Project #3 (Grades Submission) CMSC389N
Due Mon Jun 15 11:59 pm Summer 2015

Objectives

- To practice file reading and sessions.

Overview

For this project you will write an application that allows teachers submit grades for a students in a course. The application will read a text file (available in the web server) with the students' names and will display a form where a teacher can assign letter grades.

You need to implement your code in a folder named **GradesSubmission.** We should be able to execute your application by executing a script named **main.php** (which resides in the GradesSubmission folder). You can define as many php script files you understand you need.

Grading

- (10 pts) Correct response when invalid log in information provided.
- (10 pts) Once logged in we don't have log in again.
- (20 pts) Form with radio buttons to enter grades generated.
- (40 pts) Going back and correcting grades finds old grades.
- (20 pts) Table with grades to submit.

Requirements

- Your code must be object-oriented
- You must submit forms using the post method.
- You must use sessions.
- Once a person logs in, the person does not have to log in again in order to submit grades. If the person closes the browser, the person must log in again.
- Login id and password are "cmsc298s" and "terps".
- You can assume the text file with students' names is available under the GradesSubmission directory.
- Your application should handle text files named as follows:

  ```
  cmsc<COURSEID><COURSESECTION>.txt
  ```

  That means that if a teacher wants to enter grades for cmsc128, section 0101, you can expect the file with the students' names to be called **cmsc1280101.txt**

- **You can only read the students' names file once.**
- **At least one of your scripts must be a self-referencing one.**
- You can assume a teacher will enter the full course name (e.g., cmsc128 rather than just 128).
- You can assume names will always have the format <LASTNAME>_<FIRSTNAME>.
- Letter grades will be A,B,C,D and F.
- There is no default grade for a student.
- Make sure you remove any leading or trailing spaces from text fields.
- You should define at least two functions in your application (in any file).
- The formatting associated with the forms and web pages, and the confirmation message should resemble the sample output format.
- Your application must have the same set of page transitions we have in the sample output.
- At the end of the sample output below, you will see that we are keeping track of the number of submissions made (e.g., "This is submission #1".) You don't need to implement this functionality.
- Please do not add any extra functionality to the application.
- You do not need to provide comments.
- You cannot use form hidden fields.
- You can assume the user has cookies enabled.
- You do not need to validate your html forms.
- Do not post your project on the web.
- **Make sure you trim the name read from the file.**
- As with all programming assignments:
    - You must use meaningful variable names.
    - You must use good indentation.
- You must implement this project by yourself.
- Make sure you view your code using Chrome (that is the environment we will use to grade your project).
- At the end of the video you will see a message indicating mail has been sent. You don't need to generate/sent any e-mail for this project.

Sample

The link Application Video illustrates the functionality of the application you will implement. Remember that we will provide different values through the form(s) and not just the ones provided in this example. The text file used for this example is cmsc1280101.txt.

Submission

To submit your project, follow these steps:

1. Create a zip file that includes your files.
2. Upload the zip file you created in the previous step using the submit server available at: https://submit.cs.umd.edu/summer2015. Make sure you select the submit server entry (Project #3) that corresponds to this project.
3. After submitting your project, make sure you download from the submit server the submitted file and verify that what you have submitted is correct. **Important: you must uncompress (extract) what you download from the submit server. We will not accept projects sent**

via e-mail.

Academic Integrity

Please make sure you read the academic integrity section of the syllabus so you understand what is permissible in our programming projects. We want to remind you that we check your project against other students' projects and any case of academic dishonesty will be referred to the University's Office of Student Conduct