Web Development

Authentication & Authorization

Outline

We’re going to look at a number of areas that we’ve already looked at

login, registration

security issues surrounding above

We’ll fill in some Gaps particularly relevant to the project.

Sessions and creating logout option

Captcha add to registration page

Authorization & Authentication

Say you are going to a football match with a ticket

Authentication could be thought of as the process of presenting your ticket to the security staff. if you have a valid ticket you are let into the stadium.

Authorization could the be thought of as which regions of the stadium am I allowed into w.g. terrace, stand etc.

Authentication

What tools do we have to authenticate users

- Database, we store user details in a database so we can verify them as valid users later at login

Process

- Allow users to **register** with our site and store their details
- When users logs in they are verified against registration details

Authentication

Security

How can we make this process more secure:

encrypt any passwords (in php) before storing to database

password_hash($password,CONST);

CONST - algorithm type

ensure passwords are not too simple (**validation**)

Please do the tutorial on moodle that outlines this process. Lecture 7

Authentication

We can add a Captcha to a registration file to eliminate the use of robots registering on your form

I’ve added a tutorial to the moodle page

Adding a Captcha

We’ll do this at the end of the class if we have time. Otherwise do it in your own time as this is a requirement for the project

Authorization

How do we make sure an Authenticated user has access to certain areas of the application.

We can use **Sessions**.

Once a user has logged in and has been authenticated we can use sessions to keep track of data between pages

Sessions mean we can pass data around an application without having to go back to the database constantly.

Without sessions it would be difficult to maintain state i.e. who is logged in, what their user_type is, user name etc.

Authorization - Spec

We’re going to write a simple application that allows a user to login in to a form

the method the form uses is a POST

the action of the form is the same as the form file, the *login.php* file.

Once we have submitted the form we will captute the form data in some session variables using the global array variable **$_SESSION**

We will then **redirect** to a confirm page. Which will display the session data

We will **add a href link** to allow the user to **logout** of the sesssion and redirect to the login page

Authorization

Starting a Session

To start a session within an application we use the *session_start()* php function.

We place this at the top of the php files we use within the application to access the session variables, Add:

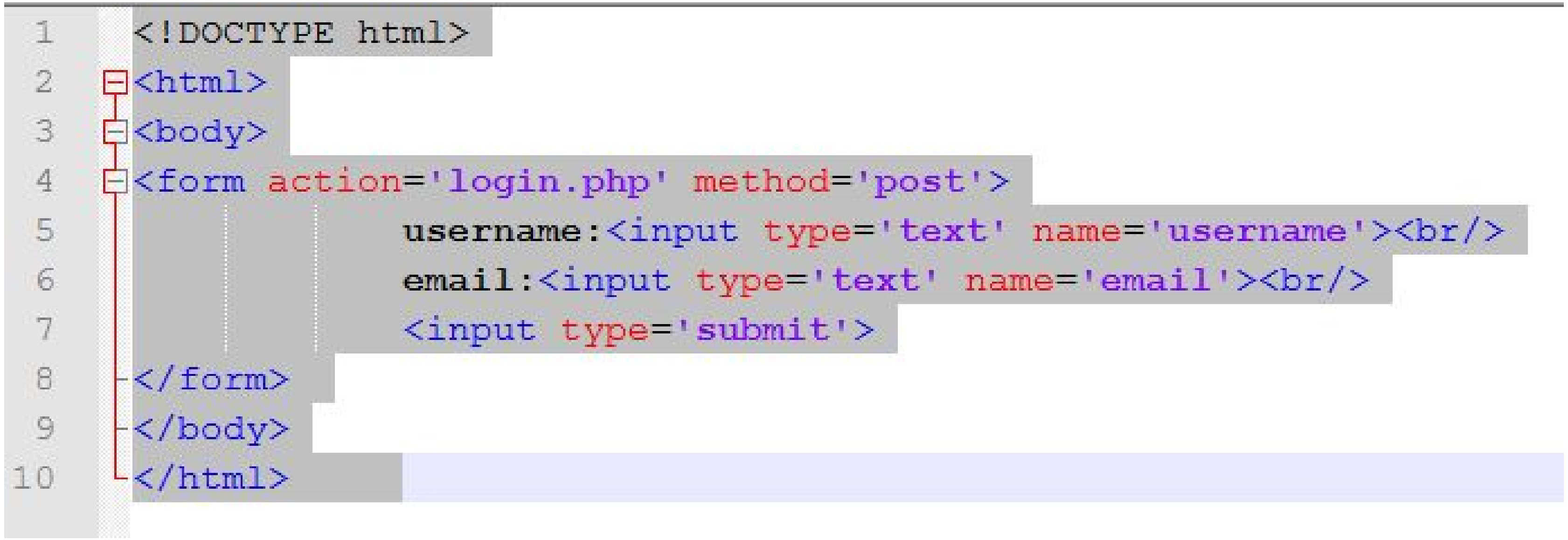
```
<?php

        session_start();
?>
```

