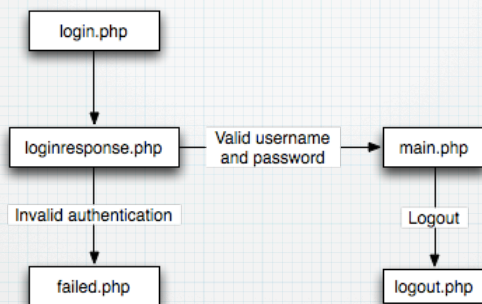


Web application session management

Demo

User: **User1** and password **qwerty**

Authentication process



login page



```
<? include "../resources/userlayout.php"; ?>
```

```
<? jokeTop("BadJokeCity.com"); ?>  
<? jokeMainBanner(); ?>  
<? jokeMainStart(); ?>  
<? jokeMainLeft(); ?>
```

..form with **username** and **password** input fields..

```
<? jokeMainFinish(); ?>  
<? jokeMainFooter(); ?>  
<? jokeBottom(); ?>
```

login page form

```
<form name="form1" id="form1" method="post" action="loginresponse.php">
  <table width="99%" height="151" border="0">
    <tr>
      <td width="23%">Username:</td>
      <td width="77%"><input name="username" type="text" id="username" /></td>
    </tr>
    <tr>
      <td height="24">Password:</td>
      <td><input name="password" type="password" id="password" /></td>
    </tr>
    <tr>
      <td></td>
      <td><input type="submit" name="Submit" value="Login" /></td>
    </tr>
  </table>
</form>
```

Response to button click is **loginresponse.php** file

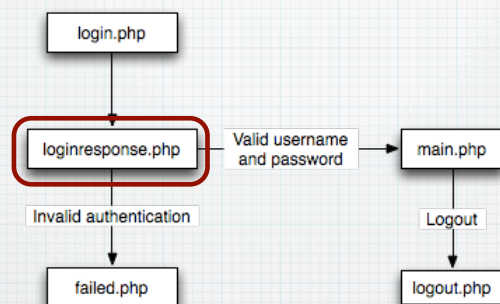
login page form

```
<form name="form1" id="form1" method="post" action="loginresponse.php">
  <table width="99%" height="151" border="0">
    <tr>
      <td width="23%">Username:</td>
      <td width="77%"><input name="username" type="text" id="username" /></td>
    </tr>
    <tr>
      <td height="24">Password:</td>
      <td><input name="password" type="password" id="password" /></td>
    </tr>
    <tr>
      <td></td>
      <td><input type="submit" name="Submit" value="Login" /></td>
    </tr>
  </table>
</form>
```

Password input type displays ********* and doesn't cache the value

Use of the **back** button does not redisplay the password

Authentication process



Where is **loginresponse.php**?

Watch the login demo again ...

loginresponse.php

Check for username and password values

If missing values then redirect to failed.php

Search database for username indicated

If mismatched values then redirect to failed.php

Store user details in a PHP session

Redirect to main.php

loginresponse.php

Check for username
and password values

```
if (($_POST['username']=="" or ($_POST['password']==""))
{
  header('Location: failed.php');
  exit();
}
```

If missing values then
redirect to failed.php

Search database for username indicated

If mismatched values then redirect to failed.php

Store user details in a PHP session

Redirect to main.php

Redirection in PHP

Use the PHP `header('location:URL')` function to
redirect to another page

Issues an appropriate HTTP request for the
indicated page

loginresponse.php

```
if (($_POST['username']=="" or ($_POST['password']==""))
{
header('Location: failed.php');
exit();
}
```

Search database for username indicated

If mismatched values then redirect to failed.php

Store user details in a PHP session

Redirect to main.php

loginresponse.php

```
if (($_POST['username']=="" or ($_POST['password']==""))
{
header('Location: failed.php');
exit();
};
```

Search database for
username indicated

```
$_result1 = queryDatabase("SELECT * FROM user where  
username='".$_strtolower($_POST['username']).'");
```

If mismatched values then redirect to failed.php

Store user details in a PHP session

Redirect to main.php

loginresponse.php

```
if (($_POST['username']=="" or ($_POST['password']==""))
{
header('Location: failed.php');
exit();
};
$_result1 = queryDatabase("SELECT * FROM user where  
username='".$_strtolower($_POST['username']).'");
```

If mismatched values then redirect to failed.php

Store user details in a PHP session

Redirect to main.php

loginresponse.php

```
if (($_POST['username']=='') or ($_POST['password']==''))
{
    header('Location: failed.php');
    exit();
};
$result1 = queryDatabase("SELECT * FROM user where
    username='".$_strtolower($_POST['username'])."");

$row1 = mysql_fetch_array($result1);
if ($_POST['password']!= $row1['password'])
{
    header('Location: failed.php');
    exit();
};
```

