CS 353 Fall 2018
Programming Assignment
(Homework 4)

Due: 30 November 2018 till 23:59

In this assignment, you will first write a simple Java application to connect to a MySQL database and modify it. Next, you are required to design a basic internet application using PHP on top of this database. In what follows, we explain each of these stages in detail.

Part 1: Connecting to the database with a Java application

For this part of the assignment, you will first connect to your database using MySQL console, and then write a simple Java application to connect and interact with this database.

- *Connecting to the database*: Connect to the dijkstra.ug.bcc.bilkent.edu.tr machine, which runs MySQL server (Use **SSH** or PUTTY as dijkstra allows only "secure" shell connections but not telnet!). Next, in the shell prompt, execute the command

    shell> mysql --user=USERNAME --pass=PASSWORD;

    To connect to the dijkstra server and MySQL, you need a username and password, which should be e-mailed to you by now. This MySQL account should contain a database whose name is the same as your username. For instance, if your username is "ahmetc" then you should have a database with the name ahmetc. Check your databases with 'show databases;' command in mysql. For the user "ahmetc" the output is as follows:

    mysql> show databases;
    +--------------------+
    | Database |
    +--------------------+
    | information_schema |
    | ahmetc |
    | test |
    +--------------------+

    If you don't have a database named as your username create it with the command: 'create database your_user_name;' (create database ahmetc; for the above example)
    Contact to the TA immediately, if you encounter any problem connecting to your database.

Next, write a Java program that does the following:

- *Connection*: Successfully connect to your MySQL database using Java and your MySQL account information. You can easily find the JDBC driver for MySQL on the Web.

- *Table creation:* Create the following relations in your database. **REMARK:** We want to set the foreign key relationships, and in MySQL this is supported with **InnoDB** tables. Do NOT forget to set table type to InnoDB and set foreign keys for each table appropriately. (Again, see MySQL reference manual for details).

  student(*sid*: CHAR(12), *sname*: VARCHAR(50), *bdate*: DATE, *telno*: CHAR(10), *scity*: VARCHAR(20), *year*: CHAR(20), *gpa*: FLOAT)

  company(*cid*: CHAR(8), *cname*: VARCHAR(20), *quota*: INT)

  apply(*sid*: CHAR(12), *cid*: CHAR(8))

- *Table population:* Insert into the newly created relations the following records.

student						
Sid	**sname**	**bdate**	**telno**	**scity**	**year**	**gpa**
21000001	Ayse	10.05.1995	5321113333	Ankara	senior	2,75
21000002	Ali	12.09.1997	5355361234	Istanbul	junior	3,44
21000003	Veli	25.10.1998	5553214455	Istanbul	freshman	2,36
21000004	John	15.01.1999	5335336622	Chicago	freshman	2,55

company		
cid	**cname**	**quota**
C101	tubitak	2
C102	aselsan	5
C103	havelsan	3
C104	microsoft	5
C105	merkez bankasi	3
C106	tai	4
C107	milsoft	2

apply	
sid	**cid**
21000001	C101
21000001	C102
21000001	C103
21000002	C101
21000002	C105
21000003	C104
21000003	C105
21000004	C107

- Execute the command "SELECT * FROM student" and print the results on screen.

- While doing the above tasks, give meaningful error messages when necessary. For instance, login information (e.g., password) may be wrong, etc.).

- Recall that your program may be executed several times during your code development and during the grading. So, you should check whether the database and/or tables actually exist before trying to create them to prevent MySQL errors. If they exist, first drop them and then re-create them.

Part 2: A simple Web based application using PHP

For this part of the assignment, you are required to design an internet application on top of the internship database you created above. In particular, you will design and implement a website that involves the middle and presentation tiers of a web application and makes use of the internship database at the data management tier. The website should include the following pages:

1. *Index page:* All users start at a common log on page, where each user is expected to enter his/her login and password. We assume that student names serve as logins and student id's serve as passwords (for instance, student "Ayse" can login by entering "ayse" and "21000001" as her login and password). Note that, login is case-insensitive. Give an appropriate error message if the log-on operation fails. In addition, give an error if one or both of the input fields left blank and "login" button is clicked (i.e., use a simple Java Script code to check).

2. *Student welcome page:* In this homework, we consider a summer internship application system. In the welcome page which shows all the companies (cid, cname and quota fields) in which student has applied for internships. Students can apply **up to 3** companies**.** Next to each of these internship applications, display a link, namely "Cancel", so that the student can cancel this application. When the student clicks on the "Cancel" link, display either an "error message" or a "successful deletion" message, and allow the student to return to student welcome page, again (which, of course, doesn't display the tuple(s) deleted from the apply table).

At the bottom of this page, also provide a link called "apply for new internship". If student has already applied for 3 companies, give an appropriate error message (either at a new page or a dialog box) when (s)he clicks on this link. Otherwise, when this link is clicked, open a new page.

At this page, show ids and names of those *candidate* companies that are not applied by this particular student (You should not show the company if all of its internship quota is filled.). Also provide an input field and a "submit" button, so that the student can choose to apply in one of these displayed companies (e.g., by typing the company id and clicking on the submit button). Again, show an appropriate message after the addition and allow user to return to the previous page. At every page, remember to put a link to an appropriate previous page, so that the student can go back without doing any modifications. Also, you may need to keep track of the current student id through pages, as well.

Finally, a logout link in all appropriate pages will be useful.

IMPLEMENTATION You should use PHP to implement the application logic (except the Java Script codes used to check blank form fields at the client side). Your application should connect to your MySQL database prepared in the first part of this assignment.

Test your web pages under your public html directory (at *dijkstra*) in a folder called *surname_name* (if the public_html directory does not exist, create it at your home directory and set its r-w-x permissions appropriately). You will copy and submit this folder as described below, and we will copy it under our own public_html and execute there. To allow this, please use *relative links* in your web pages (i.e., if you include links to an absolute page like http:// …/~ahmet/can_ahmet /index.php, it would not work under our public html folder. If this happens**, you won't get any grade** from this part of the assignment).

What to submit?

You will submit a WinRAR file including two folders, part1 and part2. Folders must be **<u>exactly</u>** named as *surname_name_p1* and *surname_name_p2 (please do not use Turkish characters)*. The RAR file must be named as *surname_name*. Each folder should include the following:

- part1 folder: In this folder, provide the Java source code file(s).
- part2 folder: This folder should include all your PHP files. We will copy them to our public_html and then execute there.

For both parts, grading will be based on the **execution** of your code.

Where to submit?

You will use the Moodle course page for submission of this assignment. If you encounter a problem during submission, please send your assignment by email to your TAs.