

IT420: Database Management and Organization

Session Control in PHP (Chapter 22 – PHP and MySQL Web Development)

Goals Today

- Reminder IT/CS Dinner Meal Registration
- Storing and Checking Passwords
- Sessions

Authentication

- Want: Allow access to a web page only to some users
- Solution: Ask for user authentication
 - log in

Step 1: Ask Login Information

Please Log In

Only valid users are allowed to see the content of this page!

Username
Password

Log In

Step 2-a: If Incorrect Information, Display Error Message

Error!

You are not authorized to view the content of this page.

Step 2-b If Correct Information, Display Secret Page

Here it is!

This is the secret content of the page.

Class Exercise

- Write a PHP script:
 - If no login info given, ask for login information
 - If username = 'user' and password = 'pass',
 - display protected content
 - Else, display error message

pass_protect.php

```
<?php
//create short names for variables
$name = $_POST['name'];
$password = $_POST['password'];

if(empty($name) | empty($password))
{
    //visitor needs to enter a name and password
?>
<h3>Please Log In!</h3>
Only valid users are allowed to see the content of this page!
<form border="1" action="pass_protect.php">
<table border="1">
<tr>
<td> Username </td> <td> <input type="text" name="name"> </td>
</tr>
<tr>
<td> Password </td> <td> <input type="password" name="password"> </td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit" value="Log In">
</td>
</tr>
</table>
</form>
<?php

else if($name == "user" && $password == "pass")
{
    // visitor's name and password combination are correct
    echo '<h3>Here is 1!</h3>';
    echo 'This is the secret content of the page..';
}
else
{
    // visitor's name and password combination are not correct
    echo '<h3>Error!</h3>';
    echo 'You are not authorized to view the content of this page..';
}
?>
```

Problems with the code

- One user-name and password hard-coded
- Password stored as plain text
- Protection for only one page
- Password transmitted as plain text

Storing Users and Passwords

- In a file on the server
- In a database
 - Users(Username, Password)
 - How do we test that user information matches the information in the database?

Encrypting Passwords

- **DO NOT** store passwords as plain text!
- Use one-way hash functions
 - *string* sha1(*string* str)
- Example: sha1('pass') == '9d4e1e23bd5b727046a9e3b4b7db57bd8d6ee684'
- Deterministic output!
 - Given same string, sha1 returns the same result every time

Example Using Encrypted Password

- Instead of
 - if (\$name == 'user' && \$pass == 'password'){

 //OK, passwords match
 }
- Use
 - if (\$name == 'user' && sha1(\$pass) == '9d4e1e23bd5b727046a9e3b4b7db57bd8d6ee684'){

 //OK, passwords match
 }

Problems with the code

- One user-name and password hard-coded
- Password stored as plain text
- Protection for only one page
- Password transmitted as plain text

Learned So Far...

- Ask login information
- Encrypt passwords
 - sha1()
- Store/get login information
 - File
 - Database

Session Control

- HTTP – no built-in way to maintain state between two transactions
- Want: Track a user during a single session on a website – remember state
 - Show content personalized to user
 - Implement shopping carts

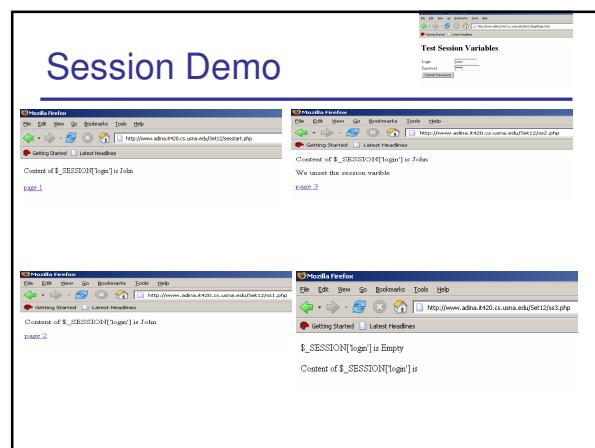
PHP Session Control

- Session ID – cryptographically random number
 - Generated for each session
 - Stored on client site
 - Cookie
 - URL
- Session variables
 - Created by PHP script
 - Stored on the server site
 - If session id visible (cookie or URL), session variables can be accessed by all scripts

Implementing Sessions in PHP

- Start a session – `session_start()`
- Register session variables
 - `$_SESSION['myvar'] = 'some value'`
- Use session variables
 - `session_start()`
 - `if (isset($_SESSION['myvar'])) { //OK code}`
- Deregister variables
 - `unset($_SESSION['myvar'])`
- Destroy session
 - `session_destroy()`

Session Demo



sesstart.php

```
<?php session_start(); //Create session  
  
//Create session variable - Save user name  
$_SESSION['login'] = $_POST['login'];  
  
//Display session variable  
include('header.inc.php');  
echo '<p>Content of $_SESSION[\'login\'] is '.  
     $_SESSION['login']. "</p>";  
echo '<p><a href="ss1.php">page 1</a></p>';  
include('footer.inc.php');  
?>
```

ss1.php

```
<?php session_start(); // Use session variable  
  
include('header.inc.php');  
echo '<p>Content of $_SESSION[\'login\'] is '.  
     $_SESSION['login']. "</p>";  
echo '<p><a href="ss2.php">page 2</a></p>';  
include('footer.inc.php');  
?>
```

ss2.php – Use, Unset

```
<?php session_start();  
include('header.inc.php');  
  
// Use session variable  
echo '<p>Content of $_SESSION[\'login\'] is '.  
     $_SESSION['login']. "</p>";  
  
// Unset session variable- should not be visible  
// anymore  
unset($_SESSION['login']);  
  
echo '<p>We unset the session variable</p>';  
echo '<p><a href="ss3.php">page 3</a></p>';  
include('footer.inc.php');  
?>
```

ss3.php – Cannot Use Session Var

```
<?php session_start();  
include('header.inc.php');  
  
//Try use session variable  
if (empty($_SESSION['login']))  
    echo '<p>$_SESSION[\'login\'] is Empty </p>';  
else  
    echo '<p>$_SESSION[\'login\'] is Not Empty </p>';  
  
echo '<p>Content of $_SESSION[\'login\'] is '.  
     $_SESSION['login']. "</p>";  
include('footer.inc.php');  
  
//Destroy session  
$_SESSION = array();  
session_destroy();  
?>
```

Class Exercise

- Given: Login page to get user info (HTML)
 - action = "login.php"
 - method = "post"
 - input fields names: user and pwd
- Write PHP to implement db authentication
 - First page: check user against the information in the database – host cs-mysqlsrv.cs.usna.edu, database IT420, table Users,
 - Next pages: display only if user logged in
 - Logout page

(extra space)