

CSCE 4523

Project - Due: Thursday, April 25, 2024; 11:59pm on Blackboard

The goal of this project is to create a website that has a web front end to a database. Students *including grads* may work in pairs. Even if you do work as a pair, *you must each have your own installation of the website at a separate url.*

Note: This project is worth 15% of the grade (roughly double the other assignments).

Objective

The goal of this project is to practice creating a relational database backend with a web front end using any technologies you prefer.

You may work help each other with any web programming issues, but acknowledge any help given in the report. You may implement the application described below, or **choose to create your own database application of similar complexity (except previous 4523 projects).**

Requirements

- DBMS: may be mysql on turing or any other **relational database** package on any other machine that is accessible 24/7 over the web
- Database: may be the database described below or any other database consisting of at least 3 tables
- Web server: may be Apache running on turing, or any other web server (AWS?) that is publically accessible 24/7
Note: If you and your partner have to pay for the Web account, you may share one installation to avoid paying twice, but you need two different sets of web pages.
- Web site: may be the web site described below, or any other web site that supports at least 7 functions, each of which is achieved through a database call. **10 point BONUS: Add a 7th, complex function**
- Programming Language: may be any programming language. Examples are provided in several languages called from php below. You do not have to use this code, any languages are fine to use.

Program Description

Create a new database to keep track of NFL football teams during the season. There should be at least three tables:

- Team: TeamId, Location, Nickname, Conference, Division; e.g., 1, Kansas City, Chiefs, AFC, West
- Game: GameId, TeamId1, TeamId2, Score1, Score2, Date
- Player: PlayerId, TeamId, Name, Position

You may create different tables or change the attributes around as you see fit. This is just a sample, simple design that is sufficient for full credit. There must be at least 7 functions that use the database, at least 3 of which modify the table contents in some way (i.e., add, insert, delete, or modify) and at least two of which combines information from multiple tables to produce the resulting web page.

Create a web interface that supports each of the following functions:

- 1) Add a game to the Game table
- 2) Add a player to Player table
- 3) View all the players on a team (you don't need to populate the tables with all the players in the league, just a few players on each team, eg quarterback, runningback, safety)
- 4) View all the a players in a given position on any team, e.g., all the quarterbacks
- 5) View all teams arranged by conference (sorted alphabetically); within each conference, sort by number of wins overall and then number of wins within the same conference
- 6) View all games played by a given team (display the Team location, their nickname, and each result. Include their opponent's location and nickame, the date of the game, and the score. Indicate whether the won or lost.)
- 7) View all results on a given date. Display the Team name, nicknames, location, and scores for the teams involved. Clearly indicate the winner.

You must also create a single home page that has links to each of the functions for your website. Each result page for the 7 functions above should have an easy way to navigate back to the home page, if necessary (without using the "Back" button").

Implementation

Create the tables on the command line or by calling a main program from the server's command line. Seed each table with at least 2 records. Create and test one function from the server's command line, then create a web page that calls it. Only after a single function is completed should you move on and use that function/page as a template for the remaining programs.

General Directions

Implement the program in a professional way, i.e., comment it, use classes and/or functions, white space, mnemonic variable names

Implement the functions one at a time and test them thoroughly. You will receive more credit for a subset of the functions that work correctly than for code which attempts all functions but is poorly implemented.

Create a short report that includes screenshots of the homepage and, for each of the 7 (or more) actions, the web page and one result page. In the report, discuss: the url for the project home page, which language(s) you used, what host server you chose, what DBMS, and your database design (tables, attributes, datatypes, constraints). Also discuss any error checking you did.

Use **tar cvf myIDNum.tar path** to put the files all together to name it myIDNum.tar and submit it via the homework submission link on Blackboard on or before the due date. A zip file is also acceptable. e.g., `tar cvf 1234.tar /home/sgauch/4523/hw1/` will make a tarfile out of all the files in my hw1 directory and call that tarfile 1234.tar.

Grading: Correctness (91 points; 13 points for each of the 7 functions), Web design (12 points), Report (12 points), Code Design (10 points), Database Design (25 points)

Where to work: Wherever you like.

What to submit: Submit one tarball *per pair*. If you worked in pairs, be sure that the report lists both names and includes both urls. The other partner should upload a short report that just says *I worked with NAME..*