Creating a blog from scratch with PHP

BLOG (BLOG), PHP & MYSQL (PHP-MYSQL), TUTORIALS (TUTORIALS)

Blog Series

Part 1 - The Build (http://daveismyname.blog/creating-a-blog-from-scratch-with-php)
Part 2 - SEO URLS (http://daveismyname.blog/creating-a-blog-from-scratch-with-php-part-2-seo-urls)
Part 3 Comments with Disqus (http://daveismyname.blog/creating-a-blog-from-scratch-with-php-part-3-comments-with-disqus)
Part 4 Categories (http://daveismyname.blog/creating-a-blog-from-scratch-with-php-part-4-categories)
Part 5 Sidebar, Categories and Archives (http://daveismyname.blog/creating-a-blog-from-scratch-with-php-part-5-sidebar-categories-and-archives)
Part 6 Pagination (http://daveismyname.blog/creating-a-blog-from-scratch-with-php-part-6-pagination)

This tutorial will cover creating a very simple blog. It will only consist of posts. The front-end will only be two pages, an index page to list all posts and a view page to view a post.

There will be a backend control panel for managing posts and admins, this guide will also include a user authentication system to login administrators.

Demo (https://daveismyname.blog/demo/creating-a-blog-from-scratch-with-php)

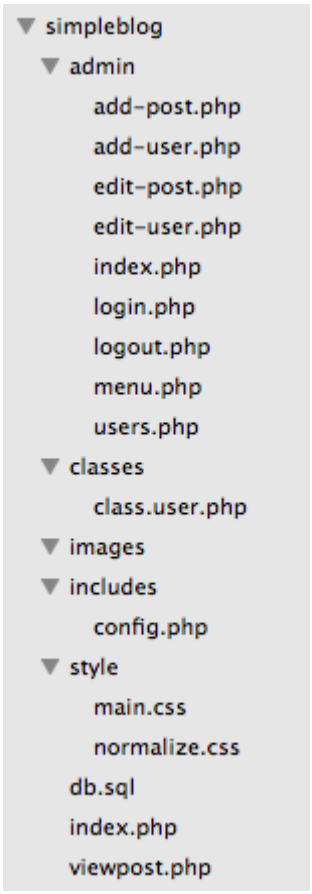
Download source files (https://github.com/daveismynamecom/simple-blog-part-1-build)

admin demo (http://www.daveismyname.blog/demos/simpleblog/admin):

username: demo
password: demo

I will reinstall the demo every hour.

The file structure will be setup as follows:



There will be two mysql tables, blog_posts and blog_members.

blog_posts structure:

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra
postID	INT	11	☑	☐	☐	☐	PRI		auto_increment
postTitle	VARCHAR	255	☐	☐	☐	☑		NULL	None
postDesc	TEXT		☐	☐	☐	☑			None
postCont	TEXT		☐	☐	☐	☑			None
postDate	DATETIME		☐	☐	☐	☑		NULL	None

blog_members structure

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra
memberID	INT	11	☑	☐	☐	☐	PRI		auto_increment
username	VARCHAR	255	☐	☐	☐	☑		NULL	None
password	VARCHAR	255	☐	☐	☐	☑		NULL	None
email	VARCHAR	255	☐	☐	☐	☑		NULL	None

Every page will need a database connection, this will be opened in config.php:

includes/config.php

Start output buffering, then headers can be used anywhere. Start sessions this will be needed for the admin area.

Define the database connection details then open a PDO connection.

Also set the timezone, adjust this as needed.

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

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

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

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

Next create an autoload function this will include any class as it is called this stops you having to manually include all classes, for this project there is only a single class but this set up the feature for more classes in future developments.

Inside the function, the name passed to is is converted into lowercase then check to see if the file exists if it does it is then included.

Lastly the user class is instantiated and passed the database connection ($db) so the class has access to the database.

```
//load classes as needed
function __autoload($class) {

   $class = strtolower($class);

   $classpath = 'classes/class.'.$class . '.php';
   if ( file_exists($classpath)) {
      require_once $classpath;
    }

   $classpath = '../classes/class.'.$class . '.php';
   if ( file_exists($classpath)) {
      require_once $classpath;
    }


}

$user = new User($db);
```

index.php

The index file will list all posts from the posts table.
A query is ran to select the columns from blog_posts then ordered by the postID in descending order.
Then the posts are looped through on each loop display the title, description date posted and a link to read the full post.

The query is wrapped inside a try catch statement so if there is any errors a PDO Exception will be used to display them.

The id for the post is past to the next page in what's called a query string ?id will become a variable in the url = assignes the value.

```
<?php
    try {

        $stmt = $db->query('SELECT postID, postTitle, postDesc, postDa
        while($row = $stmt->fetch()){

            echo '<div>';
                echo '<h1><a href="viewpost.php?id='.$row['postID'].''
                echo '<p>Posted on '.date('jS M Y H:i:s', strtotime($r
                echo '<p>'.$row['postDesc'].'</p>';
                echo '<p><a href="viewpost.php?id='.$row['postID'].'">
            echo '</div>';

        }

    } catch(PDOException $e) {
        echo $e->getMessage();
    }
?>
```

viewpost.php

viewpost.php is used to display any post that has been clicked on.

This query will use a prepared statement, the record to be selected is based upon the id been passed from a $_GET['id'] request, as such a normally query is not recommended, a prepared statement is much better. The prepare will 'prepare' the database for query to be run then when $stmt->execute is ran the items from the array will be bound and sent to the database server, at no point does the two connect to there is no way to tamper with the database.

```
$stmt = $db->prepare('SELECT postID, postTitle, postCont, postDate FRO
$stmt->execute(array(':postID' => $_GET['id']));
$row = $stmt->fetch();
```

If there is no postID coming from the database, their is no record so redirect the user to the index page.

```
if($row['postID'] == ''){
    header('Location: ./');
    exit;
}
```

Lastly display the select post in full:

```
echo '<div>';
    echo '<h1>'.$row['postTitle'].'</h1>';
    echo '<p>Posted on '.date('jS M Y', strtotime($row['postDate'])).'
    echo '<p>'.$row['postCont'].'</p>';
echo '</div>';
```

That's it for the front end! its very simple but its also very easy to expand and add more features.

Admin

Every page in the admin area will start by including the config file and checking if the admin is logged in, otherwise they are redirect to the login page.

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

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

admin/login.php

The login page will check if the user is already logged in, if they are they will be redirect to the main admin area.

The login page is very simple there is a login form which accepts a username and password.

```
<form action="" method="post">
<p><label>Username</label><input type="text" name="username" value=""
<p><label>Password</label><input type="password" name="password" value
<p><label></label><input type="submit" name="submit" value="Login"  />
</form>
```

Once submitted the username and password are collected from the form then passed to a login method in the user class (I'll come to that shortly) if this returns true, they have logged in are are taken to the admin otherwise they are shown an error.

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

    $username = trim($_POST['username']);
    $password = trim($_POST['password']);

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

        //logged in return to index page
        header('Location: index.php');
        exit;


    } else {
        $message = '<p class="error">Wrong username or password</p>';
    }

}//end if submit

if(isset($message)){ echo $message; }
```

classes/class.user.php

The user class is used to login and logout users verify their password and create a hash of their password.

The first function that will get called as soon as the class is ran is an automatic function called __construct this method is passed a database connection this is then assigned to a variable within the class so all methods will have access to it.

```
private $db;

public function __construct($db){
    $this->db = $db;
}
```

To check if a user is logged in a method is_logged_in() looks for a session called loggedin if its set and is true their is a logged in user and returns true otherwise is would return nothing.

```
public function is_logged_in(){
    if(isset($_SESSION['loggedin']) && $_SESSION['loggedin'] == true){
        return true;
    }
}
```

The get_user_hash method is used to get the columns from the database and return them.

```
private function get_user_hash($username){

    try {

        $stmt = $this->_db->prepare('SELECT MemberID, username, passwo
        $stmt->execute(array('username' => $username));

        return $stmt->fetch();

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

To verify a hash, the below method is used. pass the password from the form and the hashed password from the database this should equal to 1 otherwise they do not match.

```
if($this->password_verify($password,$user['password']) == 1){
//match
}
```

In order to verify a password matched a password given on login the hashed password needs to be fetched from the database, the username is passed to the database and the hashed password is returned.

The login method expects the users username and password then the fetches users password based on their username from the get_user_hash method in order to use the verify_hash method.
If a match is found a session is set and the method returns true.

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

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

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

        $_SESSION['loggedin'] = true;
        $_SESSION['memberID'] = $user['memberID'];
        $_SESSION['username'] = $user['username'];
        return true;
    }
}
```

admin/index.php

The blog posts are listed in a table, again using a query to select all records and display them ordered by the postID in descending order, then looped through to list all posts, each post has an edit and delete link the edit link passes the postID to edit-post.php in order to edit the selected post.
The delete link calls a javascript function (delpost) it expects the id of the

post and the title of the post, when clicked the javascript function is executed which will run a confirmation popup asking to confirm to delete the post.

