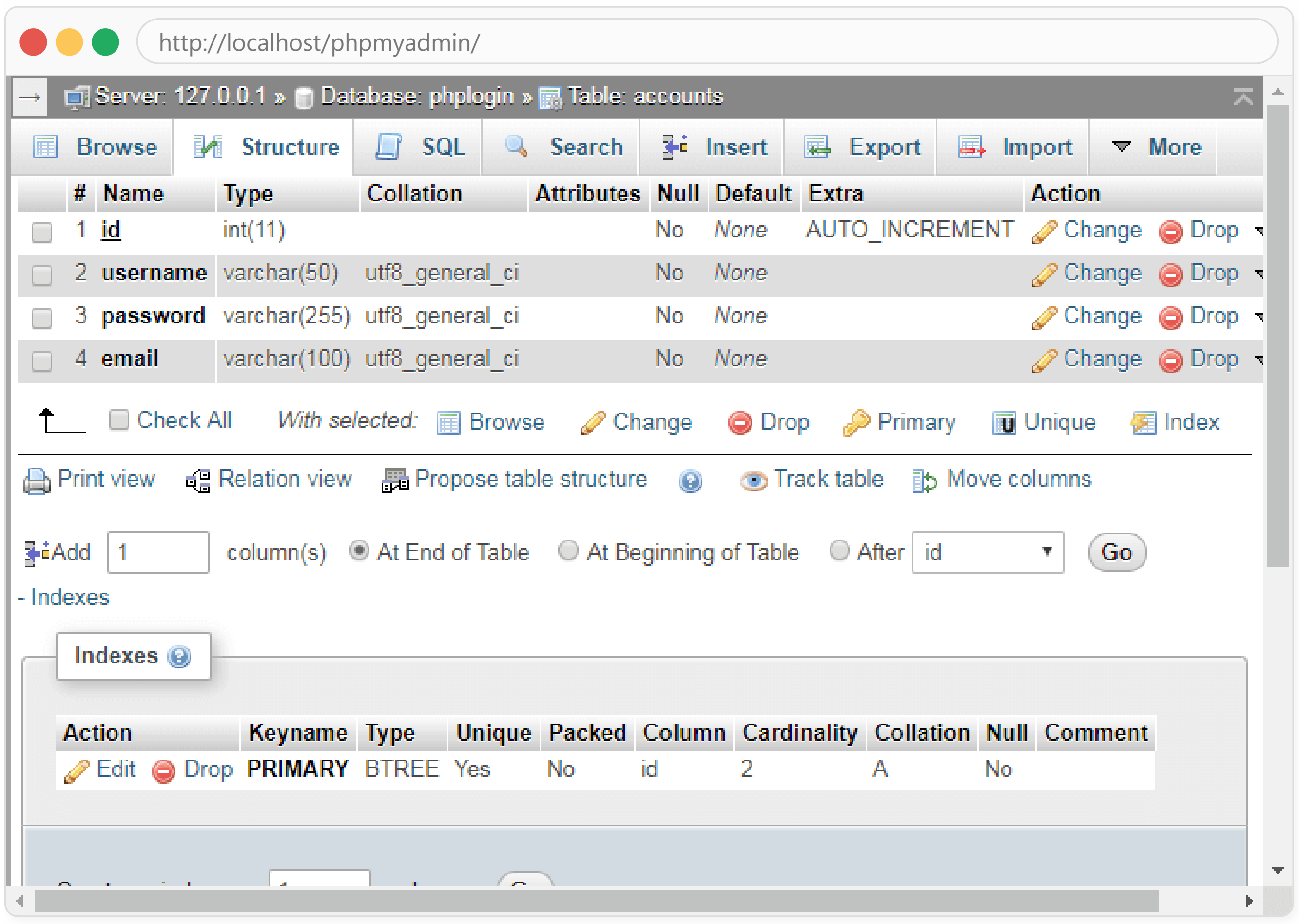
What we need now is an ***accounts*** table that will store all our accounts (usernames, passwords, emails, etc).

