Week8

Index.php

<body>

<h1>MD5 cracker</h1>

<p>This application takes an MD5 hash

of a four digit pin and check all 10,000 possible four digit pins to determine the pin</p>

<pre>

Debug Output:

**<?php**

$goodtext = "Not found";

// If there is no parameter, this code is all skipped

if ( isset($\_GET['md5']) ) {

$time\_pre = microtime(true);

$md5 = $\_GET['md5'];

// This is our alphabet

$txt = "0123456789";

$show = 15;

// Outer loop go go through the alphabet for the

// first position in our "possible" pre-hash

// text

for($i=0; $i<strlen($txt); $i++ ) {

$ch1 = $txt[$i]; // The first of two characters

// Our inner loop Note the use of new variables

// $j and $ch2

for($j=0; $j<strlen($txt); $j++ ) {

$ch2 = $txt[$j]; // Our second character

for($k=0; $k<strlen($txt); $k++ ) {

$ch3 = $txt[$k];

for($l=0; $l<strlen($txt); $l++){

$ch4 = $txt[$l];

// Concatenate the two characters together to

// form the "possible" pre-hash text

$try = $ch1.$ch2.$ch3.$ch4;

// Run the hash and then check to see if we match

$check = hash('md5', $try);

if ( $check == $md5 ) {

$goodtext = $try;

break; // Exit the inner loop

}

// Debug output until $show hits 0

if ( $show > 0 ) {

print "$check $try\n";

$show = $show - 1;

}

if($goodtext == $try){

break;

}

}

if($goodtext == $try){

break;

}

}

if($goodtext == $try) {

break;

}

}

if($goodtext == $try){

break;

}

}

// Compute ellapsed time

$time\_post = microtime(true);

print "Ellapsed time: ";

print $time\_post-$time\_pre;

print "\n";

}

**?>**

</pre>

<!-- Use the very short syntax and call htmlentities() -->

<p>Original Text: **<?=** htmlentities($goodtext); **?>**</p>

<form>

<input type="text" name="md5" size="60" />

<input type="submit" value="Crack MD5"/>

</form>

<ul>

<li><a href="index.php">Reset</a></li>

<li><a href="md5.php">MD5 Encoder</a></li>

<li><a href="makecode.php">MD5 Code Maker</a></li>

<li><a

href="https://github.com/csev/wa4e/tree/master/code/crack"

target="\_blank">Source code for this application</a></li>

</ul>

</body>

</html>

Bootstrap.php

<!-- Latest compiled and minified CSS -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" integrity="sha384-1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDDdjZlpLegxhjVME1fgjWPGmkzs7" crossorigin="anonymous">

<!-- Optional theme -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap-theme.min.css" integrity="sha384-fLW2N01lMqjakBkx3l/M9EahuwpSfeNvV63J5ezn3uZzapT0u7EYsXMjQV+0En5r" crossorigin="anonymous">

<!-- Custom styles for this template -->

<link href="starter-template.css" rel="stylesheet">

Game.php

<?php

// Demand a GET parameter

if ( ! isset($\_GET['name']) || strlen($\_GET['name']) < 1 ) {

die('Name parameter missing');

}

// If the user requested logout go back to index.php

if ( isset($\_POST['logout']) ) {

header('Location: index.php');

return;

}

// Set up the values for the game...

// 0 is Rock, 1 is Paper, and 2 is Scissors

$names = array('Rock', 'Paper', 'Scissors');

$human = isset($\_POST["human"]) ? $\_POST['human']+0 : -1;

$computer = rand(0,2); // Hard code the computer to rock

// TODO: Make the computer be random

// $computer = rand(0,2);

// This function takes as its input the computer and human play

// and returns "Tie", "You Lose", "You Win" depending on play

// where "You" is the human being addressed by the computer

function check($computer, $human) {

// For now this is a rock-savant checking function

// TODO: Fix this

if ( $human == $computer ) {

return "Tie";

} else if ( $human == 1 AND $computer == 0 ){

return "You Win";

} else if ( $human == 2 AND $computer == 1 ){

return "You Win";

} else if ( $human == 0 AND $computer == 2 ){

return "You Win";

} else if ( $human == 2 and $computer == 0 ) {

return "You Lose";

} else if ( $human == 1 and $computer == 2) {

return "You Lose";

} else if ( $human == 0 and $computer == 1 ) {

return "You Lose";

}

return false;

}

// Check to see how the play happenned

$result = check($computer, $human);

?>

<!DOCTYPE html>

<html>

<head>

<title>KRISHA JOSHI (RA1911003011053) 0c96dd40</title>

<?php require\_once "bootstrap.php"; ?>

</head>

<body>

<div class="container">

<h1>Rock Paper Scissors</h1>

<?php

if ( isset($\_REQUEST['name']) ) {

echo "<p>Welcome: ";

echo htmlentities($\_REQUEST['name']);

echo "</p>\n";

