

Homework #6: PHP Loops and Database

70 points

In this assignment you will learn and/or reinforce the following:

Reinforcing:

1. How to create dynamic web pages with PHP
2. PHP variables, concatenation, and built-in functions
3. How to implement basic secure coding
4. How to create PHP pages that validate as HTML 5

Learning:

5. How to create PHP looping structures and use counters and accumulators
6. How to access and interact with a database using PHP (PDO method)

Academic Honesty Policy:

Please be sure to adhere to the College and course academic honesty policy. More information on this is in our course syllabus.

Assignment Overview:

Now that you have the fundamentals of HTML, CSS, and PHP output, we can add further interactivity and dynamic functionality to web pages with PHP loops and database access. This is a large assignment, but we have a good deal of time to work on it. Partial credit is also provided and there are extra credit opportunities if you want to work on them.

Submission requirements:

Submit both the .php files and the **cislinux** web page addresses/URLs (copy and paste) to our Moodle web site. This way I will be able to run your programs and see their code. Both are needed for me to review and grade your assignment.

Program #1: Loop - Coin Flipping (10 pts)

Create a web page named **coin.php** using a PHP loop that randomly flips 20 coins. This program will consist of only one .php file. Use the PHP built-in random function and a PHP loop to display 20 random coin flips along with an image of the coin side. At the end the page, display the total how many heads were flips and how many tails were flipped out of the 20 flips. Display these totals as a raw numeric value and a percentage as shown in my example below. The web browser refresh button to re-flip the coins. Note: You will only need IF-ELSEIF statement for this program.

You can get started with coin.php by copying the contents of the HTML template. You will also need to download two images from Moodle: heads.png and tails.png. You will move coin.php to your main folder in cislinux.hfcc.edu: public_html. And the two images go into the img subfolder that was created earlier.

Sample screenshot at top of web page:

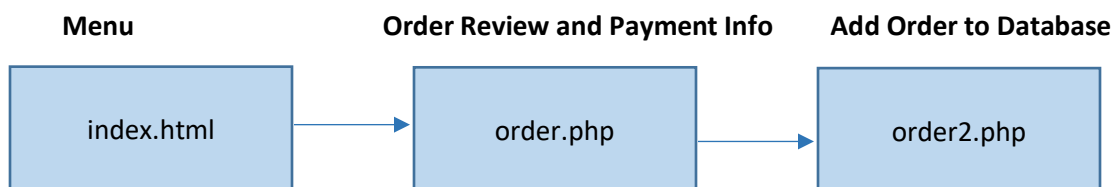


Sample screenshot at bottom of web page:



Program #2: Database Table Create and Query (60 pts)

In this program, we will create a working restaurant menu allowing orders to be placed and customer data to be reviewed by management. Below is a visual of the pages and user flow of this program.



style.css

connect.php

a. Create orders Table

- Create a new file in Notepad++ that will create an **orders** table. Name this file **cre_table1.php**
- You will need to get your MySQL username, password, and database name from the College - it is not the same as your HFC password for security reasons. Use this URL to obtain your MySQL username, password and database name: <https://cislinux.hfcc.edu/dbutils/>
- Once created, upload this cre_table1.php file to the web server, run it via the web address in the web browser, **then delete it from the server in FileZilla** once the table has been successfully created (as a precaution against inadvertently running it again).

cre_table1.php script/code to create **orders** table:

```
<!DOCTYPE html>
<html>
<body>
<?php

$servername = "localhost";
$username = "place_your_hfc_username_here";
$password = "place_your_mysql_password_here";
$dbname = "place_your_hfc_username_here";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql = "CREATE TABLE orders (
                order_id                int(10) not null auto_increment,
                payment_id              int(10) not null,
                menuitem                varchar(40),
                quantity                int,
                primary key (order_id)
            )";

    $conn->exec($sql);
    echo "Table created successfully";
}
catch(PDOException $e)
{
    echo $sql . "<br>" . $e->getMessage();
}
```

```
$conn = null;
```

```
?>
```

```
</body>
```

```
</html>
```

b. Create payment Table

- Make a copy of your cre_table1.php file and name it **cre_table2.php** We will use this file to create a second MySQL table called **payment**

Replace the SQL (structured query language) code:

```
CREATE TABLE orders (
    order_id                int(10) not null auto_increment,
    payment_id              int(10) not null,
    menuitem                varchar(40),
    quantity                int,
    primary key (order_id)
)
```

