CS 4604 Project Assignment 3

Released on Nov 1, 2018. Email PDF version of your project report to TA by 12:00 noon Dec 4, 2018. Also submit hardcopy of the report during your demos (IMP: to count as an 'on-time' submission, you have to also submit the hardcopy during your demo session, no exceptions). See Late Policy down the page. In-person Demos: There will be an in-person demo session on Dec 5. We will release instructions on how to sign-up for the demos later after Thanksgiving.

The goal of this step 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 we will use is PHP, a server-side scripting language which is very similar to Perl. Executable commands are embedded via special tags in the source of the web page. Version 5.3.2 of PHP is installed on `cs4604.cs.vt.edu`. By this facility, the PostgreSQL database that you have created can be accessed via the web, using a webserver that has been installed for us on `cs4604.cs.vt.edu`.

At this point, it is helpful to answer the following FAQs:

FAQ: *Can we do this step of the project from our own computer/web server but still access the PostgreSQL database server on `cs4604.cs.vt.edu`?*

Answer: No. Unfortunately, you can not do this. So the only options are to use either both your own web and DB servers or only use the provided infrastructure.

FAQ: *Can we use our own database server and web server?*

Answer: Yes. You are welcome to do the project completely on your own computer. Notice that this step of the project involves having a DBMS (MySQL, PostgreSQL, Oracle, SQL server etc.) and drivers for accessing the database from the programming language you will use to implement your web interface. Depending on how you choose to demo your web interface, your computer may also need to be net-accessible (so that people can access the web pages that are stored in your computer via the world wide web). In addition, you may need to install PHP (if you decide to use PHP for your web interface), and a web server that you can operate on your computer. In short, if your setup allows the facility to create web pages and make them publicly accessible on the Internet, you are in good shape. If you choose this option, you take full responsibility for ensuring that your machine supports all the desired functionalities. Unfortunately, we will NOT be able to accommodate any last minute requests for extensions because something is not working. We provide this option for your convenience, so that you can take advantage of any already existing setup for your project. Do not attempt this unless you are absolutely positively sure of what you are doing. We do not have the resources or the manpower to provide technical support and help in debugging. At the end of the day, we will grade it just like every other project.

FAQ: *Can we use ASP/Ruby-on-Rails/AJAX/COMET/substitute-the-latest-buzzword-here/ for the project?*

Answer: Certainly! However, you are on your own. We do not have facilities to support these in the lab, nor to provide technical support.

FAQ: *Can we kill any connections run by ourselves?*

Answer: Yes. See the Using Postgresql page for more details.

Preparatory Steps:

- When posted, check out Project Assignment 2 solutions on the course website.
- Select one person in your project in whose directory you will create the files for the web interface.
- Create a directory called `cs4604` in this person's home directory on `cs4604.cs.vt.edu`. You can log into this machine using your SLO id. You will be able to create/access this directory from any machine on the `rlogin` cluster as well.
- Make your home directory world-executable (`chmod a+x ~`) and the `cs4604` directory world-readable (`chmod a+r cs4604`).
- Place all your scripts and files inside this directory.
- You can access your website at the following URL: `http://cs4604.cs.vt.edu/~"selected person"/"start file name".php.`

Help with PHP

Here are some good documentation resources for PHP:

- The PHP website
- The PHP manual The manual has a newsgroup-like facility with lots of potentially useful user comments.
- A 15-part introduction to PHP (from 2004, so it is somewhat outdated).

- The TA will be hold a short recitation in-class on Nov 1 (slides here).

The best way in which you can access the PostgreSQL database from your PHP program is to use the PostgreSQL module, which supports several PostgreSQL-specific commands.

Style Guide for your Application

What we 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. We have a style guide 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 for, if you are aiming to be creative.

What you should turn in

You have to do two things:

1. Every group has to turn in a project report. Email PDF version of your report to the TA by 12:00noon Dec 4, 2018 with the subject header 'CS4604 Project Assignment 3'. Name the PDF as your-group-name.pdf. Please do NOT send any file format other than PDF. Also bring a hardcopy to your demo session on Dec 5, 2018.
2. Every group has to give an in-person demo in a 10-15 min time-slot on Dec 5.

Late submission Policy

1. NO late days allowed on this assignment.
2. Assignment submitted late every 7 hours, will lose 25% of the grade i.e. if you submit between 12:00noon-7pm Dec 4, you will be graded out of 75 points, between 7pm-2am (Dec 5) 50 points, 2am-10am Dec 5 25 points. Anything submitted later than 10am EST Dec 5 will not be graded.
3. How to submit late: Please email the PDF to the TA, and bring a hardcopy to your demo. Your submission will be timestamped according to the time of your email.

Also see below, for more information on what the project report and demo should contain.

What your project will be graded for

Here's the distribution of points:

1. (10 points) First write and run the following SQL queries on your database. These are just sample queries to give you a sense of how SQL can enable some task for your application e.g. Query 1 below can help find recent tweets for any user. For each of the problems listed below, show the following in your hardcopy:
   - Your SQL query,
   - The result you obtained,
   - The time your query took (use the \timing command in psql to obtain the time).

   While grading your solutions, we will pay attention to the quality of your queries, e.g., whether they are correct, the number of tables they reference, and the running time. Please desist from creating massive new tables to support answering these queries!
   1. (2 points) List the 10 latest tweets posted by user 'metaioDE' (without the quotes).
   2. (3 points) List the 10 latest tweets which contain the word 'conference' (again, without the quotes).
   3. (5 points) List the user who has the most number of followers. Hint: First create a view to calculate the number of followers for each user.

2. (30 points) Does everything work correctly and do all the buttons and links that we click indeed do what they are expected to do? We will see if you are really accessing the database dynamically and are not just "looking up" the data from some ready made source. We 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.

3. (20 points) A project report where you summarize all steps of your project in a single document. If possible, include snapshots of your web interface. This report can be easily created by cutting-and-pasting all the previous reports that you submitted, making sure you include the E/R diagram, the details about any normalization, your relational schema and, the queries you created (above) in the previous steps of the project. Include the role of each team member in the project. Clearly identify the scope of everybody's contribution. If there were any problems during the course of the project (due to less than complete participation by some team member), please identify them. Mention your project web-page in the first page of your report. Include your answer to the next question too. Of the 20 points, five are reserved for neatness, presentation, style, consistency etc.

4. (40 points) Implement some extra functionality (extra compared to the style guide). 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 upto 40 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 nice interface for it.
- Address other aspects of your application, such as transactions.
- Implementing user profiles and accounts
- Automatically recommend movies to a user based on what their friends like.

5. (15 points, **Extra credit**) If the extra functionality is exceptional, you can get upto 15 points of extra credit, depending on how complex your application is. For example, one thing you can do is if you find a way to integrate and use Amazon AWS services for this project.

Last Updated: Fri, Nov 2, 9am 2018