CSci 4131: A Internet Programming - Fall 2015
Assignment 5: CGI Programming and Database Connection
Due Date: November 6, 2015
This assignment MUST be done individually

Goals:

In this assignment you will learn how to connect to a MySQL database from a Python CGI program. You will use this database to store user-password information. You will also learn how to use cookies to establish a user-session so a user does not have to enter password for every interaction with your CGI program.

Problem Statement:

You will be building upon the work that you did in Assignment 4. in which you created a web application to upload photos and view/upload/delete them later. This assignment will support all those functionalities that you implemented in that assignment. In addition to those, you will include functionalities for authenticating users, and restricting operations based on the user roles.

Functionality:

You are required to support the following functionality.

Users Roles:

This application will support users in two different roles: Owner and Visitor. Every user will have a unique User-Name, Password, and one of the two roles listed above. There will be only one user in the Owner role. This information about the users will be stored in the MySQL database.

· A user in the **Visitor role** should be able to **only** view the photos. No upload or delete operation will be allowed for the users in the visitor role.

· The user in the **Owner role** will be able to add or delete users to the Visitor roles, and change their passwords. Owner is also allowed to use the upload/edit or delete photo operation from Assignment 4.

Design Requirements:

You will maintain the user-name, role, and password information in your MySQL database.

1. We suggest that you ** DO NOT ** use your CSELab/CS/X.500 User Name and Password

You will need to use HTTP Cookies to track the currently logged in user.

- Cookies will enable you to provide user specific information without requiring the user to enter User Name and Password for each interaction.
- Cookies will be given a max-age based on the period (in seconds) specified in a configuration file, as described below.
- Cookies are important part of web programming, and many languages including PHP and Python have Session Handling routines to make life easier. Session handling means that only a key (can be a random number) is stored on the client browser inside the cookie, and on the server user-specific variables will be stored inside a database or inside separate files.
- Python has several modules that deal with Session Handling jobs, and PHP has the built-in $_SESSION global variable. Since we don’t want to make this assignment a hard one, we are going

to stick to the basics in regards to cookies. You are allowed to store variables inside cookies, like user id, to indicate a user has logged in. Please look at the class notes on how to store information inside client-side cookies and how to retrieve them via Python CGI module.

Use the following SQL table structure to store the user's data. **Please do NOT deviate from this**.

CREATE TABLE Users (userID INTEGER AUTO_INCREMENT, Name CHAR(20), Role CHAR(12), Password CHAR(20), PRIMARY KEY(userID));

Roles in the database should only have two fixed values as: "Owner" and "Visitor".

Configuration File:
Your Python CGI program will read some configuration parameters from a file named **config**. This file will be stored in the same directory where your CGI program is stored. This file will contain the following three-lines of information related to
(1) your MySQL account name,
(2) password, and
(3) cookie expiration time in seconds.

An **example** structure of this file is shown below. You will use your config file with your user-name and password.

```
--------------------config file ---------------------------
MySQLuserID: C4131S14U123
MySQLpassword: 6789
expirationTime: 60
-------------------end of config file ---------------------
```

Your Python/CGI program will read this file to get the appropriate information to connect to the MySQL server.

Menu:
Given below is a description of the various menu options that you need to define as part of various forms.

Login Menu:
Input: User Name and Password

- If the user passes authentication check, then you will display one of the following:
  Owner menu, Visitor menu.
  You can use any User Name and Password for the Owner.
- Remember that you need to change your gallery.cgi, edit.cgi, delete.cgi, and upload.cgi to redirect user back to the login page, if the user is not logged in yet.

Visitor Menu:

Show gallery.cgi page without OR un-operational EDIT, DELETE, or UPLOAD buttons. Visitors do not have access to these pages, and even if the user manually enters these pages, like upload.cgi, into the browser address bar, your program should check their permission, and if they are not the owner, they should be automatically redirected back to gallery.cgi page.

Owner Menu:
Should support the following menu functions:

- Add User: Take the following items as input: User Name, Password. (All new users will be added only for the Visitor role.)
- Delete User: Input will be User Name (only a user with a Visitor role will be deleted)
- Link to gallery.cgi page (with operational UPLOAD, EDIT and DELETE buttons, but it is okay if the functionality of these buttons does not work for this assignment.)

PLEASE NOTE THE FOLLOWING REQUIREMENTS:

- The login form document will be the first page to be presented the user. This form will ask for user-name and password. **Please name this file as login.html**
- Please make sure that your code **does not contain any absolute file path-names**. All files referred to in your **code must be using relative path-names**.
- This assignment is not testing the functionality of Assignment 4 i.e. upload/edit/delete pictures. But the links to these functionalities on the Gallery page should be working.

Grading Criteria:

- 15% for redirecting from any page to Login page if the user is not yet logged in.
- 30% Correct implementation of functionality in the Visitor role (not allowed to upload, edit or delete a photo, only allowed to view photos)
  - 15% Disable/Hide Upload, Delete, and Edit Buttons on Gallery page
  - 5% Redirect from Upload page to Gallery Page
  - 5% Redirect from Edit page to Gallery Page
  - 5% Redirect from Delete page to Gallery Page
- 50% Correct functionality in the Owner role
  - 10% Correctly add a user in Visitor role
  - 10% Change password for a user
  - 10% Correctly change a user role
  - 10% Remove a user
  - 10% Correctly showing Upload, Delete, and Edit buttons on Gallery page.
- 5% Commenting the code and code readability

Submission Instructions:
Include the following files in your submission: Python CGI files and HTML files, a README file, and your config file so if needed we may connect to your database on CSELab server. In the README indicate the Student Name and ID.
Submit the tarred directory containing all these files.