CIT244 Python II Program 6

Program 6: Bootstrap

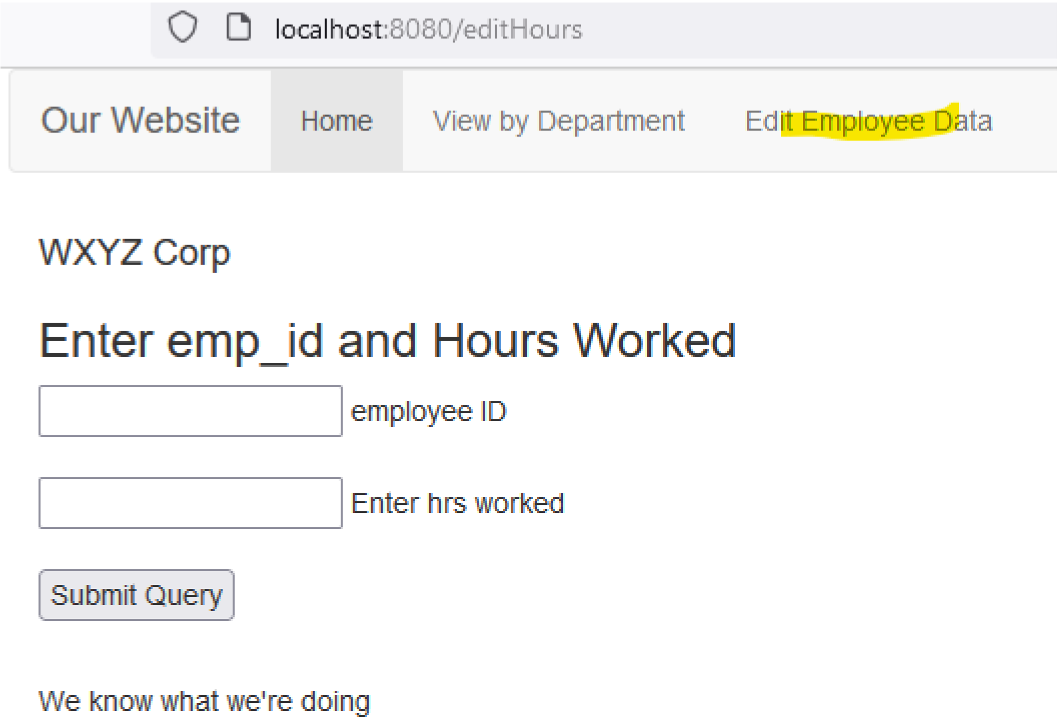
Our last program will inject proper styling and navigation to our web apps. We want to also query a database.

- The sqlite3 database is named *payroll.db*
- Two tables: one named *employees*, and it has these fields: emp_id, department, name, wage
- There are 26 employees divided into 4 departments
- The second table is named *pay_data*, with these fields: p_id, emp_id (which is a foreign key), hrs_worked, pay_period
- We want to be able to view weekly pay calculations for employees filtered by department
- We also want to be able to edit or update the hours worked for a given employee

The relationship between tables is one-to-many, not many-to-many. So don't use the movie lens example or the courses example (many students put in many classes). Look for examples in the one-to-many category. Here is a video overview of how the program should go.

click here YouTube Video **Program 6 overview**	**Program 6**

Clicking the Edit Employee Data link should show a page similar to the following which allows entering the employee ID and hours worked. Submitting this posts the information to a route that does an SQL UPDATE for the employee hours worked.



WXYZ Corp

Enter emp_id and Hours Worked

We know what we're doing

The update will overwrite a 0 or whatever value was used in the hours worked column before for that employee.

For full credit, you must:

- you need a working Bootstrap nav bar in a main layout template that allows substituting in sub templates. The nav-bar should have 3 working links.
- the default page associated with the nav-bar Home link should just say something like "You have successfully logged in.
- when you update the hours worked for a student you need to notify your user that the operation was successful.
- you must display all the employees from a given department and their weekly pay when the *View by Department* link is clicked.
- your code must allow updating any of the employee's hours worked when the *Edit Employee Dat*a link is clicked.
- your code must work with my database, so don't rename anything..

You **do not** need to have your program maintain state or log in.

If you get good and stuck, attach a compressed version of your project and mail it to me; I'll be glad to help you troubleshoot it.

Grading will follow this approach.

- 100% program runs perfectly with all required items
- 90% program meets all requirement but doesn’t always produce the correct output
- 80% program runs without errors, but is missing some requirement.
- 70% program had a good start, most but not all of the code was correct, had one or more fatal errors.
- 40-60% or below. Not very close, many parts missing, would not run, probably a late start.
- 0% no submission or code was not the student’s work.

How To Submit Your Program.

Submit compressed folder by email when ready of if you have trouble.

If you have questions let me know: mark.prather@kctcs.edu.