To the top the login.php file

Authorization

Let’s start with a simple form. Call it login.php . There’s a version on the moodle page to download if you wish. e.g. htdocs/lecture11/



```
<!DOCTYPE html>
<html>
<body>
<form action='login.php' method='post'>
		username:<input type='text' name='username'><br/>
		email:<input type='text' name='email'><br/>
		<input type='submit'>
</form>
</body>
</html>
```

Authorization

Let’s add the post data session . To some session variables.

Once set we will redirect to confirm file ***confirm.php***

So our php code now looks like

```
<?php
	if ($_POST){
		$_SESSION[‘username’]=$_POST[‘username’];
		$_SESSION[email]=$_POST[email];
		header(“Location: confirm.php”);
	}
?>
```

Authorization

confirm.php

remember we need the *session_start()* at the top of the php file to access the session variables.

And we want to print out the session variables.

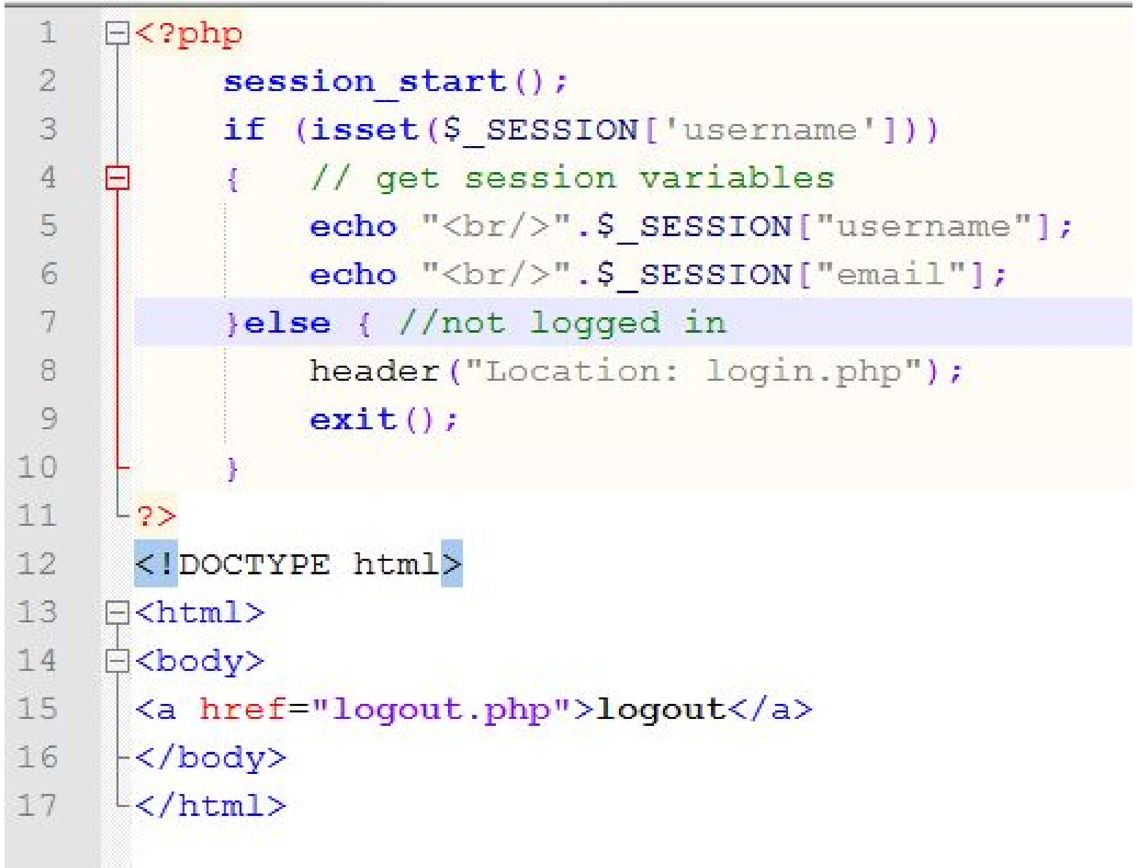
But lets add a check so that the

$_SESSION['username'] must be set to print out the values and include the logout link

otherwise

    immediately redirect user to the login page.