Click the database on the left side panel (***phplogin***) and execute the following SQL statement:

SQL Copy

```sql
CREATE TABLE IF NOT EXISTS `accounts` (
	`id` int(11) NOT NULL AUTO_INCREMENT,
  	`username` varchar(50) NOT NULL,
  	`password` varchar(255) NOT NULL,
  	`email` varchar(100) NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
```

On ***phpMyAdmin*** this should look like:


http://localhost/phpmyadmin/
Server: 127.0.0.1 » Database: phplogin » Table: accounts
Browse
Structure
SQL
Search
Insert
Export
Import
More

	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
☐	1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop
☐	2	username	varchar(50)	utf8_general_ci		No	None		Change Drop
☐	3	password	varchar(255)	utf8_general_ci		No	None		Change Drop
☐	4	email	varchar(100)	utf8_general_ci		No	None		Change Drop

Check All With selected: Browse Change Drop Primary Unique Index
Print view Relation view Propose table structure Track table Move columns
Add 1 column(s) At End of Table At Beginning of Table After id Go
- Indexes
Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	2	A	No	

The above SQL statement code will create the accounts table with the columns `id`, `username`, `password`, and `email`.

4. Registering Users with PHP & MySQL

Now we need to create the registration file that will process the form fields, check for basic validation, and insert the new account into our database.

The registration page will require a connection to our database and therefore we must include the necessary variables and MySQL functions. Edit the ***register.php*** file and add the following code:

PHP Copy

```php
<?php
// Change this to your connection info.
$DATABASE_HOST = 'localhost';
$DATABASE_USER = 'root';
$DATABASE_PASS = '';
$DATABASE_NAME = 'phplogin';
// Try and connect using the info above.
$con = mysqli_connect($DATABASE_HOST, $DATABASE_USER, $DATABASE_PASS, $DATABASE_NAME);
if (mysqli_connect_errno()) {
	// If there is an error with the connection, stop the script and display the error.
	exit('Failed to connect to MySQL: ' . mysqli_connect_error());
}
```

Don't forget to update the My

```php
// Now we check if the data was submitted, isset() function will check if the data exists.
if (!isset($_POST['username'], $_POST['password'], $_POST['email'])) {
    // Could not get the data that should have been sent.
    exit('Please complete the registration form!');
}
// Make sure the submitted registration values are not empty.
if (empty($_POST['username']) || empty($_POST['password']) || empty($_POST['email'])) {
    // One or more values are empty.
    exit('Please complete the registration form');
}
```

Now we need to check if the account already exists in the database. We can check this by selecting a record from our accounts table with the same `username` that the user has provided.

Add after:

PHP Copy

```php
// We need to check if the account with that username exists.
if ($stmt = $con->prepare('SELECT id, password FROM accounts WHERE username = ?')) {
    // Bind parameters (s = string, i = int, b = blob, etc), hash the password using the PHP passw
    $stmt->bind_param('s', $_POST['username']);
    $stmt->execute();
    $stmt->store_result();
    // Store the result so we can check if the account exists in the database.
    if ($stmt->num_rows > 0) {
        // Username already exists
        echo 'Username exists, please choose another!';
    } else {
        // Insert new account
    }
    $stmt->close();
} else {
    // Something is wrong with the sql statement, check to make sure accounts table exists with al
    echo 'Could not prepare statement!';
}
$con->close();
?>
```

Replace:

```php
// Insert new account
```

With:

PHP Copy

```php
// Username doesnt exists, insert new account
if ($stmt = $con->prepare('INSERT INTO accounts (username, password, email) VALUES (?, ?, ?)')) {
    // We do not want to expose passwords in our database, so hash the password and use password_ve
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT);
    $stmt->bind_param('sss', $_POST['username'], $password, $_POST['email']);
    $stmt->execute();
    echo 'You have successfully registered, you can now login!';
} else {
    // Something is wrong with the sql statement, check to make sure accounts table exists with al
    echo 'Could not prepare statement!';
}
```

This will insert a new account into our accounts table.

Remember in our Login System we used the `password_verify` function? As you can see in the code above we use the `password_hash` function, this will encrypt the user's password using the one-way algorithm — this will prevent your users passwords from being exposed if for somehow your database becomes vulnerable.

That's basically all we need to do to register accounts on our website.

