**Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objectives:**

* understand the natue and characteristics of internet technology database applications
* understand HTML and PHP and how SQL is used together with PHP to create an internet database application

**Reference information:**

**about HTML**

* <https://www.w3.org/MarkUp/Guide/>
* <https://www.w3.org/MarkUp/Guide/Advanced.html>

**about PHP**

* <https://www.w3schools.com/php/default.asp>

**Setup:**

You need a file editor

* + For Windows, I suggest you use a file editor such as Notepad++ to edit html and php files.
  + For Mac users, I suggest TextWrangler.

bitnami installation (apache and mysql and php)

HSD database tables and data. The sql scripts to create the table and populate with data are included in this assignment.

This assignment uses the HSD (Heather Sweeney Designs) database.  This is about a fictitious interior design company that gives seminars about kitchen remodeling and sells books and videos at the seminars.

**Parts 1 - 4 of this assignment walk you through a simple web application. In Part 5 you will modify the web application. You will submit your modified files in part 5 to receive credit for this assignment.**

**Part 1: as simple html page**

HTML consists of text to display on a browser window and tags that tell the browser how to format the text.

A simple page consists of tags for <html>, <body>, headings <h1>, <h2>, paragraph <p>, and hyper link <a>.

<!DOCTYPE html>

<html>

<body>

<h1>Welcome to my page!</h1>

<p>For information about html visit

<a href="https://www.w3.org/MarkUp/Guide/">David Raggett's Guide to HTML</a>

</p>

<p>Also see

<a href="https://www.w3.org/MarkUp/Guide/Advanced.html">Advanced Guide to HTML</a>

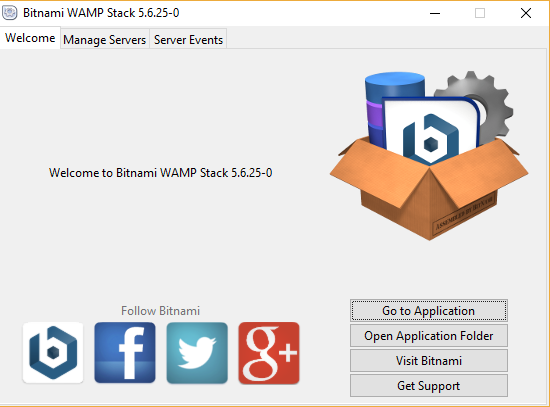
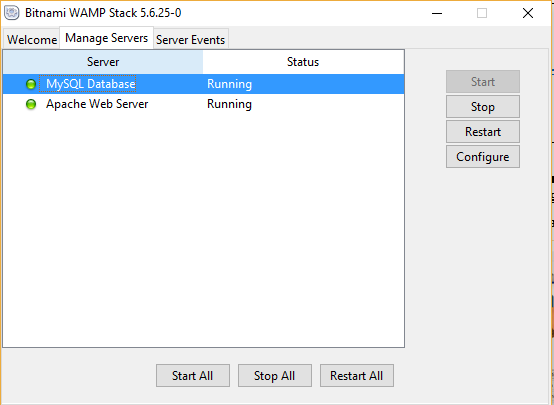
</p>

</body>

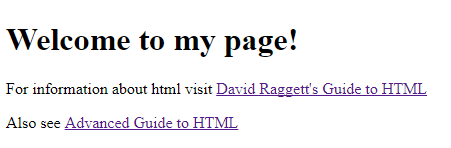
</html>

Save the above file with name **firstpage.html** and save to the directory **bitnami/apache2/htdocs/firstpage.html**

Make sure the apache server is started. Find and run the manager-windows.exe program in bitnami folder. Click on MANAGE SERVERS tab and make sure MySQL and APACHE are running. Start them as needed.

Enter the url at your browser <http://localhost/firstpage.html> and you should see your page



**Part 2: a simple php page**

The type of page we created in part 1 is a static html page. To create a more dynamic, interactive page we need to add some program logic into the html page. This can be done by using php. To write php into the html file use the tag **<?php** to mark beginning of php and **?>** to indicate the end. **date( )** is a php function that returns the current date and time. **Echo** is a php statement that outputs the current date and time.

<!DOCTYPE html>

<html>

<body>

<h1>Welcome to my page!</h1>

<p>For information about html visit

<a href="https://www.w3.org/MarkUp/Guide/">David Raggett's Guide to HTML</a>

</p>

<p>Also see

<a href="https://www.w3.org/MarkUp/Guide/Advanced.html">Advanced Guide to HTML</a>

</p>

<p> Today is

<?php

echo date("l jS \of F Y h:i:s A");

?>

</body>

</html>

Save this file as firstpage.php and then from your browser enter the url <http://localhost/firstpage.php> to see the new page with date and time. Make sure you use .php as file suffix.



