SSDP :: PHP LAB 07

OBJECTIVE   Write a secure user registration and login web application with PHP and MySQL.

REQUIREMENTS   A sample of this lab is available at http://bcitcomp.ca/ssd/lab07_sample/login.php

Write a registration / login web application. It is up to you to decide how many pages, external files, etc are required for this lab. Read these specifications thoroughly, and test your application to ensure you have met all requirements.

Ensure you have the **users** table in your MySQL bcit database, with appropriate permissions privileges for the PHP script user. If not, obtain and import into MySQL the **users.sql** table data from session 06.

Use the **dbinfo.php** file provided with the Session07 files. Update the values assigned to **DB_USER** and **DB_PASS** using values appropriate for your MySQL server, but otherwise do NOT add any additional code to this page. Use the constants defined in this file in any of your scripts that require database interaction.

Registration Page:

Users must choose a unique username (cannot exist in the database already), and a minimum 8 character password. They must enter the password twice to confirm it has been typed correctly. If the username is not yet in the database users table, add their username and password to the table and forward them to the Login page. If there was a problem with the registration, show them the registration form again, and explain what the problem was (eg: username already exists, passwords don't match, etc).

Login Page:

Display a link to the Registration page. Also display a form for username and password login. A successful login must match a case sensitive username and password against the **users** table in the **bcit** database. A successful login should start a session to keep track of this user and forward them to page 01. If the user does not login correctly (eg: is not in the database as a registered user), display the login form again along with an appropriate error message. The form on this page should also include a 'Remember Me' checkbox option, which if checked will use a cookie to store/prepopulate the username

Security:

Apply several layers of security to this application. Protect against form spoofing by carefully validating all incoming data. Protect against SQL injection attacks with **mysqli_real_escape_string( )** OR by using stored procedures. If using stored procedures, be sure to include the CREATE PROCEDURE queries in your PHP script so that your code will be portable. Use salting and client specific data to protect the session from hijacking and session guessing. Use either md5 or sha1 to encrypt the passwords for storing in the database.

Page01:

Should only be viewable by those who have logged in. Any other attempts to view the page must be forwarded to the login page. If the user is logged in, display their username in a welcome message. Also display a link to Logout.

Logout Page:

Should log the user out and provide a link back to the login page.

NOTES   The records in the **users** table initially do not use hashed passwords. Thus once you implement that feature in your scripts these records will be unusable (unless you update their passwords, or register a new user using hashed passwords).

SUBMIT   Compress all related files and folders into a **.zip** file and upload to D2L drop box before the end of day (11:59pm).