5. Validating Form Data

the codes below, add them in the `register.php` file before the following line:

```
if ($stmt = $con->prepare('SELECT id, password FROM accounts WHERE username = ?')) {
```

Email Validation

PHP Copy

```php
if (!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
    exit('Email is not valid!');
}
```

Invalid Characters Validation

PHP Copy

```php
if (preg_match('/^[a-zA-Z0-9]+$/', $_POST['username']) == 0) {
    exit('Username is not valid!');
}
```

Character Length Check

PHP Copy

```php
if (strlen($_POST['password']) > 20 || strlen($_POST['password']) < 5) {
    exit('Password must be between 5 and 20 characters long!');
}
```

You should always implement your own validation, these are just basic examples.

6. Implementing Account Activation

The account activation system will send an email to the user with the activation link when the user has registered.

The first thing we need to do is to go into ***phpMyAdmin*** and select our database, in our case this would be ***phplogin***, you can either add the column `activation_code` to the accounts table or execute the SQL statement below.

SQL Copy

```sql
ALTER TABLE accounts ADD activation_code varchar(50) DEFAULT ''
```

Now we need to edit our ***register.php*** file, search for this line of code:

```
if ($stmt = $con->prepare('INSERT INTO accounts (username, password, email) VALUES (?, ?, ?)')) {
```

Replace with:

PHP Copy

```php
if ($stmt = $con->prepare('INSERT INTO accounts (username, password, email, activation_code) VALUES
```

Search for:

```
$stmt->bind_param('sss', $_POST['username'], $password, $_POST['email']);
```

Replace with:

PHP Copy

```php
$uniqid = uniqid();
$stmt->bind_param('ssss', $_POST['username'], $password, $_POST['email'], $uniqid);
```

The `$uniqud` variable will generate a unique ID that we'll use for our activation code, this will be sent to the user's email address.

Search for:

Replace with:

PHP Copy

```php
$from    = 'noreply@yourdomain.com';
$subject = 'Account Activation Required';
$headers = 'From: ' . $from . "\r\n" . 'Reply-To: ' . $from . "\r\n" . 'X-Mailer: PHP/' . phpversi
// Update the activation variable below
$activate_link = 'http://yourdomain.com/phplogin/activate.php?email=' . $_POST['email'] . '&code='
$message = '<p>Please click the following link to activate your account: <a href="' . $activate_li
mail($_POST['email'], $subject, $message, $headers);
echo 'Please check your email to activate your account!';
```

Upon account registration, the user will need to activate their account using the activation link that is sent to their email address. You need to update both the `$from` and `$activate_link` variables.

Now we can proceed to create the activation file. The activation file will process the GET parameters and verify the email and code. The user's account will be activated if the code is valid.

Edit/create the ***activate.php*** file and add the following code:

PHP Copy

```php
<?php
// Change this to your connection info.
$DATABASE_HOST = 'localhost';
$DATABASE_USER = 'root';
$DATABASE_PASS = '';
$DATABASE_NAME = 'phplogin';
// Try and connect using the info above.
$con = mysqli_connect($DATABASE_HOST, $DATABASE_USER, $DATABASE_PASS, $DATABASE_NAME);
if (mysqli_connect_errno()) {
    // If there is an error with the connection, stop the script and display the error.
    exit('Failed to connect to MySQL: ' . mysqli_connect_error());
}
// First we check if the email and code exists...
if (isset($_GET['email'], $_GET['code'])) {
    if ($stmt = $con->prepare('SELECT * FROM accounts WHERE email = ? AND activation_code = ?')) {
        $stmt->bind_param('ss', $_GET['email'], $_GET['code']);
        $stmt->execute();
        // Store the result so we can check if the account exists in the database.
        $stmt->store_result();
        if ($stmt->num_rows > 0) {
            // Account exists with the requested email and code.
            if ($stmt = $con->prepare('UPDATE accounts SET activation_code = ? WHERE email = ? AND
                // Set the new activation code to 'activated', this is how we can check if the user
                $newcode = 'activated';
                $stmt->bind_param('sss', $newcode, $_GET['email'], $_GET['code']);
                $stmt->execute();
                echo 'Your account is now activated! You can now <a href="index.html">login</a>!';
            }
        } else {
            echo 'The account is already activated or doesn\'t exist!';
        }
    }
}
?>
```

If the code reflects the one in the database that is associated with the user's account then the value of the `activation_code` column will be updated to `activated`.

