



1ST FEB 2014 | IN PHP (C-PHP-BC), TUTORIALS (C-TUTORIALS-BC) BY DAVID CARR

Login And Registration System With PHP

This tutorial will cover creating a login system with registration upon registering an activation link will be emailed containing a link to activate the account. Once active the user can login, a reset option is available to reset the password.

Demo ([Http://Www.Daveismyname.Com/Demos/Loginandregister](http://www.Daveismyname.Com/Demos/Loginandregister))

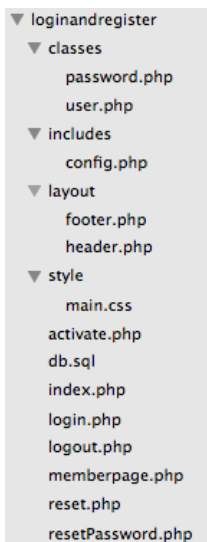
Download

Please click either the Facebook Like button, the Google +1 button or send a tweet to Download the file.

Tweet 104

+1 36

The file structure will be setup as follows:



The database will require a table to store the members, create a table called members:

```
CREATE TABLE `members` (  
  `memberID` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(255) NOT NULL,  
  `password` varchar(60) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `active` varchar(255) NOT NULL,  
  `resetToken` varchar(255) DEFAULT NULL,  
  `resetComplete` varchar(3) DEFAULT 'No',  
  PRIMARY KEY (`memberID`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

In the classes folder there are two files: `password.php` and `user.php`. `password.php` is used to provide the same hashing capability that exists within PHP 5.5; it uses the same function names so versions 5.3 – 5.5 can use the same functions.

`user.php` is a class that contains methods to return the user's hash (hashed password) as well as logging in, checking if a user is logged in, session already exists, and logging the user out.

I'll be going through the `user.php` methods as they are put to use.

Config.php

`Config.php` will be included into all pages to enable sessions and turn on output buffering; this way headers can be used anywhere in the project.

Set the timezone and define the credentials for the database; next attempt to make a new PDO connection; if the connection fails, display the error and kill the page.

Next include the user class and make an instance of it; pass in the database object to the class to make use of the database.

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

//set timezone
date_default_timezone_set('Europe/London');

//database credentials
define('DBHOST','localhost');
define('DBUSER','database username');
define('DBPASS','password');
define('DBNAME','database name');

//application address
define('DIR','http://domain.com/');
define('SITEEMAIL','noreply@domain.com');

try {

    //create PDO connection
    $db = new PDO("mysql:host=".DBHOST.";port=8889;dbname=".DBNAME, DBUSER, DBPASS);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

} catch(PDOException $e) {
    //show error
    echo '<p class="bg-danger">'. $e->getMessage(). '</p>';
    exit;
}

//include the user class, pass in the database connection
include('classes/user.php');
$user = new User($db);
?>
```

Next I have a folder called layout in there is a header.php and footer.php these will contain any layout code that will be used on every page, this saves having to include the stylesheet each time. header.php is a typical header file, notice the title expects a \$title variable, this will be created in the pages and made available to this file, also making use of Bootstrap this is optional and is not required.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title><?php if(isset($title)){ echo $title; }?></title>
    <link href="//netdna.bootstrapcdn.com/bootstrap/3.1.0/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="style/main.css">
</head>
<body>
```

Next footer.php this simply closed the body and html, that would be a good place for placing tracking code or any javascript includes.

```
</body>
</html>
```

Index.php

This is the root page the system loads by default, on this page there is a form for users to register to the site, along with links to the login page, if they are already a member. Also if the user is already logged in they will be redirected to the members page.

How these pages start is by including the config file then checking if the user should be redirected or not.

a call is made to the user object \$user->is_logged_in() this will return true or false if the user is logged in.

```
<?php
//include config
require_once('includes/config.php');

//check if already logged in move to home page
//if logged in redirect to members page
if( $user->is_logged_in() ){ header('Location: memberpage.php'); }
```

The title and header.php file is also included on every page

```
//define page title
$title = 'Demo';

//include header template
require('layout/header.php');
```

For new registrations display a form consisting of username, email, password and confirm password

