CS275 Web and Mobile App Development

Assignment 5

As with all prior assignments, you must work on this assignment individually.

Objective

The purpose of Assignment 5 is three-fold:

1. Get experience with MySQL databases, including how to install, set up a database, create and populate relational tables through the MySQL command line interface.

2. Interface with a MySQL database through a Nodejs server.

3. Gain experience in designing an application/website. As such, this assignment description will not be as explicit as earlier assignments regarding design items such as the presentation of the web page, flow of data between client and server and the organization of the web service code. It will be your responsibility to define the overall application design. Keep in mind your toolset: JavaScript, JQuery, JSON, AJAX, Serving Static Pages, Server Endpoints, Modules, Emitters, etc.…

Assignment 5 Requirements

In this assignment, we will create a MySQL database application with the following functionality/specs:

1. Your app must include a web service to:
   - Serve static web pages as needed
   - Interact with the MySQL database
   - Process the required features described below

2. The following database (3 table) schema will be created and utilized to enable interrogation of a university /student course grade repository:
   A. STUDENT reference table
      - Student id (key)
      - First name
      - Last name
      - Date of birth
      - Major of study (CS, IS, IT, CE, etc)

   B. COURSE table
      - Course id (key) – e.g. cs275
      - Course description – e.g. Web and Mobil app Development

   C. GRADES table
      - Id column (key auto_increment)
      - Course id (the key from the course table)
      - Student id (the key from the student table)
      - Term / year taken (part of key) – e.g. Fall16, Winter17, Spring17, Summer17
      - Grade (A, B, C, D, F)
      - We want the combination of (course id, student id, term) to be UNIQUE in this table so after creating the table we can add a constraint similar to:

        ```
        ALTER TABLE GRADES ADD UNIQUE(courseid, studentid, term);
        ```

3. The application should be able to conduct the following activities:
   A. *Display a table* - User will select one of the 3 tables above from a list and the application query the database and display that table in the web page (all rows and columns.

      Note: for all select options described in these specs, it is OK to "hard code" the choices in your "dropdown" list based on what you know is in your database (STUDENT, COURSE and GRADES would be the set of choices for this database).

   B. *Student "transcript" search* - User will select a student from a "dropdown" list (as stated above, OK to "hard code" the list of students in the database) plus select one term /year (also via dropdown box), and a report containing the following will be produced:

Student ID	**First Name**	**Last Name**	**Term/Year**	**Course ID**	**Descriptoin**	**Grade**
abc21	Dave	Tannen	Wi17	cs275	Web and …	A
abc21	Dave	Tannen	Wi17	cs260	Data Str…	B

Part 1: Installation and configuration of the MySQL environment

MySQL installers and software can be found at: https://dev.mysql.com/downloads

Download and install the "MySQL Community Server" edition for your operating system. Check out slide 8 from the lecture slide set "Interactions Between Node.js & Relational Databases" for additional installation and database setup tips.

Part 2 : Assignment 5 Activities

1. Before you begin to code or populate your database, create a design of your application's components (web page, Nodejs program and database contents)
2. From the **mysql** > command prompt,
   A. Create a database
   B. Create the 3 tables (including schema) in that database
   C. Manually populate the 3 database tables with a representative sampling of information. Be sure to include enough entries to enable comprehensive reports (e.g. multiple Winter17 grades for student xxx to produce a multi-row "transcript").
3. Develop and test code to accomplish the two activities described in the **Requirements** section:
   - Display a table
   - Student "transcript" search

Hints:

- The NodeJS mysql module returns JSON object, so you might want to print to console to see what it's like.
- In general, you may want to copy and paste your queries into your mysql console to make sure they work.
- You can use the `eval` function in JavaScript (and NodeJS) to evaluation a string representation of a command.
- You may not want to store your SQL password in your server script and/or make sure not to have your web server's root directory be where the server script and/or password files are.

What to submit

- A screen cast video detailing a thorough code walk through, followed by a demo of all features
- Your source code, well internally documented
- README file on how to run your application
- **YOU DO NOT NEED TO UPLOAD YOUR DATABASE**
- All of the above **ZIPPED** together and that single compressed file uploaded to bbLearn

Part 3: Extra Credit (20 points)

To receive these points, please enhance your application with the following features /improvements:

1. *Improvement (10 extra credit points)*
   Generally, if possible, it is preferred to enable application users to select choices from a drop down box as opposed to manually entering into a text box. This serves two purposes:
   a. Eliminate entry of a miss-spelled or bogus name, team/year, etc.
   b. Thwart the possibility of a SQL injection attack

   In addition, a truly robust database application would allow for the selections to contain dynamic, up to date choices as opposed to having them frozen and hard coded at a certain point in time. In our assignment, for example, we would want to display the latest set of students in the database so that recent additions can be considered for selection.

   For ten extra credit points, modify the student selection feature to enable dynamic choice updates. This could be accomplished by:
   a. First querying the database for a table / list of current student names and,
   b. Using this result to create an html <selection> element

   You could demo this feature by initially running with the current set of students, then add a new student (via the mysql command line client, or if you complete the second extra credit part, via that method) followed by a display of the updated student choices.

2. *New feature (10 extra credit points)*
   Develop a feature to add a new student to the STUDENT table (along with the associated columns) from the web page as opposed to manually entering at the **mysql>** command line prompt.

   It is not necessary to allow for web page based population of associated course grades for that student (although a fully functional app would also contain this capability).

   Although you might want to think about….
   - Should data be sent to the server via GET or POST?
   - Might we need to worry about SQL injections!?
   - Do we need to worry about duplicate entries?
   - Is there some way we can show a status of the request?

Grading (50+20) Points

- 40 points: program correctness along with adherence to the stated requirements
  - 10pts – Schemas and tables
  - 10pts – Web service
  - 10pts - Activities
- 5 points: quality of internal documentation, code style and overall app design
- 5 points: README file
- 20 points extra credit
  - 10pts – Improvement
  - 10pts - New feature