```
<table>
<tr>
    <th>Title</th>
    <th>Date</th>
    <th>Action</th>
</tr>
<?php
    try {

        $stmt = $db->query('SELECT postID, postTitle, postDate FROM bl
        while($row = $stmt->fetch()){

            echo '<tr>';
            echo '<td>'.$row['postTitle'].'</td>';
            echo '<td>'.date('jS M Y', strtotime($row['postDate'])).'<
            ?>

            <td>
                <a href="edit-post.php?id=<?php echo $row['postID'];?>
                <a href="javascript:delpost('<?php echo $row['postID']
            </td>

            <?php
            echo '</tr>';

        }

    } catch(PDOException $e) {
        echo $e->getMessage();
    }
?>
</table>
```

The javascript confirm function, once confirmed a command window.locaation.href is ran which will redirect the page, in this case it goes to index.php again but appends ?delpost= and the id of the post to be deleted, which in turn will execute a php function.

```
<script language="JavaScript" type="text/javascript">
function delpost(id, title)
{
  if (confirm("Are you sure you want to delete '" + title + "'"))
  {
      window.location.href = 'index.php?delpost=' + id;
  }
}
</script>
```

PHP Delete function

If the get request delpost has been sent then a prepared statement is ran to delete the post where the postID matches the id passed in the array. Then the page is reloaded passing a status to the url in indexphp?action= the action is used on the page to confirm the deletion.

```
if(isset($_GET['delpost'])){

    $stmt = $db->prepare('DELETE FROM blog_posts WHERE postID = :postI
    $stmt->execute(array(':postID' => $_GET['delpost']));

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

if there has been an action passed in a $_GET request then display it.

```
if(isset($_GET['action'])){
    echo '<h3>Post '.$_GET['action'].'.</h3>';
}
```

There is an admin menu that will be display on every page, whilst this could be added to each page it makes more sense to add links to a separate file called menu.php that can be included into every file, that way any changes only need to be applied once.

```
<?php include('menu.php');?>
```

admin/menu.php

The menu is very simple for this site the links are inside a ul list, the view website links back to the root of the project by going back a directory using ../ in the href path.

```
<h1>Blog</h1>
<p>Logged in as <?=$_SESSION['username'];?></p>
<ul id='adminmenu'>
    <li><a href='index.php'>Blog</a></li>
    <li><a href='users.php'>Users</a></li>
    <li><a href="../" target="_blank">View Website</a></li>
    <li><a href='logout.php'>Logout</a></li>
</ul>
<div class='clear'></div>
<hr />
```

admin/add-post.php

The form to add a post is made up of input's and textareas, each section has a name which will become a variable in php when the form is submitted.
The forms also use what's called sticky forms meaning if validation fails then show all content entered into the form.

```
<form action='' method='post'>

    <p><label>Title</label><br />
    <input type='text' name='postTitle' value='<?php if(isset($error))

    <p><label>Description</label><br />
    <textarea name='postDesc' cols='60' rows='10'><?php if(isset($erro

    <p><label>Content</label><br />
    <textarea name='postCont' cols='60' rows='10'><?php if(isset($erro

    <p><input type='submit' name='submit' value='Submit'></p>

</form>
```

For textarea's rather then making the admins enter the html for the text themselves its better to use an editor, I've chosen to use a popular one called TinyMCE To use it you would normally have to download the files from tinyMCE's website upload them and configure the config, thankfully they have recently released a CDN version so you can include tinyMCE by simply referencing the CDN and then your setup options:

This will convert all textarea's into editors.

```
<script src="//tinymce.cachefly.net/4.0/tinymce.min.js"></script>
<script>
        tinymce.init({
            selector: "textarea",
            plugins: [
                "advlist autolink lists link image charmap print previ
                "searchreplace visualblocks code fullscreen",
                "insertdatetime media table contextmenu paste"
            ],
            toolbar: "insertfile undo redo | styleselect | bold italic
        });
</script>
```

To process the form data once its been submitted is a simple process first make sure the form has been submitted. Then remove any slashed in the $_POST array. Then extract all posts items inside $_POST by using extract($_POST) any post element is then accessible by using just its name so $_POST['postTitle'] becomes $postTitle.

Next validate the data, these are very basic validation rules. These can be improved upon, if any of the if statements are true then an error is needed, adding an error to an array called error is a simple way to collect multiple errors.

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

    $_POST = array_map( 'stripslashes', $_POST );

    //collect form data
    extract($_POST);

    //very basic validation
    if($postTitle ==''){
        $error[] = 'Please enter the title.';
    }

    if($postDesc ==''){
        $error[] = 'Please enter the description.';
    }

    if($postCont ==''){
        $error[] = 'Please enter the content.';
    }
```

Next if no error has been set then insert the data into the database, this is using prepared statements the place holders :postTitle, :postDesc etc are using to bind the matching array elements when execute to add the data into the correct columns. Once inserted the user is redirected back to the admin a action status is appended to the url ?action=added.

```
if(!isset($error)){

    try {

        //insert into database
        $stmt = $db->prepare('INSERT INTO blog_posts (postTitle,postDe
        $stmt->execute(array(
            ':postTitle' => $postTitle,
            ':postDesc' => $postDesc,
            ':postCont' => $postCont,
            ':postDate' => date('Y-m-d H:i:s')
        ));

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

    } catch(PDOException $e) {
        echo $e->getMessage();
    }

}
```

If there has been any errors set then loop through the error array and display them.

```
if(isset($error)){
    foreach($error as $error){
        echo '<p class="error">'.$error.'</p>';
    }
}
```

admin/edit-post.php

This is very much like the add page except before the menu a query must be ran to select the correct post from the database to populate the form with.

The query needs to select the record where the postID matched the id passed in the $_GET['id'] request, as this can be manipulated a prepared statement is used.
The form also has an hidden field with the name postID and a value from the database this is used when updating the record to determine which post to make the changes to.

```
<?php
try {

    $stmt = $db->prepare('SELECT postID, postTitle, postDesc, postCont
    $stmt->execute(array(':postID' => $_GET['id']));
    $row = $stmt->fetch();

} catch(PDOException $e) {
    echo $e->getMessage();
}
?>

<form action='' method='post'>
    <input type='hidden' name='postID' value='<?php echo $row['postID'

    <p><label>Title</label><br />
    <input type='text' name='postTitle' value='<?php echo $row['postTi

    <p><label>Description</label><br />
    <textarea name='postDesc' cols='60' rows='10'><?php echo $row['pos

    <p><label>Content</label><br />
    <textarea name='postCont' cols='60' rows='10'><?php echo $row['pos

    <p><input type='submit' name='submit' value='Update'></p>

</form>
```

When the form is submitted the same checks are done that were done on the add post page:

```
if(isset($_POST['submit'])){

    $_POST = array_map( 'stripslashes', $_POST );

    //collect form data
    extract($_POST);

    //very basic validation
    if($postID ==''){
        $error[] = 'This post is missing a valid id!.';
    }

    if($postTitle ==''){
        $error[] = 'Please enter the title.';
    }

    if($postDesc ==''){
        $error[] = 'Please enter the description.';
    }

    if($postCont ==''){
        $error[] = 'Please enter the content.';
    }

    if(!isset($error)){
```

To update the post a prepared statement is ran this time an update command is used updating the specified columns by the matching place holders/ executed array. Once completed the user is redirected to index with an action status.

```
try {

    //insert into database
    $stmt = $db->prepare('UPDATE blog_posts SET postTitle = :postTitle
    $stmt->execute(array(
        ':postTitle' => $postTitle,
        ':postDesc' => $postDesc,
        ':postCont' => $postCont,
        ':postID' => $postID
    ));

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

} catch(PDOException $e) {
    echo $e->getMessage();
}
```

admin/users.php

This page lists all administrators of the blog.

While looping through the admins edit and delete links are created, the first admin will have an id of 1 this user should not be deleted, therefor an if statement is used to make sure the delete link is not shown for the first admin.

```
$stmt = $db->query('SELECT memberID, username, email FROM blog_members
while($row = $stmt->fetch()){

    echo '<tr>';
    echo '<td>'.$row['username'].'</td>';
    echo '<td>'.$row['email'].'</td>';
    ?>

    <td>
        <a href="edit-user.php?id=<?php echo $row['memberID'];?>">Edit
        <?php if($row['memberID'] != 1){?>
            | <a href="javascript:deluser('<?php echo $row['memberID']
        <?php } ?>
    </td>

    <?php
    echo '</tr>';

}
```

Deleting a user is the same as a post:

```
<script language="JavaScript" type="text/javascript">
function deluser(id, title)
{
  if (confirm("Are you sure you want to delete '" + title + "'"))
  {
      window.location.href = 'users.php?deluser=' + id;
  }
}
</script>
```

```
if(isset($_GET['deluser'])){

    //if user id is 1 ignore
    if($_GET['deluser'] !='1'){

        $stmt = $db->prepare('DELETE FROM blog_members WHERE memberID
        $stmt->execute(array(':memberID' => $_GET['deluser']));

        header('Location: users.php?action=deleted');
        exit;

    }
}
```

admin/add-user.php

Adding and editing users is very similar to posts I will go over only what's different.

Present the form to be filled in, notice the password have a type of password this stops the password being show in the form.

```
<form action='' method='post'>

    <p><label>Username</label><br />
    <input type='text' name='username' value='<?php if(isset($error)){

    <p><label>Password</label><br />
    <input type='password' name='password' value='<?php if(isset($erro

    <p><label>Confirm Password</label><br />
    <input type='password' name='passwordConfirm' value='<?php if(isse

    <p><label>Email</label><br />
    <input type='text' name='email' value='<?php if(isset($error)){ ec

    <p><input type='submit' name='submit' value='Add User'></p>

</form>
```

As part of processing the form new users will need a hash to be created from the user class this is done by passing the password from the form into the class object and use the create_hash method.
Next a normal insert statement is ran in the array the password is not used but instead the hashedpassword is given.