If the page is not working, check for php error messages in the file **apache2/logs/error.log**

**Part 3: a php page with sql**

Using mysql workbench, create the following table in a database. It will keep a count of visitors to a web page.

create table visit (id varchar(30) primary key, count int not null);

insert into visit values("firstpage",0);

Now modifiy the php application to update the count in the database and display the new value.

You will have to modify user, password and database values for your installation.

<!DOCTYPE html>

<html>

<body>

<h1>Welcome to my php sql page!</h1>

<p>For information about html visit

<a href="https://www.w3.org/MarkUp/Guide/">David Raggett's Guide to HTML</a>

</p>

<p>Also see

<a href="https://www.w3.org/MarkUp/Guide/Advanced.html">Advanced Guide to HTML</a>

</p>

<p> Today is

<?php

echo date("l jS \of F Y h:i:s A");

echo "<br>";

$host = "localhost";

$user = "root";

$password = "cst363YOU";

$database = "cst438";

$port = 3306;

// create connection

$conn = new mysqli($host, $user, $password, $database, $port);

if ($conn->connect\_errno) {

exit ("Failed to connect: (" . $conn->connect\_errno . ") " . $conn->connect\_error );

}

// update page visit count

$sqlu = "update visit set count=count+1 where id='firstpage' ";

if ($conn->query($sqlu) != TRUE){

exit ("Update failed". $conn->error . " sql=" . $sqlu );

}

// read the new visit count

$sqlr = "select count from visit where id='firstpage' ";

$res = $conn->query($sqlr);

if (!$res) {

exit ("Select failed: (" . $conn->errno . ") " . $conn->error . " sql=" . $sqlr);

}

// fetch the first (and only) row returned by select

$row = $res->fetch\_assoc();

// get the value of the "count" column

$c = $row["count"];

// commit transation and close connection

$conn->commit();

$conn->close();

// output message

echo "You are visitor number ". $c . " to this page.";

?>

</p>

</body>

</html>

Save this file as apache2/htdocs/firstpagevisit.php

Enter the url <http://localhost/firstpagevisit.php>



If the page is not working, check any messages in **apache2/logs/error.log**

**Part 4: html forms and php applications**

We now want to allow the user to enter information or make selections and have this information passed to the php program. HTML forms allow user to enter information.

We will write a php application that displays a list of items for sale using HTML table tag. To keep things simple for now, the user can pick only one item and enter a quantity. A second php program will process the order and display acknoledgement to the user.

We need to create tables for items for sale and customer orders and insert some items for sale.

create table items (

id int primary key,

name varchar(30) not null,

price double not null);

create table orders (

id int auto\_increment primary key,

name varchar(30),

item\_id int not null,

quantity int not null,

total double not null);

insert into items values

(1, 'TI-30XS calculator', 12.85),

(2, 'NetGear wireless router', 54.90),

(3, 'Great White printer paper', 3.99);

Here is the shop.php program. Notice the **<form>** tag. This defines what entry fields will be displayed and the program **purchase.php** to be called when the user submits the form.

<?php

$host = "localhost";

$user = "root";

$password = "cst363YOU";

$database = "cst438";

$port = 3306;

// create connection

$conn = new mysqli($host, $user, $password, $database, $port);

if ($conn->connect\_errno) {

exit ("Failed to connect: (" . $conn->connect\_errno . ") " . $conn->connect\_error );

}

// read the items for sale

$sql = "select id, name, price from items order by id";

$res = $conn->query($sql);

if (!$res) {

exit ("Select failed: (" . $conn->errno . ") " . $conn->error . " sql=" . $sql);

}

?>

<!DOCTYPE html>

<html>

<body>

<h1>Welcome to my store!</h1>

<p>Items for sale.</p>

// fetch each row and display using HTML table

<table>

<?php

while ( $row = $res->fetch\_assoc() ) {

echo "<tr> <td> ".$row['id']."</td> <td>".$row['name']."</td> <td>" . $row['price']. "</td> </tr>";

}

// commit transation and close connection

$conn->commit();

$conn->close();

?>

</table>

<hr>

**<form method="post" action="purchase.php">**

**Enter the item you wish to purchase, the quantity and your name.**

**<br>**

**<table>**

**<tr><td>Item number</td><td>**

**<input type="number" name="id"/></td></tr>**

**<tr><td>Quantity</td><td>**

**<input type="number" name="quantity"/></td></tr>**

**<tr><td>Name</td><td>**

**<input type="text" name="name"/></td></tr>**

**</table>**

**<input type="submit" value="Place Order"/>**

**</table>**

**</form>**

</body>

</html

Here is the purchase program.