If mismatched values
then redirect to
failed.php

Store user details in a PHP session

Redirect to main.php

loginresponse.php

```
if (($_POST['username']=='') or ($_POST['password']==''))
{
    header('Location: failed.php');
    exit();
};
$result1 = queryDatabase("SELECT * FROM user where
    username='".$_strtolower($_POST['username'])."");

$row1 = mysql_fetch_array($result1);
if ($_POST['password']!= $row1['password'])
{
    header('Location: failed.php');
    exit();
};
```

Store user details in a PHP session

Redirect to main.php

PHP Sessions

Sessions allow web servers to save state (typically the value of variables) in a file stored by the web server - linked to a cookie in the browser

Can be used to pass values between pages

Session configuration done through **php.ini** file, things like location, default behavior etc.

PHP Sessions - Apache and Firefox demo

Logging in creates a cookie with a distinct name in Firefox (client side)

See current cookies using Preferences - Privacy - Show Cookies (PHPSESSID)

Simultaneously, a text file with the Cookie ID is created in the web server

See current cookies using MAMP - tmp - php

Session functions

<code>session_start ()</code>	creates the session
<code>session_register ('name')</code>	register a variable inside the cookie / session
<code>session_unregister ('name')</code>	remove a variable from the cookie / session
<code>session_is_registered ('name')</code>	check to see if variable exists

Many other functions to manipulate sessions (see PHP help files)

Storing the user authentication state

Here the users `userid` and `username` is stored in the session (should really be encrypted)

Use an initial `session_start ()` to create a new session

```
session_start();  
session_register('userid');  
session_register('username');  
$userid=$_row1['userid'];  
$username=strtolower($_POST['username']);  
header('Location: main.php');  
exit();
```

Storing the user authentication state

Here the users userno and username is stored in the session (should really be encrypted)

```
session_start();
session_register('userno');
session_register('username');
$userno=$_row1['userno'];
$username=strtolower($_POST['username']);
header('Location: main.php');
exit();
```

Use session_register ('variable') to indicate the variable that should be stored in the session

Storing the user authentication state

Here the users userno and username is stored in the session (should really be encrypted)

```
session_start();
session_register('userno');
session_register('username');
$userno=$_row1['userno'];
$username=strtolower($_POST['username']);
header('Location: main.php');
exit();
```

Assign values to the variables as normal

Storing the user authentication state

Here the users userno and username is stored in the session (should really be encrypted)

```
session_start();
session_register('userno');
session_register('username');
$userno=$_row1['userno'];
$username=strtolower($_POST['username']);
header('Location: main.php');
exit();
```

Redirect to the logged in page 'main.php'

So how do we use this on 'secure' pages?

```
session_start();
if (!$HTTP_SESSION_VARS['username']!="")
{
    header('Location: failed.php');
    exit();
};
```

Every 'secure' page has a block of code to check for the session values

If the correct session variable exists allow access to the page, otherwise redirect to the **failed.php** file

The only way for the session variables to have been created is through **loginresponse.php** page

main.php Main log in page

```
<?
include "../resources/session.php";
include "../resources/userlayout.php";
?>

<? jokeTop("BadJokeCity.com"); ?>
<? jokeUserBanner(); ?>
<? jokeMainStart(); ?>
<? jokeUserLeft(); ?>

<h1>Authenticated</h1>

<p>
Welcome user <? print $username; ?>, you have logged
in.
</p>

<? jokeMainFinish(); ?>
<? jokeUserFooter(); ?>
<? jokeBottom(); ?>
```

Check for correct authentication (redirects out to failed.php if invalid)

Display main logged in message

logout.php

```
<?
session_start();
session_destroy();
include "../resources/userlayout.php";
?>
```

Remove session variables (note - have to start a session before destroying it)

```
<? jokeTop("BadJokeCity.com"); ?>
<? jokeMainBanner(); ?>
<? jokeMainStart(); ?>
<? jokeMainLeft(); ?>
```

```
<h1>Log out request</h1>
```

Displays goodbye message

```
<p>You have logged out</p>
```

```
<? jokeMainFinish(); ?>
<? jokeMainFooter(); ?>
<? jokeBottom(); ?>
```


failed.php

```
<?
include "../resources/userlayout.php";
?>
```

```
<? jokeTop("BadJokeCity.com"); ?>
<? jokeMainBanner(); ?>
<? jokeMainStart(); ?>
<? jokeMainLeft(); ?>
```

```
<h1>User access problem</h1>
<p>The page you have requested is not
available as you have not
authenticated correctly.</p>
```

**Display a failed
authentication message**

```
<? jokeMainFinish(); ?>
<? jokeMainFooter(); ?>
<? jokeBottom(); ?>
```