CIS 1152 - Lab #8 PHP Sessions

S. Ruegsegger

Modified with permission by Peter Chapin

Objective

Use PHP Sessions to have password protected spaces on an app – and <u>not</u> store plain text passwords.

Task 1 – Database Setup

In this lab you will build off the course schedule database in the previous lab. The idea is to provide a page where a student can log in and then select a semester (Spring/Fall and year) to see their schedule for that semester. The application should have three main pages: a login page, a page where the student selects the semester (i.e., a form), and a third page where the student views their schedule. There should also be a "logout" button on the form page and on the schedule page. When selected, the student is logged out and a fourth page is presented with a confirmation message ("You have been logged out") and a link that returns to the login page.

You should be able to reuse the database structure from Lab #7, but you will need to create a new table that associates student names with hashed passwords. You will also need another page (so, yes, a 5th page overall), that an "administrator" can use to set a student's password. This extra page is not intended to be used by the students themselves, so don't worry if you can set any student password with the page. In fact, that is the intention. You do not need to provide a way for a student to change their password.

Consider Task 1 complete when you've added the extra table to the database, created the administrative page, and have given passwords to the students in the database. You will want to use the PHP `password_hash()` function to compute the salted, 60-byte string (type CHAR(60)).

Task 2 – Login Page with Authentication

In this task you will create the login page so that a student can login and see the form for selecting their desired semester.

<u>Requirements</u>:

- Write a login script which accepts a student name and password.
- When the user clicks "login" --- authenticate by…
    - First, retrieve the salted (60-char) password stored in the database.
    - Next, use PHP's `password_verify()` to compare the user's plain text password to the salted password from above.
    - If they match – the user is authenticated. If not, the user's login is rejected.
- If authenticated, put the user's name into $_SESSIONS.
- Display a form that lets the user select their semester of interest. Ideally this form would only contain options that are relevant for the student, but for a first approximation you can hard-code semesters and years from a reasonable range.

Task 3 – Display Schedule

In this task create a page to display the students schedule that is activated by submitting the form you created in Task 2. You will need to retrieve the student's name from the $_SESSIONS array to make the proper query to the database.