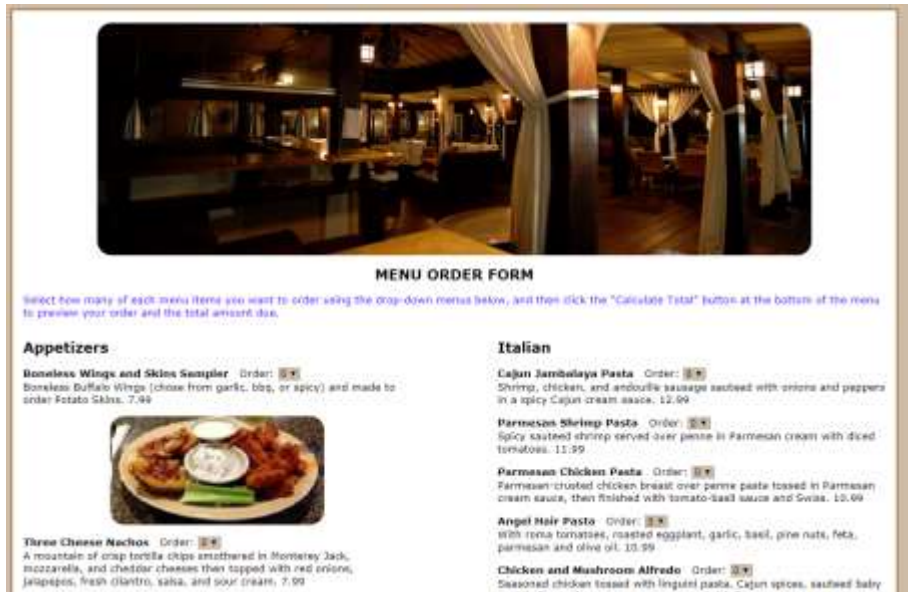
With this new SQL code to create the payment table:

```
CREATE TABLE payment (
    payment_id              int(10) not null auto_increment,
    firstname               varchar(30),
    lastname                varchar(30),
    phone1                  varchar(3),
    phone2                  varchar(3),
    phone3                  varchar(4),
    email                   varchar(35),
    cardtype                varchar(20),
    cardnumber              varchar(20),
    zipcode                 varchar(5),
    subtotal                decimal(6,2),
    taxes                   decimal(6,2),
    discount                decimal(6,2),
    total                   decimal(6,2),
    couponcode              varchar(5),
    orderdate               datetime,
    primary key (payment_id)
)
```

- Upload this file to the web server, in your public_html folder, run it via the web address in the web browser. If you accidentally run this again after you have created and populated your database tables, the previous contents could be wiped out, so as a precaution, **delete it from the server in FileZilla** once the table has been successfully created. In addition, this script contains your database password, so it's a good idea to delete after use.

c. **Download the menu order files from Moodle.**

Upload them into a separate sub-folder called **menu** under your `public_html/` folder. The main menu order page is called `index.html`. Your web address will be <https://cislinux.hfcc.edu/~username/menu/> where username is your HFC username. Also download `reports.css` from Moodle and store in the menu folder.



The screenshot shows a web page titled "MENU ORDER FORM" with a background image of a restaurant interior. Below the title, there is a instruction: "Select how many of each menu item you want to order using the drop-down menus below, and then click the 'Calculate Total' button at the bottom of the menu to preview your order and the total amount due." The menu is divided into two columns: "Appetizers" and "Italian".

Appetizers

- Boneless Wings and Skins Sampler** Order:
 Boneless Buffalo Wings (choose from garlic, bbq, or spicy) and made to order Potato Skins. 7.99
- 
 Three Cheese Nachos Order:
 A mountain of crisp tortilla chips smothered in Monterey Jack, mozzarella, and cheddar cheeses then topped with red onions, jalapeños, fresh cilantro, salsa, and sour cream. 7.99

Italian

- Cajun Jambalaya Pasta** Order:
 Shrimp, chicken, and andouille sausage sautéed with onions and peppers in a spicy Cajun cream sauce. 12.99
- Parmesan Shrimp Pasta** Order:
 Spicy sautéed shrimp served over penne in Parmesan cream with diced tomatoes. 11.99
- Parmesan Chicken Pasta** Order:
 Parmesan-crusted chicken breast over penne pasta tossed in Parmesan cream sauce, then finished with tomato-basil sauce and Swiss. 10.99
- Angel Hair Pasta** Order:
 With roma tomatoes, roasted eggplant, garlic, basil, pine nuts, feta, parmesan and olive oil. 10.99
- Chicken and Mushroom Alfredo** Order:
 Seasoned chicken tossed with linguini pasta, Cajun spices, sautéed baby mushrooms, and a creamy Alfredo sauce. 12.99

After the user chooses their menu items and clicks the "Calculate Total" button on the menu page, the order data is sent to the **order.php** page where it can be reviewed by the user and **payment information entered**.



