ELEC 6220 Information Networks and Technology  
LAB 2 REPORT

Xiao Lin & Jie Zou

# Abstract

In this lab, we have learnt to write basic PHP and HTML codes and also learnt how to mix them work together. We write basic PHP and HTML codes to create some websites in XAMPP in exercise 1. In the other exercises, we get the knowledge of how to get the information entered by user by using the method $\_GET and $\_POST. Beside in the last exercise, we following the guide try to combine MySQL with PHP and HTML, and we realize the website combined with MySQL database using basic MySQL query method in PHP and HTML.

# Introduction

We can write either HTML or PHP codes in XAMPP but onething we should attention is the version of PHP we installed is different from the version useed in guides, so something may be different. Following the guides, we store all HTML and PHP files in the directory C:\XMAPP\htdocs\html, and we can visit these HTML and PHP files by the URL: http://localhost/html\+filename.

# Design

### Exercise 1

In the first exercise we try to write a basic “Hello Word” program in HTML and PHP and use web browser to check the result of these codes.

Here is the code of PHP, and we can check it by the URL: <http://localhost/html/new_php_file.php>

<html>

<head>

<title>PHP Test</title>

</head>

<body>

<?php echo ("<p>Hellow World</php>");

?>

</body>

</html>

The screen shot is like this

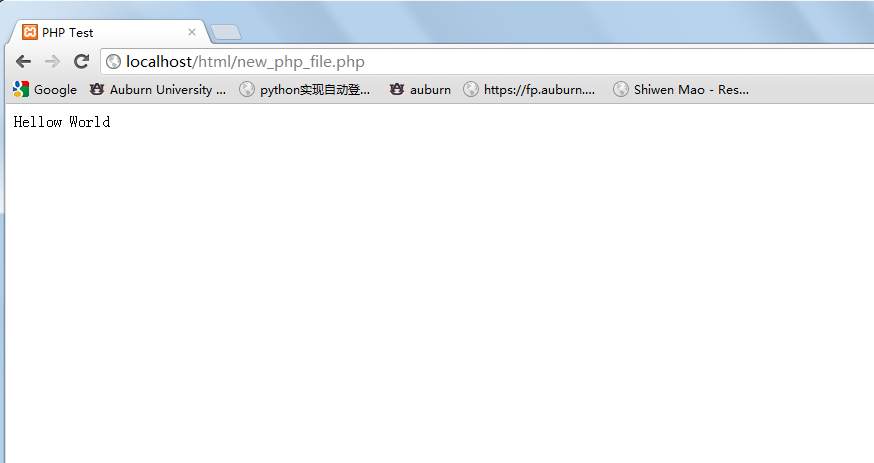


Figure 1-1: screen shot of new\_php\_file.php

And here is the HTML code for the same function it can be check by URL: <http://localhost/html/new_html_file.html>

<html>

<head>

<title>PHP Test</title>

</head>

<body>

<p>Hello world</p>

</body>

</html>

Screen shot like this

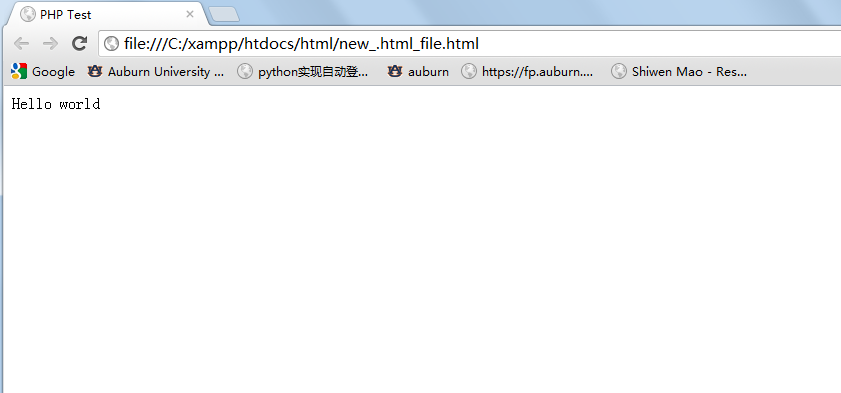


Figure 1-2: screen shot of new\_html\_file.html

And following the guide we add one line to code to show the internal information of PHP itself

Here is the code:

<html>

<head>

<title>PHP Test</title>

</head>

<body>

<?php echo ("<p>Hellow World</php>");

phpinfo();?>

</body>

</html>

Here is the screen shot

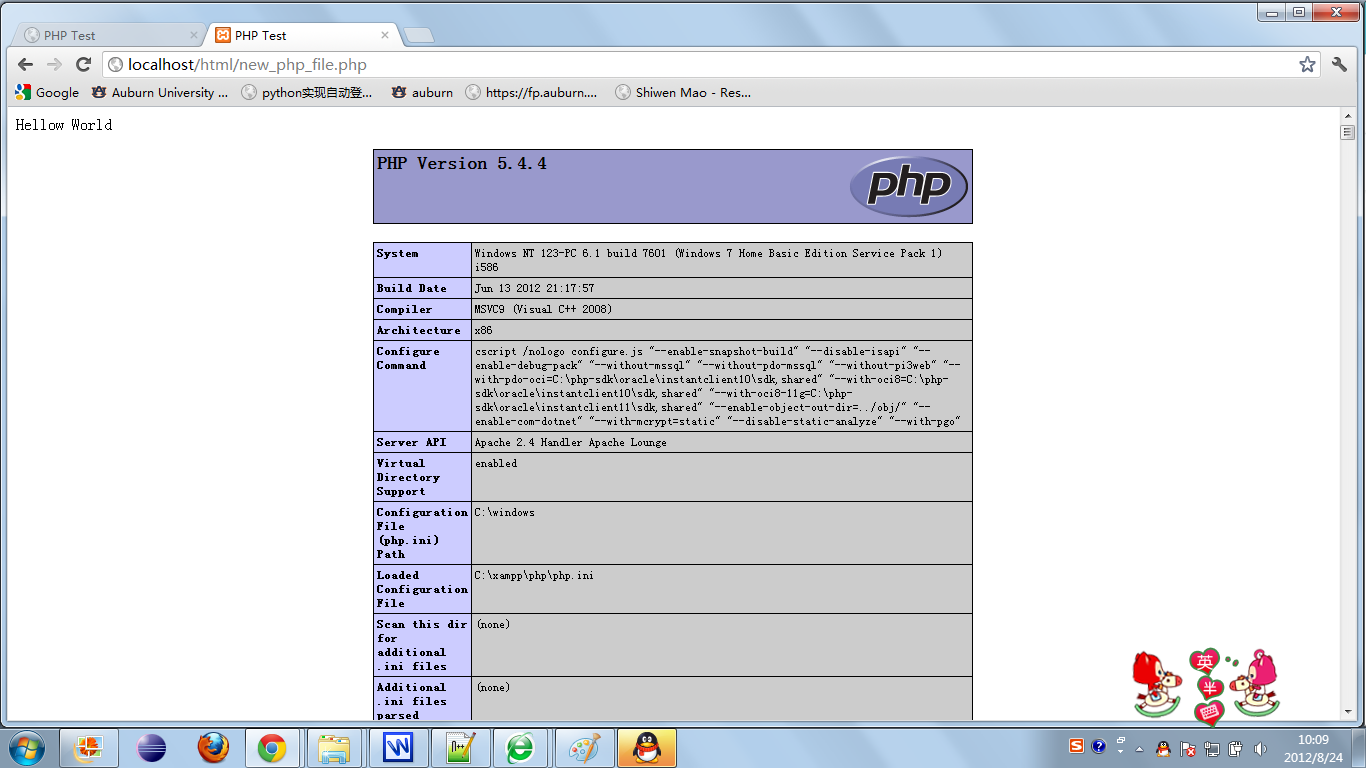


Figure 1-3: screen shot of PHP internal information

Following the guide, we write codes to show information about ourselves both in PHP and HTML.

Here is the version of HTML:

<html>

<head>

<title>my information</title>

</head>

<body>

<p><big>My Information</big></p>

<p><i>My Name : Lin Xiao</i></p>

<p><b>My Age : 21</b></p>

<p style="color:purple"><small> My Favorate Color : purple </small></p>

</body>

</html>

And here is the screen shot

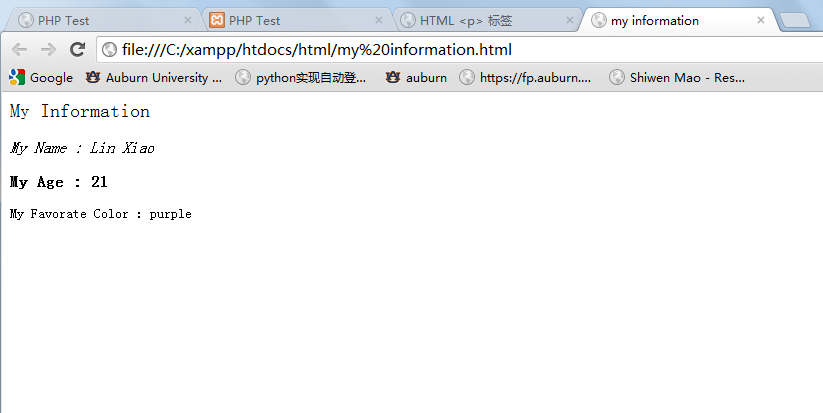


Figure 1-4: screen shot of my\_information.html

Here is the version of PHP

<html>

<head>

<title>my information</title>

</head>

<body>

<?php

echo "<p style='color:red'><big>My Information</big></p>";

