

DEVELOPING AN E-COMMERCE DATABASE WEB APPLICATION FOR UNIVERSITY SPORT CONCESSION EVENT: A TEACHING CASE

Thomas L. Ngo-Ye, Alabama State University, tngoye@alasu.edu
Jae J. Choi, Pittsburg State University, jchoi@pittstate.edu
Dexter Gittens, Alabama State University, dgittens@alasu.edu

ABSTRACT

This case study is developed to teach students the basics of developing a database web application. The background of the case is that a university wants to extend its sport concession event sale operation online. An e-commerce database web application is needed to capture online customer order, calculate Subtotal, Tax, and Total, and store order data into a database for future analysis. Students are asked to construct a MySQL database and a table to hold customer order data. Students also need to create a web page to take customer order. Moreover, students need to build a PHP application program to process the customer order, store data in the database, and display the new online order as well as all orders in the database table. The case study gives students a hands-on opportunity to develop a functional database web application using a popular platform – PHP and MySQL.

Keywords: Teaching Case, Database Web Application Development, PHP, MySQL, E-Commerce

INTRODUCTION

This teaching case is targeted to undergraduate Computer Information Systems (CIS) major students. The ideal target students would be in junior or senior year, and already took courses of Introduction to Database, HTML Website Development, and Introduction to Programming. The course type for this teaching case is the second course in programming or the second course in Database. Preferably this project should be used at the end of a semester. This teaching case has been used in a classroom once, in the spring 2019 semester. The estimated time for completion by students is three to four class meetings, with 75 minutes per class meeting. For a professor who has already used MySQL database, PhpMyAdmin, and PHP before, it may take him/her about two hours to prepare teaching using this case. However, for a professor who has not used MySQL database, PhpMyAdmin, and PHP, but familiar with Microsoft Access database and a programming language such as Java or C++, it may take about four hours to get ready for teaching with this case. There are multiple valuable and free online resources that provide PHP 5 and MySQL Tutorials (PHP, 2019; PHP Tutorial, 2019; PHP 5 Tutorial, 2019). In addition to these online tutorials, we also recommend the following book (Powers, 2010) to learn PHP and MySQL.

List of Prerequisite Key Terms, Concepts, Skills/Techniques:

Students should have the following basic understanding before working on this case project.

- Understanding basics of relational database, table, record (row), field (column), primary key
- Understanding different data types for table columns
- Understanding that if a column's data type is AutoNumber, DMBS automatically generates a number by increment 1 and users do not need to manually provide value for this column
- Understanding basics of SQL SELECT and INSERT statements
- Understanding basics of HTML Version 5, HTML elements such as heading level, table (<tr>, <th>, <td>), form (method="post", action file), input and type attribute (type="text", type="email", type="submit", type="reset")
- Understanding basics of CSS Version 3, use embedded style to set background-image and other format styles
- Understanding basics of PHP programming constructs, variables, statements, procedural functions, String Concatenation Operator – dot (.) operator, if ... else condition, while loop

CASE BACKGROUND

A state University XXX holds several sport concession events annually. During the events, hot dogs, hot chicken wings, hamburgers, as well as some other items are sold. A food vendor contracted with the university provides all the pre-cooked food, food container, hot oven to keep food warm, refrigerator to keep drinks cold, casher register machine, as well as all other necessary supplies. The university offers its physical facility to the food vendor for concession events. University faculties and student volunteers sell food and drink in concession events. Depending on the type of sports game, number of team fans, game day weather, and other factors, the sales revenue varies among games, ranging from about \$5000 to \$20,000 in a game day. A pre-negotiated percentage of the sales proceed goes to the university, which is used to fund student organization activities.

XXX University plans to extend the physical sale of sport concession events to online. Customers can order food and drink online using their smart phone or tablets during the concession events. The concession event e-commerce webpage asks customers for their full name, email address, phone number, and the food type and quantity. After receiving the online order, concession stand workers will prepare the food based on the order. Then customers can come to the concession stand to pay the order and pick up the food. This new e-commerce operation will reduce the human labor and human error to take customer orders. It also decreases customers' waiting time. Moreover, the sales data collected electronically can be further analyzed to understand the demand of various food, the traffic distribution of different time of the game day. The insight gained can help optimize future operation and improve decision making.

XXX University decides to construct a pilot concession event e-commerce application. The web application will collect order information, calculate Subtotal, tax, and total for each order. The order date and time will also be automatically tracked. A unique order number is automatically generated for each order. All the information of every order will be recorded in a database. The web application is able to display the information of the newly placed online order on a web page. Moreover, the web application can also present information of all the orders on a web page.

