CS 3380 Lab Assignment 7

1 Directions

This assignment must be completed by **Sunday, October 11th at 11:59 PM**. You must upload your PHP code to Blackboard. The uploaded file should be named `index.php` (or a compressed file as described in Section 3.3.5). Your code must also be hosted on your Azure account. Your lab 7 submission should be reachable and functional through the following URL:

`http://cs3380-PAWPRINT.cs.missouri.edu/lab7/index.php`

If that URL does not work you will lose points. Late submissions, either for the files or the URL, will not be accepted.

2 Goals

- Using PHP to create a functional web application
- Query using prepared statements with `mysqli_prepare`
- Using the LIKE operator
- SQL `INSERT`, `UPDATE` and `DELETE` statements

3 Tasks

3.1 Download

Begin by downloading an SQL dump file by executing the following commands in your terminal:

```
sudo mkdir /var/www/html/lab7
cd /var/www/html/lab7
sudo wget http://cs3380-mapp86.cloudapp.net/lab5/world.sql .
```

Note that you might not be able to copy-paste the above commands. You may need to type them manually into your terminal.

Next, run the `mysql` command to run the lab5.sql file and import to tables into your database.

3.2 Inspect the Data

The `data.sql` file will create three tables within your database.

3.3 Implementation

3.3.1 Main Page

You will be responsible for creating a PHP script that allows a user to execute a search based on country, city or language name. These correspond to the "name", "name" and "language" fields in the "country", "city" and "language" tables respectively.

Your code **must** use `mysqli_prepare` for your queries against the database tables in this assignment. **If you use `mysql_query` you will receive a 0 on the assignment.** The point of this assignment is to use prepared statements.

Your page should prompt the user to select what they wish to search for by selecting a radio button[1]. It should also contain a text box that allows for user entered search text to be provided. The query string should be used to perform a prefix matching search using the `LIKE` operator within your SQL query[2]. For example, this means that a search term of "Ta" should match the countries Taiwan, Tajikistan and Tanzania. Make sure you consider both uppercase and lowercase characters.

You should print the number of rows returned by a query. All results should be ordered in ascending order by the field that was used to do the search. This means that countries should be sorted using the name field, cities using the name field and languages using the language field.

When presenting search results, there should be two options at the beginning of each row: update and delete. These options should be HTML buttons that can be clicked to execute each action.

Additionally, the `index.php` page for this assignment should have a link to a page that allows for the creation of a new city within the database.

3.3.2 Update

The update button should present a page where various values can be edited and stored to the database.

The following fields in each table should be editable:

- **country** - local_name, government_form, indep_year, population
- **city** - population, district
- **country_language** - is_official, percentage

The update page should show all data for a particular record, but only allow the specified values for each table to be edited.

3.3.3 Delete

The delete button shown in search results should trigger a deletion of the corresponding record from the table.

3.3.4 Insert

Finally, a user should be allowed to add records to the city table. The id for a city should be populated with the default value (which is the next value from the associated sequence). The name, district and population should be provided by the user in text boxes. The country code should be determined by selection from a drop down box. The drop down box should contain all country **names** (notice that I said names, not codes). This list of country names should be pulled dynamically such that it changes as countries are added or deleted from the database.

3.3.5 Miscellaneous

- All queries should list the number of records returned. This is particularly useful when zero records have been returned, so a user is aware the an error did not occur.
- All appropriate error checking should be implemented. I shouldn't be able to "break" your code by providing invalid data for any user input.
- All insert, update, delete actions should provide a useful status message to the user indicating that they either occurred successfully or failed.
- You may find it useful to break your code into multiple PHP files. If you do this, create a tar or zip file containing **all** of your PHP source code and upload that single tar/zip file to Blackboard.
- As always, be sure that your code is not suspect to SQL injection attacks. You do this by passing any user input that's being used in a SQL statement through the PHP `htmlspecialchars` function.

[1] http://www.echoecho.com/htmlforms10.htm
[2] https://dev.mysql.com/doc/refman/5.0/en/string-comparison-functions.html