echo "<p style='font-family:verdana;font-size:80%;color:green'><i>My Name : Lin Xiao</i></p>";

echo "<p style='font-family:courier'><b>My Age : 21</b></p>";

echo "<p style='color:purple'><small> My Favorate Color : purple </small></p>";?>

</body>

</html>

Here is the screen shot

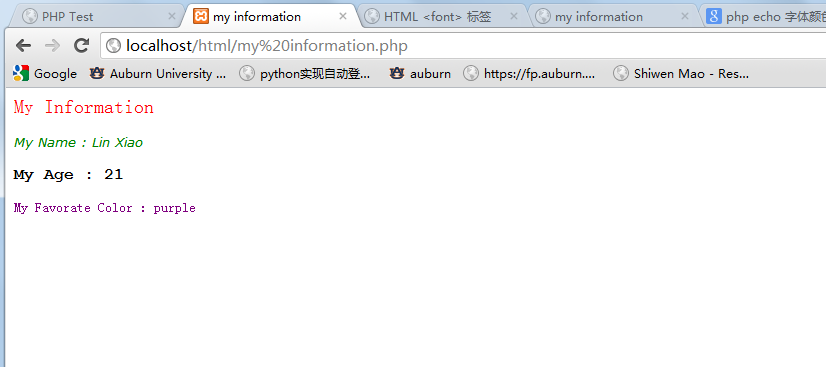


Figure 1-5: the screen shot of my\_information.php

### Exercise 2

in exercise 2 we try to mix PHP codes with HTML codes within a singal file.

Here is the code for detecting the web browser user use. If user use IE9, it will show You are using Internet Explorer 9, if not it will show You are not using Internet Explorer 9.

Here is the code of browser\_test.php:

<?php

if (strpos($\_SERVER['HTTP\_USER\_AGENT'], 'MSIE 9'))

{

?>

<h1>strpos must have returned true</h1><center><b>You are using Internet Explorer 9</b></center>

<?php

}

else {

?>

<h1>strpos must have returned false</h1><center><b>You are not using Internet Explorer 9</b></center>

<?php } ?>

The screen shot is like this:



Figure 2-1: screen shot when using Chrome

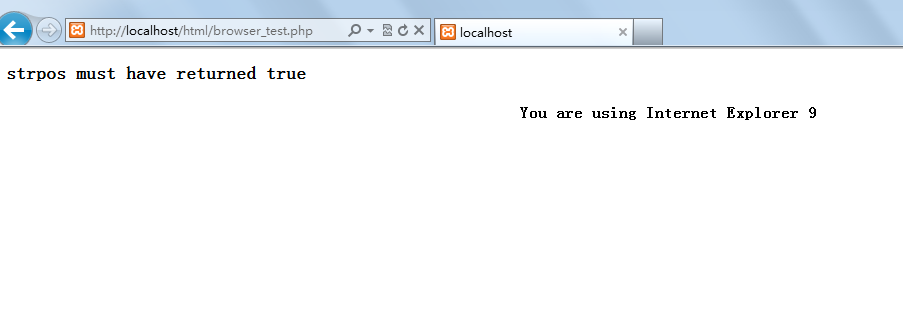


Figure 2-2: screen shot when using IE9

The HTML code above can be replace by PHP code by using the method echo, and show the same contents, here is the code of browser\_test2.php

<?php

if (strpos($\_SERVER['HTTP\_USER\_AGENT'], "MSIE 9"))

{

echo"<h1 style='color:red'>strpos must have returned true</h1><center><b style='color:yellow'>You are using Internet Explorer 9</b></center>";

}

else

{

echo"<h3 style='color:blue'>strpos must have returned false</h3><center><b style='color:green'>You are not using Internet Explorer 9</b></center>";

}?>

Here is the screen shot:

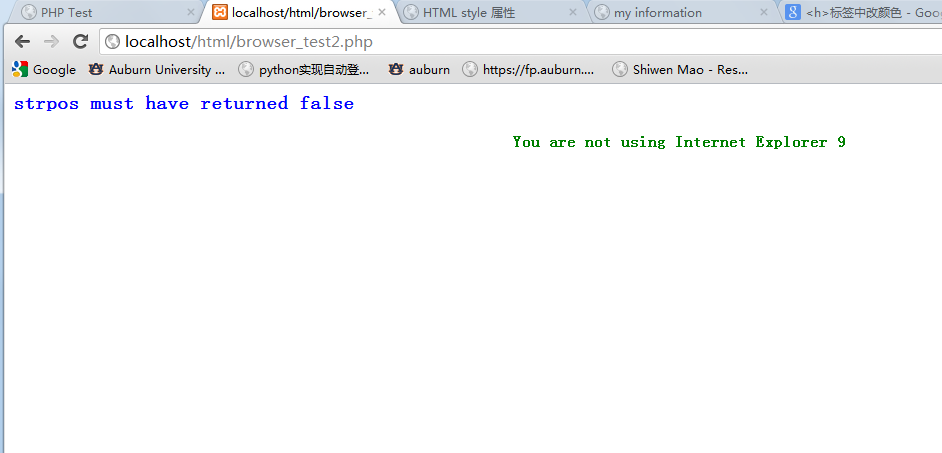


Figure 2-3: screen shot when using Chrome

### Exercise 3

In exercise 3 we use the <form> in HTML to get the information user entered and deal with the information we get in PHP.

We create a file inputG.html to get the information user entered and send the information to inputGet.php by the method \_GET. Here is the code:

<html>

<head>

<title>Get From Input HTML</title>

</head>

<body>

Please enter your name and email address, and then click Enter:</br>

<form action="inputGet.php" method="get">

Name: <input type="text" name="name" /> </br>

Email address: <input type="text" name="email" />

<input type="submit" value="Enter" />

</form>

</body>

</html>

Here is the code of inputGet.php which is used to receive the information and show the information:

<html>

<head>

<title>PHP Form</title>

</head>

<body>

Welcome <?php echo $\_GET["name"]; ?>. <br />

Your email address is <?php echo $\_GET["email"]; ?>

</body>

<html>

And the following is the screen shot:

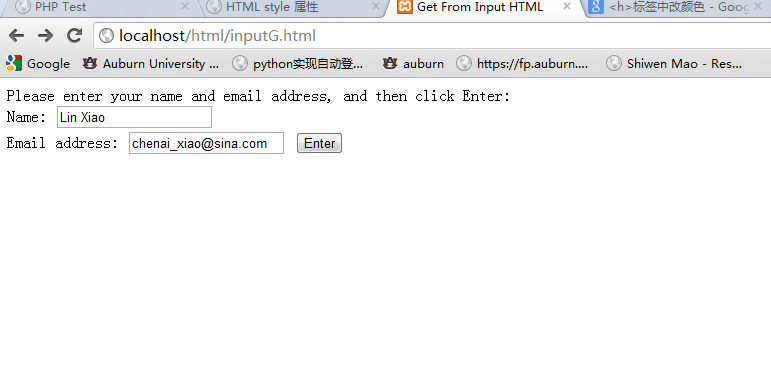


Figure 3-1: type in information in inputG.html and click the button enter

And we can get the following website.

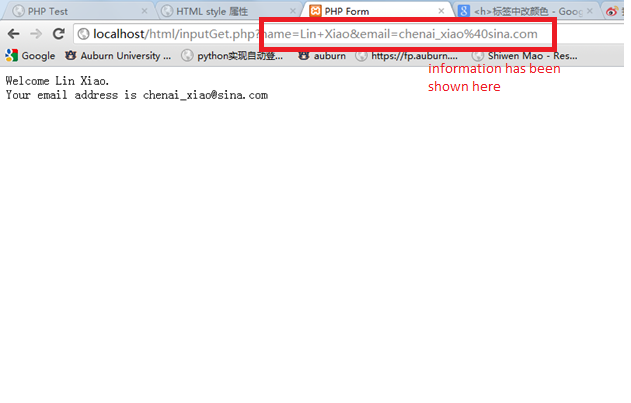


Figure 3-2: screen shot of inputGet.php

Following the guide we also learn another method to get user information and we write a similar code with \_POST method.