Please review and submit your order

Menu Item	Quantity	Price	Sub Total
Spinach Artichoke Dip	1	\$6.99	\$6.99
Santa Fe Chicken Quesadilla	2	\$7.99	\$15.98
Sub-Total			\$22.97
Taxes			\$1.38
10% Discount			\$2.30
Total			\$22.05

Enter payment information below or [go back](#) and change the order.

Payment Information

First Name

Last Name

Phone Number

E-Mail Address

Credit Card No. CCV No.

* We only accept Mastercard, Visa, Discover, and American Express

Examine the code in the order.php page.

- It receives and sanitizes the data from the index.html page, e.g. wings, nachos, etc.
- It creates five variables to store total data, e.g. \$itemtotal, \$subtotal, etc.
- It creates one HTML user input form to send data to the next page (order2.php).
 - The post method is used
 - Hidden HTML fields are used to store the quantity of each item ordered, which will be saved in the orders table.
 - A subtotal variable is calculated stored in a variable and field.
 - Taxes are calculated and stored in a variable and field.
 - A discount amount is calculated if a valid coupon code was entered by the user and stored in a variable and field.
 - The order total is calculated and stored in a variable and field, i.e. \$total.
 - HTML input fields for payment information are present to collect customer data for the payment table.

d. Create order2.php page (insert into database)

- In your **order2.php** page, you will need to:
- Retrieve and sanitize all your **data** from the order.php page, e.g.:

```
$firstname = substr(filter_input(INPUT_POST, 'firstname', FILTER_SANITIZE_STRING),0,30);
```

- Retrieve the **system/current date** using PHP code such as this (this will be used for the date the order was placed, i.e. the date built-in function:

```
$date = date('Y-m-d H:i:s');
```

- For security purposes, point to a separate file called `connect.php` in your username folder (three folders higher) that will contain your MySQL login credentials. This line in your `order2.php` page will link to your login credentials:

```
include('.././connect.php');
```

Contents of **connect.php** file (with your MySQL database information):

```
$host="localhost";  
$username=" ";  
$password=" ";  
$database=" ";
```

- **Establish database connection** with PDO.

```
// Establish database connection with PDO  
try {  
    $DBH = new PDO("mysql:host=$host;dbname=$database", $username, $password);  
}  
catch(PDOException $e) {  
    echo $e->getMessage();  
}
```

- **Insert the Payment Information** into the MySQL payment table using the PDO PHP method:

```
// Use of PHP PDO prepared statements to prevent SQL injection  
// Insert into payment table  
$sql = $DBH->prepare("INSERT INTO payment VALUES (NULL, :firstname, :lastname,  
:phone1, :phone2, :phone3, :email, :CardType, :CardNumber, :zipcode, :subtotal, :taxes,  
:discount, :total, :couponcode, :date)");
```

```
$sql->execute( array(':firstname'=>$firstname, ':lastname'=>$lastname,  
':phone1'=>$phone1, ':phone2'=>$phone2, ':phone3'=>$phone3, ':email'=>$email,  
':CardType'=>$CardType, ':CardNumber'=>$CardNumber, ':zipcode'=>$zipcode,  
':subtotal'=>$subtotal, ':taxes'=>$taxes, ':discount'=>$discount, ':total'=>$total,  
':couponcode'=>$couponcode, ':date'=>$date ) or die(print_r($sql->errorInfo(),  
true));
```

- Get the **payment_id** from newly Placed Order so we can use it in orders table to link two tables together:

```
// Get payment_id from newly placed order  
$query = $DBH->prepare("SELECT * FROM payment WHERE firstname = :firstname AND  
lastname = :lastname AND orderdate = :date");  
$values = [':firstname' => $firstname, ':lastname' => $lastname, ':date' => $date];  
$query->execute($values);  
$payments = $query->fetchAll(PDO::FETCH_CLASS, 'StdClass');
```

```
foreach ($payments as $payment) {
    $payment_id = $payment->payment_id;
}
```

