Project 7: Membership

CS 212 • 20 Points Total
Due Monday, November 7, 2016

Objectives

- Create and manage SQL tables.
- Create a membership database for your site.
- Allow users to log in and out and keep track of the activity.

Assignment

A common function is to create a membership database so that users can log on when they come to a website. This function has many purposes: special privileges, protection of sensitive data, or even maintaining some session data between visits. Whatever the purpose, the requirements of the dataset and the associated pages are tightly bound. We can't explore all the possibilities but it's a good exercise in MySQL to build a simple database.

We're going to add membership functionality to our websites by allowing users to register, storing their information, and then allowing them to login to see the rest of our site. Here are the tasks you need to accomplish:

- Design and build your SQL tables. You will need at least a table of **users** and a table to **log user activity** (signing in and out and anything else you feel is interesting).
  - The fields you use are up to you, as long as you can at least service the requirements of this assignment. You can use more tables, too.
  - You can use a GUI-based tool, a shell tool, or PHP scripts to build your tables and enter test data.
- Create a set of forms for a user to the site. The required forms are:
  - **Registration** – get the user's info, assign a unique id, and put the record in your members table. Optionally, you can log the user in without making them use a login form.
  - **Change User Info** – invariably, they want to change something. Usually it's their password, but why not let them change other things, too?
  - **Delete User** – They want to remove their info. If you simply remove their record, what will that do to your log table? You can either clean them out of the log table (probably not the best idea) or have a field in the user table that allows you to flag a user as inactive without actually deleting their info.
  - **Login** – should prompt for their id and password. If successful, a record is entered in the log table with a timestamp and some kind of status code that tells you what happened. Maybe you also want to log failed attempts (catch the guys trying to hack in, maybe?)
- For all the forms above, don't forget to create the accompanying script(s) – assuming that you aren't combining your forms with your PHP code. Don't just make the form and then have it do nothing; it should execute the proper PHP and SQL commands when it is used.

- In addition, you will want a few scripts that run on a button press or hyper-link (be sure to put the buttons or links on the site so they can be used):
    - **Logout** – logs the user off (creates a record in the logging table). Don't really need a form for that, do we?
    - **Print** or **Display Log Table** – either way (or both), just so we can see what's going on.
    - **Archive Log Table** – dump all log entries into a CSV file and then clear the table. Better still if you clear only those entries older than a given date (like archiving all records over a week old.)
    - **Optionally** you can make scripts to do other things as well, like automatically building your tables in case something removes them, or restoring data from archive.
- All scripts **must validate input** before sending it to the database.
- You will want to change the name of your current home page and create a new one that does not allow any linking unless the user first registers and/or logs in, after which they get to the original home page.
- **Optionally**, but strongly encouraged, you should create a user for yourself and some kind of special privileges that allows you to link to a sysop page that has the links to scripts only you can perform. You don't want to let just anyone view your log file.
- You should comment your code well. If something is going wrong, neither I nor the grader will spend a lot of time debugging your work to give you partial credit. Improve your chances of partial credit by good comments so we can quickly figure out your intent.
- Work together to figure things out but, as always, turn in your own code.

Project Report

The final step of this assignment is to create a report consisting of a cover page, an overview of the project, sample output, and the source code. **For your source code**, you can simply zip up your site files. Otherwise, we can't see the content of your scripts. The database should be presented as a table view, using EXPLAIN tablename, or printing the layout from a GUI-based tool. We don't care about the data in it, especially if you've created lots of records. Just the table layouts will be fine. For the rest, see Assignment Policies on either the class website or Bb Learn. Just as a reminder, the **cover page** should contain the following information. Correct the date to be the day you actually turn in the project.

Project 7: Membership

Your Name
November 7, 2016

CS 212