

CSCI 466 GROUP PROJECT

DATABASE DRIVEN WEB APPLICATION

THE GOAL

In groups of between three and five people (groups of other size must be explicitly approved by the professor, and there must be a *very* good reason), you will be applying all of the topics we've learned all semester to develop, from a set of requirements, a database driven web application using MariaDB and PHP.

THE REQUIREMENTS

We will be continuing on with the Fitness Tracker application that was used in the first assignment, so you can refer back to that document for details. In that assignment, you only had to design the ER diagram. Assignment 1 was graded very leniently, so it would probably behoove you to return to the drawing board, applying your more recently acquired knowledge to the ER diagram, rather than just choosing someone's assignment 1 submission and assuming it will work.

We will be adding some functionality that actually need to be implemented. Previously, the necessary views were just examples to give an idea what data needed to be stored; now they are requirements. The exception is that you are not required to implement the drawing of graphs. Kudos to groups that figure out a way to do them anyway.

In addition to the requirements from the first assignment, the following are now required:

- You must implement a page that facilitates the addition of new foods/drinks into the database. This should allow any of the relevant information (calories, macros, micronutrients, etc.) to be added.
- You must implement a page that allows the user to update their weight.
- You must implement a page that allows the user to enter the foods/drinks they have consumed, and in what quantities (tracking page). This page must allow the amount consumed to be specified in any relevant type of unit, and the rest of your app should be able to handle those conversions.
- You must implement a page that shows a table that lists of all the food consumed over a selected period of time, along with the calories and macros that are contained in the quantity of food specified. You must implement this page in such a way that clicking on the column headings will make your page sort based on that column. First click should sort in ascending order, a second should sort in descending order, and subsequent clicks just toggle that back and forth.
- You must implement a page that allows the user to enter their workouts.
- You must implement a page showing workouts over a given time period, including estimated calories burned, and any other relevant information, for each workout. Statistics such as total calories and average calories per workout should be displayed on this page as well.

FAQ's

Q. Can I work alone?

A. No.

Q. Can I use X library?

A. I have no problem with you using CSS or JavaScript to make your pages pretty or more functional. However, all of the code that implements the required page functionality must be written by you. Therefore, things like Bootstrap are OK, but a "library" written by your buddy that's not in your group that makes everything work would not be.

Q. Can I use AJAX or similar?

A. Yes. However, just as before, the code that does the work must be written by the people in the group.

WHAT TO SUBMIT?

Your group will be expected to submit the following (I will make an announcement about the final submission method):

1. A PDF file of the ER diagram you use to implement the database. This must be a valid ER diagram, so no cheating with things that are not available on ER diagrams. For example, there is no such thing as a foreign key in the conceptual model, so if two entities are connected somehow, there must be a relationship. All cardinalities must be marked, and identifiers must be indicated.
2. A PDF file that gives a description of each of the attributes present in the ER diagram. The description should include the purpose of the data, as well as a description of its domain.
3. A PDF file with the relational schema developed by converting the ER diagram using the steps from class. This should be in third normal form.
4. SQL scripts that can be run to create the tables described in the previous step.
5. SQL scripts that insert example data into the tables created in the previous set of scripts. There should be enough example data to ensure that your PHP code can run meaningfully with the data provided.
6. PHP code implementing the application, using PDO to interface with the database designed in the previous steps.
7. A link to a URL where the application is set up already and working.
8. **I am still working on the logistics for this, but there was a presentation element to the group projects in previous semesters, where the functionality of the page had to be demonstrated. Watch out for announcements on how we might still accomplish this from home.**