<?php

// retrieve user data

$item\_id= $\_POST["id"];

$quantity = $\_POST["quantity"];

$order\_name = $\_POST["name"];

$host = "localhost";

$user = "root";

$password = "cst363YOU";

$database = "cst438";

$port = 3306;

// create connection

$conn = new mysqli($host, $user, $password, $database, $port);

if ($conn->connect\_errno) {

exit ("Failed to connect: (" . $conn->connect\_errno . ") " . $conn->connect\_error );

}

// check that item\_id entered by user is valid

$sql = "select name, price from items where id = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("i", $item\_id);

$stmt->execute();

$stmt->store\_result();

if ($stmt->num\_rows <= 0) {

exit ("Error. Item id entered is not valid." . $conn->connect\_errno . ":" . $conn->connect\_error);

}

// retrieve item\_name and price

$stmt->bind\_result($item\_name, $price);

$stmt->fetch();

// calculate total purchase

$total = $quantity \* $price;

// store order

$sqli = "insert into orders values( null, ?, ?, ?, ? )";

$stmti = $conn->prepare($sqli);

$stmti->bind\_param("siid", $order\_name, $item\_id, $quantity, $total);

if (!$stmti->execute()) {

exit ("Error. Unable to place order." . $conn->error);

}

// commit transaction and close connection

$conn->commit();

$conn->close();

?>

<!DOCTYPE html>

<html>

<head>

<style>

table {

border-collapse: collapse;

width: 100%;

}

td, th {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

</style>

</head>

<body>

<h1>Thank you for shopping!</h1>

<p>Your order</p>

<?php

echo date("l jS \of F Y h:i:s A");

?>

<br>

<p>Name = <?= $order\_name;?> </p>

<br>

<table>

<tr><th>Item Id</th><th>Item Name</th><th>Quantity</th><th>Price</th></tr>

<tr><td><?= $item\_id;?></td> <td> <?= $item\_name; ?> </td> <td> <?= $quantity; ?> </td> <td> <?= $price; ?> </td></tr>

</table>

<p>Total = $<?= $total;?></p>

</body>

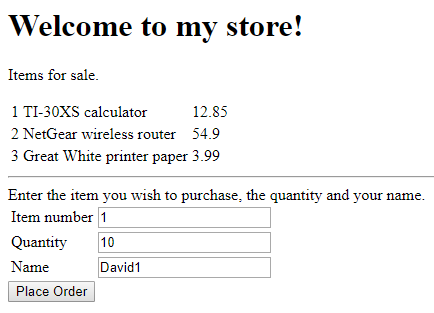
</html>

The purchase program makes of **prepared sql statements**. The sql contains question marks ? as place holders for the user values. The bind statement then specifies the php variables that will supply the values.

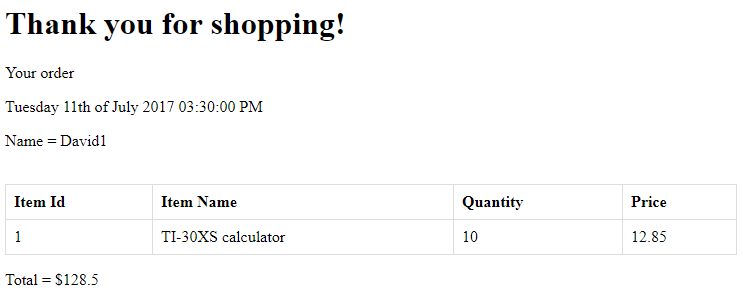
For security reasons to prevent against malicious web site attacks, it is important to use prepared statements (instead of using string concatenation to insert the user value into an sql string) to prevent sql-injection attacks.

* Find ***SQL injection attack*** in the texbook index and read about it or google it.
* Do a google search on ***SQL injection attack*** and read about it from the internet.

Enter the url <http://localhost/shop.php> which will display



If the page does not work, be sure to look at error messages in **apache2/logs/error.log**



**Part 5:**

* Modify the shop.php program to allow the user to enter a CODE field.
* Modify the purchase.php program
  + if the user enter the value SAVE10, give the user a 10% discount on the order.
  + If the user enters a value but it is not SAVE10, then give the user a 5% discount if the order if over $25.00
  + if the user does not enter any value in the CODE field, then do not give any discount.
  + Add a column to the orders table to record the value of the discount. Insert the value 0, 5 or 10 indicating the discount given to this user. use the sql statement ALTER TABLE ORDERS ADD COLUMN DISCOUNT INT.
* Upload and submit your modified shop.php and purchase.php program to get credit for this assignment.