CS 157A: Project

PDF versions of any documents described in "what to turn in" must be uploaded to Canvas by the posted deadline for each student in the group **or you will receive zero points**.

The goal of this part of the project is to interface your database with a programming language and to make it web-accessible. This is the stage where you embed your database into a complete application. The programming language that I recommend is PHP, a popular, server-side scripting language. Executable commands are embedded via special tags in the source of the web page. (See the class lecture slides for more details.) PHP should be installed via XAMPP on the one of your group member's machine as which includes the Apache web server. Using PHP, the MySQL database that you have created can be accessed via the installed web server.

Disclaimer: You can use ASP/Ruby-on-Rails/AJAX/Javascript/substitute-the-latest-buzzword-here/ for the project. However, should you decide to use <insert buzzword here>, you are on your own. If you choose to use <insert buzzword here>, it better work, as it will not be supported by the instructor.

Style Guide for your Application

What I want to see at the end of the day is a single web page that provides a nice interface to your database. Feel free to do more fancy and creative things, but do so **only after completing this "core" requirement**. The project template is uploaded to homework folder on Canvas for what this page should look like. If you look at the template carefully, notice that we are only querying the database. We are not doing any inserts, updates or deletes. (That's one direction to look at, if you are aiming to be creative.)

Some how-tos of scripting

Help with PHP

Here are some good documentation resources:

- The PHP website
- The PHP manual. The manual has a newsgroup-like facility with lots of potentially useful user comments.
- W3Schools.com PHP tutorials
- PHP 101: Down the Rabbit Hole. A 15-part introduction to PHP.
- Our library has a subscription to the Safari Books On-line digital library. O'Reilly publishers have all their books available on Safari, and they publish some excellent PHP books. Titles like "Head First PHP & MySQL", "Learning PHP, MySQL and JavaScript", "Build Your Own Database Driven Web Site Using PHP & MySQL", "PHP and MySQL For Dummies" and many others look promisingly relevant to this project. (Your professor has not read all these books and cannot speak to which is best. Past students have spoken

well of "Head First" books, and personal experience informs me that the books published by O'Reilly Media are frequently well put together.)
- Execute PHP MySQL On-line, a PHP testing facility provided by http://www.tutorialspoint.com/(courtesy of C Busby)

There are different ways in which you can access the MySQL database from your PHP program.

- PHP supports several APIs for accessing MySQL. Here is a guide to selecting the API you prefer.
- dbx is a database abstraction layer that provides a nice high-level interface to databases but is restricted in terms of functionality. MySQL is a supported database.
- The MySQL Improved Extension (mysqli) allows you to access the functionality provided by MySQL 4.1 and above.

Should you find another good PHP/scripting/MySQL documentation/help WWW page, please let me know and I will add a link here for the whole class to benefit from.

What your project will be graded for:

Here's the distribution of points:

1. (100 points) Does everything work correctly and do all the buttons and links that I click indeed do what they are expected to do. I will see if you are really accessing the database dynamically and are not just "looking up" the data from some ready-made source. I will ensure this by typing ad-hoc queries. You will also need to make sure that all query results have column names neatly presented and that data is clearly tabulated.

2. (20 points) Explain whether or not somebody who is truly working in the domain of your application would use your web-enabled database, or if there is something missing or intrinsically complex that will deter them from using it. (Is your web-enabled database not powerful enough for the real world, or too complicated for the real world?) For example, if you are in the books domain, would a bookseller use your database? It would be ideal if you had access to a real application domain person; you could just ask him/her to visit your web page and give you feedback. Alternately, look at "similar" systems or web pages and see if they are doing something differently and if your approach is better (or worse) and why. Include your answer to this question in your project report.

   Please note that there is no "right" answer to this question nor will you be penalized for saying that there is something missing from your project. What I want to see is if you can discuss your domain, your database, and the implications of the design and implementation decisions you made in your system for the application domain that you have chosen.

3. (20 points) Neatness, presentation, grammar, style, consistency etc.

4. (60 points) Implement some extra functionality. In your project report, clearly list exactly what you have implemented and how we can test it. Note that you must implement some extra functionality, to have a chance of getting up to 15 points. Here are some suggestions for this step:
    - Implement more complicated DB functionality: inserts, updates, deletes, etc., or some more complex queries.
    - Create a more visual or sophisticated interface for your web application, not just the vanilla style template shown above. If there is a more "natural" way for your application, maybe you will write some cute little interface for it.
    - Address other aspects of your application, such as transactions

5. (Extra Credit points) If the extra functionality is remarkable, you can get up to 15 points of extra credit, depending on how complex your application is.

What to turn in:

1. A list of your SQL **CREATE TABLE** commands (with **PRIMARY KEY**, **NOT NULL**, **DEFAULT**, **FOREIGN KEY**, and **CHECK** constraints). This is the list of create table statements your group creates as part of your work on question 2 of homework 3.
2. The source code files for your web application and any special instructions.
3. Required but not graded: Include one sentence per group member summarizing each group member's contributions. These sentences will be required in all project parts. They are not for part of any student grades. They will be used to monitor group dynamics, and to try to intercede in troubled groups (if any) before troubles get out of hand.