- **Insert menu items ordered** into orders table (repeat for each menu item):

```
// Insert records into orders table
if (!empty($_POST['wings'])) {
    $wings=$_POST['wings'];
    $sql=$DBH->prepare("INSERT INTO orders VALUES (NULL, :payment_id, 'Boneless
Wings and Skins Sampler', :wings)");
    $sql->execute( array(':payment_id'=>$payment_id, ':wings'=>$wings) )or
die(print_r($sql->errorInfo(), true));
}
```

- Display message and close the database connection:

```
echo "<p>Order successfully placed.</p>";
```

```
// Close database connection
$DBH = null;
```

e. Place at least 5 orders using your restaurant menu order web site

f. Create Management Reports menu page

Name the main reports menu page **index.html** and save it in a new sub-folder under menu called reports, i.e. <https://cislinux.hfcc.edu/~username/menu/reports/>



g. Create a menu display all orders report

- Name this file **displayorders.php**

- Use the template PHP code below to display the contents (all records and all fields) of the MySQL table to a web page. Your name

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="keywords" content="CIS Restaurant, Lunch Menu">
    <meta name="description" content="CIS Restaurant: Lunch Menu">
    <title>CIS Restaurant Menu</title>
    <link rel="stylesheet" type="text/css" href=" ../style.css">
    <link rel="stylesheet" type="text/css" href=" ../reports.css">
</head>

<body>

<div id="container">

<h2 class="blue center">CIS 122 Restaurant - Menu Orders</h2>

<?php
// database login credentials
include('../../connect.php');

// Establish database connection with PDO
try {
    $DBH = new PDO("mysql:host=$host;dbname=$database", $username, $password);
}
catch(PDOException $e) {
    echo $e->getMessage();
}

// Run Query
$sql= "SELECT * FROM payment";
$stmt = $DBH->prepare($sql);
$stmt->execute();
$result = $stmt->fetchAll();

echo "<table class='table2 tablectr'>";
echo "<tr><th>Payment No</th><th>Order Date/Time</th><th>Last Name</th><th>First
Name</th><th>Phone No</th>
    <th>Email</th><th>Sub-
Total</th><th>Taxes</th><th>Discount</th><th>Total</th><th>Details</th></tr>";
foreach($result as $row)
{
    echo '<tr>';
    echo "<td>". htmlspecialchars($row['payment_id']) . "</td>";
    $newdate = date( 'm/d/y g:i A', strtotime($row['orderdate']));
    echo "<td>". $newdate . "</td>";
```

```

        echo "<td>" . htmlspecialchars($row['lastname']) . "</td>";
        echo "<td>" . htmlspecialchars($row['firstname']) . "</td>";
        echo "<td>(" . htmlspecialchars($row['phone1']) . ") ";
        echo htmlspecialchars($row['phone2']) . "-";
        echo htmlspecialchars($row['phone3']) . "</td>";
        echo "<td>" . htmlspecialchars($row['email']) . "</td>";
        echo "<td>$" . htmlspecialchars($row['subtotal']) . "</td>";
        echo "<td>$" . htmlspecialchars($row['taxes']) . "</td>";
        echo "<td>$" . htmlspecialchars($row['discount']) . "</td>";
        echo "<td>$" . htmlspecialchars($row['total']) . "</td>";
        echo "<td><a href=\"displayorders2.php?payment_id=" . htmlspecialchars($row['payment_id'])
. "\">Details</a></td>";
        echo "</tr>";
    }
    echo "</table>";

