****

CS234 - Database and Web Development

Semester project

****

Deliverable

You must upload to Moodle a single .zip archive file of your site folder. Use a utility that produces .zip files (Windows: right click | Send to | compress, Mac: right click | compress).

Objectives

To design, code and publish a database driven web site of your choosing.

Project setup

- Using VS Code, create the folder project. This folder is referred to from here on out as the <site> folder. This is the folder you will .zip and upload to Moodle (classes.cs.siue.edu).

Project Requirements

The web site you are going to develop over the next one month. The site should be designed and coded with the following set of minimum requirements in mind.

MySQL Database

- The site must be database driven and as such you should use MySQL. You should export your final database into the file project.sql and add it to your site folder for delivery.
- The database should be named **project** and should include a **registration** table to store registered users of your site.
- Include at least two more tables in a one-to-many relationship.
- You must provide full admin CRUD functionality of your site, i.e. the admin should be able to - add a record, retrieve a record, update a record and delete a record. You may offer some or all of the CRUD functionality to a registered user as well.

Login form

- You must include a **login** form to allow a registered user to log in. This is the very first page any user to your site must be allowed to land on.
- You must ask for the username and password.
- An attempt to visit any other page by a user not logged in should be redirected to the login page. Once a registered user is logged in, they are free to visit any other page on the site.

Register form

- You must include a **registe**r form to allow a user to register before they log in.
- As a minimum, ask for the username and password.
- If the username is already taken by another registered user, inform the user of such and allow them to enter a unique username.

Home page

- The home page is the first page a registered user sees once they successfully log in. Do not allow a user to access this page if they have not logged in.

Other pages

- Design and code any other page as you see fit.
- Do not allow the user to access any page if they have not logged in. If a user attempts to land on any such page, redirect them to the login page.

Server side scripting

- You must use PHP to handle server side scripting.
- Use PHP PDO to access the database and to execute all needed queries.
- Validate the forms (login, register) making use of regular expressions where necessary.
- Authenticate a registered user trying to log in. This means verify the user credentials given are the same as they are in DB for that user.
- All passwords saved in the DB must be hashed. Use bcrypt to create the hash digest.
- When registering a user, make sure that their username is unique, i.e. the username given during registration is not already taken by another registered user.
- Use sessions to prevent a user from accessing any page directly if they have not logged in first

Presentation

- You can use the W3.CSS framework to style your site. Please do not use any other framework to style. You may use your own CSS style rules when/if overwriting/customizing a W3.CSS rule you don't like.

Grading Rubric	
Deliverable	**100%**
MySQL database	25%
Login form	5%
Register form	5%
Home page	10%
Server side scripting:	
PDO	10%
Validation	5%
Authentication	10%
Password hash	5%
Username unique	5%
Use of sessions	10%
W3.CSS	10%