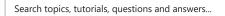
REFERENCES EXAMPLES FAQ



HOME



SQL



< >

SNIPPETS



Online HTML Editor

ADVERTISEMENTS





 $\triangleright \times$



PHP Introduction

PHP Getting Started

HTML5

CSS3 JAVASCRIPT

JQUERY

BOOTSTRAP4

PHP Syntax

PHP Variables

PHP Constants

PHP Echo and Print

PHP Data Types

PHP Strings

PHP Operators

PHP If...Else

PHP Switch...Case

PHP Arrays

PHP Sorting Arrays

PHP Loops

PHP Functions

PHP Math Operations

PHP GET and POST

PHP ADVANCED

PHP Date and Time

PHP Include Files

PHP File system

PHP Parsing Directories

PHP File Upload

PHP File Download

PHP Cookies

PHP Sessions

PHP Send Email

PHP Form Handling

PHP Form Validation

PHP Filters

PHP Error Handling

PHP Classes and Objects



PHP MySQL Login System

PHP7



Implementing User Authentication Mechanism

User authentication is very common in modern web application. It is a security mechanism that is used to restrict unauthorized access to member-only areas and tools on a site.

In this tutorial we'll create a simple registration and login system using the PHP and MySQL. This tutorial is comprised of two parts: in the first part we'll create a user registration form, and in the second part we'll create a login form, as well as a welcome page and a logout script.

Building the Registration System

In this section we'll build a registration system that allows users to create a new account by filling out a web form. But, first we need to create a table that will hold all the user data.

Step 1: Creating the Database Table

Execute the following SQL query to create the users table inside your MySQL database.







PHP Magic Constants
PHP JSON Parsing
PHP Regular Expressions
PHP Exception Handling

PHP & MySQL DATABASE

PHP MySQL Introduction PHP MySQL Connect PHP MySQL Create Database PHP MySQL Create Table PHP MySQL Insert PHP MySQL Prepared PHP MySQL Last Inserted ID PHP MySQL Select PHP MySQL Where PHP MySQL Limit PHP MySQL Order By PHP MySQL Update PHP MySQL Delete PHP MySQL CRUD Application PHP MySQL Ajax Search PHP MySQL Login System

PHP EXAMPLES

PHP Practice Examples
PHP FAQ's Answers

PHP REFERENCE

PHP String Functions
PHP Array Functions
PHP File System Functions
PHP Date/Time Functions
PHP Calendar Functions
PHP MySQLi Functions
PHP Filters
PHP Error Levels
More references

Please check out the tutorial on SQL CREATE TABLE statement for the detailed information about syntax for creating tables in MySQL database system.

Step 2: Creating the Config File

After creating the table, we need create a PHP script in order to connect to the MySQL database server. Let's create a file named "config.php" and put the following code inside it.

```
Example
              Procedural
                        Object Oriented
                                     PDO
                                                                          ± Download
     define('DB_NAME', 'demo');
 8
 9
     /* Attempt to connect to MySQL database */
     $link = mysqli connect(DB SERVER, DB USERNAME, DB PASSWORD, DB NAME);
10
11
12
     // Check connection
     if($link === false){
13
14
         die("ERROR: Could not connect. " . mysqli_connect_error());
15
    }
16
   ?>
```

If you've downloaded the Object Oriented or PDO code examples using the download button, please remove the text "-oo-format" or "-pdo-format" from file names before testing the code.



Note: Replace the credentials according to your MySQL server setting before testing this code, for example, replace the database name 'demo' with your own database name, replace username 'root' with your own database username, specify database password if there's any.

Step 3: Creating the Registration Form

Let's create another PHP file "register.php" and put the following example code in it. This example code will create a web form that allows user to register themselves.

This script will also generate errors if a user tries to submit the form without entering any value, or if username entered by the user is already taken by another user.

```
PDO
                                                                      Example
                         Object Oriented
                                                                             ± Download
104
               .wrapper{ width: 350px; padding: 20px; }
105
          </style>
106
      </head>
107
      <body>
108
          <div class="wrapper">
109
               <h2>Sign Up</h2>
```



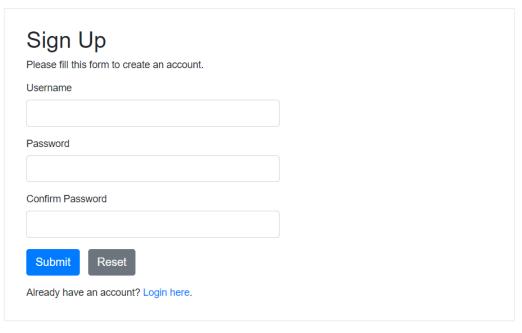
Καλύτερο από κουμπαρό





Creating a User Login System with PHP and MySQL - Tutorial Republic

— The output of the above example (i.e. signup form) will look something like this:



In the above example, we've used the PHP password_hash() function to create password hash from the password string entered by the user (*line no-75*). This function creates a password hash using a strong one-way hashing algorithm. It also generates and applies a random salt automatically when hashing the password; this means that even if two users have the same passwords, their password hashes will be different.

At the time of login we'll verify the given password with the password hash stored in the database using the PHP password_verify() function, as demonstrated in the next example.

We've used the Bootstrap framework to make the form layouts quickly and beautifully. Please, checkout the Bootstrap tutorial section to learn more about this framework.