Here is the code of inputP.html:

<html>

<head>

<title>Post Form Input HTML file</title>

</head>

<body>

Please enter your name and email address, and then click Enter: </br>

<form action="inputP.php" method="post">

Name: <input type="text" name="name" /> </br>

Email address: <input type="text" name="email" />

<input type="submit" value="Enter" />

</form></body>

</html>

And here is the code of inputP.php which is used to deal with the information:

<html>

<head>

<title>PHP Form</title>

</head>

<body>

Welcome <?php echo $\_POST["name"]; ?>.<br />

Your email address is <?php echo $\_POST["email"]; ?>.</body>

</html>

And here is the screen shot when type in the information



Figure 3-3: type in information in inputP.html

And we click enter we get this screen shot

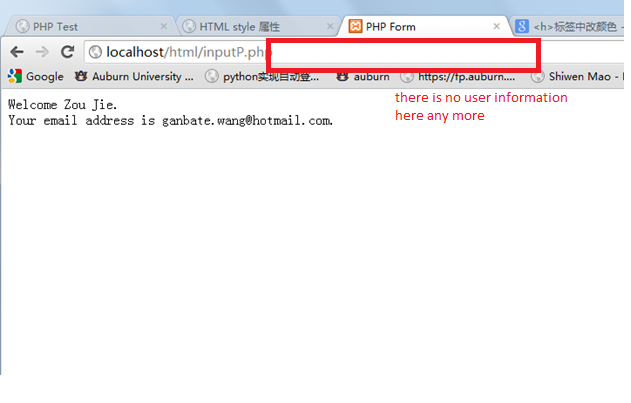


Figure 3-4: the screen shot of inputP.php

Following the guide we also try many other form tags and look how them works, here is the code we to test other form tags:

<html>

<head>

<title>Form Test</title>

</head>

<body>

<form>

<input type="text" length="30" maxlength="45">

<input type="password" length="20" maxlength="20">

<input type="radio" name="radio\_1" value="Option 1">

<input type="radio" name="radio\_1" value="Option 2">

<textarea name="comment\_box" rows="10" cols="40"></textarea>

<input type="checkbox" name="checkbox\_1" value="Yes">

<select name="pulldown\_box\_1">

<option value="option\_1">Option 1</option>

<option value="option\_2">Option 2</option></select>

<input type="hidden" name="hidden\_field\_1" value="This is hidden"></form>

</body>

</html>

Here is the screen shot when we visit this html file:

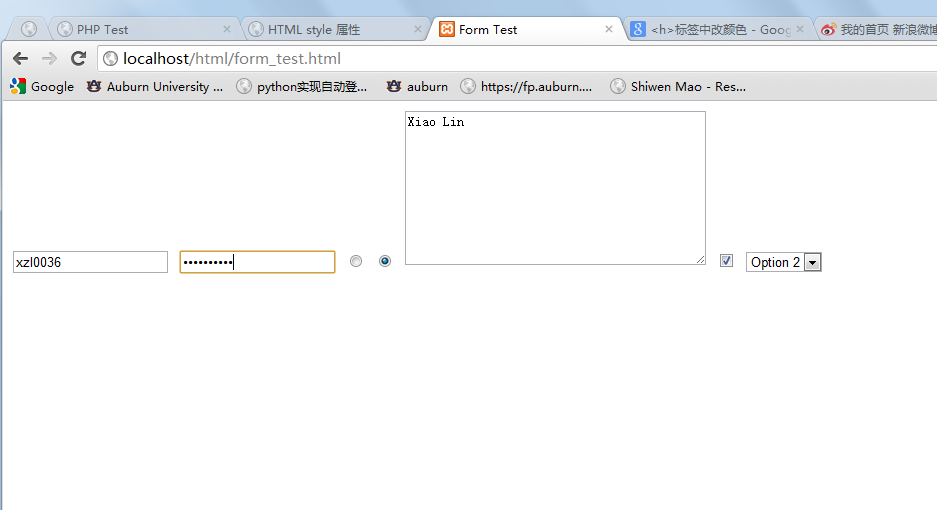


Figure 3-5: screen shot of form\_test.html

In our own exercise, we create a file with many different form tags which is used to ask for user’s information, the tags we use include: text box, password box, radio button, checkbox, pull down menu and text area. After we get the information of user, we use the method \_GET send the information to a php file(My exercise3.php). Here is the code of My exercise3.html:

<html>

<head>

<title>My exercise3</title>

</head>

<body>

<form action="My exercise3.php" method="get">

Name : <input type="text" name="name" length = "30"/></br>

Password : <input type="password" name="password" length = "20"/></br>

Sex:</br> Male<label><input type="radio" checked="checked" name ="Sex" value="male"/></label></br>

