CSCI 4000 Assignment 5

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.

You need to use the database created in Assignment 4, Q1&2. So if you have completed Assignment 4, you do not need to do Q1&2 below. They are the same as Assignment 4, Q1&2.

Q1: **SQL Script and Database** [5 points] –MySQL, phpMyAdmin – Same as Assignment 4, Q1.
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: 0 hours (same as Assignment 4, Q1)

Q2: **Database Connection and Error** [5 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: 0 hours (same as Assignment 4, Q2)

Q3: **AJAX Search and Display** [45 points] – JavaScript, AJAX, PHP, MySQL, phpMyAdmin
Write a web page (program) `<my name>ajaxsearch.htm` that
- Display **<my name>'s Student GPA Search Page** as heading (e.g. h1 tag)
- Give it page title `<my name> Q3 AJAX Search and Display.`
- Ask user to enter a `GPA` number (textbox).
- Create a button with value "Search for students with higher GPA".
- Create a div tag on the page displaying "Search results will be displayed here".
- Create an `XMLHttpRequest` object in a JavaScript function, which is triggered by user clicking the button.
- The `XMLHttpRequest` object will use a php file `<my name>ajaxsearch.php` to search the database.
- `<my name>ajaxsearch.php` will retrieve all the data from the `<my name>StudentGPA` database table, with `GPA` value higher than the user input (`GPA` number). 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.
- Use html table (or css) to format the output table (into rows and columns).
- The `XMLHttpRequest` object will use the div tag to display all retrieved data (without reloading `<my name>ajaxsearch.htm`).
- Use an external *.css file to format the .htm 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: **AJAX Search and Display** [45 points] - JavaScript, AJAX, PHP, MySQL, phpMyAdmin
Write a PHP web page (program) `<my name>ajaxsearch_single.htm` that
- Display **<my name>'s Student ID Search Page** as heading (e.g. h1 tag)
- Give it page title `<my name> Q4 AJAX Search and Display.`
- Ask user to enter a `studentID` number (textbox).
- Create a button with value "Search for a student with the exact student ID".
- Create four read-only textboxes for outputting `name, gender, major, GPA.` (or you can use any suitable html tag)
- Create an `XMLHttpRequest` object in a JavaScript function, which is triggered by user clicking the button.
- The `XMLHttpRequest` object will use a php file `<my name>ajaxsearch_single.php` to search the database.
- `<my name>ajaxsearch_single.php` will retrieve a row from the `<my name>StudentGPA` database table, with the `studentID` value matching the user input (`studentID`). 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.

- The XMLHttpRequest object will use the four textboxes to display all retrieved data (without reloading <my name>ajaxsearch.htm).
- Use an external *.css file to format the .htm 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

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_assignment05.pdf**.
2. All program files. Zip your files into a single zip file (or rar file) **lastname_firstname_assignment05.zip**. The zip file should contain:
- <my name>Assignment.sql (same as Assignment 4)
- <my name>database.php (same as Assignment 4)
- <my name>database_error.php (same as Assignment 4)
- <my name>ajaxsearch.htm
- <my name>ajaxsearch.php
- <my name>ajaxsearch_single.htm
- <my name>ajaxsearch_single.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**.