If we want to check if the user has activated their account, we can add the following code to the pages we want to restrict non-activated users:

PHP Copy

```php
if ($account['activation_code'] == 'activated') {
    // account is activated
    // Display home page etc
} else {
    // account is not activated
```

For the above code to work, you will need to connect to your MySQL database and select the user's account.

Also, take note PHP mail function will only work if your computer or server supports it. If it doesn't send an email, check your configuration or install a mail server such as Postfix.

Conclusion

Congratulations! You've successfully created a Login System (if you followed the previous tutorial) and registration system with PHP and MySQL. You're free to use the code in this tutorial and adapt it for your own projects.

Remember that this is just a secure base that you should work from. Consider changing or implementing your own validation, and implement your own features.

If you would like more of this tutorial series, feel free to drop a comment and suggest to us what we could create next.

Enjoy coding!

If you would like to support us, consider purchasing the advanced secure login & registration system below as it will greatly help us create more tutorials and keep our website up and running. The advanced package includes improved code and more features.

ADVANCED	BUNDLE (SAVE 53%) Sale
$ 20.00	$ 119.00 ~~255.00~~
view more details	view more details
Source Code ✓	Secure Login & Registration System ✓
Database SQL File ✓	Shopping Cart System ✓
Secure Login & Registration System ✓	CRUD Application ✓
Home, Profile & Edit Profile Pages ✓	Ticketing System ✓
Account Activation Feature ✓	Gallery System ✓
Remember Me Feature ✓	Event Calendar System ✓
AJAX Integration ✓	Poll and Voting System ✓
PDO, MVC OOP & Basic Versions ✓	Commenting System ✓
Admin Panel ✓	Review System ✓
Add-on: Forgot Password ✓	Contact Form ✓
Add-on: Brute Force Protection ✓	Live Support Chat System ✓
Add-on: CSRF Protection ✓	Newsletter & Mailing System ✓
Add-on: Two-factor Authentication ✓	Access to future scripts ✓
Add-on: Captcha ✓	
Responsive Design (mobile-friendly) ✓	
SCSS File ✓	

Free Updates & Support (minor issues) ✓

User Guide ✓

Extra: Tutorial Source Code ✓

PayPal
Download

Stripe
Download

Crypto
Download

PayPal
Download

Stripe
Download

Crypto
Download

ABOUT AUTHOR

David Adams

Enthusiastic website developer, I've been designing and developing web applications for over 10 years, I enjoy the creativity I put into my projects and enjoy what others bring to the awesome web. My goal is to help newcomers learn the ways of the web.

form mysql mysqli php programming register registration tutorials

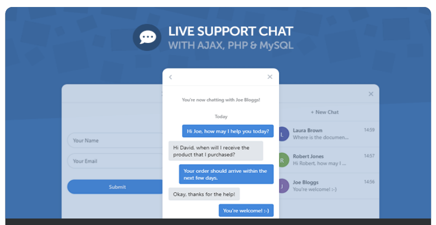
← Secure Login System with PHP and MySQL

Pure CSS3 and HTML5 Tooltips →

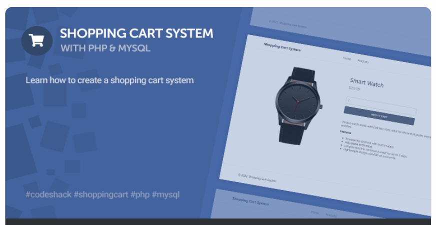
RELATED POSTS

LOGIN & REGISTRATION SYSTEM
PYTHON FLASK & MYSQL
Learn how to create a login & registration system
Register
Login

Login System with Python Flask and MySQL



Live Support Chat with AJAX, PHP and MySQL



Shopping Cart System with PHP and MySQL

G Join the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS ?