As one of the required projects of CIS XXX course, students are tasked to develop the new concession event e-commerce application. The web application includes the customer order database and table, a web page that customers can place an online order, and an application program that processes the online order.

FUNCTIONAL REQUIREMENTS

In this section, we present the functional requirements of this project. XXX University wants to keep track of the following items for each sport concession event customer order: order identifier, customer full name, customer email, customer phone number, number of hot dogs ordered, number of boxes of hot chicken wings ordered, number of hamburgers ordered, subtotal, tax, total, and order date time. The price for a hot dog is \$3. The price for a box of hot chicken wings is \$7. Note each box of hot chicken wings contains six wings. The price for a 6 oz. hamburger is \$6. The sale tax rate in the city is 10%.

The e-commerce application has a web page that enables customers to enter their own online order data. The e-commerce application also has an application program that calculates subtotal, tax, and total. The application program assigns a unique number to each online order. The order date and time of each order also needs to be captured by the application program. The application program stores all above order information to a database. It displays current order as well as all other order information on a webpage.

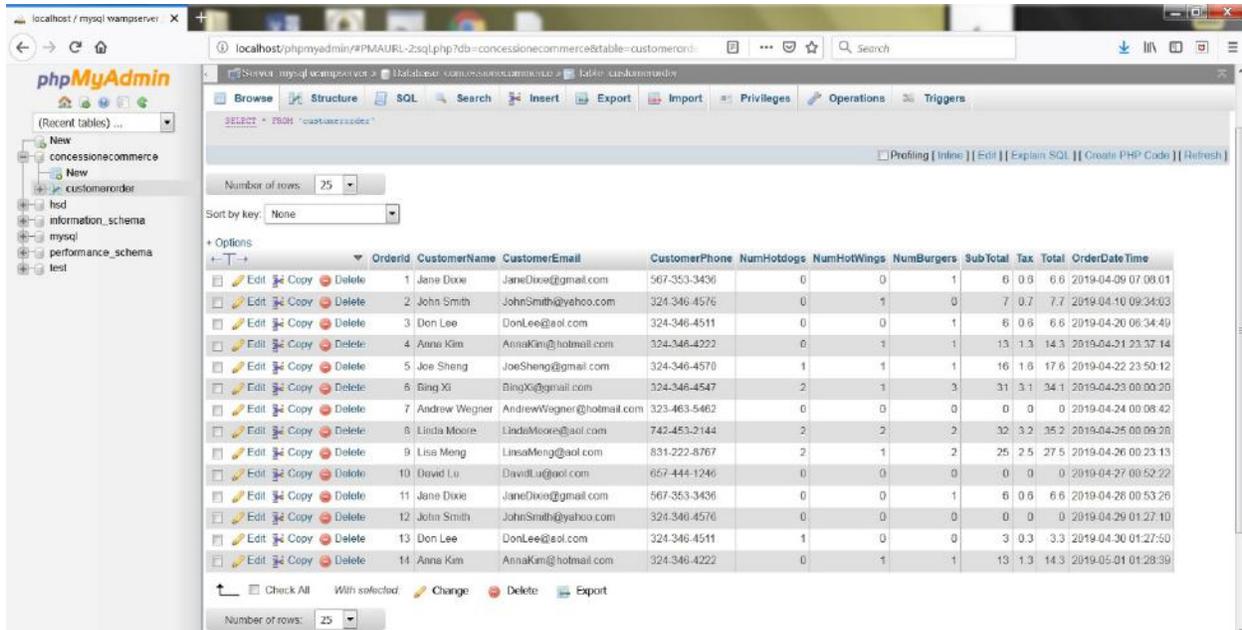
The initial web page enables data entry for a new customer order with data: *CustomerName*, *CustomerEmail*, *CustomerPhone*, *NumHotdogs*, *NumHotWings*, and *NumBurgers*. The new customer order record is added to a table ***CustomerOrder*** in a database ***ConcessionECommerce***. Table name is hard-coded as ***CustomerOrder***. *Tax rate* is hard-coded as 0.1. Table columns names are also hard-coded.