```
$hashedpassword = $user->create_hash($password);

try {

    //insert into database
    $stmt = $db->prepare('INSERT INTO blog_members (username,password,
    $stmt->execute(array(
        ':username' => $username,
        ':password' => $hashedpassword,
        ':email' => $email
    ));

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

} catch(PDOException $e) {
    echo $e->getMessage();
}
```

admin/edit-user.php

To edit a user first the user needs to be retrieved from the database using a prepared statement where the memberID matches the id passed in the get request.

The password fields are not populated, this are only filled in to change the password.

```
<?php
 try {

        $stmt = $db->prepare('SELECT memberID, username, email FROM bl
        $stmt->execute(array(':memberID' => $_GET['id']));
        $row = $stmt->fetch();

    } catch(PDOException $e) {
        echo $e->getMessage();
    }
?>

<form action='' method='post'>
    <input type='hidden' name='memberID' value='<?php echo $row['membe

    <p><label>Username</label><br />
    <input type='text' name='username' value='<?php echo $row['usernam

    <p><label>Password (only to change)</label><br />
    <input type='password' name='password' value=''></p>

    <p><label>Confirm Password</label><br />
    <input type='password' name='passwordConfirm' value=''></p>

    <p><label>Email</label><br />
    <input type='text' name='email' value='<?php echo $row['email'];?>

    <p><input type='submit' name='submit' value='Update User'></p>

</form>
```

When running validation the password checks should only run if a password has been entered.

```
if( strlen($password) > 0){

    if($password ==''){
        $error[] = 'Please enter the password.';
    }

    if($passwordConfirm ==''){
        $error[] = 'Please confirm the password.';
    }

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

}
```

When updating the database a check is made to see if the password has been set, if so then the password is updated otherwise another update is ran without updating the password.
When the password is to be updated a new hash is created from the user object.

```
if(isset($password)){

    $hashedpassword = $user->create_hash($password);

    //update into database
    $stmt = $db->prepare('UPDATE blog_members SET username = :username
    $stmt->execute(array(
        ':username' => $username,
        ':password' => $hashedpassword,
        ':email' => $email,
        ':memberID' => $memberID
    ));


} else {

    //update database
    $stmt = $db->prepare('UPDATE blog_members SET username = :username
    $stmt->execute(array(
        ':username' => $username,
        ':email' => $email,
        ':memberID' => $memberID
    ));

}
```

That's all the notable differences the files in full are available in the download I'll also list the full files below:

Source Files

index.php

```
<?php require('includes/config.php'); ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Blog</title>
    <link rel="stylesheet" href="style/normalize.css">
    <link rel="stylesheet" href="style/main.css">
</head>
<body>

    <div id="wrapper">

        <h1>Blog</h1>
        <hr />

        <?php
            try {

                $stmt = $db->query('SELECT postID, postTitle, postDesc
                while($row = $stmt->fetch()){

                    echo '<div>';
                        echo '<h1><a href="viewpost.php?id='.$row['pos
                        echo '<p>Posted on '.date('jS M Y H:i:s', strt
                        echo '<p>'.$row['postDesc'].'</p>';
                        echo '<p><a href="viewpost.php?id='.$row['post
                    echo '</div>';

                }

            } catch(PDOException $e) {
                echo $e->getMessage();
            }
        ?>

    </div>
```

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

viewpost.php

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

$stmt = $db->prepare('SELECT postID, postTitle, postCont, postDate FRO
$stmt->execute(array(':postID' => $_GET['id']));
$row = $stmt->fetch();

//if post does not exists redirect user.
if($row['postID'] == ''){
    header('Location: ./');
    exit;
}

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Blog - <?php echo $row['postTitle'];?></title>
    <link rel="stylesheet" href="style/normalize.css">
    <link rel="stylesheet" href="style/main.css">
</head>
<body>

    <div id="wrapper">

        <h1>Blog</h1>
        <hr />
        <p><a href="./">Blog Index</a></p>


        <?php
            echo '<div>';
                echo '<h1>'.$row['postTitle'].'</h1>';
                echo '<p>Posted on '.date('jS M Y', strtotime($row['po
                echo '<p>'.$row['postCont'].'</p>';
            echo '</div>';
        ?>

    </div>
```

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

includes/config.php

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

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

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


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

//load classes as needed
function __autoload($class) {

   $class = strtolower($class);

    //if call from within /assets adjust the path
   $classpath = 'classes/class.'.$class . '.php';
   if ( file_exists($classpath)) {
      require_once $classpath;
    }

    //if call from within admin adjust the path
   $classpath = '../classes/class.'.$class . '.php';
   if ( file_exists($classpath)) {
      require_once $classpath;
    }

    //if call from within admin adjust the path
   $classpath = '../../classes/class.'.$class . '.php';
   if ( file_exists($classpath)) {
      require_once $classpath;
    }
```

```
}

$user = new User($db);
?>
```

classes/class.user.php

```
<?php

class User{

    private $db;

    public function __construct($db){
        $this->db = $db;
    }


    public function is_logged_in(){
        if(isset($_SESSION['loggedin']) && $_SESSION['loggedin'] == tr
            return true;
        }
    }

    public function create_hash($value)
    {
        return $hash = crypt($value, '$2a$12.substr(str_replace('+', '
    }

    private function verify_hash($password,$hash)
    {
        return $hash == crypt($password, $hash);
    }

    private function get_user_hash($username){

        try {

            //echo $this->create_hash('demo');

            $stmt = $this->db->prepare('SELECT password FROM blog_memb
            $stmt->execute(array('username' => $username));

            $row = $stmt->fetch();
            return $row['password'];

        } catch(PDOException $e) {
```

```
            echo '<p class="error">'.$e->getMessage().'</p>';
        }
    }


    public function login($username,$password){

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

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

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


    public function logout(){
        session_destroy();
    }

}

?>
```

admin/index.php

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

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

//show message from add / edit page
if(isset($_GET['delpost'])){

    $stmt = $db->prepare('DELETE FROM blog_posts WHERE postID = :postI
    $stmt->execute(array(':postID' => $_GET['delpost']));

    header('Location: index.php?action=deleted');
    exit;
}

?>
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Admin</title>
  <link rel="stylesheet" href="../style/normalize.css">
  <link rel="stylesheet" href="../style/main.css">
  <script language="JavaScript" type="text/javascript">
  function delpost(id, title)
  {
      if (confirm("Are you sure you want to delete '" + title + "'"))
      {
          window.location.href = 'index.php?delpost=' + id;
      }
  }
  </script>
</head>
<body>

    <div id="wrapper">

    <?php include('menu.php');?>
```

```
<?php
//show message from add / edit page
if(isset($_GET['action'])){
    echo '<h3>Post '.$_GET['action'].'.</h3>';
}
?>

<table>
<tr>
    <th>Title</th>
    <th>Date</th>
    <th>Action</th>
</tr>
<?php
    try {

        $stmt = $db->query('SELECT postID, postTitle, postDate FRO
        while($row = $stmt->fetch()){

            echo '<tr>';
            echo '<td>'.$row['postTitle'].'</td>';
            echo '<td>'.date('jS M Y', strtotime($row['postDate'])
            ?>

            <td>
                <a href="edit-post.php?id=<?php echo $row['postID'
                <a href="javascript:delpost('<?php echo $row['post
            </td>

            <?php
            echo '</tr>';

        }

    } catch(PDOException $e) {
        echo $e->getMessage();
    }
?>
</table>
```

```
    <p><a href='add-post.php'>Add Post</a></p>

</div>

</body>
</html>
```

admin/login.php

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


//check if already logged in
if( $user->is_logged_in() ){ header('Location: index.php'); }
?>
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Admin Login</title>
  <link rel="stylesheet" href="../style/normalize.css">
  <link rel="stylesheet" href="../style/main.css">
</head>
<body>

<div id="login">

    <?php

    //process login form if submitted
    if(isset($_POST['submit'])){

        $username = trim($_POST['username']);
        $password = trim($_POST['password']);

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

            //logged in return to index page
            header('Location: index.php');
            exit;


        } else {
            $message = '<p class="error">Wrong username or password</p
        }

    }//end if submit
```

```
    if(isset($message)){ echo $message; }
    ?>

    <form action="" method="post">
    <p><label>Username</label><input type="text" name="username" value
    <p><label>Password</label><input type="password" name="password" v
    <p><label></label><input type="submit" name="submit" value="Login"
    </form>

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

admin/logout.php

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

//log user out
$user->logout();
header('Location: index.php');

?>
```

admin/menu.php

```
<h1>Blog</h1>
<ul id='adminmenu'>
    <li><a href='index.php'>Blog</a></li>
    <li><a href='users.php'>Users</a></li>
    <li><a href="../" target="_blank">View Website</a></li>
    <li><a href='logout.php'>Logout</a></li>
</ul>
<div class='clear'></div>
<hr />
```

admin/add-post.php

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

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: login.php'); }
?>
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Admin - Add Post</title>
  <link rel="stylesheet" href="../style/normalize.css">
  <link rel="stylesheet" href="../style/main.css">
  <script src="//tinymce.cachefly.net/4.0/tinymce.min.js"></script>
  <script>
          tinymce.init({
              selector: "textarea",
              plugins: [
                  "advlist autolink lists link image charmap print pre
                  "searchreplace visualblocks code fullscreen",
                  "insertdatetime media table contextmenu paste"
              ],
              toolbar: "insertfile undo redo | styleselect | bold ital
          });
  </script>
</head>
<body>

<div id="wrapper">

    <?php include('menu.php');?>
    <p><a href="./">Blog Admin Index</a></p>

    <h2>Add Post</h2>

    <?php

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