```

<form role="form" method="post" action="" autocomplete="off">

    <div class="form-group">
        <input type="text" name="username" id="username" class="form-control input-lg" placeholder="User Name" value="<?php if(isset($error)){ echo $_POST['username']; } ?>" tabindex="1">
    </div>
    <div class="form-group">
        <input type="email" name="email" id="email" class="form-control input-lg" placeholder="Email Address" value="<?php if(isset($error)){ echo $_POST['email']; } ?>" tabindex="2">
    </div>
    <div class="row">
        <div class="col-xs-6 col-sm-6 col-md-6">
            <div class="form-group">
                <input type="password" name="password" id="password" class="form-control input-lg" placeholder="Password" tabindex="3">
            </div>
        </div>
        <div class="col-xs-6 col-sm-6 col-md-6">
            <div class="form-group">
                <input type="password" name="passwordConfirm" id="passwordConfirm" class="form-control input-lg" placeholder="Confirm Password" tabindex="4">
            </div>
        </div>
    </div>

    <div class="row">
        <div class="col-xs-6 col-md-6"><input type="submit" name="submit" value="Register" class="btn btn-primary btn-block btn-lg" tabindex="5"></div>
    </div>
</form>

```

This is a standard form, one thing to note I make use of sticky forms which means if there has been a validation error the fields that have been filled out will be populated again with the supplied data, except for passwords. Username and email would be restored.

This is done by doing an if statement, if the array \$error is set meaning it exists then retrain the \$_POST

```
value="<?php if(isset($error)){ echo $_POST['email']; } ?>"
```

If an error has been created it will be stored in an error array to display them loop through the array:

```

//check for any errors
if(isset($error)){
    foreach($error as $error){
        echo '<p class="bg-danger">'. $error. '</p>';
    }
}

```

Once the new registration has been saved the form will post back to the same page appending a `$_GET` key on the end of the URL the key will be called `action` it will have a value of `joined`

(this technique is used through the project)

```
if(isset($_GET['action']) && $_GET['action'] == 'joined'){
    echo "<h2 class='bg-success'>Registration successful, please check your email to activate your account.</h2>";
}
```

The form should only be processed if it has been submitted this can be checked by an if statement:

```
//if form has been submitted process it
if(isset($_POST['submit'])){
```

This way only if the form has been submitted does the validation start and database interactions commence.

Validation

The validation used is fairly basic and can be improved upon

This example checks the length of the username if it's less than 3 characters an error is created, if the first check passes the username is looked up to see if it already exists by passing the username to the database if a record is found an error is created.

```
if(strlen($_POST['username']) < 3){
    $error[] = 'Username is too short.';
} else {
    $stmt = $db->prepare('SELECT username FROM members WHERE username = :username');
    $stmt->execute(array(':username' => $_POST['username']));
    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    if(!empty($row['username'])){
        $error[] = 'Username provided is already in use.';
    }
}
```

These check the password to make sure the email has not been used, it's important the email address is only used once, in the event the user wants to reset their password a link will be emailed to that user.

```

if(strlen($_POST['password']) < 3){
    $error[] = 'Password is too short.';
}

if(strlen($_POST['passwordConfirm']) < 3){
    $error[] = 'Confirm password is too short.';
}

if($_POST['password'] != $_POST['passwordConfirm']){
    $error[] = 'Passwords do not match.';
}

//email validation
if(!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)){
    $error[] = 'Please enter a valid email address';
} else {
    $stmt = $db->prepare('SELECT email FROM members WHERE email = :email');
    $stmt->execute(array(':email' => $_POST['email']));
    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    if(!empty($row['email'])){
        $error[] = 'Email provided is already in use.';
    }
}
}

```

After the validation if no errors have been created then carry on

The password provided cannot be stored as it is, that would be a huge security concern instead it's hashed by passing it to the user object inside a password_hash call this returns a hashed password which can then be stored in the database, this way no one can know what the password was apart from the user who entered it.

If your wondering how can the system login a user in without knowing the password; what happens when the user fills in the login form the password they enter is again hashed and then compared with the hash to see if its a match.

We also want to send an activation link to the user when they register to ensure their email address is active, for this we generate an activation code it will be sent in the emails and will form part of a url to validate the email address.

```

//if no errors have been created carry on
if(!isset($error)){

    //hash the password
    $hashedpassword = $user->password_hash($_POST['password'], PASSWORD_BCRYPT);

    //create the activation code
    $activation = md5(uniqid(rand(),true));
}

```

Next the user's details are saved to the database using a prepared statement, the first page of the query tells MySQL what action to perform in this case to add a new row and the table,columns to insert into.

where their are columns starting with : like :username these are place holders that will be used to bind the username value to \$stmt->execute call. This is done to avoid passing user provided data to the query directly and avoid chances of MySQL Injection.

calling `lastInsertId` followed by the primary key will return the id of the record just saved, this is needed for the next step.