TECHNICAL ENVIRONMENT

Next, we discuss the technical environment of this project. PHP and MySQL are popular technologies for database driven web application used not only in real-world industries but also used in teaching environment (Ngo-Ye & Wang, 2017). Moreover, PHP and MySQL are free and easy to learn. Therefore, PHP and MySQL are recommended as the programming language and database for this e-commerce application. To develop and test this web application, students need the following technical environment. First, students need a Windows PC with administrator right to install and run software programs. Second, students need to download and install WampServer on a PC. WampServer is a popular and free (under GPML license) Windows web development environment for Apache web server, MySQL databases, and PHP programs. WampServer also includes PhpMyAdmin, a free web-based tool to manage MySQL databases (WampServer, 2019). Students can download WampServer from <http://www.wampserver.com/en/>. Students should use the default setting of WampServer. Third, students need to use a HTML/PHP program editor. Adobe Dreamweaver CC 2019 (19.0.1.11212) is an excellent choice with rich features and tight integration with WampServer. However, Adobe Dreamweaver is a proprietary web development tool and students have to purchase license to use it. Another choice of HTML/PHP program editor is Notepad++, which is free source code editor governed by [GPL](https://www.gnu.org/licenses/gpl-3.0.html) License (Notepad++, 2019). Students can download Notepad++ version 7.6.6 from <https://notepad-plus-plus.org/repository/7.x/7.6.6/npp.7.6.6.Installer.exe>.

REQUIRED TASKS

In this section, we describe the required tasks of this project. The tasks include: 1) create a MySQL database, 2) create a webpage, which is the entry point for the web application, 3) create a PHP application program, and 4) testing the PHP database application. Along with the tasks, we provide some sample codes as tips for students. Depending on the level of students' programming knowledge, we can adjust the amount of sample codes given to students.



The screenshot shows the phpMyAdmin interface for a MySQL database named 'concessioncommerce'. The 'customerorder' table is selected, and its data is displayed in a table format. The table has 14 rows of data, each representing a customer order. The columns include OrderId, CustomerName, CustomerEmail, CustomerPhone, NumHotdogs, NumHotWings, NumBurgers, SubTotal, Tax, Total, and OrderDate Time. Each row also has interactive icons for Edit, Copy, and Delete.

OrderId	CustomerName	CustomerEmail	CustomerPhone	NumHotdogs	NumHotWings	NumBurgers	SubTotal	Tax	Total	OrderDate Time
1	Jane Dooe	JaneDooe@gmail.com	567-353-3436	0	0	1	6	0.6	6.6	2019-04-09 07:08:01
2	John Smith	JohnSmith@yahoo.com	324-346-4576	0	1	0	7	0.7	7.7	2019-04-10 09:34:03
3	Don Lee	DonLee@aol.com	324-346-4511	0	0	1	6	0.6	6.6	2019-04-20 06:34:40
4	Anna Kim	AnnaKim@hotmail.com	324-346-4222	0	1	1	13	1.3	14.3	2019-04-21 23:37:14
5	Joe Sheng	JoeSheng@gmail.com	324-346-4570	1	1	1	16	1.6	17.6	2019-04-22 23:50:12
6	Bing Xi	BingXi@gmail.com	324-346-4547	2	1	3	31	3.1	34.1	2019-04-23 00:00:20
7	Andrew Wegner	AndrewWegner@hotmail.com	323-463-5462	0	0	0	0	0	0	2019-04-24 00:08:42
8	Linda Moore	LindaMoore@aol.com	742-453-2144	2	2	2	32	3.2	35.2	2019-04-25 00:09:28
9	Lisa Meng	LisaMeng@aol.com	831-222-8767	2	1	2	25	2.5	27.5	2019-04-26 00:23:13
10	David Lu	DavidLu@aol.com	657-444-1246	0	0	0	0	0	0	2019-04-27 00:52:22
11	Jane Dooe	JaneDooe@gmail.com	567-353-3436	0	0	1	6	0.6	6.6	2019-04-26 00:53:26
12	John Smith	JohnSmith@yahoo.com	324-346-4576	0	0	0	0	0	0	2019-04-29 01:27:10
13	Don Lee	DonLee@aol.com	324-346-4511	1	0	0	3	0.3	3.3	2019-04-30 01:27:50
14	Anna Kim	AnnaKim@hotmail.com	324-346-4222	0	1	1	13	1.3	14.3	2019-05-01 01:28:39

Figure 1. Data in CustomerOrder Table, ConcessionECommerce Database

Create a MySQL Database

1. Start WampServer on a Windows computer.

2. Manually create a new MySQL database **ConcessionECommerce** in MySQL Server using phpmyadmin (<http://localhost/phpmyadmin>).
3. Create a new table named **CustomerOrder** in the new database.
Table columns:
OrderId int(10) unsigned NOT NULL AUTO_INCREMENT
CustomerName VARCHAR 50 NOT NULL
CustomerEmail VARCHAR 50 NOT NULL
CustomerPhone VARCHAR 50 NOT NULL
NumHotdogs int(10) DEFAULT NULL
NumHotWings int(10) DEFAULT NULL
NumBurgers int(10) DEFAULT NULL
SubTotal FLOAT NOT NULL
Tax FLOAT NOT NULL
Total FLOAT NOT NULL
OrderDateTime DateTime NOT NULL
4. Perform data entry in MySQL (using phpmyadmin) and enter at least two records in table **CustomerOrder**.