```
$_POST = array_map( 'stripslashes', $_POST );

//collect form data
extract($_POST);

//very basic validation
if($postTitle ==''){
    $error[] = 'Please enter the title.';
}

if($postDesc ==''){
    $error[] = 'Please enter the description.';
}

if($postCont ==''){
    $error[] = 'Please enter the content.';
}

if(!isset($error)){

    try {

        //insert into database
        $stmt = $db->prepare('INSERT INTO blog_posts (postTitl
        $stmt->execute(array(
            ':postTitle' => $postTitle,
            ':postDesc' => $postDesc,
            ':postCont' => $postCont,
            ':postDate' => date('Y-m-d H:i:s')
        ));

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

    } catch(PDOException $e) {
        echo $e->getMessage();
    }

}
```

```
    }

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

    <form action='' method='post'>

        <p><label>Title</label><br />
        <input type='text' name='postTitle' value='<?php if(isset($err

        <p><label>Description</label><br />
        <textarea name='postDesc' cols='60' rows='10'><?php if(isset($

        <p><label>Content</label><br />
        <textarea name='postCont' cols='60' rows='10'><?php if(isset($

        <p><input type='submit' name='submit' value='Submit'></p>

    </form>

</div>
```

admin/edit-post.php

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

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: login.php'); }
?>
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Admin - Edit Post</title>
  <link rel="stylesheet" href="../style/normalize.css">
  <link rel="stylesheet" href="../style/main.css">
  <script src="//tinymce.cachefly.net/4.0/tinymce.min.js"></script>
  <script>
          tinymce.init({
              selector: "textarea",
              plugins: [
                  "advlist autolink lists link image charmap print pre
                  "searchreplace visualblocks code fullscreen",
                  "insertdatetime media table contextmenu paste"
              ],
              toolbar: "insertfile undo redo | styleselect | bold ital
          });
  </script>
</head>
<body>

<div id="wrapper">

    <?php include('menu.php');?>
    <p><a href="./">Blog Admin Index</a></p>

    <h2>Edit Post</h2>


    <?php

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

```
$_POST = array_map( 'stripslashes', $_POST );

//collect form data
extract($_POST);

//very basic validation
if($postID ==''){
    $error[] = 'This post is missing a valid id!.';
}

if($postTitle ==''){
    $error[] = 'Please enter the title.';
}

if($postDesc ==''){
    $error[] = 'Please enter the description.';
}

if($postCont ==''){
    $error[] = 'Please enter the content.';
}

if(!isset($error)){

    try {

        //insert into database
        $stmt = $db->prepare('UPDATE blog_posts SET postTitle
        $stmt->execute(array(
            ':postTitle' => $postTitle,
            ':postDesc' => $postDesc,
            ':postCont' => $postCont,
            ':postID' => $postID
        ));

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

```
        } catch(PDOException $e) {
            echo $e->getMessage();
        }

    }

}

?>


<?php
//check for any errors
if(isset($error)){
    foreach($error as $error){
        echo $error.'<br />';
    }
}

    try {

        $stmt = $db->prepare('SELECT postID, postTitle, postDesc,
        $stmt->execute(array(':postID' => $_GET['id']));
        $row = $stmt->fetch();

    } catch(PDOException $e) {
        echo $e->getMessage();
    }

?>

<form action='' method='post'>
    <input type='hidden' name='postID' value='<?php echo $row['pos

    <p><label>Title</label><br />
    <input type='text' name='postTitle' value='<?php echo $row['po

    <p><label>Description</label><br />
    <textarea name='postDesc' cols='60' rows='10'><?php echo $row[
```

```
        <p><label>Content</label><br />
        <textarea name='postCont' cols='60' rows='10'><?php echo $row[

        <p><input type='submit' name='submit' value='Update'></p>

    </form>

</div>

</body>
</html>
```

admin/users.php

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

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

//show message from add / edit page
if(isset($_GET['deluser'])){

    //if user id is 1 ignore
    if($_GET['deluser'] !='1'){

        $stmt = $db->prepare('DELETE FROM blog_members WHERE memberID
        $stmt->execute(array(':memberID' => $_GET['deluser']));

        header('Location: users.php?action=deleted');
        exit;

    }
}

?>
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Admin - Users</title>
  <link rel="stylesheet" href="../style/normalize.css">
  <link rel="stylesheet" href="../style/main.css">
  <script language="JavaScript" type="text/javascript">
  function deluser(id, title)
  {
      if (confirm("Are you sure you want to delete '" + title + "'"))
      {
          window.location.href = 'users.php?deluser=' + id;
      }
  }
  </script>
</head>
```

```
<body>

    <div id="wrapper">

    <?php include('menu.php');?>

    <?php
    //show message from add / edit page
    if(isset($_GET['action'])){
        echo '<h3>User '.$_GET['action'].'.</h3>';
    }
    ?>

    <table>
    <tr>
        <th>Username</th>
        <th>Email</th>
        <th>Action</th>
    </tr>
    <?php
        try {

            $stmt = $db->query('SELECT memberID, username, email FROM
            while($row = $stmt->fetch()){

                echo '<tr>';
                echo '<td>'.$row['username'].'</td>';
                echo '<td>'.$row['email'].'</td>';
                ?>

                <td>
                    <a href="edit-user.php?id=<?php echo $row['memberI
                    <?php if($row['memberID'] != 1){?>
                        | <a href="javascript:deluser('<?php echo $row
                    <?php } ?>
                </td>

                <?php
                echo '</tr>';
```

```
            }

        } catch(PDOException $e) {
            echo $e->getMessage();
        }
    ?>
    </table>

    <p><a href='add-user.php'>Add User</a></p>

</div>

</body>
</html>
```

admin/add-user.php

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

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: login.php'); }
?>
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Admin - Add User</title>
  <link rel="stylesheet" href="../style/normalize.css">
  <link rel="stylesheet" href="../style/main.css">
</head>
<body>

<div id="wrapper">

    <?php include('menu.php');?>
    <p><a href="users.php">User Admin Index</a></p>

    <h2>Add User</h2>

    <?php

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

        //collect form data
        extract($_POST);

        //very basic validation
        if($username ==''){
            $error[] = 'Please enter the username.';
        }

        if($password ==''){
            $error[] = 'Please enter the password.';
        }
```

```
        if($passwordConfirm ==''){
            $error[] = 'Please confirm the password.';
        }

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

        if($email ==''){
            $error[] = 'Please enter the email address.';
        }

        if(!isset($error)){

            $hashedpassword = $user->create_hash($password);

            try {

                //insert into database
                $stmt = $db->prepare('INSERT INTO blog_members (userna
                $stmt->execute(array(
                    ':username' => $username,
                    ':password' => $hashedpassword,
                    ':email' => $email
                ));

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

            } catch(PDOException $e) {
                echo $e->getMessage();
            }

        }

    }

    //check for any errors
    if(isset($error)){
```

```
        foreach($error as $error){
            echo '<p class="error">'.$error.'</p>';
        }
    }
    ?>

    <form action='' method='post'>

        <p><label>Username</label><br />
        <input type='text' name='username' value='<?php if(isset($erro

        <p><label>Password</label><br />
        <input type='password' name='password' value='<?php if(isset($

        <p><label>Confirm Password</label><br />
        <input type='password' name='passwordConfirm' value='<?php if(

        <p><label>Email</label><br />
        <input type='text' name='email' value='<?php if(isset($error))

        <p><input type='submit' name='submit' value='Add User'></p>

    </form>

</div>
```

admin/edit-user.php

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

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: login.php'); }
?>
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Admin - Edit User</title>
  <link rel="stylesheet" href="../style/normalize.css">
  <link rel="stylesheet" href="../style/main.css">
</head>
<body>

<div id="wrapper">

    <?php include('menu.php');?>
    <p><a href="users.php">User Admin Index</a></p>

    <h2>Edit User</h2>


    <?php

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

        //collect form data
        extract($_POST);

        //very basic validation
        if($username ==''){
            $error[] = 'Please enter the username.';
        }

        if( strlen($password) > 0){

            if($password ==''){
```

```
            $error[] = 'Please enter the password.';
        }

        if($passwordConfirm ==''){
            $error[] = 'Please confirm the password.';
        }

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

    }


    if($email ==''){
        $error[] = 'Please enter the email address.';
    }

    if(!isset($error)){

        try {

            if(isset($password)){

                $hashedpassword = $user->create_hash($password);

                //update into database
                $stmt = $db->prepare('UPDATE blog_members SET user
                $stmt->execute(array(
                    ':username' => $username,
                    ':password' => $hashedpassword,
                    ':email' => $email,
                    ':memberID' => $memberID
                ));


            } else {

                //update database
                $stmt = $db->prepare('UPDATE blog_members SET user
```