Female<label><input type="radio" name="Sex" value = "female"/></label></br>

Favorite color:

<select name="color">

<option value="blue">blue</option>

<option value="green">green</option>

<option value="yellow">yellow</option>

<option value="purple">purple</option>

</select></br>

Do you have the following?</br>

<label><input type="checkbox" name="Car" value="Car"/></label>Car</br>

<label><input type="checkbox" name="Bike" value="Bike"/></label>Bike</br>

Comments</br>

<textarea name ="commentbox"rows="10" cols="30"></textarea></br>

<input type="submit" value = "Submit"/></br>

</form>

</body>

</html>

When type in information, the page screen shot is like this:

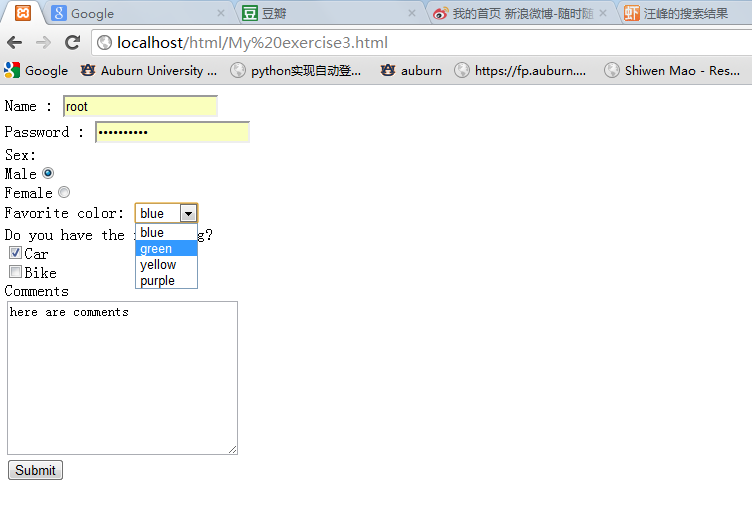


Figure 3-6: type in My exercise3.html

After we get the information we send to php file, here is the code of php file:

<html>

<head>

<title>My exercise3.php</title>

</head>

<body>

Welcom <?php echo $\_GET["name"];?></br>

Your password is <?php echo $\_GET["password"];?></br>

Sex you select is <?php echo $\_GET["Sex"];?></br>

Your favorite color is <?php echo $\_GET["color"];?></br>

Do you have a car? <?php if (isset($\_GET["Car"])) {echo "YES";} else {echo "NO";}?></br>

Do you have a bike? <?php echo isset($\_GET["Bike"])?"YES":"NO";?></br>

Your comments is : <?php echo $\_GET["commentbox"];?></br>

</body>

</html>

Here is the result of user informantion displayed:



Figure 3-7: the result in My exercise3.php

### Exercise 4

In exercise 4 we learn how to connect to MySQL database using PHP and how to query database as well as how to update the information in database.

Here is the code of breed\_lookup.html for user to type in the dog breed:

<html>

<head>

<title>Input form for Breed Lookup</title>

</head>

<body>

<form action="breed\_lookup.php" method="GET">

Dog Breed: <input type="text" name="breed\_input"><br>

<input type="submit">

</form>

</body>

</html>

And here is the screen shot when type in web browser:

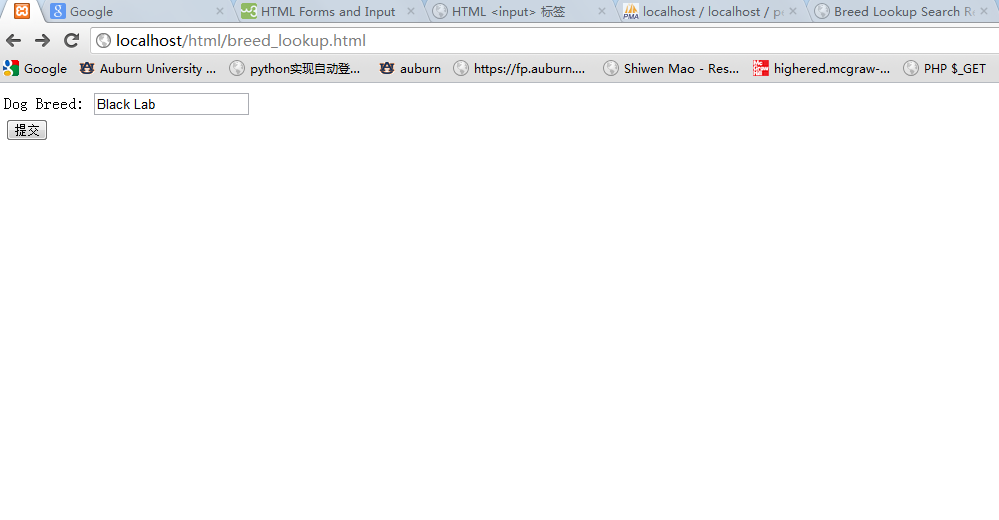


Figure 4-1: screen shot of breed\_lookup.html

And when we click the button, we will turn to the page of breed\_lookup.php, here is the code of breed\_lookup.php:

<html>

<head>

<title>Breed Lookup Search Results</title>

</head>

<body>

<?php

// Connect to the mySQL server and select our database

$dbcnx = mysql\_connect("localhost", "root","lx64498908") or die("Could not connect to the database server.");

mysql\_select\_db("pet\_store",$dbcnx) or die("Can not select database");

// Now let's do our mySQL query to lookup the information

$query = "SELECT \* FROM animals WHERE breed=\"$\_GET[breed\_input]\" ";

// We needed the slashes (\) to go with the PHP syntax, but

// to actually complete the query we need to get rid of them.

$query = stripslashes($query);

// Actually run the query

$result = mysql\_query($query) or die(mysql\_error());

// Return how many fields we selected with our \*

$number\_cols = mysql\_num\_fields($result);

// Set up the initial table and header rows

echo "<table border=\"1\" cellspacing=\"0\" cellpadding=\"5\"><tr><td>Id</td><td>Species</td><td>Breed</td><td>Quantity</td><td>Price</td></tr>";

// This next line creates an array called $row which will

// contain all of the items in a row returned by the mySQL

// query

while ($row = mysql\_fetch\_row($result))

{

echo "<tr>";

for ($i=0;$i<$number\_cols;$i++)

{

echo "<td>$row[$i]</td>";

}

echo "</tr>";

}

?>

</body>

</html>

And we get a web page like this:

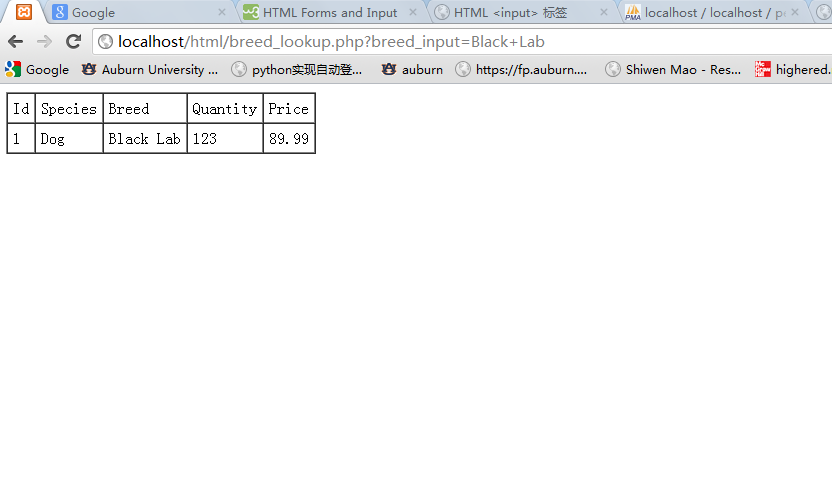


Figure 4-2: web page of breed\_lookup.php

In our own exercise, we create a page ask user for pet id and updating its quantity with HTML form. We chose the \_POST method to send information to php file. Here is the code:

<html>

<head>

<title>My exercise4</title>

</head>

<body>

<form action="My exercise4.php" method="POST">

Field ID #:<input type="text" name="Field\_ID"></br>

New Quantity : <input type="text" name="New\_Quantity"></br>

<input type="submit" value="submit" name="submit"></br>

</form>

</body>

</html>

The screen shot of the page is like this:

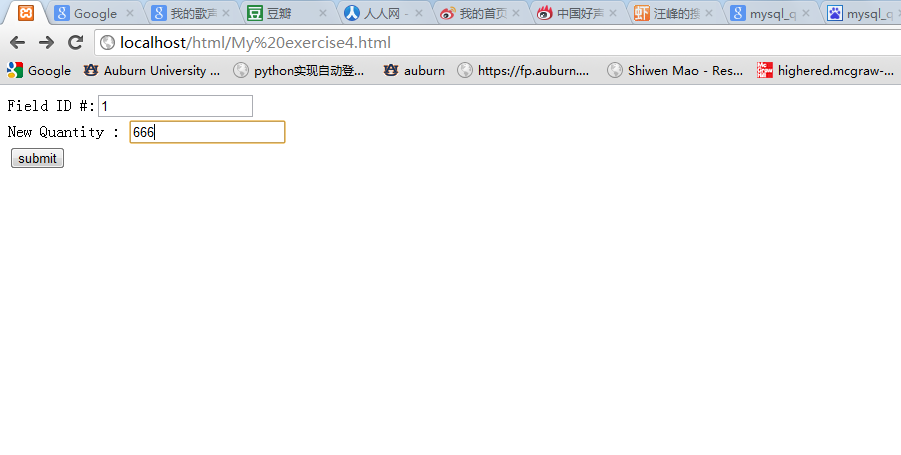


Figure 4-3: screen shot of My exercise4.html

And after we enter ID and new quantity and click the submit button we turn to a new page, the code of the new page is here:

<html>

<head>

<title>My exercise</title>

</head>

<body>

<?php

// Connect to the mySQL server and select our database

$dbcnx = mysql\_connect("localhost", "root","lx64498908") or die("Could not connect to the database server.");

mysql\_select\_db("pet\_store",$dbcnx) or die("Can not select database");

// Now let's do our mySQL query to lookup the information

$query = "SELECT \* FROM animals WHERE id\_num=\"$\_POST[Field\_ID]\" ";

// We needed the slashes (\) to go with the PHP syntax, but

// to actually complete the query we need to get rid of them.

$query = stripslashes($query);

// Actually run the query

$result = mysql\_query($query) or die(mysql\_error());

// Return how many fields we selected with our \*

$number\_cols = mysql\_num\_fields($result);

// Set up the initial table and header rows

echo "Old Table";

echo "<table border=\"1\" cellspacing=\"0\" cellpadding=\"5\"><tr><td>Id</td><td>Species</td><td>Breed</td><td>Quantity</td><td>Price</td></tr>";

// This next line creates an array called $row which will

// contain all of the items in a row returned by the mySQL

// query

while ($row = mysql\_fetch\_row($result))

{

echo "<tr>";

for ($i=0;$i<$number\_cols;$i++)

{

echo "<td>$row[$i]</td>";

}

echo "</tr></table></br>";

}

//update

$Mysql\_update="update animals set quantity = \"$\_POST[New\_Quantity]\" where id\_num = '1'" ;

$Mysql\_update= stripslashes($Mysql\_update);

mysql\_query($Mysql\_update) or die(mysql\_error());

//redraw the new table

echo "New Table";

$query = "select \* from animals where id\_num=\"$\_POST[Field\_ID]\" ";

$query = stripslashes($query);

$result = mysql\_query($query) or die(mysql\_error());

$number\_cols = mysql\_num\_fields($result);

echo "<table border=\"1\" cellspacing=\"0\" cellpadding=\"5\"><tr><td>Id</td><td>Species</td><td>Breed</td><td>Quantity</td><td>Price</td></tr>";

while ($row = mysql\_fetch\_row($result))

{

echo "<tr>";

for ($i=0;$i<$number\_cols;$i++)

{

echo "<td>$row[$i]</td>";

}

echo "</tr></table>";

}

?>

</body>

</html>

In this code, we query database by ID input by user, and display a form of old information in database. After this we updata the database according to the new quantity entered by user and display the new informtion in a new form. The screen show just like this:



Figure 4-4: screen shot of My exercise4.php after update information

# Result

All of the steps in this part of lab have been completed without significant problems. In these four exercises, we have learnt how to write PHP, HTML codes as well as their mixed codes. Besides we also learn how to get information by HTML form and how to pass information to PHP codes by the method \_GET and \_POST and their differences. In last exercise we get the knowledge of how to query MySQL database, show the information in form as well as how to update the information in database. One thing we should pay attention to is that the PHP version we use is different from the guide, so there is some small problem but we have modified them and go though all exercises.

# Conclusion

This lab is very helpful in starting with HTML and PHP programming and we are excited when we run our first web page successfully. And we find it is a very easy way to use MySQL database by HTML and PHP, so we think that will be powerful and significant in our future study. But we think the screen shot and codes in guides should be updated, because the version of PHP we use nowadays is different from that one. It may not cause big problems, but we think should make everything exactly.

# References

* PHP, <http://www.apachefriends.org/en/xampp.html>
* HTML Specification, <http://www.w3.org/TR/html4/>