CSCI 4000 Assignment 4

Important: For this assignment, you must use **PDO** (PHP Data Object) to interface with the MySQL database. If you do not use PDO, you will receive 0 points for Q2, 3, 4 & 5.

Q1: **SQL Script and Database** [20 points] –MySQL, phpMyAdmin
Create a text file `<my name>Assignment.sql` with SQL statements. These SQL statements should perform the following actions on your MySQL database server (when imported via phpMyAdmin).
- Create a MySQL database `<my name>StudentDB`. Drop the database if it already exists.
- Create a database table `<my name>StudentGPA` in the created database.
- Create the following fields (columns) for the created table: `studentID, name, gender, major, GPA`. Choose the appropriate date types for the fields. Make `studentID` the primary key, and auto_increment. `GPA` needs to have 2 decimal places.
- Insert 10 rows of data into the database table. Make the data as realistic as possible. (e.g. points will be deducted if you call a student "123" or "donut" etc)
- Create a MySQL database username "`<my name>gpa_user`" with password "`lifeisgreat`", with privileges "select, insert, update, delete" for `<my name>StudentDB` database only.
- Do screen captures of the database structure, table structure and privileges for submission. (You may use phpMyAdmin or any other tool.)

Important: If you do not put `<my name>` to the above mentioned fields (database related field, page title and filename), **you will get 0 point for the question.**

Estimated time: 2 hours

Q2: **Database Connection and Error** [20 points] - PHP, MySQL, phpMyAdmin
Write a PHP program `<my name>database.php` that
- Use PDO (PHP Data Object) to interface with your MySQL database.
- Connect to the `<my name>StudentDB` MySQL database, using the MySQL database username "`<my name>gpa_user`" and password "`lifeisgreat`".
- Use `try {} catch () {}` exception handling to detect if the connection is successful.
- If the connection is not successful, show an error page `<my name>database_error.php`.
- Show the error message generated by PDOException on `<my name>database_error.php`.
- Use an external *.css file to format `<my name>database_error.php`. Create your own *.css file. Do NOT use the provided *.css file in D2L. No two students should use the same *.css file (**both will get 0 for this question**).

Important: If you do not put `<my name>` to the above mentioned fields (database related field, page title and filename), **you will get 0 point for the question.**

Estimated time: 1 hour

Q3: **Display** [20 points] - PHP, MySQL, phpMyAdmin

Write a PHP web page (program) `<my name>index.php` that

- Use `<my name>database.php` file (created in Q2) to connect to your MySQL database (hint: `require_once()`). Do not recreate PDO object. Use what you created in Q2.
- Give it page title `<my name> Index`.
- Display `<my name>`'s Student GPA Page as heading (e.g. h1 tag).
- Retrieve all the data from the `<my name>StudentGPA` database table (10 rows inserted in Q1).
- Use html table (or css) to format the output table (into rows and columns).
- Use an external *.css file to format the .php file. Create your own *.css file. Do NOT use the provided *.css file in D2L. No two students should use the same *.css file (**both will get 0 for this question**).

Important: If you do not put `<my name>` to the above mentioned fields (database related field, page title and filename), **you will get 0 point for the question.**

Estimated time: 2 hours

Q4: **Delete** [20 points] - PHP, MySQL, phpMyAdmin

Update the PHP web page (program) `<my name>index.php` (from Q3)

- Display a "delete" link or button next to each data item (each row of data).
- When the "delete" link or button is clicked, the corresponding row of data should be removed from the `<my name>StudentGPA` database table.
- You would need to create an additional *.php file `<my name>delete.php` to do the deletion.
- Use `<my name>database.php` file (created in Q2) to connect to your MySQL database (hint: `require_once()`). Do not recreate PDO object. Use what you created in Q2.
- Automatically refresh the `<my name>index.php` page. The deleted data item (row of data) should disappear.

Important: If you do not put `<my name>` to the above mentioned fields (database related field, page title and filename), **you will get 0 point for the question.**

Estimated time: 2 hours

Q5: **Add** [20 points] - PHP, MySQL, phpMyAdmin

Add a link "Add Student" to `<my name>index.php` (created in Q3) that connect to `<my name>add_form.php`.

- Give `<my name>add_form.php` a page title: `<my name> Add Form`.
- `<my name>add_form.php` contains a simple web form.
- In the form, allow user to add new data items: `name, gender, major, GPA`.
- Use html table (or css) to format the input fields (into rows and columns).
- Submit the form to another php file `<my name>add.php`. It should use the $_POST array to get all the form fields/inputs. Add submitted data to the `<my name>StudentGPA` database table.
- Use `<my name>database.php` file (created in Q2) to connect to your MySQL database (hint: `require_once()`). Do not recreate PDO object. Use what you created in Q2.

- Automatically refresh (or show user) the `<my name>index.php` page. The new data item (row of data) should appear.
- Use an external *.css file to format `<my name>add_form.php`. Create your own *.css file. Do NOT use the provided *.css file in D2L. No two students should use the same *.css file (**both will get 0 for this question**).

Important: If you do not put `<my name>` to the above mentioned fields (database related field, page title and filename), **you will get 0 point for the question.**

Estimated time: 2 hours

Submission instructions:
You need to test the above programs (questions) separately, and provide **two test cases** (if applicable) for each program (question). Do a screen capture of the input and related output for each test case. Use any graphic editing software (e.g. Microsoft Paint, Adobe Fireworks, GIMP) to cut out the program input and output (from the screen capture), paste them into a word document under a related question number, save the document as a pdf file.
You only need to do **one set** of screen captures of the **database structure, table structure and privileges**.

You need to submit the following:
1. A pdf file containing the screen captures of program input and output of all test cases, name the file **lastname_firstname_assignment04.pdf**.
2. All program files. Zip your files into a single zip file (or rar file) **lastname_firstname_assignment04.zip**. The zip file should contain:
- `<my name>Assignment.sql`
- `<my name>database.php`
- `<my name>database_error.php`
- `<my name>index.php`
- `<my name>delete.php`
- `<my name>add_form.php`
- `<my name>add.php`
- related *.css file
- any other files you may need

Please submit electronic copy (the above mentioned **two files**) to D2L digital dropbox.

Grading guidelines (programming questions):
Your programs will be judged on several criteria, which are shown below.
- Correctness (50%): Does the program compile (run) correctly? Does the program do what it's supposed to do?
- Design (20%): Are operations broken down in a reasonable way (e.g. classes and methods)?
- Style (10%): Is the program **indented** properly? Do variables have **meaningful names**?
- Robustness (10%): Does the program handle erroneous or unexpected input gracefully?
- Documentation (10%): Do all program files begin with a **comment** that identifies the author, the course code, and the program date? Are all the classes, methods and data fields clearly **documented (comments)?** Are unclear parts of code **documented (comments)?** (Some items mentioned may not apply to some languages)

A program that does not compile (run) will get at most **50% of the possible points**.