The database and table should look like Figure 1.

Create a Webpage

1. Create a new blank HTML version 5 webpage called **CustomerOrderInsertPHP.html**
2. Use the text **ConcessionECommerce Project** as the title of the webpage.
3. Download the image file **trilliumgradient.png** from Blackboard to the computer.
4. Set the web page background image to **trilliumgradient.png** by using embedded CSS.
5. Use the largest font for the heading **Concession Event E-Commerce Online Order**.
6. Use the second largest font for the following three lines:
*Tax = SubTotal * TaxRate (10%)*
Total = SubTotal + Tax
Please enter the following information when placing order
7. Create a Form with method="post" and action="**CustomerOrderInsertPHP.php**"
<form name="form1" id="form1" method="post" action="**CustomerOrderInsertPHP.php**">
8. Create a Table within the Form. Set Table border to 1 pixel
9. Create 6 rows and two columns in the table and type the following text in the first column of the Table:
Your Full Name
Your Personal E-mail
Your Personal Phone
Number of Hotdogs (\$3 each)
Number of Hot Wings (\$7 each)
Number of 6 OZ. Burgers (\$6 each)
10. Create 6 text-boxes (Input element) in the second column. Make sure to have name and id attributes and set them with same value. The value should be meaningful (such as *CustomerName*, *CustomerEmail*, *CustomerPhone*, *NumHotdogs*, *NumHotWings*, and *NumBurgers*)

11. All six text-boxes are required. Put a red * to the right of the text-box. Use HTML Input element attribute required to force users to enter something for all six text-boxes.
12. For Your Personal E-mail text-box, use a special HTML 5 Input element attribute type to force users to enter something similar to an email address with @ in the middle. Use attribute value type="email".

```
<tr>
  <td><label for="CustomerEmail">Your Personal E-mail</label></td>
  <td><input type="email" name="CustomerEmail" id="CustomerEmail"
    required="required"><span class="error">*</span></td>
</tr>
```
13. For *NumHotdogs*, *NumHotWings*, and *NumBurgers* text-boxes, use a special HTML 5 Input element attribute type to force users to enter non-negative integers. Use attribute value type="number" to force input as integer. Use attribute value min="0" to force input as non-negative.

```
<input type="number" min="0" name="NumHotdogs" id="NumHotdogs" required="required">
```
14. Below the table, add a Submit button and a Reset button
15. Save the web page as *CustomerOrderInsertPHP.html*

The *CustomerOrderInsertPHP.html* should look like Figure 2 below.