```
$stmt = $db->prepare('INSERT INTO members (username,password,email,active) VALUES (:username, :password, :email, :active)');
$stmt->execute(array(
    ':username' => $_POST['username'],
    ':password' => $hashedpassword,
    ':email' => $_POST['email'],
    ':active' => $activation
));
$id = $db->lastInsertId('memberID');
```

Next send an email to the newly created user. Two constants defined in `config.php` will be used here

`DIR` – contains the full website address

`SITEEMAIL` – the email address used for emails

in the body of the email is a link `activate.php?x=$id&y=$activation` this link is passing the id of the user `$id` and also the activation code when the user received this email, clicking the link will activate their account.

```
$to = $_POST['email'];
$subject = "Registration Confirmation";
$body = "Thank you for registering at demo site.nn To activate your account, please click on this link:nn ".DIR."activate.php?x=$id&y=$activationnn Regards Site Admin nn";
$additionalheaders = "From: <".SITEEMAIL.">\r\n";
$additionalheaders .= "Reply-To: $".SITEEMAIL."";
mail($to, $subject, $body, $additionalheaders);
```

The last step is to redirect the page back to itself and adding an action with the value of `joined` so the page know if to show a success message.

```
header('Location: index.php?action=joined');
exit;
```

Activate.Php

This page checks for the id and activation code being passed from the url (this happens when the user clicks the link from their email)

once the data has been verified the users record is updated, the column `active` is changed from the token to hold 'Yes' to say they are active, this will only happen if the id and token passed match what's stored against that user.

```
<?php
require('includes/config.php');

//collect values from the url
$memberID = trim($_GET['x']);
$active = trim($_GET['y']);

//if id is number and the active token is not empty carry on
if(is_numeric($memberID) && !empty($active)){

    //update users record set the active column to Yes where the memberID and active va
    lue match the ones provided in the array
    $stmt = $db->prepare("UPDATE members SET active = 'Yes' WHERE memberID = :memberID
    AND active = :active");
    $stmt->execute(array(
        ':memberID' => $memberID,
        ':active' => $active
    ));

    //if the row was updated redirect the user
    if($stmt->rowCount() == 1){

        //redirect to login page
        header('Location: login.php?action=active');
        exit;

    } else {
        echo "Your account could not be activated.";
    }

}
?>
```

Login.Php

Now users can register they need a way to login, start off with a form that expects their username and password


```

<form role="form" method="post" action="" autocomplete="off">

    <div class="form-group">
        <input type="text" name="username" id="username" class="form-control input-lg"
placeholder="User Name" value="<?php if(isset($error)){ echo $_POST['username']; } ?>"
tabindex="1">
    </div>

    <div class="form-group">
        <input type="password" name="password" id="password" class="form-control input-
lg" placeholder="Password" tabindex="3">
    </div>

    <div class="row">
        <div class="col-xs-9 col-sm-9 col-md-9">
            <a href='reset.php'>Forgot your Password?</a>
        </div>
    </div>

    <hr>
    <div class="row">
        <div class="col-xs-6 col-md-6"><input type="submit" name="submit" value="Login"
class="btn btn-primary btn-block btn-lg" tabindex="5"></div>
    </div>
</form>

```

The login page will be used to show messages if the users account has been activated or password has been changed, the page will know which message to show based on the value contained inside `$_GET['action']`

```

if(isset($_GET['action'])) {

    //check the action
    switch ($_GET['action']) {
        case 'active':
            echo "<h2 class='bg-success'>Your account is now active you may now log in.
</h2>";
            break;
        case 'reset':
            echo "<h2 class='bg-success'>Please check your inbox for a reset link.</h2>";
            break;
        case 'resetAccount':
            echo "<h2 class='bg-success'>Password changed, you may now login.</h2>";
            break;
    }
}

```

Next attempt to log the user in. Collect the username and password from the form pass them to the users object in the login method this internally will fetch the users hash by looking for the username in the database once the hash is returned it's then passed to password_verify if the hash and user's hash match it returns true which in turns sets a

session \$_SESSION['loggedin'] to true otherwise false is returned.

```

public function login($username,$password){

    $hashed = $this->get_user_hash($username);

    if($this->password_verify($password,$hashed) == 1){

        $_SESSION['loggedin'] = true;
        return true;
    }
}

<pre lang="php">
//process login form if submitted
if(isset($_POST['submit'])){

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

    if($user->login($username,$password)){

        header('Location: memberpage.php');
        exit;

    } else {
        $error[] = 'Wrong username or password or your account has not been activated.'
    }
;
}

} //end if submit

```

Logout.Php

To log a user out its very easy:

```

//logout
$user->logout();

```

Once the user is logged out redirect them.

Memberpage.Php

Once the user is logged in redirect them to the members only page (optional). To ensure a user can only access the page if logged in do a check:

```

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: login.php'); }

```

In this example their is not a lot to the members page namely:

```
<h2>Member only page</h2>
<p><a href='logout.php'>Logout</a></p>
```

Reset.Php

every system need the ability to reset a password in case it's forgotten, how this will work is a user enters their email address, a check is made to make sure its belongs to a user.

Next a token is created and saved to the users record, an email is sent to them containing a link to when clicked the token from the link is verified, if it passed the user is provided with a form to enter their new password, its then saved to the database.

This may seem like a long winded approach but it does prevent the password being sent by email which is not recommended.

To start with the form:

```
<form role="form" method="post" action="" autocomplete="off">
  <div class="form-group">
    <input type="email" name="email" id="email" class="form-control input-lg" place
holder="Email" value="" tabindex="1">
  </div>

  <hr>
  <div class="row">
    <div class="col-xs-6 col-md-6"><input type="submit" name="submit" value="Sent R
eset Link" class="btn btn-primary btn-block btn-lg" tabindex="2"></div>
  </div>
</form>
```

If their is an \$_GET['action'] show the correct message

```
<?php
if(isset($_GET['action'])) {

    //check the action
    switch ($_GET['action']) {
        case 'active':
            echo "<h2 class='bg-success'>Your account is now active you may now log in.
</h2>";
            break;
        case 'reset':
            echo "<h2 class='bg-success'>Please check your inbox for a reset link.</h2>
";
            break;
    }
}
?>
```

Next process the form ensure the email matches a user:

```
//email validation
if(!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)){
    $error[] = 'Please enter a valid email address';
} else {
    $stmt = $db->prepare('SELECT email FROM members WHERE email = :email');
    $stmt->execute(array(':email' => $_POST['email']));
    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    if(empty($row['email'])){
        $error[] = 'Email provided is not on recognised.';
    }

}
}
```

Create the token

```
//create the activation code
$token = md5(uniqid(rand(), true));
```

Next update the users record and set resetToken to the value of the token and resetComplete to No that will be needed if the link is clicked and password has been changed. Send an email to the user containing a link that points to resetPassword.php?key=\$token passing the token.

```
$stmt = $db->prepare("UPDATE members SET resetToken = :token, resetComplete='No' WHERE email = :email");
$stmt->execute(array(
    ':email' => $row['email'],
    ':token' => $token
));

//send email
$to = $row['email'];
$subject = "Password Reset";
$body = "Someone requested that the password be reset. nnIf this was a mistake, just ignore this email and nothing will happen.nnTo reset your password, visit the following address: ".DIR."resetPassword.php?key=$token";
$additionalheaders = "From: <".SITEEMAIL.">\r\n";
$additionalheaders .= "Reply-To: $".SITEEMAIL."";
mail($to, $subject, $body, $additionalheaders);

//redirect to index page
header('Location: login.php?action=reset');
exit;
```

ResetPassword.Php

First check the token been passed to the page matches a user

```

$stmt = $db->prepare('SELECT resetToken, resetComplete FROM members WHERE resetToken = :token');
$stmt->execute(array(':token' => $_GET['key']));
$row = $stmt->fetch(PDO::FETCH_ASSOC);

//if no token from db then kill the page
if(empty($row['resetToken'])) {
    $stop = 'Invalid token provided, please use the link provided in the reset email.';
} elseif($row['resetComplete'] == 'Yes') {
    $stop = 'Your password has already been changed!';
}

```

If \$stop has been set then display that

```

if(isset($stop)){
    echo "<p class='bg-danger'>$stop</p>";
}

```

If no errors have been created show a form to change the password

```

<form role="form" method="post" action="" autocomplete="off">
    <div class="row">
        <div class="col-xs-6 col-sm-6 col-md-6">
            <div class="form-group">
                <input type="password" name="password" id="password" class="form-control input-lg" placeholder="Password" tabindex="1">
            </div>
        </div>
        <div class="col-xs-6 col-sm-6 col-md-6">
            <div class="form-group">
                <input type="password" name="passwordConfirm" id="passwordConfirm" class="form-control input-lg" placeholder="Confirm Password" tabindex="1">
            </div>
        </div>
    </div>

    <hr>
    <div class="row">
        <div class="col-xs-6 col-md-6"><input type="submit" name="submit" value="Change Password" class="btn btn-primary btn-block btn-lg" tabindex="3"></div>
    </div>
</form>

```

Once the form has been submitted validate the data then hash the password update the users row and set resetComplete to Yes to indicate the process is finished if the reset link is clicked again from email the process will be halted.