```
                $stmt->execute(array(
                    ':username' => $username,
                    ':email' => $email,
                    ':memberID' => $memberID
                ));

            }


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

        } catch(PDOException $e) {
            echo $e->getMessage();
        }

    }

}

?>


<?php
//check for any errors
if(isset($error)){
    foreach($error as $error){
        echo $error.'<br />';
    }
}

    try {

        $stmt = $db->prepare('SELECT memberID, username, email FRO
        $stmt->execute(array(':memberID' => $_GET['id']));
        $row = $stmt->fetch();

    } catch(PDOException $e) {
        echo $e->getMessage();
```

```
        }

    ?>

    <form action='' method='post'>
        <input type='hidden' name='memberID' value='<?php echo $row['m

        <p><label>Username</label><br />
        <input type='text' name='username' value='<?php echo $row['use

        <p><label>Password (only to change)</label><br />
        <input type='password' name='password' value=''></p>

        <p><label>Confirm Password</label><br />
        <input type='password' name='passwordConfirm' value=''></p>

        <p><label>Email</label><br />
        <input type='text' name='email' value='<?php echo $row['email'

        <p><input type='submit' name='submit' value='Update User'></p>

    </form>

</div>

</body>
</html>
```

By David Carr (https://daveismyname.blog/author/david-carr) On 7th Jun 2013 08:25 PM

Want to chat about any of my tutorials or other like minded developers?

Join my Slack channel (https://join.slack.com/t/daveismynameblog/shared_in

(https://daveismyname.blog/author/david-carr)

David Carr (https://daveismyname.blog/author/david-carr)

For the past 10 years, I've been developing applications for the web using mostly PHP. I do this for a living and love what I do as every day there is something new and exciting to learn.

In my spare time, the web development community is a big part of my life. Whether managing online programming groups and blogs or attending a conference, I find keeping involved helps me stay up to date. This is also my chance to give back to the community that helped me get started, a place I am proud to be part of.

Besides programming I love spending time with friends and family and can often be found together going out catching the latest movie, staying in playing games on the sofa or planning a trip to someplace I've never been before.

 (https://twitter.com/daveismynamecom) (https://www.facebook.com/davidcarrblog)
 (https://www.youtube.com/c/Daveismynamecom)
 (https://www.instagram.com/daveismynamecom/)
 (https://plus.google.com/b/113363703862364375355/+Daveismynamecom)

 Previous Post

Detecting and replacing bad words in PHP

(https://daveismyname.blog/detecting-and-replacing-bad-words-in-php)

Next Post

Creating a blog from scratch with PHP - Part 2 SEO URLS

(https://daveismyname.blog/creating-a-blog-from-scratch-with-php-part-2-seo-urls)

327 Comments Dave is my name Primus

Recommend 22 Share Sort by Best

Join the discussion…

Emmeline Ge • 2 years ago

How to remove the administrator privilege for all users? I want only the user who login at that time can edit and add their own post

7 • Reply • Share ›

David Carr Mod ➚ Emmeline Ge • 2 years ago

you could check the user id of the loggin in user and only allow that user id access to add/edit better yet have a level column in the database table set it to int and default to 2 and set the admin user to 1 then when logging in store the level in a session then you can give/deny actions based on the users level.

• Reply • Share ›

deepz513 . • 4 years ago

hi, thank you so much for the tutorial ..even for a beginner like its a great help!..I had a small issue though,,while I try to edit the post in the Content using the the editor, It doesnt display the image . I made the same project using MVC architecture. any suggestion? btw I insert image by using the editor's Insert Image. Thanks!

7 • Reply • Share ›

KARA • 4 years ago

Thank you so much! I've been looking for this everywhere. I'm 15 and looking to develop my own blog from scratch, hopefully this will help me tons.

Thanks again ! :3

12 • Reply • Share ›

serverbattlecraft • 4 years ago

THIS SEEMS TO BE THE PERPECT THING FOR ME! LOVE IT LOVE IT LOVE IT. THANK YOU SOOOOO MUCH!

3 • Reply • Share ›

DanCurtisMusic • 4 years ago

Thank you for this incredible tutorial. I have learned a lot! I'm having problems with the login on my website. I didn't get how to put in the first username and password so I just put it into mysql directly. But when I log in to admin, it says I have the wrong username or password. Can you tell me what I might be doing wrong?

2 ^ | v • Reply • Share ›

David Carr Mod ➚ DanCurtisMusic • 4 years ago

The password needs to be hashed using the function from the user class or it won't be recognised.

to change the password the hashed version will need adding to the database,

in classes/class.user.php edit the construct function and add echo $this->create_hash('demo');

like this:

```
public function __construct($db){
$this->db = $db;
echo $this->create_hash('demo');
}
```

then go to the blog you should see a long string at the top of the page, copy that and save it in the password field in the database.

Once you have done that you should be able to login, then remove echo $this->create_hash('demo'); from the class.

To change the password edit the demo user from the admin panel.

^ | v • Reply • Share ›

Guest ➚ David Carr • 3 years ago

I have an error after doing this, what could be the problem:

Fatal error: Cannot access private property User::$db in classes/class.password.php on line 10

^ | v • Reply • Share ›

Nicolae Iosif ➚ David Carr • 4 years ago

I can't login,where to copy there row?

^ | v • Reply • Share ›

Nikola Petrovic • 3 years ago

I have problem with admin login...

I created member in sql and i can't login... why?

3 ^ | v • Reply • Share ›

Mohamed ➚ Nikola Petrovic • 3 years ago

Don't save the password as plaintext in the db.

echo password_hash('nikola123', PASSWORD_DEFAULT);
copy the output,
insert it in the db (UPDATE blog_members SET password = 'password_hash_here');
and then try logging in from the website.

NOTE:
To generate hash, use password_hash($password, PASSWORD_DEFAULT);
To verify password, use password_verify($entered_password, $saved_hash);

1 ^ | v • Reply • Share ›

Lizette Chokmah ➚ Nikola Petrovic • 3 years ago

i have the same problem

^ | v • Reply • Share ›

christopher • a year ago

Fatal error: Uncaught exception 'PDOException' with message 'could not find driver' in C:\wamp64\www\blog\includes\config.php on line 11(!) PDOException: could not find driver in C:\wamp64\www\blog\includes\config.php on line 11

2 ^ | v • Reply • Share ›

David Carr Mod ➚ christopher • 9 months ago

that means wamp does not have the pdo driver installed see https://stackoverflow.com/q...

^ | v • Reply • Share ›

Dimosthenis Pagakis ➚ David Carr • 8 months ago

I've been getting the same error. I've spent my last two days trying

I've been getting the same error. I've spent my last two days trying to solve the problem while looking at the link you sent but finding several others as well. Nothing seems to do the trick. Any additional advice?

Reply • Share ›

Jatinder Kumar • 2 months ago

sir i make a new data base but i face problem wrong password

1 Reply • Share ›

Duke André Miller • 9 months ago

Hi, how can I echo the username of the current user that is logged into the page?

1 Reply • Share ›

David Carr Mod → Duke André Miller • 7 months ago

I've updated the tutorial and github sources to add the memberID and username to a session so you can get either by using $_SESSION['username'] or $_SESSION['memberID']

1 Reply • Share ›

Jeffrey • a year ago

Hi, how can I echo the username of the current user that is logged into the page? So that the user see this own name?

Thank you,
- Jeffrey

1 Reply • Share ›

Elite Game Group • a year ago

Not able to log in via admin/login.php password and username incorrect? How do I add an user? As the page login.php will always showed first?

1 Reply • Share ›

Elite Game Group → Elite Game Group • a year ago

Fixed.

Reply • Share ›

J. → Elite Game Group • a year ago

How?

Reply • Share ›

Elite Game Group → J. • a year ago

Well it has been a long time ago for me, but I thought I did change the condition where if u not logged in the page will redirect. If you remove that, you will be able to go to the admin zone where u can add users. Tell me if that works

1 • Reply • Share ›

Lens Alexis ➚ Elite Game Group • 9 months ago

Hey, can you elaborate please? I am having the same difficulties

• Reply • Share ›

Md Imran Hossain ➚ Lens Alexis • 7 months ago

I am also

• Reply • Share ›

richard marin • 2 years ago

Hello Dave,

This series of tutorial is amazing, not to mention your good will of sharing all of the codes. Thank you very much! You alone is helping me and my work to be great and fast. I really appreciate you, your blog and your tutorials.

Greetings from Brazil.

1 • Reply • Share ›

David Carr Mod ➚ richard marin • 2 years ago

Thanks!

• Reply • Share ›

Robert Weatherall • 2 years ago

This is great Dave Thank you!
I am also having an issue with admin login.
demo demo says wrong username or password. (I did not change them)
So I removed the login from admin, added a new user (admin) and set password. (admin123)
The new user was added but had password field blank in db.
I think that means it is not finding the "classes.php" ?
I have classes folder in the root, should it be elsewhere?

1 • Reply • Share ›

Robert Weatherall ➚ Robert Weatherall • 2 years ago

I change config file of classes path and it returned error of not finding user, so the config is finding the classes.php but I still can not login with the demo and when I add a user the password field is always blank in db?
I just imported the sql from downloaded files, was this the right way?

1 • Reply • Share ›

Robert Weatherall ➚ Robert Weatherall • 2 years ago

Also, I edited Demo account's password, then checked db and the field was empty.

I must have some error with the hash or something?

Reply • Share ›

Robert Weatherall ➚ Robert Weatherall • 2 years ago

I deleted everything including db and restarted. same issue, tried both 5 and 5.5 database, same while adding a user I switch the insert $email and $hashedpassword

then email was blank, so it has to do with $hashedpassword?

Reply • Share ›

Robert Weatherall ➚ Robert Weatherall • 2 years ago

NEEDS PHP 5.3 OR HIGHER

Reply • Share ›

Rafael Rodriguez • 2 years ago

Great post!! Im wondering... what do you think about to compare a "from scratch blog" vs WordPress blog? What advantages could I get working from scratch with php and mysql? Perhaps many people are using CMS but I prefer to try a harder way because 'control'. Some bloggers said to me that I am crazy because I dont want to use WordPress and its free resources.... am I wrong? I think CMS is not the better option in speed term. WP is more heavy than this "from scratch" blog, isnt it?

1 Reply • Share ›

David Carr Mod ➚ Rafael Rodriguez • 2 years ago

a build yourself blog is never going to complete with wordpress in terms of features, WP is huge having said that I really don't like the admin interface and prefer to build my own system. Building your own is a lot more work but you end up with a system that does exactly what you need without the bloat.

also if you need to add a new feature you build it, if something goes wrong you will know where to go to fix it or find more details. The downside is you will miss out on the huge WP resources but generally php resources can still apply.

With composer you can easily add third part libraries to your code base, this blog is built using my own custom cms made from my own framework http://simpemvcframework.com.