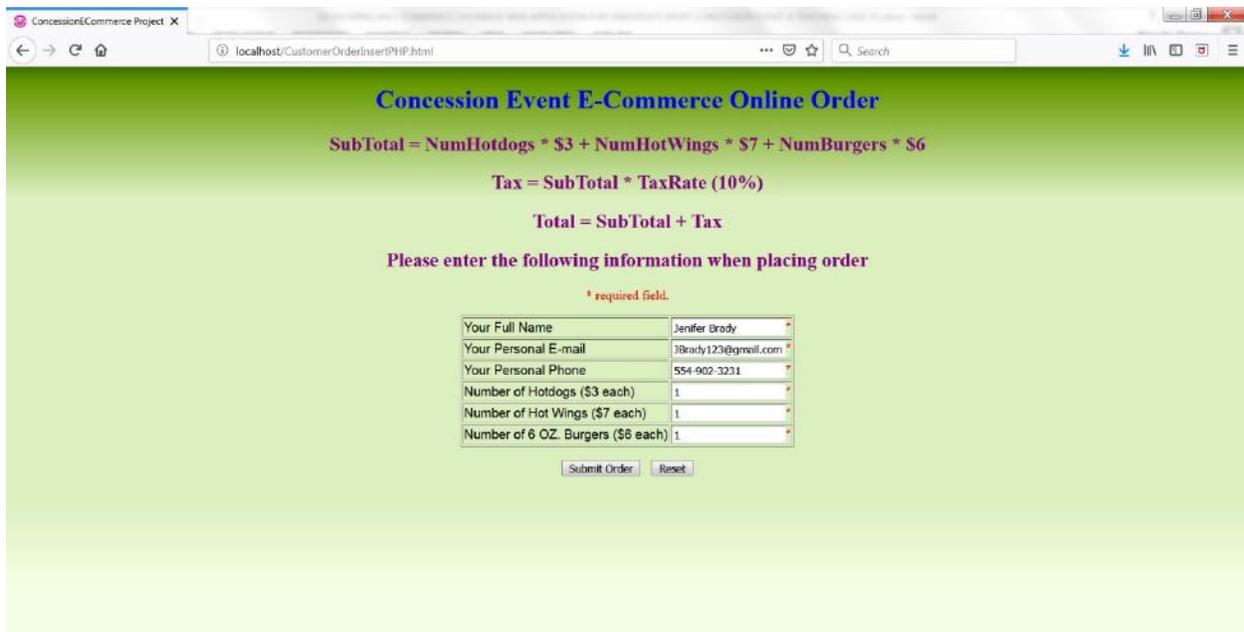


Figure 2. Data Entry in CustomerOrderInsertPHP.html

Create a PHP Application Program

Create the application program that processes data entered from the web form. The application program will fulfill the following functions. Students should use PHP's MySQLi extension to interface with MySQL database. Moreover, students should use PHP Procedural version to interact with MySQLi Procedural API.

1. Create a new file *CustomerOrderInsertPHP.php* and edit *CustomerOrderInsertPHP.php* to add PHP code.

PHP superglobal variable `$_POST[]` is widely used to collect web form data after submitting an HTML form with `method="post"`. Use `$_POST[]` to retrieve user form input data from `CustomerOrderInsertPHP.html` and assign the values to new variables. The new variables should have the same name as the name attribute value in Input element of ***CustomerOrderInsertPHP.html***. If we have a text input field in a web form called *CustomerName*, its value can be retrieved from `$_POST['CustomerName']`.

After a user click the submit button in ***CustomerOrderInsertPHP.html***, the user input form data are posted to the action page ***CustomerOrderInsertPHP.php***. At this point, the value of `($_SERVER["REQUEST_METHOD"] == "POST")` will be true. Then we can retrieve value of user input through `$_POST[]`.

Use `$_POST[]` to retrieve value of user input for *CustomerName*, *CustomerEmail*, *CustomerPhone*, *NumHotdogs*, *NumHotWings*, and *NumBurgers*. Then assign the retrieved value to a new PHP variable of the same name. For example, `$CustomerName = $_POST['CustomerName'];`

We also need to use proper PHP form validation to protect our web form from hackers and spammers, especially SQL Injection Attacks. At least, we need to use the `htmlspecialchars()` function to convert special characters to HTML entities, which means that `htmlspecialchars()` function replaces HTML characters like `<` and `>` with `<` and `>`; `htmlspecialchars()` function prevents attackers from exploiting the code by injecting JavaScript or HTML code (Cross-site Scripting attacks) in web forms.

2. Write PHP code like the following to retrieve user input.

```
function clean_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
If ($_SERVER["REQUEST_METHOD"] == "POST") {
    $CustomerName = clean_input($_POST['CustomerName']);
    ...
}
Else {
    $CustomerName = "";
    ...
    $NumHotdogs = 0;
    ...
}
```

3. If somehow ***CustomerOrderInsertPHP.php*** is invoked directly without submit button being clicked in ***CustomerOrderInsertPHP.html***, the value of `($_SERVER["REQUEST_METHOD"] == "POST")` will be false. If this is the case, we initialize value of `$CustomerName` to "" and `$NumHotdogs` to 0.

4. If users enter empty space(s) in `$_POST['NumHotdogs']`, we need to set `$NumHotdogs` to 0.

5. Write PHP code like the following to perform the calculations.

```
$TaxRate = 0.1; // hard-code TaxRate value in the application program
$SubTotal = $NumHotdogs * 3 + $NumHotWings * 7 + $NumBurgers * 6;
$Tax = $SubTotal * $TaxRate;
$Total = $SubTotal + $Tax;
```

6. *OrderDateTime* is the current Date/Time when the new record is inserted into the MySQL database. We have to manually set value for *OrderDateTime* the application program. Write PHP code like the following to set value for *OrderDateTime*

```
date_default_timezone_set('America/Chicago');
$Now = date('Y/m/d H:i:s');
```

7. Use MySQLi to make connection between **CustomerOrderInsertPHP.php** and the MySQL database. Since we run WampServer on a local PC, write PHP code like the following to connect to the MySQL database:

```
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "ConcessionECommerce";
$conn = mysqli_connect($servername, $username, $password, $dbname);
```