```

//if form has been submitted process it
if(isset($_POST['submit'])){

    //basic validation
    if(strlen($_POST['password']) < 3){
        $error[] = 'Password is too short.';
    }

    if(strlen($_POST['passwordConfirm']) < 3){
        $error[] = 'Confirm password is too short.';
    }

    if($_POST['password'] != $_POST['passwordConfirm']){
        $error[] = 'Passwords do not match.';
    }

    //if no errors have been created carry on
    if(!isset($error)){

        //hash the password
        $hashedpassword = $user->password_hash($_POST['password'], PASSWORD_BCRYPT);

        try {

            $stmt = $db->prepare("UPDATE members SET password = :hashedpassword, resetC
omplete = 'Yes' WHERE resetToken = :token");
            $stmt->execute(array(
                ':hashedpassword' => $hashedpassword,
                ':token' => $row['resetToken']
            ));

            //redirect to index page
            header('Location: login.php?action=resetAccount');
            exit;

            //else catch the exception and show the error.
        } catch(PDOException $e) {
            $error[] = $e->getMessage();
        }

    }

}
}

```

Conclusion

That covers the foundations, this can be used as a starting point to build members based sites or even a start to an admin panel.

I'll making a copy of this tutorial that is compatible with **Simple MVC Framework** (<http://www.simplemvcframework.com>) for those that prefer the MVC approach.

Install From Download Github

- Go to <https://github.com/daveismynamecom/loginregister> (<https://github.com/daveismynamecom/loginregister>)
- Download the files
- import db.sql into your database
- Open includes/config.php and add database details

Tweet 104  1 Share 1 810

Enter Email Address

Subscribe by Email

Follow @daveismynamecom



AdChoices

The Real-Time Cloud  TRY NOW

← Connecting to Gmail with imap_open (<http://daveismyname.com/connecting-to-gmail-with-imap-open-bp>)

lator to convert currency with PHP → (<http://daveismyname.com/use-google-finance-calculator-to-convert-currency-with-php-bp>)

ALSO ON DAVE IS MY NAME

WHAT'S THIS?

Display Tweets from Twitter for API 1.1 2 comments

Quick way to add hours and minutes with PHP - David Carr - Web Developer 2 comments

Free Bootstrap Admin Themes 16 comments

Building a content management system from scratch - SEO URLs - David Carr PHP Web Developer Hull 4 comments

80 Comments Dave is my name

Login ▾

Sort by Best ▾

Share  Favorite ★



Join the discussion...



belhelhe • 5 months ago

if anyone is getting this error: SQLSTATE[HY000] [2002] ; just change "\$db = new PDO("mysql:host=".DBHOST.";port=8889;dbname=".DBNAME, DBUSER, DBPASS);" to "\$db = new PDO("mysql:host=".DBHOST.";port=3306;dbname=".DBNAME, DBUSER, DBPASS);" in config.php file ;)

4 ^ | ▾ • Reply • Share ▾



Abu Awal Md. Shoeb → belhelhe • a month ago



thanks!

• Reply • Share ›

**Wilson** • 6 months ago

Thanks for this great tutorial, i will still have to go through it one more time and try it out but i have always been

ob_start(); and still dont know it use. :(

2 • Reply • Share ›

**David Carr** Mod Wilson • 6 months ago

it means turn on output buffering that way you can call a header function anywhere on the page.

• Reply • Share ›

**Cheebs** • 6 months ago

Hello Dave,

Thanks for the great tutorial!

1 • Reply • Share ›

**Guest** • 6 months ago

Hello Dave, everything is working now. The issue regarding the password not being saved into the database has been solved. For some, changing the collation into utf8 will work out just fine but not for me. So, i decided to change my hosting, i guess it has something to do with the version. By the way, in the memberspage, can i ask a little favor? I just wanted to output soething like this, welcome user! So how will i do the trick?

1 • Reply • Share ›

**David Carr** Mod Guest • 6 months ago

in the user class edit the login method to add a session to store the username

```
public function login($username,$password){

    $hashed = $this->get_user_hash($username);

    if($this->password_verify($password,$hashed) == 1){

        $_SESSION['loggedin'] = true;
        $_SESSION['username'] = $username;

        return true;

    }
}
```

then in the welcome page you can do

```
echo $_SESSION['username'];
```

• Reply • Share ›

**Sundeep Next** • 2 days ago

SQLSTATE[HY000] [1045] Access denied for user 'database username'@'localhost' (using password: YES)

i create database with "db" name and table name "members"

and

change this