}

?>

<form method="post">

<select name="human">

<option value="-1">Select</option>

<option value="0">Rock</option>

<option value="1">Paper</option>

<option value="2">Scissors</option>

<option value="3">Test</option>

</select>

<input type="submit" value="Play">

<input type="submit" name="logout" value="Logout">

</form>

<pre>

<?php

if ( $human == -1 ) {

print "Please select a strategy and press Play.\n";

} else if ( $human == 3 ) {

for($c=0;$c<3;$c++) {

for($h=0;$h<3;$h++) {

$r = check($c, $h);

print "Human=$names[$h] Computer=$names[$c] Result=$r\n";

}

}

} else {

print "Your Play=$names[$human] Computer Play=$names[$computer] Result=$result\n";

}

?>

</pre>

</div>

</body>

</html>

Login.php

<?php // Do not put any HTML above this line

if ( isset($\_POST['cancel'] ) ) {

// Redirect the browser to game.php

header("Location: index.php");

return;

}

$salt = 'XyZzy12\*\_';

$stored\_hash = '1a52e17fa899cf40fb04cfc42e6352f1'; // Pw is meow123

$failure = false; // If we have no POST data

// Check to see if we have some POST data, if we do process it

if ( isset($\_POST['who']) && isset($\_POST['pass']) ) {

if ( strlen($\_POST['who']) < 1 || strlen($\_POST['pass']) < 1 ) {

$failure = "User name and password are required";

} else {

$check = hash('md5', $salt.$\_POST['pass']);

if ( $check == $stored\_hash ) {

// Redirect the browser to game.php

header("Location: game.php?name=".urlencode($\_POST['who']));

return;

} else {

$failure = "Incorrect password";

}

}

}

// Fall through into the View

?>

<!DOCTYPE html>

<html>

<head>

<?php require\_once "bootstrap.php"; ?>

<title>KRISHA JOSHI (RA1911003011053) 0c96dd40</title>

</head>

<body>

<div class="container">

<h1>Please Log In</h1>

<?php

// Note triple not equals and think how badly double

// not equals would work here...

if ( $failure !== false ) {

// Look closely at the use of single and double quotes

echo('<p style="color: red;">'.htmlentities($failure)."</p>\n");

}

?>

<form method="POST">

<label for="nam">User Name</label>

<input type="text" name="who" id="nam"><br/>

<label for="id\_1723">Password</label>

<input type="text" name="pass" id="id\_1723"><br/>

<input type="submit" value="Log In">

<input type="submit" name="cancel" value="Cancel">

</form>

<p>

For a password hint, view source and find a password hint

in the HTML comments.

<!-- Hint: The password is the 3 character language we are learning (all lower case) followed by 123. -->

</p>

</div>

Ind.php

<?php // Do not put any HTML above this line

if ( isset($\_POST['cancel'] ) ) {

// Redirect the browser to game.php

header("Location: index.php");

return;

}

$salt = 'XyZzy12\*\_';

$stored\_hash = '1a52e17fa899cf40fb04cfc42e6352f1'; // Pw is php123

$failure = false; // If we have no POST data

// Check to see if we have some POST data, if we do process it

if ( isset($\_POST['who']) && isset($\_POST['pass']) ) {

if ( strlen($\_POST['who']) < 1 || strlen($\_POST['pass']) < 1 ) {

$failure = "User name and password are required";

} else {

$check = hash('md5', $salt.$\_POST['pass']);

if ( $check == $stored\_hash ) {

// Redirect the browser to game.php

header("Location: game.php?name=".urlencode($\_POST['who']));

return;

} else {

$failure = "Incorrect password";

}

}

}

// Fall through into the View

?>

<!DOCTYPE html>

<html>

<head>

<?php require\_once "bootstrap.php"; ?>

<title>KRISHA JOSHI (RA1911003011053) 0c96dd40</title>

</head>

<body>

<div class="container">

<h1>Please Log In</h1>

<?php

// Note triple not equals and think how badly double

// not equals would work here...

if ( $failure !== false ) {

// Look closely at the use of single and double quotes

echo('<p style="color: red;">'.htmlentities($failure)."</p>\n");

}

?>

<form method="POST">

<label for="nam">User Name</label>

<input type="text" name="who" id="nam"><br/>

<label for="id\_1723">Password</label>

<input type="text" name="pass" id="id\_1723"><br/>

<input type="submit" value="Log In">

<input type="submit" name="cancel" value="Cancel">

</form>

<p>

For a password hint, view source and find a password hint

in the HTML comments.

<!-- Hint: The password is the four character sound a cat

makes (all lower case) followed by 123. -->

</p>

</div>

</body>