8. Build SQL insert statement with PHP code. Insert the new customer order record (*CustomerName*, *CustomerEmail*, *CustomerPhone*, *NumHotdogs*, *NumHotWings*, *NumBurgers*, *SubTotal*, *Tax*, *Total*) explicitly into table **CustomerOrder**. **OrderId** is the primary key and its data type is AutoNumber. So users do not need to provide its value. MySQL DBMS automatically inserts a new value for **OrderId**. Note **\$Now** matches for column **OrderDateTime** in table **CustomerOrder**. Students are provided with a similar sample code and are asked to modify the code to fit the project requirement.

```
$$SQLINSERT = "INSERT INTO STUDENT (firstname, lastname, email) VALUES ('$firstname', '$lastname', '$email')";
```

9. Write PHP code like the following to execute the SQL insert statement.

```
$Result = mysqli_query($Conn, $$SQLINSERT);
```

10. Write PHP code like the following to display **New Customer Order Added** on webpage using table element. Present the value of each variable in each table data cell by setting value. The value should come from text-box input of CustomerOrderInsertPHP.html. Do this for all 6 rows (6 text-box/Input). Moreover, present value of SubTotal, Tax, and Total for the new order.

```
echo "<h1>New Customer Order Added:</h1>";
    <table class='output' border='1' align='center'>
        <tr>";
            <td>CustomerName:</td>";
            echo "<td>" . $CustomerName . "</td>";
        </tr>";
    ...
    echo "</table>";
```

11. Write SQL statement in PHP code to retrieve all the records in table **CustomerOrder**.

```
$$SQL = "SELECT * FROM CustomerOrder";
```

12. Write PHP code like the following to execute the SQL select statement.

```
$RecordSet = mysqli_query($Conn, $$SQL);
```

13. Write PHP code like the following to display all the records in table **CustomerOrder**.

```
echo "<h1>Customer Order Table</h1>";
echo "<table class='output' border='1' align='center'>";
    <tr>
        <th>OrderId</th>
        ...
    while($RecordSetRow = mysqli_fetch_assoc($RecordSet))
    {
        echo "<tr>";
        echo "<td>" . $RecordSetRow["OrderId"] . "</td>";
        ...
        echo "</tr>";
    }
echo "</table>";
```

14. Write PHP code like the following to close the PHP's MySQLi database connection.
`mysqli_close($Conn);`
15. Save the PHP web application program as *CustomerOrderInsertPHP.php*.

Testing the PHP Database Web Application

1. Make sure copy both *CustomerOrderInsertPHP.html* and *CustomerOrderInsertPHP.php* to `C:\wamp\www\` and start WampServer on a PC. `C:\wamp\www\` is the default folder to hold all webpages and PHP programs in WampServer.
2. Open a web browser and type in <http://localhost/CustomerOrderInsertPHP.html> and type in meaningful data (Your real name, email, phone, and your order). Now the *CustomerOrderInsertPHP.html* should look like Figure 2.
3. After entering data at <http://localhost/CustomerOrderInsertPHP.html> click submit button. It should be directed to <http://localhost/CustomerOrderInsertPHP.php> with the order result.

The result after running the *CustomerOrderInsertPHP.php* should look like Figure 3 below.

