CS 1520 / CoE 1520: Programming Languages for Web Applications (Spring 2012)

Department of Computer Science, University of Pittsburgh

Assignment #3: MySQL

Released: March 26th, 2012 **Due:** 11:59pm, Monday, April 9th, 2012

Goal

Gain familiarity with MySQL and PHP.

Description

In this assignment, you will implement a simple crowd-sourcing application, where a user will be shown an image along with a question and a set of possible answers from which he/she is asked to pick the most appropriate one. You will use PHP and MySQL for this assignment.

Web Pages

We consider two types of users:

- **Administrator** – this is you and requires a password that is hardcoded in your program. Store the password in variable `$ADMINPASS` (so that we can find it easily during grading).
- **Authenticated user** – this a user that has registered and provided a user name and password. You should use either cookies or special hidden fields in the generated forms for authentication in subsequent accesses from the same user. You should also show the username on the top of the screen.

You are asked to implement six different PHP scripts with the following functionality:

1. **global.php** – This should be included by all your other scripts. It contains global variables that are used throughout your programs (e.g., your database server credentials and variables `$ADMINPASS` and `$XMLURL`, described below).

2. **setup.php** – This is the "admin" page for your program. You should use this page to:
   - reset the tables used by this application in your database (i.e., delete all tuples from all tables, but leave the table definitions intact), and
   - upload an XML document with the list of images, questions, and possible answers. An example of the expected document is at
     `http://db.cs.pitt.edu/courses/cs1520/spring2012/assign/a3.input.xml`
     Your script should simply download the document from a URL that is hardcoded in your program. Store this URL in variable `$XMLURL` (which we will change during grading).

     Note: You should use a PHP library to parse the XML document, such as the one used for Recitation #7. Also, you can expect the document to be well-formed and conforming to the DTD specified.

   For security, you should have a form with a single password field. After the user clicks the submit button, the submitted password is checked against a hardcoded password in your script, stored in `$ADMINPASS` and both operations (reset and upload) are executed.

3. **newuser.php** – A simple form to add new users into the database. You should collect username, password, email address, and optionally first and last name and store them in the database. You should check if the username is unique, but should not worry about any other verification.

4. **question.php** – For authenticated users: Display an image (selected at random), along with a question (random selection) and all the possible answers as a form with a multiple-choice format. Record the answer in the database after the user clicks submit. Reload the form again, with another random selection of image/question. Repeat.

   Provide a link to questionstats.php and userstats.php at the bottom of the page.

5. **questionstats.php** – For authenticated users: Show a list of all questions ordered by the number of answers provided (highest first). For each question, provide a percentage breakdown for every possible answer, as a pie-chart, using Google Image Charts:
   `http://code.google.com/apis/chart/image/docs/making_charts.html` and including the chart as an external image.

   Provide a link to question.php and userstats.php at the bottom of the page.

6. **userstats.php** – For authenticated users: Show a list of users (username, first and last name), ordered by the number of questions answered (highest first). For every user, show the total number of questions answered and their overall **accuracy**. To compute accuracy, you need to identify what is the most popular answer for every image/question combination (over all users) which we assume to be the "correct" answer. Then, for every user, compute the percentage of correct answers from all the answers he/she has provided. If the same image/question combination has been answered more than once by the user, consider all answers from the user in order to compute his/her accuracy (i.e., 2 correct answers and 2 wrong answers will lead to 50% accuracy for this image/question combination).

   Provide a link to question.php and questionstats.php at the bottom of the page.

   Note: You are allowed to use database views or temporary tables to answer this question. You should store the accuracy value for every user in the `accuracy` field in the user table.

You are also encouraged to add some form of personalization to your assignment (e.g., a header on the web page with your name and other information) and utilize CSS to simplify your HTML page design (although these components of your program will receive minimal attention during grading).

Please note that you should NOT use Javascript in your program.

Database Schema

You should use the database schema provided. You have been give access to the `crowdsourcing` database in the class MySQL server, which has all the required tables and you can simply copy to your own database. Alternatively, you can create the tables with the SQL script located at:
`http://db.cs.pitt.edu/courses/cs1520/spring2012/assign/a3.setup.sql`

What to submit

Although you will be testing your PHP code with the PHP and MySQL setup provided, you must submit all the required files so that your program can be executed at a different web server. This includes the PHP scripts specified above, any additional PHP scripts/library files that you have used, and, if needed, any additional SQL definitions (beyond this in the `a3.setup.sql` file) that are needed by your program (e.g., view definitions), as a separate SQL file called `extra.sql`.

Make sure that all paths are **relative**. Submit all files as a single zip file.

Academic Honesty

The work in this assignment is to be done *independently*, by you and only you. Discussions with other students on the assignment should be limited to understanding the statement of the problem. **Cheating in any way, including giving your work to someone else, will result in an F for the course and a report to the appropriate University authority for further disciplinary action.**

How to submit your assignment

We will use a Web-based assignment submission interface. To submit your assignment:

- Go to the class web page `http://db.cs.pitt.edu/courses/cs1520/spring2012` and click the Submit button.
- Use your pittID as the username and the password you specified at the contact information form for authentication. There is a reminder service via email if you forgot your password. You must have already submitted your contact information, if you have not yet you need to do so now.
- Upload your assignment file to the appropriate assignment (from the drop-down list).
- Check (through the web interface) to verify what is the file size that has been uploaded and make sure it has been submitted in full. **It is your responsibility to make sure the assignment was properly submitted.**

You must submit your assignment before the due date (11:59pm, Monday, April 9th, 2012) to avoid getting any late penalty. The timestamp of the electronic submission will determine if you have met the deadline. There will be no late submissions allowed after 11:59pm, Wednesday, April 11th, 2012.

[Last updated on March 26, 2012 at 12:43am EST]