You get a much greater understanding of applications when you build them yourself.

1 Reply • Share ›

Rafael Rodriguez ➚ David Carr • 2 years ago

Thanks a lot. I agree with you. And what do you think about speed? If the blog gets more and more and more traffic... WP may be a little heavy solution?

Reply • Share ›

David Carr Mod → Rafael Rodriguez • 2 years ago

when you get lots of traffic, its more down to your hosting then the site itself, using caching helps and optimize the output and database queries also helps.

Once a site is getting lots of traffic a VPS or dedicated server is a good call, they have more resources available then the typical shared server does.

Reply • Share ›

Apak™ • 2 years ago

Hi, I've a problem : Fatal error: in C:\xampp\htdocs\mpress\includes\config.php on line 11

1 Reply • Share ›

ahmad → Apak™ • 2 years ago

remove port=8889 from the line

1 Reply • Share ›

Apak™ → ahmad • 2 years ago

Thx

Reply • Share ›

Oluflourish Emmanuel • 2 years ago

I really like this! But I have a question. How can I add the date the post was published to the slug so that I will have something in this format www.example.com/2015/10/25/.... I tried including $text = date("Y/m/d").'/'.$text; to the function slug($text) but didn't seem to work well. It worked well when I used date("Y-m-d") but I want to use "/" not "-". Please how I can I do it? Thanks in advance

1 Reply • Share ›

Aditya Tripathi • 3 years ago

OMG THIS IS BEST BLOG TUTORIAL EVER!!

1 Reply • Share ›

lonesomedreams • 4 years ago

I just wanna say that this is perfect, just the way I want it to be. Thank you! Ahhhhh, I wish I have time to study PDO. I'm having troubles in using your codes into mine. I prefer the procedural way given my current knowledge. Anyway,

thank you very much good Sir.

1 ^ | v • Reply • Share ›

Bea Getalado • 7 days ago

Hi Dave! I tried doing the log in to access the admin and I get this error:

Warning: A non-numeric value encountered in C:\xampp\htdocs\davidcarr_tut\classes\class.user.php on line 22

Warning: crypt() expects at most 2 parameters, 4 given in C:\xampp\htdocs\davidcarr_tut\classes\class.user.php on line 22

How do I fix it? Thank you :)

^ | v • Reply • Share ›

Elizabeth Herbert • a month ago

I had a little bit of trouble following at first, but that's because I am very much a newbie. Great tutorial and exactly what I was looking for!

^ | v • Reply • Share ›

Amit Kumar • a month ago

I have problem with admin login...

^ | v • Reply • Share ›

Sugih Liawan • 3 months ago

great tutorial, and great job at Nova Framework dave

^ | v • Reply • Share ›

Blair • 3 months ago

test

^ | v • Reply • Share ›

Craig Lewis • 4 months ago

Has anybody else had this script hacked ??

^ | v • Reply • Share ›

Landry Demingnou • 4 months ago

Well done david, but I've met some error on addind new post

Notice: Undefined variable: checked in C:\wamp\www\categorie\admin\add-post.php on line 137 but the code still working

see more

Reply • Share ›

frenchbeans • 5 months ago

The admin demo doesn't seem to be working given the credentials in this tutorial.

Reply • Share ›

Load more comments

ALSO ON **DAVE IS MY NAME**

My top VS Code Extensions (Plugins) - David Carr | Web Developer Blog

1 comment • a month ago

Isaac Kayode Daramola — Thank you Dave. I love this. I've been wondering how to add some stuffs. You just made ...

Get a free SSL certificate using Cloudflare - David Carr | Web …

1 comment • a year ago

Bhavik Solanki — Thank you for sharing this great information. This information helpful and useful for many people ...

Run Python 3 as default when Python 2 is installed on a Mac

1 comment • a year ago

John wright — Thank you for sharing this great information. In this post gave good explanation that helpful and ...

HTML 5 Submit form - multiple actions with formaction

2 comments • a year ago

David Carr — if your post to another page but want go to a different url the if else would have to be on the page

Categories

Design (https://daveismyname.blog/categories/design)
Development (https://daveismyname.blog/categories/development)

Networking (https://daveismyname.blog/categories/networking)
Personal (https://daveismyname.blog/categories/personal)
Reviews (https://daveismyname.blog/categories/reviews)
SEO (https://daveismyname.blog/categories/seo)
Social Media (https://daveismyname.blog/categories/social-media)
Tools (https://daveismyname.blog/categories/tools)
Tutorials (https://daveismyname.blog/categories/tutorials)
- CSS (https://daveismyname.blog/categories/css)
- Flash (https://daveismyname.blog/categories/flash)
- Htaccess (https://daveismyname.blog/categories/htaccess)
- HTML (https://daveismyname.blog/categories/html)
- Javascript (https://daveismyname.blog/categories/javascript)
- PHP & MySQL (https://daveismyname.blog/categories/php-mysql)
  - Nova Framework (https://daveismyname.blog/categories/nova-framework)
  - IMAP (https://daveismyname.blog/categories/imap)
  - CMS (https://daveismyname.blog/categories/cms)
  - Blog (https://daveismyname.blog/categories/blog)
  - Laravel Framework (https://daveismyname.blog/categories/laravel-framework)
  - API (https://daveismyname.blog/categories/api)
  - Stripe API (https://daveismyname.blog/categories/stripe-api)
- Wordpress (https://daveismyname.blog/categories/wordpress)
- Demos (https://daveismyname.blog/categories/demos)
- Bash (https://daveismyname.blog/categories/bash)
- Python (https://daveismyname.blog/categories/python)

Wordpress Plugins (https://daveismyname.blog/categories/wordpress-plugins)

Authors

David Carr (https://daveismyname.blog/author/david-carr)
Dan Sherwood (https://daveismyname.blog/author/dan-sherwood)

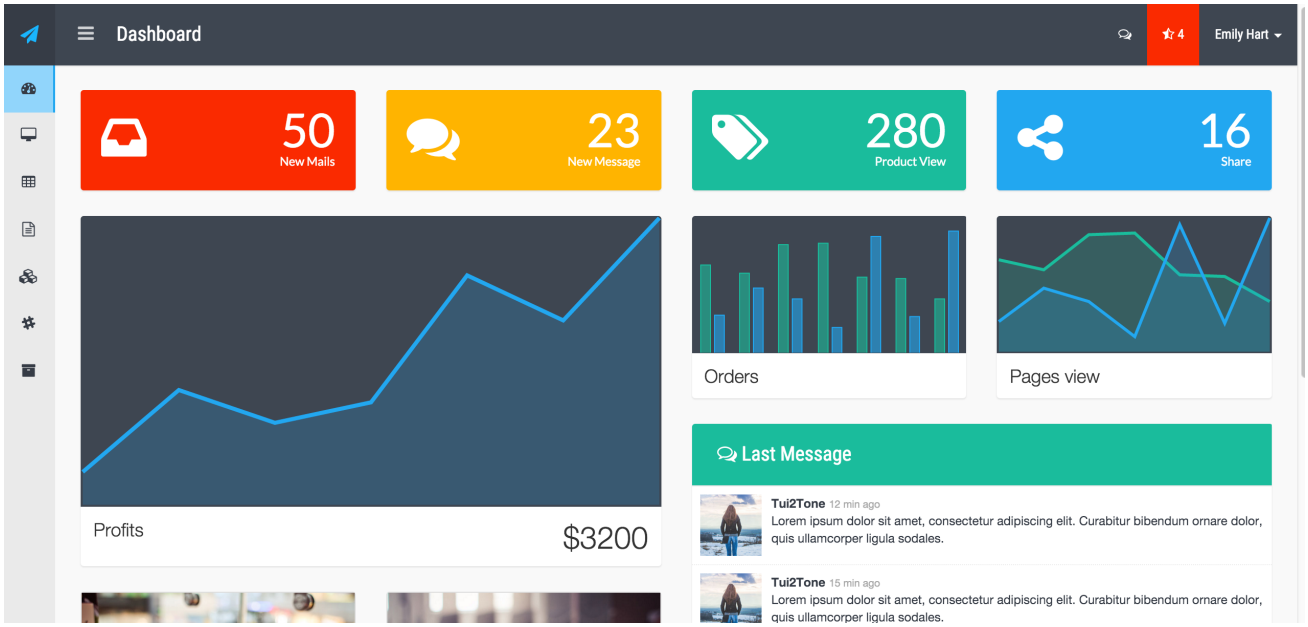
SUBSCRIBE BY EMAIL

Your email address

SUBSCRIBE NOW

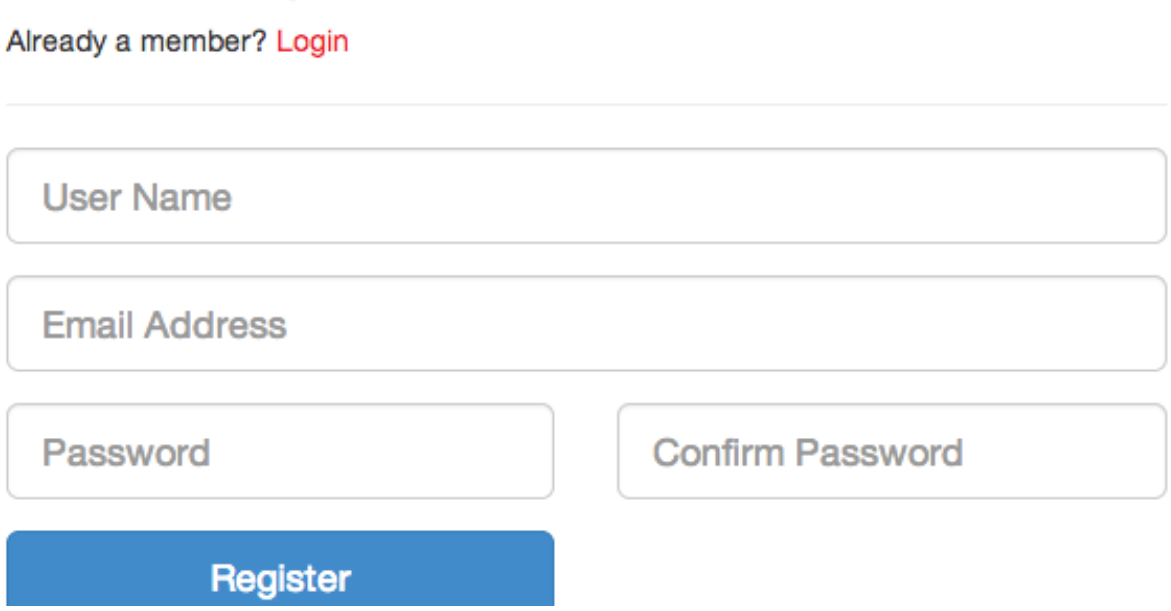
POPULAR POSTS

Free Bootstrap Admin Themes (https://daveismyname.blog/free-bootstrap-admin-themes)



14th Nov 2015

Login and Registration system with PHP (https://daveismyname.blog/login-and-registration-system-with-php)



1st Feb 2014

Creating a blog from scratch with PHP (https://daveismyname.blog/creating-a-blog-from-scratch-with-php)

7th Jun 2013

Autocomplete with PHP, MySQL and Jquery UI (https://daveismyname.blog/autocomplete-with-php-mysql-and-jquery-ui)

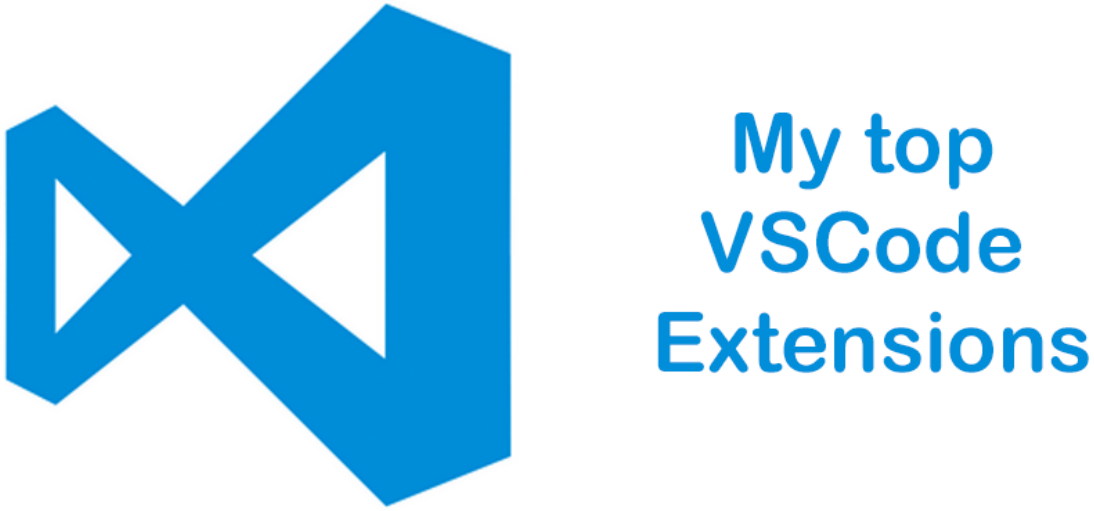
27th Feb 2013

Building a content management system from scratch (https://daveismyname.blog/building-a-content-management-system-from-scratch)

6th May 2011

Latest Posts

My top VS Code Extensions (Plugins) (https://daveismyname.blog/my-top-vs-code-extensions-plugins)



14th Feb 2018

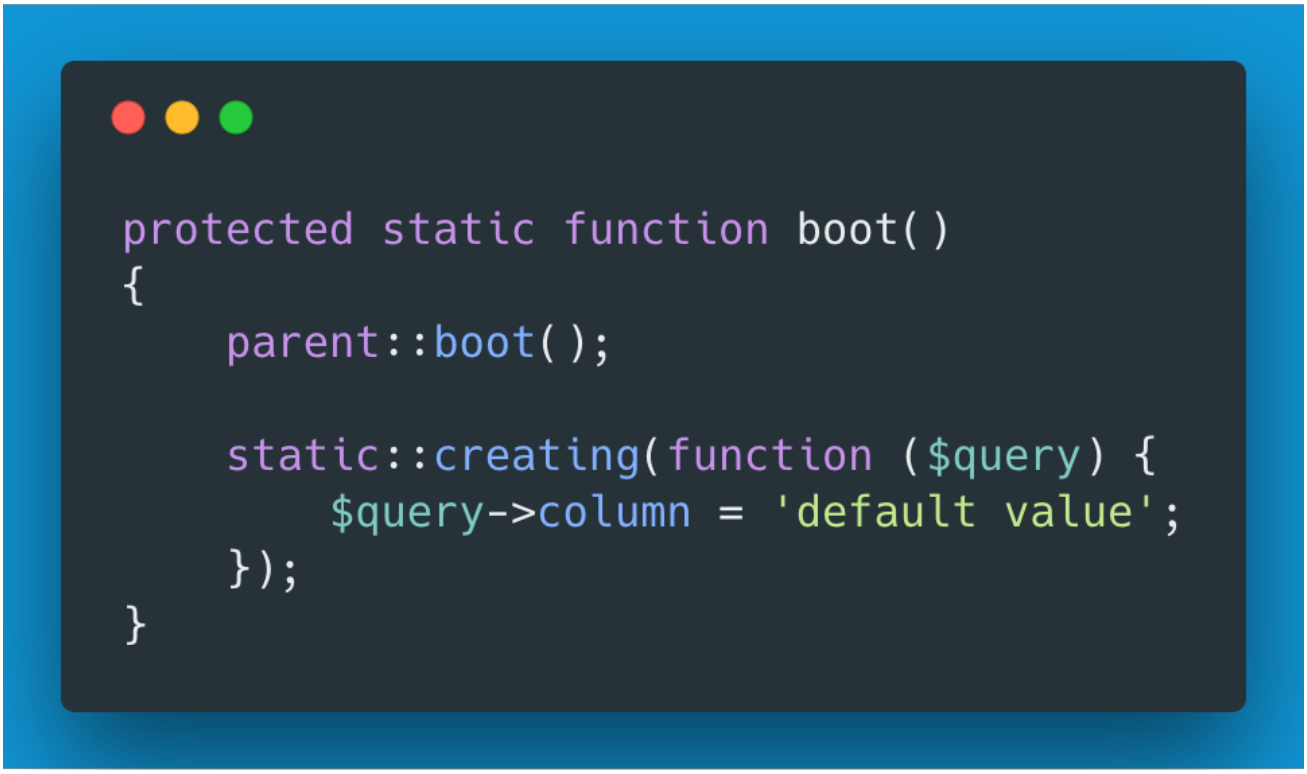
Laravel apply a query constraint with global scopes (https://daveismyname.blog/laravel-apply-a-query-constraint-with-global-scopes)