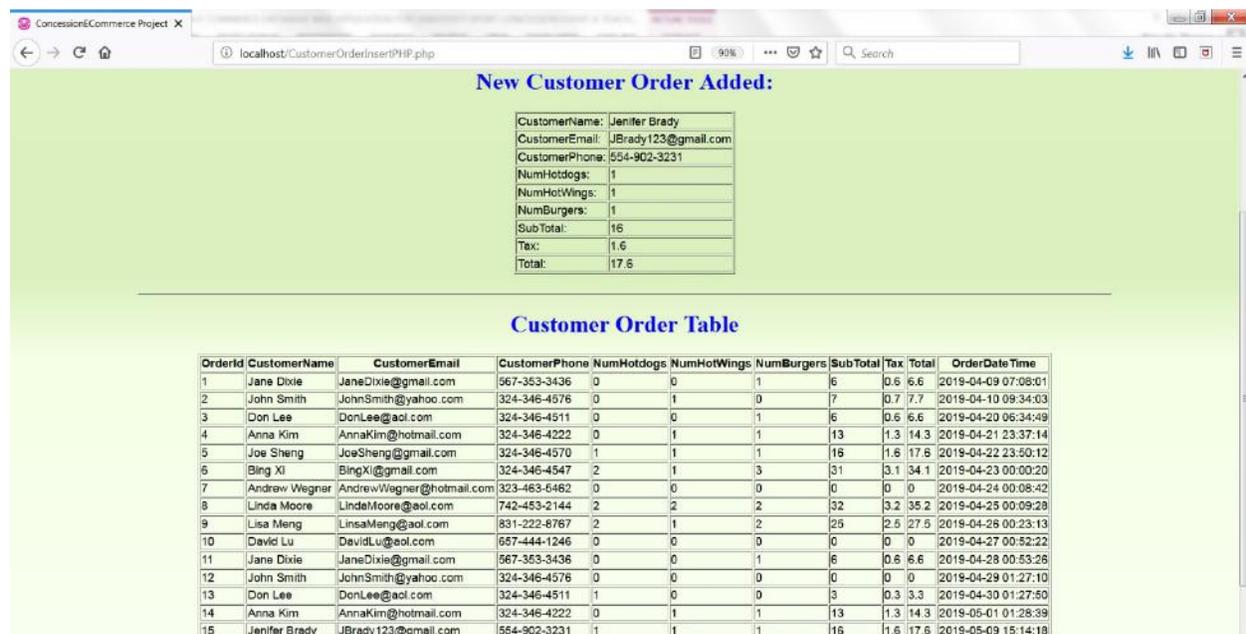


Figure 3. Results after Running CustomerOrderInsertPHP.php

DISCUSSION

In this section, we provide teaching tips based on piloting of the case. We highlight one particular area that seems especially difficult for students to grasp. We also raise questions to students encouraging them to think further to extend the project.

Manually constructing SQL INSERT statement in PHP is very difficult for many students. To mitigate the issue, we provide a sample PHP code of SQL INSERT statement to students, so they can modify the PHP code to match the project requirement.

```
$$SQLINSERT = "INSERT INTO STUDENT (firstname, lastname, email) VALUES ('$firstname', '$lastname', '$email')";
```

For the PHP application program, we provide adequate amount of sample code to students. In other words, we take our completed PHP program and break it down to several small modules. We remove some code in the modules. Based on the code snippets that we provide to students, students need to write additional PHP code to complete the project. For example, we only provide a single statement “`$CustomerName = clean_input($_POST['CustomerName']);`” and ask students to write PHP code for the remaining five statements for *CustomerEmail*, *CustomerPhone*, *NumHotdogs*, *NumHotWings*, and *NumBurgers*.

At the end of the project, we ask students questions such as “What future features do you believe are necessary, add additional value, or will be nice to have?” or simply “What comes next?”

We also raised the question “How to expand this e-commerce database web application to include sales of other products such as turkey legs and popcorn?”

A possible response is that we need to modify the MySQL database table *CustomerOrder* to include the new columns such as *NumTurkeyLegs* and *NumPopcorn*. In both *CustomerOrderInsertPHP.html* and *CustomerOrderInsertPHP.php*, we need to modify the source code to add *NumTurkeyLegs* and *NumPopcorn* as new `<input>` elements, variables, calculation of SubTotal, SQL INSERT statement, and table `<th>` and `<td>`.

COMPARISON TO RELEVANT LITERATURE

In this section, we describe relevant literature and position this paper in the body of practical resource of teaching case for Information Systems. Baugh (2018) reports the endeavor of teaching multiple programming languages concurrently for a large programming project in a beginning program course and satisfying course results. Ngo-Ye and Wang (2017) also adopt multiple programming languages (Java, PHP, and ASP.NET) in a single database application programming course. Ali (2017) discusses the selection of paradigms (such as Explorative vs. Learning by Doing) for a web design course in light of the new and most recent technological advances. Wynn (2018) develops a database design teaching case for supporting a fictional new commuter airline’s operations. The case provides detailed business information such as airline destinations, tickets and bookings, passenger, flight, planes, and crew information, as well as required tasks for students to accomplish and desired outputs. Serapiglia (2018) presents a database project teaching case based on a single season in the FIA Formula One (F1) World Championship of auto racing. The case gives thorough introduction of how F1 operates and shows a real F1 race calendar. Then the case describes various assignments for data modeling, database construction, data extraction, transformation, and loading, querying data, and reporting. Schwieger and Liu (2018) develop a data modeling teaching case focusing on the revenue and expense cycle processes of a health supply wholesale company. The case introduces the business background of a health supply company and provides detailed information of its warehouse, purchasing, accounts receivable, and accounts payable. Then the case lists various assignments for process modeling, system analysis and design, and database development. Consistent with the structure and format of the above teaching cases, we present our case in a similar manner. While the previous cases in the literature concentrate on database modeling and construction, our case focuses on database driven web application development. We believe that our paper contributes to the literature by providing a real-world practical web application development example.