```
//database credentials
define('DBHOST','localhost');
define('DBUSER','database username');
define('DBPASS','password');
define('DBNAME','db');
```

but same error....

pls guide me...

• Reply • Share ›

**roxstarsxi** • 2 days ago

I am getting problem with login after activation as many have posted below.Can you please suggest some good solution.?

• Reply • Share ›

**Scott Leitch** • 3 days ago



Hi Dave,

Thanks for this little tutorial.

All seems to be working fine apart from the activation link in the registration email. I have updated the database credentials and application address in the config.php to point to my server. But when I click on the link in the generated email I get a 404 page.

Any ideas?

^ | v • Reply • Share ›

**Scott Leitch** → Scott Leitch • 3 days ago

My bad, I forgot to update the filepath...

Rookie mistake.

^ | v • Reply • Share ›

**eDnS** • 13 days ago

hello,

I would like to know how to perform the addition of account validation but only by an admin with sending mail to the user that the account is validated please?

sorry for my English

^ | v • Reply • Share ›

**Sarah Khan** • 20 days ago

Please tell me what to do after downloading these files from GitHub? Anyone please tell me how to install it on website? There is MyphpAdmin and Sql option in my hosting but I don;t know how to create Registration and logging system. :(Can anyone help please?? mail me iAmAamirRana@gmail.com . m waiting.

^ | v • Reply • Share ›

**David Carr** Mod → Sarah Khan • 20 days ago

upload the files to your server, import the sql file into your database using phpmyadmin then update the settings in includes/config.php

^ | v • Reply • Share ›

**Pfft** • 23 days ago

Hi Dave! I published the site and its working fine except for the send mail ? does it have anything to do with the hosting?

^ | v • Reply • Share ›

**David Carr** Mod → Pfft • 23 days ago

the script uses php's mail function, some servers require special setup for sending email.

^ | v • Reply • Share ›

**Pfft** → David Carr • 23 days ago

I see, I believe they're (web hosting) is using lamp, at my end i just edited these lines

```
//application address
define('DIR','http://www.xxx.com/');
define('SITEEMAIL','xxx@xxx.com');
```

I'm relatively new to php (and web hosting for that matter), any help will be appreciated. Thanks :)

^ | v • Reply • Share ›

**David Carr** Mod → Pfft • 22 days ago

if their running this locally they would need to configure their setup to be able to send emails if their using online hosting this should be in place already.

If using wamp/xampp or anything like that then search for email setup on google their should be plenty of articles.

^ | v • Reply • Share ›

**Abu Awal Md. Shoeb** • a month ago

I have problem in login.php page showing "Wrong username or password or your account has not been activated." Can you advise what to do please?

Btw, this is a fantastic code! Thanks

^ | v • Reply • Share ›

**kennethjwoodard** → Abu Awal Md. Shoeb • 8 hours ago

Has this been corrected? I am having the same issue. Also, my activate.php doesn't seem to be redirecting to my login.php.

^ | v • Reply • Share ›

**David Carr** Mod → Abu Awal Md. Shoeb • a month ago

did you click on the activation link in the email?

^ | v • Reply • Share ›

**kennethjwoodard** → David Carr • 8 hours ago

Has this been corrected? I am having the same issue. Also, my activate.php doesn't seem to be redirecting to my login.php.

^ | v • Reply • Share ›

**Abu Awal Md. Shoeb** → David Carr • a month ago

yes and the value in the db got changed once it is activated but still doesn't work. later i changed the password hashing to md5 to continue. thanks a lot for the scripts.

^ | v • Reply • Share ›

**roxstarsxi** → Abu Awal Md. Shoeb • 3 days ago

yes same problem..can you please provide correct code !!

^ | v • Reply • Share ›

**Richard Nash** • a month ago

Great Script is there a quick way to show the user information on the members.php?

^ | v • Reply • Share ›

**David Carr** Mod → Richard Nash • a month ago

doing a query to return the desired info would be how I would do it.

^ | v • Reply • Share ›

**Rick** → David Carr • a month ago

thanks for the reply :), any chance you could show me a quick example

^ | v • Reply • Share ›

**David Carr** Mod → Rick • a month ago

when the user logs in save the memberID into a session then you can do:

```
$stmt = $db->prepare('SELECT * FROM members WHERE memberID = :memberID');
```

```
$stmt->execute(array(':memberID' => $_SESSION['memberID']));
```

```
$row = $stmt->fetch(PDO::FETCH_ASSOC);
```

then you have access to all the user info inside \$row

^ | v • Reply • Share ›

**Rick** → David Carr • a month ago

thx :)

^ | v • Reply • Share ›

**santosh** • 2 months ago

Sir, i am beginner with PHP and your tutorial has helped me a lot. I would like to know how to direct a user to a specific page admin.php, officer.php, dataoperators.php depending on his role like admin, officer, dataoperators.

^ | v • Reply • Share ›

**David Carr** Mod → santosh • a month ago

you could have a role column in the database that stored an ID or the role name then when the user logs in redirect depending on the role:

```

switch ($role) {
case 'admin':
$loc = 'admin.php';
break;
case 'officer':
$loc = 'officer.php';
break;
case 'dataoperators':
$loc = 'dataoperators.php';
break;
}

```

```

header('Location: '.$loc);
exit();

```

^ | v • Reply • Share ›

**Bob Ramey** • 3 months ago

Hi Dave - I really like this script. I'm using it to build a simple CMS for a client. It is installed in the 'login' directory of the site. I've run into a problem I'm hoping you might be able to help me with.

I added some tables to the db to keep track of some things in the site. I created a protected page where the logged in user can update some form fields with HTML and then click submit to update the database. Seems pretty straight forward. On this particular page, I have two fields, and I'm using tinyMCE as an editor. When I edit the first field and submit, everything works fine. DB is updated and all is right with the world. But, when I add any kind of HTML tags to the 2nd field (the one called Links) and hit submit, I get redirected back to the homepage in the root folder. The db is not updated. And to add to the weirdness, the URL of the site remains '<http://coastcms2.dandelion-pat...>' and not '<http://coastcms2.dandelion-pat...>', which is the page that is clearly being displayed. I just can't explain this.

I thought it might be a problem with tinyMCE. But I removed that script and just edited the text areas by hand. Same result. Adding HTML to the 2nd field (Links) results in the weird redirect, and no db update.

Head over to my testing site at <http://coastcms2.dandelion-pat...> and login in with admin/admin. Click on the Edit Header link to go to the page in question.

Any thoughts or ideas would be greatly appreciated!

^ | v • Reply • Share ›

**David Carr** Mod → Bob Ramey • 3 months ago

Hi this is strange it appears to be loading index.php when the links box is used. Looking at the site I'm unable to see what the cause could be.

I could look over the files if you would allow me FTP access or email me them and I'll be able to see what's causing it.

My email is dave@daveismyname.com

^ | v • Reply • Share ›

**Bob Ramey** → David Carr • 3 months ago

Just emailed you the files. Thanks David!

^ | v • Reply • Share ›

**Stevie** • 5 months ago

The same problem as Adrian. Cannot log in even though the users record is activated - "Wrong username or password or your account has not been activated."

^ | v • Reply • Share ›

**ravi2143** → Stevie • 22 days ago

hi, i also got the same problem but i got it, its a silly mistake we did in the place of USER NAME we are entering email id go to you data base and see under user name email id is not user name . every thing is working fine even the password need not to change to md5 its a grate script it helped a lot it took lot of time to understand coz i am new to PHP, any way thanks a lot.

^ | v • Reply • Share ›

**rusty** • 5 months ago

hey bro! i need a help! can u give me ur skype please! here is my gmail ----> rusty.ctp@gmail.com please contact me as fast u can thanks bro)

^ | v • Reply • Share ›

**Tang William** • 5 months ago

this is a form which i found out the most super great useful and easy to understand. And i have tested in real server and it runs well. May i ask do you know how to add a specify file or a folder according the user name who has click register successfully?

^ | v • Reply • Share ›

**Guest** • 6 months ago

Hey Dave,

Thanks for this tutorial, I just get this error SQLSTATE[HY000] [1045] Access denied for user 'database username'@'localhost' (using password: YES).

Would you please give me some advice?:D

^ | v • Reply • Share ›

**David Carr** Mod → Guest • 5 months ago

the database credentials need changing to your own.

1 ^ | v • Reply • Share ›

**Hawks** • 6 months ago

Hello Dave,

What's need to change for merge login and register in just One Page? I've tryed but got error: Warning: Invalid argument supplied for foreach() .../login/index.php

^ | v • Reply • Share ›



Suzie • 6 months ago

I've downloaded your files from Github, have created a database and changed the details in the config.php file but when I try to register a user I get a blank screen. Nothing is created in the database either.

^ | v • Reply • Share ›



Rick → Suzie • a month ago