Name

11 Share Best Newest Oldest

JV **Jos Velema**
15 days ago
Hi! I just wanted to say that the MVC version is very impressive. I thought I knew a bit about MVC but looking at your code I realize I know nothing XD. A lot of things going on and hard to run down what's what. Guess I have to brush op my knowledge (again) to really get a grip on this and be able to extend it with the other packages. cheers
1 0 • Reply • Share ›

David Adams Mod → Jos Velema
14 days ago
Thanks, @Jos Velema!
0 0 • Reply • Share ›

LH **Ludovic Hoareau**
16 days ago
Je me suis tromper de rubrique, du coup j'ai du acheter 2 versions, le PHP et le nodeJS que je ne vais jamais utilser :'(.
0 0 • Reply • Share ›

LH **Ludovic Hoareau**
16 days ago
Bonjour, j'ai acheter le mauvais fichier, j'ai pris la version nodeJS au lieu du php, il y a t'il un moyen d'échanger svp?
0 0 • Reply • Share ›

David Adams Mod → Ludovic Hoareau
14 days ago
Bonjour @Ludovic Hoareau, thank you for purchasing the advanced package! I'll sort this out. Bear with me.
0 0 • Reply • Share ›

LH **Ludovic Hoareau** → David Adams
11 days ago
Bonjour, merci pour tout David!
Le package en php fonctionne a merveille, je le recommande a tout le monde. :D .
0 0 • Reply • Share ›

David Adams Mod → Ludovic Hoareau
11 days ago
That's awesome, @Ludovic Hoareau! Merci!
1 0 • Reply • Share ›

PC **Phil Caruso**
23 days ago
Hi David, this is great work and love these scripts. I just wanted to know why you chose not to combine your html and php within register.php and instead created a seperate html file. Was this purely cosmetic? The reason I ask is that I'm now having trouble displaying my authentication message as a modal within the same register.html page, which would be ideal. Any suggestions? Thankyou!
0 0 • Reply • Share ›

David Adams Mod → Phil Caruso
21 days ago
Thanks, @Phil Caruso! The feedback is appreciated. The reason for the seperate files is so the developer can differentiate them and understand that executing POST requests from an HTML file doesn't require the file to be a PHP file. Basically, the idea is to keep the client-side code seperated from the server-side code, so the developer can understand the logic.
0 0 • Reply • Share ›

Adam
23 days ago
Hi David, I've been using the Advanced Secure Login & Registration System since March 2021 and haven't

21 days ago
Thanks David.

1 0 • Reply • Share ›

William van de Laar
25 days ago edited
Hi David, love your scripts. only thing i can't solve is that i add 2 extra form fields, firstname and lastname, i think i changed everything good in the register.php but it doesn't work. i got no error message only a blank screen, can you please tell me where the mistake is. Thank you!!

// We need to check if the account with that username exists.
if ($stmt = $con->prepare('SELECT id, password FROM accounts WHERE username = ?')) {
// Bind parameters (s = string, i = int, b = blob, etc), hash the password using the PHP password_hash function.
$stmt->bind_param('s', $_POST['username']);
$stmt->execute();
$stmt->store_result();
// Store the result so we can check if the account exists in the database.
if ($stmt->num_rows > 0) {
// Username already exists
echo 'Username exists, please choose another!';
} else {
// Username doesnt exists, insert new account
if ($stmt = $con->prepare('INSERT INTO accounts (username, password, email, firstname, lastname)

see more

0 0 • Reply • Share ›

David Adams Mod → William van de Laar
25 days ago
Hi @William van de Laar,

I don't see an issue with your PHP code. Did you add the name attribute to your new form fields? They should reflect your POST variable names. I assume you added "firstname" and "lastname" to your MySQL accounts table?

0 0 • Reply • Share ›

William van de Laar → David Adams
25 days ago
Thank you for the respond David, And yes i put the name attributes and offcourse i put the two new in the accounts table.. I'm struggling 3 days with this and i can't find a mistake.
Here is mine register.html, maybe there is an mistake.
<body>
<div class="register">
<h1>Register</h1>
<form action="register.php" method="post" autocomplete="off">

<label for="username">

</label>
<input type="text" name="username" placeholder="Username" id="username" required="">

<label for="password">

</label>

see more

0 0 • Reply • Share ›

David Adams Mod → William van de Laar

Add the following to your registration file:

```
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
```

It should help you identify the error. It could be a typo.

0 0 • Reply • Share ›

WV **William van de Laar** David Adams
22 days ago edited

Thank you david, it was a typo. Now the next problem. In Xamp it works now but if i try it on the server from Strato i get this error:

Could not prepare statement! Fatal error: Uncaught Error: Call to a member function close() on bool in /mnt/web207/c3/44/512126744/htdocs/register.php:54 Stack trace: #0 {main} thrown in /mnt/web207/c3/44/512126744/htdocs/register.php on line 54

Line 54 is $stmt->close();

0 0 • Reply • Share ›

FB **Fred Bonani**
a month ago

Thanks for your response David.
To make sure I understand your response...I would change the code in the register.php code in that location to what you wrote. I ask because the changed code then says to process login instead of continuing with the registration. I want this check at the registration point so that the person can't register if they are not on the eligible_usernames list.