PROJECT LIMITATIONS AND FURTHER ENHANCEMENTS

Upon completion of this project, students have developed a functional web application, including a MySQL database, a web based data entry for a new customer order, and an application program that handles and processes the user input. It displays the resulting outcomes in the end. We need to acknowledge that after all it is just an introductory exercise for the students. However, it is a practical hands-on lab activity of developing a database web application using PHP/MySQL in a realistic setting that is familiar for most students. We hope that through completing this project, students can demonstrate that they have mastered the working knowledge of developing a PHP/MySQL web application in e-commerce setting.

Obviously, the XXX university sport concession event e-commerce web application outlined here is just a starting point. For the three text-boxes of *NumHotdogs*, *NumHotWings*, and *NumBurgers*, we do not check whether user input is natural integer. If users insert a real number, a negative number, or a text string, the PHP application program will crash. We have not secured this application with encryption and password protected authentication. We also have not setup password for the account *root* in MySQL server. In a real production system, further input validation and security measures should be applied. We assume payment feature is not part of this e-commerce application to simplify the features of the application. However, in a real-world scenario, it is very much likely that some customers prefer to pay the order online. If online payment feature is included, the resulting application will be more useful. Of course, additional program needs to be written to interact with a payment system such as a credit card transaction web service or a PayPal web service. Moreover, we need to further verify the identity of customers picking up their orders. In addition to the *OrderId*, *CustomerName*, *CustomerEmail*, *CustomerPhone*, we may need a second factor such as a code sent via text message to a customer. In a future advanced database programming project, we will extend this application and address these issues. Some additional possible extensions of this project include Point-of-Sale (POS) integration and order online with in person pickup. Nowadays more and more fast food restaurant chains start adopting the practice of online/mobile order with in person pickup (McDonalds, 2019). However, this substantial extension is beyond the scope of one single CIS course. Other CIS courses such as Systems Analysis and Design, Project Management, and capstone courses might need to be coordinated with a wider scope of integration. We believe that further enhancements and extensions of this project will challenge and benefit CIS students. We are currently developing the second version of this project, expecting to report new insights and experiences in near future.

REFERENCES

- Ali, A. (2017). Back to Teaching a Web Design Course Again After Some Break. *Issues in Information Systems*, 18(1), 44-51. Retrieved June 28, 2019, from http://www.iacis.org/iis/2017/1_iis_2017_44-51.pdf
- Baugh, J. M. (2018). Teaching Multiple Languages Concurrently. *Issues in Information Systems*, 19(3), 1-10. Retrieved June 28, 2019, from http://www.iacis.org/iis/2018/3_iis_2018_1-10.pdf
- McDonalds. (2019). *McDonald's Mobile Ordering*. Retrieved May 9, 2019, from McDonalds.com: <https://www.mcdonalds.com/us/en-us/mobile-order-and-pay.html>
- Ngo-Ye, T. L., & Wang, J. (2017). Triangulation Approach to Teach Database Application Programming. *Issues in Information Systems*, 18(2), 171-181. Retrieved May 9, 2019, from http://www.iacis.org/iis/2017/2_iis_2017_171-181.pdf
- Notepad++. (2019). Retrieved May 9, 2019, from notepad-plus-plus.org: <https://notepad-plus-plus.org/>
- PHP. (2019). Retrieved May 9, 2019, from [php.net](http://www.php.net/): <https://www.php.net/>
- PHP 5 Tutorial. (2019). Retrieved May 9, 2019, from [w3schools.com](http://www.w3schools.com/php/default.asp): <https://www.w3schools.com/php/default.asp>
- PHP Tutorial. (2019). Retrieved May 9, 2019, from [tutorialspoint.com](http://www.tutorialspoint.com/php/): <http://www.tutorialspoint.com/php/>
- Powers, D. (2010). *Adobe Dreamweaver CS5 with PHP: Training from the Source* (1st ed.). Adobe Press.
- Schwieger, D., & Liu, Z. (2018, April). System Development and Data Modeling for Stevens' Wholesale Health Supplies. *Information Systems Education Journal*, 16(2), 25-33. Retrieved June 28, 2019, from <http://isedj.org/2018-16/n2/ISEDJv16n2p25.pdf>
- Serapiglia, A. (2018, April). Formula One – a database project from start to finish. *Information Systems Education Journal*, 16(2), 34-39. Retrieved June 28, 2019, from <http://isedj.org/2018-16/n2/ISEDJv16n2p34.pdf>
- WampServer. (2019). Retrieved May 9, 2019, from [wampserver.com](http://www.wampserver.com/en/): <http://www.wampserver.com/en/>

Wynn, D. J. (2018, April). Dragon Air: A Database Design Case. *Information Systems Education Journal*, 16(2), 18-24. Retrieved June 28, 2019, from <http://isedj.org/2018-16/n2/ISEDJv16n2p18.pdf>