this happened to me :) check the script for the PHP tags if its

^ | v • Reply • Share ›



Hawks → Suzie • 6 months ago

look for php version... need to 5.3 or higher

^ | v • Reply • Share ›



Suzie → Hawks • 6 months ago

I have 5.5.3

1 ^ | v • Reply • Share ›



Hawks → Suzie • 6 months ago

Ok. Show any errors?

2014-04-16 19:42 GMT-03:00 Disqus <notifications@disqus.net>:

^ | v • Reply • Share ›



Mr.RicoJ • 6 months ago

Thanks for this great tutorial....

How to add user picture profile?

Can u add with upload / edit image & store the path file into mysql?

Appreciate so much if you can.... :D

^ | v • Reply • Share ›



Hawks • 6 months ago

Congrats for this great job! I want know how to change password to md5 hash? I mean, whats files and line need to add script?
md5((\$_POST['password']))

Best Regards,

^ | v • Reply • Share ›



David Carr Mod → Hawks • 6 months ago

MD5 is not secure at all.

To use MD5 replace password_hash() and password_verify function calls with MD5(\$password).

^ | v • Reply • Share ›



Hawks → David Carr • 6 months ago

Thanks in advance, David.

so, which hash do you prefer at all?

sha512? bcrypt?

2014-04-10 7:18 GMT-03:00 Disqus <notifications@disqus.net>:

^ | v • Reply • Share ›

Load more comments

✉ Subscribe

D Add Disqus to your site

🔒 Privacy

Recent Posts

Select previous and next rows in MySQL (<http://daveismyname.com/select-previous-and-next-rows-in-mysql-bp>)

5th Oct 2014



(<http://daveismyname.com/select-previous-and-next-rows-in-mysql-bp>)



My TOP Sublime Text plugins (<http://daveismyname.com/my-top-sublime-text-plugins-bp>)

28th Sep 2014

(<http://daveismyname.com/my-top-sublime-text-plugins-bp>)



Backup to Dropbox with PHP (<http://daveismyname.com/backup-to-dropbox-with-php-bp>)

20th Sep 2014

(<http://daveismyname.com/backup-to-dropbox-with-php-bp>)



Running mysqldump with PHP (<http://daveismyname.com/running-mysqldump-with-php-bp>)

20th Sep 2014

(<http://daveismyname.com/running-mysqldump-with-php-bp>)



Using closures and \$this within a class with PHP (<http://daveismyname.com/using-closures-and-this-within-a-class-with-php-bp>)

12th Sep 2014

(<http://daveismyname.com/using-closures-and-this-within-a-class-with-php-bp>)

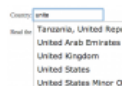
Popular Posts



Free Bootstrap Admin Themes (<http://daveismyname.com/free-bootstrap-admin-themes-bp>)

3rd May 2014

(<http://daveismyname.com/free-bootstrap-admin-themes-bp>)



Autocomplete with PHP, MySQL and JQuery UI (<http://daveismyname.com/autocomplete-with-php-mysql-and-jquery-ui-bp>)

27th Feb 2013

(<http://daveismyname.com/autocomplete-with-php-mysql-and-jquery-ui-bp>)



Creating a blog from scratch with PHP (<http://daveismyname.com/creating-a-blog-from-scratch-with-php-bp>)

7th Jun 2013

(<http://daveismyname.com/creating-a-blog-from-scratch-with-php-bp>)



Login and Registration system with PHP (<http://daveismyname.com/login-and-registration-system-with-php-bp>)

1st Feb 2014

(<http://daveismyname.com/login-and-registration-system-with-php-bp>)




Building a content management system from scratch (<http://daveismyname.com/building-a-content-management-system-from-scratch-bp>)

6th May 2011

(<http://daveismyname.com/building-a-content-management-system-from-scratch-bp>)

Categories

Design (<http://daveismyname.com/c-design-bc>)
 Development (<http://daveismyname.com/c-development-bc>)
 Networking (<http://daveismyname.com/c-networking-bc>)
 Reviews (<http://daveismyname.com/c-reviews-bc>)
 SEO (<http://daveismyname.com/c-seo-bc>)
 Simple MVC Framework (<http://daveismyname.com/c-simple-mvc-framework-bc>)
 Social Media (<http://daveismyname.com/c-social-media-bc>)
 Tools (<http://daveismyname.com/c-tools-bc>)
 Tutorials (<http://daveismyname.com/c-tutorials-bc>)
 CSS (<http://daveismyname.com/c-css-bc>)
 Flash (<http://daveismyname.com/c-flash-bc>)
 Htaccess (<http://daveismyname.com/c-htaccess-bc>)
 HTML (<http://daveismyname.com/c-html-bc>)
 Javascript (<http://daveismyname.com/c-javascript-bc>)
 PHP (<http://daveismyname.com/c-php-bc>)
 Wordpress (<http://daveismyname.com/c-wordpress-bc>)
 Wordpress Plugins (<http://daveismyname.com/c-wordpress-plugins-bc>)



(<http://daveismyname.com/login-and-registration-system-with-php-bp#.VD-ocnWSxCU>)

source=followbutton&varia