Tip: Password salting is a technique which is widely used to secure passwords by randomizing password hashes, so that if two users have the same password, they will not have the same password hashes. This is done by appending or prepending a random string, called a salt, to the password before hashing.

Building the Login System

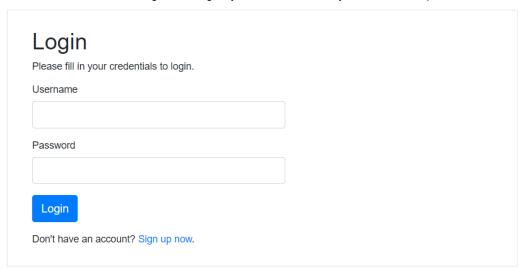
In this section we'll create a login form where user can enter their username and password. When user submit the form these inputs will be verified against the credentials stored in the database, if the username and password match, the user is authorized and granted access to the site, otherwise the login attempt will be rejected.

Step 1: Creating the Login Form

Let's create a file named "login.php" and place the following code inside it.

```
PDO
Example
              Procedural
                         Object Oriented
                                                                    ± Download
      <?php echo (!empty($password_err)) ? 'is-invalid' : ''; ?>">
122
                      <span class="invalid-feedback"><?php echo $password_err; ?>
      </span>
123
                  </div>
                  <div class="form-group">
124
125
                      <input type="submit" class="btn btn-primary" value="Login">
126
                  </div>
127
                  >Don't have an account? <a href="register.php">Sign up</a>
      now</a>.
128
              </form>
129
          </div>
      </body>
130
131
      </html>
```

— The output of the above example (i.e. login form) will look something like this:



Step 2: Creating the Welcome Page

Here's the code of our "welcome.php" file, where user is redirected after successful login.

```
Example
                                                                       ± Download
23
         <h1 class="my-5">Hi, <b><?php echo
     htmlspecialchars($_SESSION["username"]); ?></b>. Welcome to our site.</h1>
24
25
             <a href="reset-password.php" class="btn btn-warning">Reset Your
     Password</a>
26
             <a href="logout.php" class="btn btn-danger ml-3">Sign Out of Your
     Account</a>
27
         </body>
28
29
     </html>
```

If data comes from external sources like form filled in by anonymous users, there is a risk that it may contain malicious script indented to launch cross-site scripting (XSS) attacks. Therefore, you must escape this data using the PHP htmlspecialchars() function before displaying it in the browser, so that any HTML tag it contains becomes harmless.

For example, after escaping special characters the string <script>alert("XSS")</script> becomes <script>alert("XSS")</script> which is not executed by the browser.

Step 3: Creating the Logout Script

Now, let's create a "logout.php" file. When the user clicks on the log out or sign out link, the script inside this file destroys the session and redirect the user back to the login page.

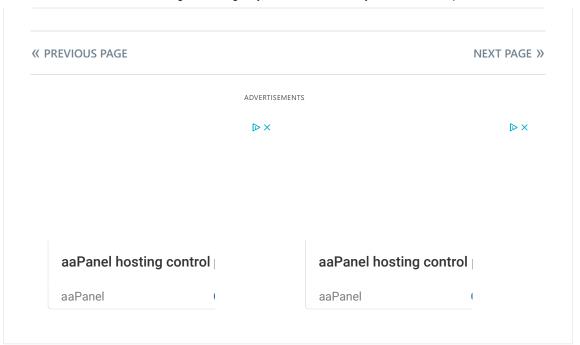
```
Example
                                                                         ± Download
 1
     <?php
     // Initialize the session
     session_start();
 4
     // Unset all of the session variables
     $_SESSION = array();
 7
     // Destroy the session.
     session_destroy();
 9
10
     // Redirect to login page
11
     header("location: login.php");
12
13
     exit;
     ?>
14
```

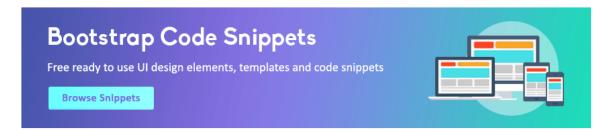
Adding the Password Reset Feature

Finally, in this section we will add the password reset utility to our login system. Using this feature *logged in users* can instantly reset their own password for their accounts.

Let's create a file named "reset-password.php" and place the following code inside it.

```
± Download
Example
              Procedural
                        Object Oriented
                                     PDO
26
         } else{
27
              $new_password = trim($_POST["new_password"]);
28
29
         // Validate confirm password
30
31
         if(empty(trim($_POST["confirm_password"]))){
32
              $confirm_password_err = "Please confirm the password.";
33
              $confirm_password = trim($_POST["confirm_password"]);
34
35
              if(empty($new_password_err) && ($new_password !=
     $confirm_password)){
                  $confirm_password_err = "Password did not match.";
36
37
38
```





Is this website helpful to you? Please give us a like, or share your feedback to help us improve. Connect with us on Facebook and Twitter for the latest updates.

ABOUT US	CONTACT	INTERACTIVE TOOLS		
Our Story	Contact Us	Font Awesome Icon Search Utility	HTML Formatter	
Terms of Use	Report Error	Title & Meta Length Calculator	HTML Color Picker	
Privacy Policy	Advertise	Bootstrap Button Generator	HTML Editor	👨 Buy us a coffee
		SQL Playground		U UU

Copyright © 2021 Tutorial Republic. All Rights Reserved.

Share This: **f**