CSCI 311 Spring 2020: Lab 5

Learning Objectives:

- Create a mysql database
- Connect to database using PHP
- Use database contents to dynamically build web content
- Break up php code across multiple files

What to hand in:

- submit the following files to VIU Learn no later than March 7, 18:00:
  - Zip Together:
    - Lab5.php
    - Lab5Insert.php
    - Lab5Select.php
    - front.php
    - back.php
    - any other css or php files you use
- late submissions will be penalized 20% per day

All work must be individual.
Plagiarized work will result in a mark of 0. Further penalties may apply.

Marking Scheme:

- Specifications: 4 marks
- Requirements: 4 marks
- Code standards: 4 marks

Instructions:

For this lab you will create and set up your database. Then you will write php files that will output the results of a query to a table of the database and also insert into the database.

1. Create and set up your database:

- Use the following command on the command line to login to mysql:
  - `mysql -h wwwstu.csci.viu.ca -p`
  - enter your mysql password when prompted (which you will get in lab)
- Create a database for your labs:
  - You can only create databases that following the following naming convention:
    - USERNAME_*
    - where USERNAME is your csci username
    - and * can be 0 or more characters
  - `create database USERNAME_labs;`
- Indicate that you're going to use this database:
  - `use USERNAME_labs;`
- You can *show* the databases to check if it worked:
  - `show databases;`

- o   you should now see:

```
+-------------------------+
|   Database              |
+-------------------------+
| information_schema      |
| USERNAME_labs           |
+-------------------------+
2 rows in set (0.00 sec)
```

- add a table to the database:

```
create table Autos(
     auto_id SERIAL,
     auto_name VARCHAR(30) NOT NULL,
     date_added DATE NOT NULL,
     make VARCHAR(20) NOT NULL,
     model VARCHAR(20) NOT NULL,
     description VARCHAR(255),
     price DECIMAL(7,2),
     pix VARCHAR(20) NOT NULL DEFAULT "missing.jpg",
     PRIMARY KEY(auto_id)
);
```

- To add entries to your database, use an insert query of the following form:
    - o   *INSERT INTO Autos(auto_name, date_added, make, model, description, price) VALUES ("herby", NOW(), "VW", "Beetle", "the love bug", 550.00);*
    - o   **note: replace items in quotes with actual values**
- If you like, you can use the Lab5FillAutos.sql file found on the course page to fill your database
    - o   download the file and save it
    - o   while in *mysql* enter the following command:
        - ▪   `\. pathToFile/autos.sql`
- To check if values were entered correctly, run the following command:
    - o   SELECT * from Autos;

2. create your dbinfo.inc file
- create a file called dbinfo.inc, and save it to your Lab5 directory
- this file will contain the connection information needed to connect to the database. It should contain:

```php
<?php
     $host="wwwstu.csci.viu.ca";
     $user="USERNAME";
     $password="PASSWORD";
     $database="USERNAME_labs";
?>
```

- Replace USERNAME with your user name (lower case)
- Replace PASSWORD with your mysql password
- use chmod to set the permissions of this file to 600

3. create your front.php file
- This file should include the opening html tag

- It should contain the head tags, as well as all tags in the head
  - Fill the title using the $user variable found in the dbinfo.inc file
- Add the opening body tag

4. create your back.php file
- This file should contain a footer with copyright info
- Closing body and html tags

5. create your Lab5Select.php file (will be done together in lab)
- Include dbinfo.inc at the top of the file using the require command
- Next, include the front.php file at the top of the file, using the PHP `require` command
- In the body of the file:
  - Add a heading with the following text: Using PHP to access MySQL database
  - Add the following php code:
  - Connect to the database using PDO, and output a bit of info about each entry in the db:

```
try {
    $dbh = new PDO("mysql:host=$host;dbname=$database",
        $user, $password);

    $result = $dbh->query('SELECT * from Autos');
    foreach($result as $row){
        $nameVal = $row['auto_name'];
        $price = $row['price'];
        print_r("<p>".$nameVal. " " . $price."</p>");
    }
}catch(PDOException $e){
    print "Error!" . $e->getMessage()."<br/>";
}
```

- Include back.php below this code, and run your code
  - You should see the contents of your database output into your html (ugly!)
  - For an example, see the demo:
    - wwwstu.csci.viu.ca/~sarahstu/Lab5/Lab5.php
- What does this code do?:
  - Saves the results of the query in a variable, before the foreach loop
    - `$result = $dbh->query('SELECT * from Autos');`
  - You can iterate over each row of this database using a foreach loop:
    - `foreach($result as $row){ ... }`
  - Uses the field names to access each entry's:
    - Name
    - price
- Modify this to also output the name, make, model, price and description of each row.
  - to access these values for each row we could use:
    - `$nameVal = $row['auto_name'];`
    - `$modelVal = $row['model];`
    - `$makeVal = $row['make'];`

- $desc = $row['description'];
- $price = $row['price'];
  - o Generate a table or a grid of divs to output the above values for each element of the database
  - o Format the output nicely
- Style your output using external css
- Make sure all php files have the correct permissions (600)
- You can see an example here:
  - wwwstu.csci.viu.ca/~sarahstu/Lab5/Lab5Select.php

6. create your Lab5Insert.php file (to be done on your own)

This file will provide a form to let someone add a new car to the Autos table. It should accept, at minimum, the following information:

- Name
- Make
- Model
- Description
- Price

- Include dbinfo.inc at the top of the file using the require command
- Next, include the front.php file at the top of the file, using the PHP `require` command
- Check if the post variable has been set, and if it has:
  - o Get the form data and validate it
  - o Set an error if it is not valid
  - o Set the appropriate variables it is all valid
  - o Add a date variable and set it
- In the body of the file:
  - o Add a heading with the following text: Insert into the Autos Table
  - o If the error variable is set, output it meaningfully to the user
  - o If the variables were set in front.php
    - Connect to the database using PDO
    - Form a query using the values the user entered
    - Insert the value into the db
    - Output an error if appropriate, or a success message
  - o Put a form with the following:
    - Let the user input the Make, Model, and Vehicle name
- Include back.php below this code

7. create your Lab5.php file and add links

- This file should provide links to Lab5Select.php and Lab5Insert.php
- Lab5Select.php and Lab5Insert.php should include links to all Lab5 html/php files

Bonus: add paging functionality, so that you only output 5 rows per page with forward/back buttons that allow the user to page through the items. Add more items to your table to test this functionality.

Notes:

- remember, you can view your php errors and logs in the terminal with the command:
  - o ssh wwwstu
- There are good examples in the Music Demo files on the course page

Specifications:

- At least 5 rows of data inserted correctly into Autos database
- Lab5Select.php and Lab5Insert.php should successfully connect to the database
- Contents of Autos database output in a table or grid
- Database is updated correctly by Lab5Insert.php
- PHP code is correctly split across front.php, back.php, dbinfo.inc, Lab5Select.php, Lab5Insert.php and Lab5.php

Requirements:

- All pages are located and work as specified on csci server, and correct working link to Lab5.php is provided with submission
- All file names must be consist solely of a-z, A-Z, 0-9, _, and .
- All files and folders have correct permissions
- All files must be error free
- All pages must have an *appropriate* title set
- No broken links, No missing images, No hot-linked images
- Accessibility needs are met

Code Readability, and Comments:

- Consistent use of indentation (except php-generated code)
- Variables and functions have meaningful names
- Good modularization
- Comments meet course standards

Resources:

- http://www.mysqltutorial.org/mysql-resources.aspx
- https://www.w3schools.com/PHP/php_mysql_intro.asp