```php
protected static function boot()
{
    parent::boot();

    //apply condition to all queries
    static::addGlobalScope('tenent', function (Builder $builder) {
        $builder->where('tenant_id', session('tenant_id'));
    });
}
```

9th Feb 2018

Laravel set default value for creating model instances (https://daveismyname.blog/laravel-set-default-value-for-creating-model-instances)

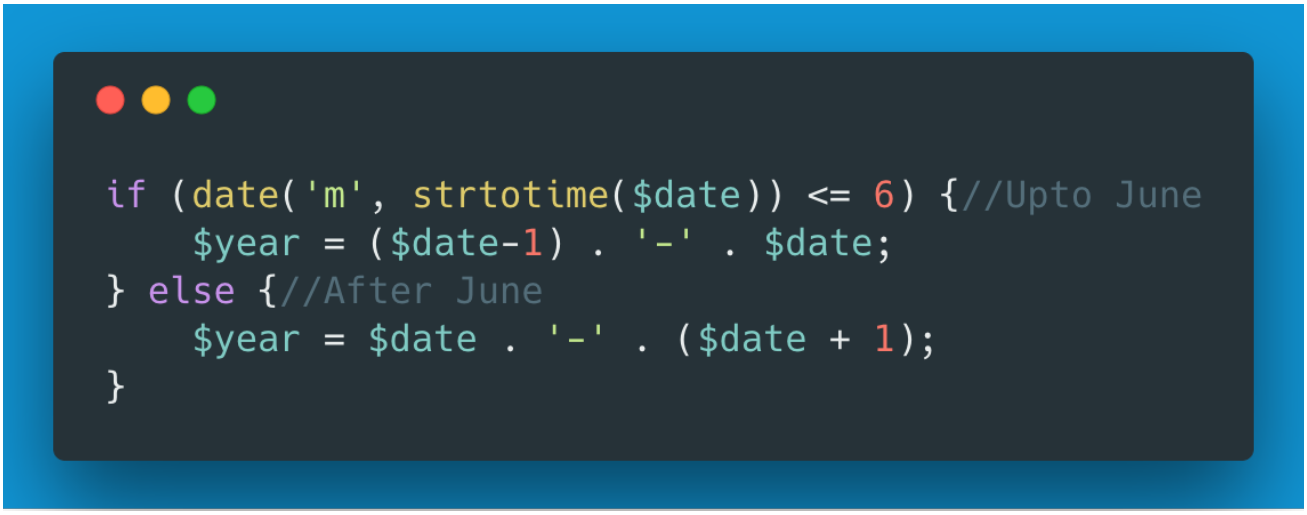


```php
protected static function boot()
{
    parent::boot();

    static::creating(function ($query) {
        $query->column = 'default value';
    });
}
```

8th Feb 2018

Financial year select menu (https://daveismyname.blog/financial-year-select-menu)



```php
if (date('m', strtotime($date)) <= 6) {//Upto June
    $year = ($date-1) . '-' . $date;
} else {//After June
    $year = $date . '-' . ($date + 1);
}
```

8th Feb 2018

Regular expression to convert usernames into links like Twitter (https://daveismyname.blog/regular-expression-to-convert-usernames-into-links-like-twitter)

```php
$subject = 'Hello, @daveismyname twitter handle is clickable but not this email someone@domain.com';

echo preg_replace('/\B\@([a-zA-Z0-9_]{1,254})/', '<a href="user/$1">$0</a>', $subject)
```

8th Feb 2018

Archives

February 2018 (https://daveismyname.blog/archive/February/2018)
January 2018 (https://daveismyname.blog/archive/January/2018)
December 2017 (https://daveismyname.blog/archive/December/2017)
November 2017 (https://daveismyname.blog/archive/November/2017)
October 2017 (https://daveismyname.blog/archive/October/2017)
September 2017 (https://daveismyname.blog/archive/September/2017)
August 2017 (https://daveismyname.blog/archive/August/2017)
July 2017 (https://daveismyname.blog/archive/July/2017)
June 2017 (https://daveismyname.blog/archive/June/2017)
May 2017 (https://daveismyname.blog/archive/May/2017)
April 2017 (https://daveismyname.blog/archive/April/2017)
March 2017 (https://daveismyname.blog/archive/March/2017)
February 2017 (https://daveismyname.blog/archive/February/2017)
January 2017 (https://daveismyname.blog/archive/January/2017)
December 2016 (https://daveismyname.blog/archive/December/2016)

November 2016 (https://daveismyname.blog/archive/November/2016)
October 2016 (https://daveismyname.blog/archive/October/2016)
September 2016 (https://daveismyname.blog/archive/September/2016)
July 2016 (https://daveismyname.blog/archive/July/2016)
June 2016 (https://daveismyname.blog/archive/June/2016)
May 2016 (https://daveismyname.blog/archive/May/2016)
April 2016 (https://daveismyname.blog/archive/April/2016)
March 2016 (https://daveismyname.blog/archive/March/2016)
February 2016 (https://daveismyname.blog/archive/February/2016)
January 2016 (https://daveismyname.blog/archive/January/2016)
December 2015 (https://daveismyname.blog/archive/December/2015)
November 2015 (https://daveismyname.blog/archive/November/2015)
October 2015 (https://daveismyname.blog/archive/October/2015)
September 2015 (https://daveismyname.blog/archive/September/2015)
August 2015 (https://daveismyname.blog/archive/August/2015)
July 2015 (https://daveismyname.blog/archive/July/2015)
June 2015 (https://daveismyname.blog/archive/June/2015)
April 2015 (https://daveismyname.blog/archive/April/2015)
March 2015 (https://daveismyname.blog/archive/March/2015)
February 2015 (https://daveismyname.blog/archive/February/2015)
January 2015 (https://daveismyname.blog/archive/January/2015)
December 2014 (https://daveismyname.blog/archive/December/2014)
November 2014 (https://daveismyname.blog/archive/November/2014)
October 2014 (https://daveismyname.blog/archive/October/2014)
September 2014 (https://daveismyname.blog/archive/September/2014)
August 2014 (https://daveismyname.blog/archive/August/2014)
July 2014 (https://daveismyname.blog/archive/July/2014)
June 2014 (https://daveismyname.blog/archive/June/2014)
May 2014 (https://daveismyname.blog/archive/May/2014)
April 2014 (https://daveismyname.blog/archive/April/2014)
March 2014 (https://daveismyname.blog/archive/March/2014)
February 2014 (https://daveismyname.blog/archive/February/2014)
January 2014 (https://daveismyname.blog/archive/January/2014)
December 2013 (https://daveismyname.blog/archive/December/2013)
October 2013 (https://daveismyname.blog/archive/October/2013)
September 2013 (https://daveismyname.blog/archive/September/2013)
August 2013 (https://daveismyname.blog/archive/August/2013)
July 2013 (https://daveismyname.blog/archive/July/2013)
June 2013 (https://daveismyname.blog/archive/June/2013)
May 2013 (https://daveismyname.blog/archive/May/2013)
April 2013 (https://daveismyname.blog/archive/April/2013)
March 2013 (https://daveismyname.blog/archive/March/2013)
February 2013 (https://daveismyname.blog/archive/February/2013)
January 2013 (https://daveismyname.blog/archive/January/2013)
October 2012 (https://daveismyname.blog/archive/October/2012)
September 2012 (https://daveismyname.blog/archive/September/2012)

August 2012 (https://daveismyname.blog/archive/August/2012)
July 2012 (https://daveismyname.blog/archive/July/2012)
June 2012 (https://daveismyname.blog/archive/June/2012)
May 2012 (https://daveismyname.blog/archive/May/2012)
April 2012 (https://daveismyname.blog/archive/April/2012)
March 2012 (https://daveismyname.blog/archive/March/2012)
February 2012 (https://daveismyname.blog/archive/February/2012)
January 2012 (https://daveismyname.blog/archive/January/2012)
November 2011 (https://daveismyname.blog/archive/November/2011)
September 2011 (https://daveismyname.blog/archive/September/2011)
July 2011 (https://daveismyname.blog/archive/July/2011)
June 2011 (https://daveismyname.blog/archive/June/2011)
May 2011 (https://daveismyname.blog/archive/May/2011)
April 2011 (https://daveismyname.blog/archive/April/2011)
March 2011 (https://daveismyname.blog/archive/March/2011)
February 2011 (https://daveismyname.blog/archive/February/2011)
January 2011 (https://daveismyname.blog/archive/January/2011)
December 2010 (https://daveismyname.blog/archive/December/2010)
November 2010 (https://daveismyname.blog/archive/November/2010)
October 2010 (https://daveismyname.blog/archive/October/2010)
September 2010 (https://daveismyname.blog/archive/September/2010)
August 2010 (https://daveismyname.blog/archive/August/2010)
July 2010 (https://daveismyname.blog/archive/July/2010)
June 2010 (https://daveismyname.blog/archive/June/2010)
May 2010 (https://daveismyname.blog/archive/May/2010)
April 2010 (https://daveismyname.blog/archive/April/2010)
March 2010 (https://daveismyname.blog/archive/March/2010)
February 2010 (https://daveismyname.blog/archive/February/2010)
January 2010 (https://daveismyname.blog/archive/January/2010)
December 2009 (https://daveismyname.blog/archive/December/2009)
November 2009 (https://daveismyname.blog/archive/November/2009)
October 2009 (https://daveismyname.blog/archive/October/2009)

DONATE

Support my work by donating with PayPal.

LATEST TWEETS

Tweets by daveismynamecom (https://twitter.com/daveismynamecom?ref_src=twsrc%5Etfw)

FOLLOW ME ON FACEBOOK

David Carr Blog (https://www.facebook.com/davidcarrblog/)