

Creating a Logout Page

In the previous tutorial, you protected pages so that only logged-in users could access them. In this tutorial, you will create a logout page that destroys the user's session and returns them to the login page.

Why Do We Need Logout?

When a user logs in, their information is stored in a PHP session.

For example:

```
user_id = 1
username = admin
```

To completely sign the user out, we need to remove this session information.

Create the Logout Page

Create a new file called:

```
logout.php
```

Add the following code:

```
<?php

session_start();

session_destroy();

header("Location: login.php");
exit();
```

```
?>
```

Save the file.

Understanding the Code

The following line starts the current session:

```
session_start();
```

The following line removes all session data:

```
session_destroy();
```

Finally, the user is redirected back to the login page:

```
header("Location: login.php");  
exit();
```

Test the Logout Page

First, log in to your application.

You should be redirected to:

```
http://localhost/members.php
```

Now manually visit:

```
http://localhost/logout.php
```

You should immediately be redirected to:

```
http://localhost/login.php
```

Verify the Session Has Been Removed

After visiting:

```
http://localhost/logout.php
```

attempt to access:

```
http://localhost/members.php
```

You should no longer have access.

Instead, you should be redirected back to:

```
http://localhost/login.php
```

This confirms that the session was successfully destroyed.

Add a Logout Link

Open:

```
members.php
```

Add the following code near the bottom of the page:

```
<p>
  <a href="logout.php">Logout</a>
</p>
```

Example:

```
<h1>Members Area</h1>

<p>Welcome,
<?php echo $_SESSION["username"]; ?>
</p>
```

```
<p>User ID:  
<?php echo $_SESSION["user_id"]; ?>  
</p>  
  
<p>  
    <a href="logout.php">Logout</a>  
</p>
```

Save the file.

Test the Logout Link

1. Log in.
2. Open the Members Area.
3. Click the Logout link.
4. Confirm you are returned to the login page.
5. Attempt to revisit members.php.

You should be redirected back to the login page.

Complete logout.php File

```
<?php  
  
session_start();  
  
session_destroy();  
  
header("Location: login.php");  
exit();  
  
?>
```

Complete Updated members.php File

```
<?php

session_start();

if (!isset($_SESSION["user_id"])) {

    header("Location: login.php");
    exit();

}

?>

<!DOCTYPE html>
<html>
<head>
    <title>Members Area</title>
</head>
<body>

<h1>Members Area</h1>

<p>Welcome,
<?php echo $_SESSION["username"]; ?>
</p>

<p>User ID:
<?php echo $_SESSION["user_id"]; ?>
</p>

<p>
    <a href="logout.php">Logout</a>
</p>
```

```
</body>
```

```
</html>
```

You now have a complete authentication system that allows users to:

- Register an account
- Log in
- Stay logged in using sessions
- Access protected pages
- Log out

Next tutorial: **Adding Role-Based Access Control (Admin, Teacher and Student Accounts).**

Revision #1

Created 2026-06-08 08:53:47 UTC by Mr Napper

Updated 2026-06-08 08:55:10 UTC by Mr Napper