0 0 • Reply • Share ›

David Adams Mod Fred Bonani
a month ago

Yup! You can add the code below the validation.

0 0 • Reply • Share ›

FB **Fred Bonani** David Adams
a month ago

I think I am making progress, but not quite there.

This is the error message I get..."Username exists, please choose another!"

Here is the code I changed per your suggestion:

if ($stmt = $con->prepare('SELECT id, firstName FROM hphs64_reg WHERE lastName = ?')) {
// Bind parameters (s = string, i = int, b = blob, etc), hash the password using the PHP password_hash function.
$stmt->bind_param('s', $_POST['lastName']);
$stmt->execute();
$stmt->store_result();
// Store the result so we can check if the account exists in the database.
if ($stmt->num_rows > 0) {
// Username already exists
echo 'Username exists, please choose another!';
} else {

The hphs64_reg table only contains id, firstName, lastName

I have added firstName, lastName to the registration.html file, and edited the register.php file to handle the two additional field names. I have also added the two additional field names to the accounts table.

Where am I making the mistake?

1 0 • Reply • Share ›

FB **Fred Bonani**
a month ago

What I would like to do is have the registration form check the name being registered against a list of eligible members. If they don't match, then the registration is denied. Is that possible with this form?

0 0 • Reply • Share ›

David Adams Mod Fred Bonani
a month ago

Sure, @Fred Bonani! I assume these eligible usernames will be stored in a database? If so, you can simply use the code that I've already implemented and change it accordingly:

```
$stmt = $con->prepare('SELECT * FROM eligible_usernames WHERE username = ?');
$stmt->bind_param('s', $_POST['username']);
$stmt->execute();
$stmt->store_result();
```

```
if ($stmt->num_rows > 0) {
        // continue to process login
} else {
        exit('Username not eligible!');
}
```

0 0 • Reply • Share ›

FB **Fred Bonani** ➦ David Adams
a month ago
In re-reading this response David, are you saying that the code you provide here is in addition to the existing code in register.php or a replacement for that portion of the code in register.php?

0 0 • Reply • Share ›

David Adams Mod ➦ Fred Bonani
a month ago
Yes, I apologize for the confusion! You can duplicate the current code that I've already implemented and change the query. Basically, copy and paste the existing code.

0 0 • Reply • Share ›

FB **Fred Bonani** ➦ David Adams
a month ago
Does the duplicated code go before or after the original code? Perhaps it goes somewhere else entirely? I am having trouble getting this to work properly.

0 0 • Reply • Share ›

David Adams Mod ➦ Fred Bonani
a month ago
You can place it after or before the validation code.

1 0 • Reply • Share ›

FB **Fred Bonani** ➦ David Adams
a month ago
Thank you very much David...it seems that it is working. I just have to "pretty up" the page that provides the message that the registrant is not eligible to register.

1 0 • Reply • Share ›

David Adams Mod ➦ Fred Bonani
a month ago
Brilliant! :-)

0 0 • Reply • Share ›

C **CITR**
2 months ago edited
Great system! Thank you. The forgot password in the advanced system works great except I'm not getting a email reset. Any suggestions? It's not sending to my email at all or in my spam.

0 0 • Reply • Share ›

David Adams Mod ➦ CITR
2 months ago
Hi @CITR,

Thank you for purchasing the advanced package!

You need a working SMTP mail server to send mail with PHP. If you upload the code to a production server, you shouldn't have this problem because your provider will have one installed and configured.

0 0 • Reply • Share ›

ianhaneyit
2 months ago
Hi

I purchased the Secure Login System with PHP and MySQL - Advanced Package yesterday and using the advanced_pdo version and loving it so far, I just have a quick question if ok

1) Can I add more code in so the user who is signed in edit more details, for example I want to add in address fields and so the user signed in can edit their address details? What file or files do I add the code in to do that please?

