CSC30500 Project #3

DUE: 11:59 PM, Friday, December 1, 2023

NOTE: This project is optional!

1 Objective

To build a set of web pages that interface with a MySQL database and utilizes transactions where necessary.

2 Problem Statement

Your web pages will:

- connect to and use a specified database.
- create underlying table(s) as needed.
- allow addition of recipes/ingredients for those recipes.
- display ingredients needed for a recipe when given a recipe name.
- allow purchase of ingredients required for a recipe from the store.
- add more ingredients into the inventory of the store, noting that such may actually just be an increase in existing inventory for an ingredient.

In order to do this, you will first need to get the user of the web based application to specify their username and password as well as the IP address of the MySQL database server. *You may either prompt the user to enter their database credentials (i.e. username and password) as part of every form that might need them to be processed properly or you could have the application start page prompt for their database credentials and then store such in a session (and then display the "main menu")... do not do both of these!*

You will then need to build several web pages, including:

- a **main menu** page. This page should simply be a list of links to other web pages in your application. They must include links labeled:
  - Create a recipe or add ingredient to a recipe
  - List a recipe's ingredients
  - Buy all recipe ingredients from the store
  - Add ingredients to the store

  Each of these is described in more detail below. Note that your application should ensure that the tables needed for this application exist before trying to use them. *If the tables do not yet exist, your application should create them.* The following is a brief description of the two tables needed for this application and their associated attributes:

  - the `Recipes` table, which contains three attributes:
    - `RecipeName`, which is a string representing a recipe name
    - `Ingredient`, which is a string representing an ingredient found in the recipe specified by the `RecipeName` attribute.
    - `Quantity`, which represents the amount of `Ingredient` needed in recipe `RecipeName`.

    Note that `RecipeName`, `Ingredient` $\rightarrow$ `Quantity`.

  - the `Inventory` table, which contains two attributes:
    * `Ingredient`, which is a string representing an ingredient name that is available in the store.
    * `Quantity`, which represents the amount of this ingredient available in the store.

  Note that these tables should each have 0 records in them immediately after they have been created. You are welcome to add as many views as you like to the database, provided the above tables exist.

  You should then add the following web pages:

- a **Create a recipe or add ingredient to a recipe** page. When the equivalent link is clicked from the main menu page, this page should be displayed. It should:
  - Prompt the user to enter a recipe name, a new ingredient name, and the quantity of that new ingredient required in the recipe.
  - The underlying table(s) should then be updated to reflect this data.
- a **List a recipe's ingredients** page. When the corresponding link is clicked from the main menu, this page should be displayed. It should:
  - Prompt the user to enter a recipe name.
  - List the ingredients and required quantities for that recipe name.
- a **Buy all recipe ingredients from the store** page. When the corresponding link is clicked on the main menu page, this page should appear. It should:
  - Prompt the user to enter a recipe name.
  - For each of the ingredients in the recipe and quantity needed, the corresponding amount should be removed from the inventory of the store.
  - Note that this should be an all or nothing transaction - if there is enough inventory in the store to satisfy the *entire* recipe request, it should be satisfied. If not, none of the store's inventory should change, and a message indicating failure of the "purchase" should occur. You should also guard against issues that might come up when two (or more) users simultaneously attempt to purchase ingredients for recipe(s).
- an **Add ingredients to the store** page. When the corresponding link from the homepage is clicked, this page should:
  - Prompt the user to enter an ingredient name and a quantity. The corresponding ingredient in the inventory should have this quantity added to it. A new ingredient can be added to the store using this option.

3 Submission

Your web pages MUST work on the Lindenwood University Computer Science Linux machines. Anyone failing to do so will receive a zero for the associated project. NO EXCEPTIONS!!!! While you can develop your code anywhere you like, the end result must run on the department's Linux machines. Be warned that the process of porting code from Windows or your own other-O/S based machine to these Linux machines can be a *very* time consuming process - I've seen people work

40-50 hours on porting web pages that they wrote in less than 30 minutes. An inability to work with the Linux machines will *NOT* be considered a valid excuse for not completing this project - *NO EXCEPTIONS !!!*

You should post to Canvas either a `zip`, a `tar`, or a `tgz` file containing *all* of your `html` and `php` files as well as a `read.me` file; these files shouls be f=placed inside of a single folder. Not knowing how to do such will not be considered a valid excuse for not getting this project done.

The `read.me` file should include information about your project including (but not limited to):

- your name
- the date
- any special steps needed to execute your project
- any bugs your code has
- a brief summary of how you approached the problem

You might also want to consider adding things like a "software engineering log" or anything else you utilized in getting the project done.

4 Grading Breakdown

Correct Submission	25%
Following Directions	25%
Correct Execution	40%
Comments/read.me	10%
Extra Credit	5%

5 Final Hints

- Be warned that web development can be a *HUGE* time sink when one tries to "beautify" a web page. Note that absolutely ZERO points are being associated with how your web page looks; the only thing that matters here is that you meet the project specifications and generate correct results. Do not waste time trying to make things look super-spiffy (although things like displaying results in a readable format instead of "RunningThemAllTogether" would certainly put the grader in a better mood.)
- Have you started this project yet? If not, start now!
- If you have any questions about this project, contact me as soon as possible.
- Have you started this project yet? If not, **start NOW** !