Confirm.php



```
<?php
	session_start();
	if (isset($_SESSION['username']))
	{	// get session variables
		echo "<br/>".$_SESSION["username"];
		echo "<br/>".$_SESSION["email"];
	}else { //not logged in
		header("Location: login.php");
		exit();
	}
?>
<!DOCTYPE html>
<html>
<body>
<a href="logout.php">logout</a>
</body>
</html>
```

logout.php

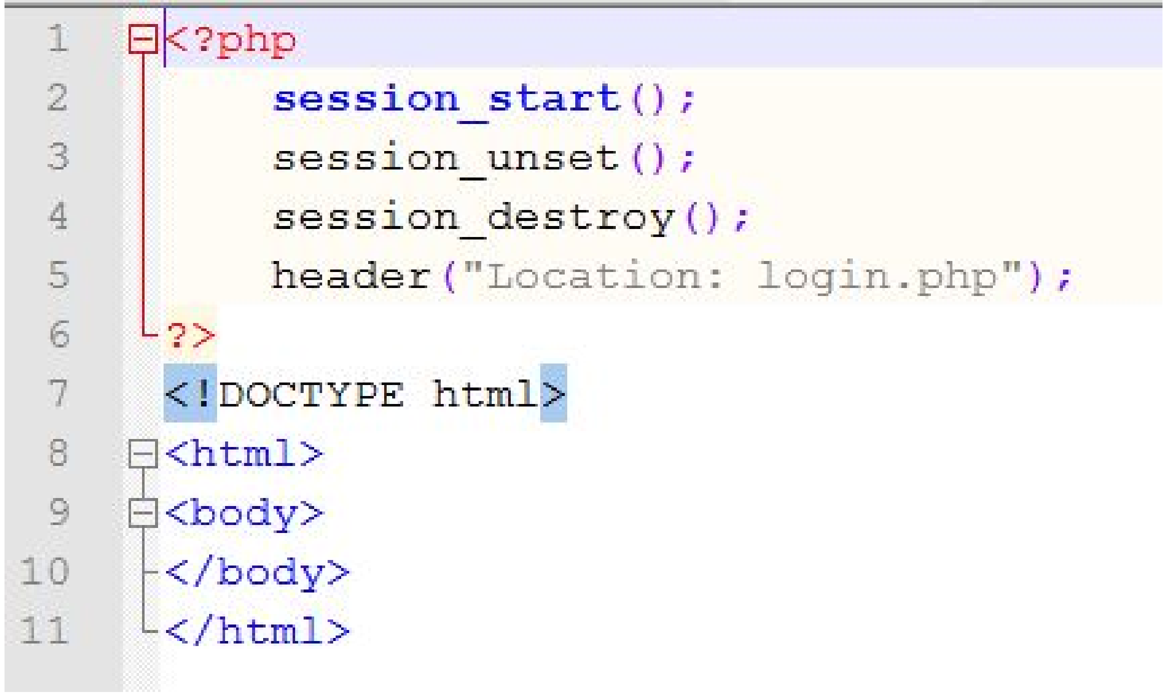
Let’s add the ***logout.php*** file.

Here we want to destroy the Session we have created and return to the login page, removing all the session variables.

Two new session functions

session_unset()	-	destroys all the session variables for that session
session_destroy()	-	removes files associated with the session

logout.php



```
<?php
	session_start();
	session_unset();
	session_destroy();
	header("Location: login.php");
?>
<!DOCTYPE html>
<html>
<body>
</body>
</html>
```

logout.php

Prove that the session has been destroyed

Once logged out try and bringup the confirm.php file in the browser

i.e.

    localhost/lecture11/confirm.php

You should immediately be redirected back to the login page as the $_SESSION[‘username’] should no longer be set

Data Management

What other ways can we transfer data from one php file to another ?

Form POST and GET via the $_POST and $_GET global variables

Passing parameters in a href link

```
<a href=”confirm.php?id=1”>link</a>
```

Then use $_GET to retreive parameter ***id*** in confirm.php

Cookies, similar to sessions but less secure. We’ll look at Cookies in the next lecture