IDG2003 : Back-end Web Development 1

2020 Autumn

Assignment I

Submission Deadline: 1 November 2020

(Submit a version before 11 October for feedback)

Luvin M Ragoo

September 20, 2020

1 Objectives

Students should demonstrate that they know and able to use:

- basic PHP constructs such as expressions and loops,
- functions, arrays, and objects,
- file management
- time manipulation(optional),

and create a basic dynamic website using these knowledge and skills.

2 Description

In this assignment, you are expected to create a part of a course management system for a university managing data about *students, courses, instructors,* and *grades.*

You need to write *three php files*.

The first php file is named 'students.php', and when called it should first display the total number of unique students. Then a table for each student should display student number, name, surname, birthdate, total number of courses completed, total number of courses failed, and GPA (grade point average), and status for each student. Students should be sorted with respect to GPA in descending order.

The second php file is named 'courses.php', and when called it should first display the number of unique courses. Then a table for each course should present course code, course name, course year, course semester, instructor name, number of credits, number of students registered, number of students passed, number of students failed, and average grade taken. Courses should be sorted with respect to number of students in ascending order.

The third php file is named 'data.php' and should allow you to upload new data from a file. Each line in the input file should include: student number, student name, student surname, student birthdate, course code, course year, course semester, instructor name, number of credits, and grade.

The GPA of a student equals to sum(course_credit x grade) / sum(credits_taken). A student can get grades from A (5) to F (0), and max GPA can be 5. Status info should be 'unsatisfactory' if GPA is between 0-1.99, 'satisfactory' if GPA is between 2-2.99, 'honour' if GPA is between 3-3.99, and high honour if GPA is between 4-5. Each course could have either 5, 7 or 10 credits and a semester could be 'Fall', 'Spring' and 'Summer'.

You need to *design a set of text files*, e.g., in coma-separated values (CSV) format, to store your data (e.g., studentsDatabase.csv, coursesDatabase.csv etc.). You should avoid unnecessarily repeating data (i.e., your text database should be normalised). The order and structure of descriptions given in this text does not necessarily reflect how you should organise your text files and code.

A user can upload data multiple times; therefore, you should check the validity of data and display an error message for example against duplicate records or student/course/instructor info mismatch. An example might be that you have two students registered to the same course at the same year; however, the course titles are different, or another example is when a student has different birthdate in two different input records.

3 OTHER REQUIREMENTS

- The use of classes, objects, and arrays are mandatory.
- You should read from your files to objects, whenever possible.

- Look and feel of the pages is under your discretion.
- Your code should work without assuming any fixed number of records.
- Your code should be well indented and commented.
- Your code should use proper variable names for readability.
- Prepare an example input data set for testing.
- Your class declarations should be in a separate file.
- Create a sensible folder structure/naming pattern for your files (e.g., data, classes etc.).
- Provide a 'readme.txt' file explaining how to use your application.
- Zip all your code into a single zip file.
- All date fields should be stored as unix timestamps.
- This assignment has to be done individually!

4 HINTS

- In order to update a file, consider using *file_get_contents*, *file_put_contents*, and *str_replace functions*.
- You should consider to implement a small library of functions to read and write files (should just meet your needs and not be an exhaustive library of course!).

Good luck!