0 0 • Reply • Share ›

David Adams Mod ➦ ianhaneyit
2 months ago
Hi @ianhaneyit,

Thank you for purchasing the advanced package! I appreciate the support!

I've responded to your email.

0 0 • Reply • Share ›

Jamie Hoult ➜ David Adams
2 months ago
I'm having this issue too. I'm trying to add first and last name in the system as purchased your pro package and everytime I save details it doesn't update to the database. Any advice on how to do this? I'm using the PDO version

0 0 • Reply • Share ›

ianhaneyit ➜ David Adams
2 months ago
Hi

Thank you, I'll reply to your email in a min

0 0 • Reply • Share ›

CITR
2 months ago
Where is the css to make the login box and registration box larger? It appears very small on mobile devices.

0 0 • Reply • Share ›

David Adams Mod ➜ CITR
2 months ago
The tutorial source doesn't include the responsive code, but you can add the following meta tag to the head section to change the size on mobile devices:

```
<meta name="viewport" content="width=device-width,minimum-scale=1">
```

0 0 • Reply • Share ›

CITR ➜ David Adams
2 months ago
This is already in the code but still the form appears very small on mobile devices. Your style sheets do not affect the size of the box.

0 0 • Reply • Share ›

Manolo
2 months ago
If we want to check if the user has activated their account, we can add the following code to the pages we want to restrict non-activated users: ¿ Help create page ?

0 0 • Reply • Share ›

David Adams Mod ➜ Manolo
2 months ago
All you need to do is implement the code on pages you want to restrict access to, and execute a query to retrieve the activation code column. Search the below comments. I'm pretty sure I provided an example.

0 0 • Reply • Share ›

pomegranate
3 months ago
Hi David,
The "forgot password code" is mixed with "remember me code". Could you please show me only "forgot password code"?
Thank you.

0 0 • Reply • Share ›

David Adams Mod ➜ pomegranate
3 months ago
Hi @pomegranate,

Are you referring to the advanced package? If so, which version are you using? The "remember me" functionality is not included in the "forgot password" add-on as it's a default feature for the package.

0 0 • Reply • Share ›

pomegranate ➜ David Adams
3 months ago
[add-ons-advanced-Forgotpassword-forgotpassword.php, resetpassword.php, forgotpassword.sql] however “include ‘main.php’ and ‘config.php”’
could you please make each one page without include?
Thank you.

0 0 • Reply • Share ›

Alexandre Marleau
4 months ago
hi david , an idea fo a next tutorial would be to make a payable event management platform i think a lot of beginners would appreciate this kind of feature (me included)

0 0 • Reply • Share ›

David Adams Mod → Alexandre Marleau
4 months ago
Thanks for the suggestion! I'll consider it in the future. I'm working on a newsletter tutorial at the moment.

0 0 • Reply • Share ›

L

Lorenzo
5 months ago
Hey David,
do you mind if I ask you for the ERM / RM for the Database stuff for the advanced version?
THX, Cheers

0 0 • Reply • Share ›

David Adams Mod → Lorenzo
5 months ago
Hi @Lorenzo, unfortunately, I don't have the ERM design for the advanced package as there isn't sufficient tables to create a relationship. However, if you're using phpMyAdmin, you can select a database and click the "Designer" tab to view the database in design view and create relationships between tables (see here as an example: https://i.imgur.com/eoD7MHs....

0 0 • Reply • Share ›

D

Darren
5 months ago
Morning,
Im having an issue with the capture code. the file is a broken img link the src to file is correct but it does not show any text. Has any one else had this issue (i am at the moment using localhost). Thanks

0 0 • Reply • Share ›

David Adams Mod → Darren
5 months ago
Hi @Darren, you need to make sure the gd extension is enabled on your server. Follow the below instructions.
1. Open the "php.ini" file. Should be able to access it from the XAMPP control panel.
2. Search for: ;extension=gd
3. Remove ; then restart the server

0 0 • Reply • Share ›

D

Darren → David Adams
5 months ago
Hi again David (lol). That unfortunately hasnt worked.

0 0 • Reply • Share ›

David Adams Mod → Darren
5 months ago
Hmm... Try to add the following to the "php.ini" file:

```
extension=gd2
```

See if that works.

0 0 • Reply • Share ›

Load more comments