// Close database connection
$DBH = null;
?>
<br>
</div>
<br>

</body>
</html>

```

Sample Output of **displayorders.php** page

CIS 122 Restaurant - Menu Orders										
Payment No	Order Date/Time	Last Name	First Name	Phone No	Email	Sub-Total	Taxes	Discount	Total	Details
7	03/23/19 10:52 AM	Smith	Dan	(313) 111-0101	smith@smith.com	\$21.97	\$1.32	\$0.00	\$23.29	Details
8	03/23/19 10:52 AM	Bazzi	Ali	(313) 393-0001	alib@gmail.com	\$23.97	\$1.44	\$2.40	\$23.01	Details
9	03/23/19 10:54 AM	Tillman	Mary	(248) 999-0191	tilman23@mail.com	\$36.95	\$2.22	\$0.00	\$39.17	Details
10	03/23/19 10:55 AM	Davis	George	(313) 392-3030	gdavis01@gmmal.com	\$47.94	\$2.88	\$0.00	\$50.82	Details

- h. Create a page called **displayorders2.php** that displays the order details when the user clicks the "Details" hyperlink in the previous report.

Order Details		
Order No	Menu Items	Quantity
1	Boneless Wings and Skins Sampler	2
2	Three Cheese Nachos	1

i. Implement .htaccess security on your reports/ folder.

- Create a text file named `.htaccess` and save/upload it in your **reports/ folder**. Add this code to it with your HFC username where *username* is

```
AuthType Basic
AuthName "My Protected Area"
AuthUserFile /home/username/.htpasswd
Require valid-user
```

- Go to this web site, enter a username and password that will be credentials to log into the reports folder. **Important: do not a password that you use for the College or any other web site or system**. Provide me with this username and password so I can log into and access your reports folder.

<https://www.htaccesstools.com/htpasswd-generator/>

Save the .htpasswd file it creates and upload it to your **username folder**, e.g.
`/home/username`

Extra Credit #1

Create these three additional management reports for **3 extra credit points per report**.

- j. Create a page called **displaytotal2.php** that displays the five most popular menu items sold and five least popular items sold.

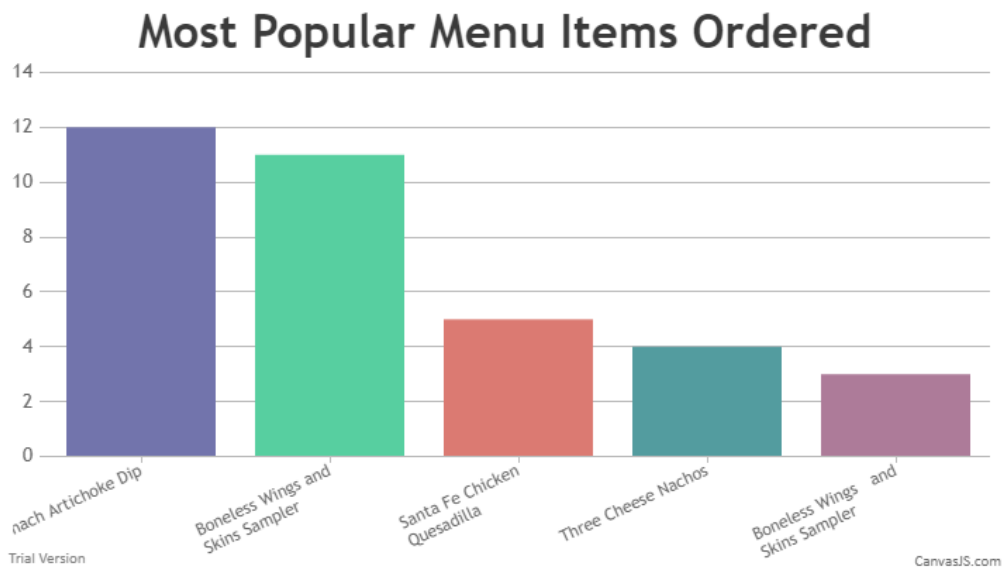
CIS 122 Restaurant - Most Popular and Least Popular Menu Items	
Least Popular Menu Items	Quantity
Boneless Wings and Skins Sampler	1
Spinach Artichoke Dip	28
Three Cheese Nachos	31
Boneless Wings and Skins Sampler	32
Most Popular Menu Items	Quantity
Boneless Wings and Skins Sampler	32
Three Cheese Nachos	31
Spinach Artichoke Dip	28
Boneless Wings and Skins Sampler	1

- k. Create a page called **displaysales2.php** that displays the five most popular menu items sold and five least popular items sold.

Total Sales by Month/Year

Month	Sub-Total	Taxes	Total
March 2019	\$130.83	\$7.86	\$136.29
April 2019	\$197.74	\$11.87	\$208.01

- l. Below is a dynamic, database-backed bar chart report using the canvasjs script we used earlier.



- m. Integrate the search-as-you-type functionality from Homework #3 into your displayorders.php search page.

CIS 122 Restaurant - Menu Orders

Search:

Payment No	Order Date/Time	Last Name	First Name	Phone No	Email	Sub-Total	Taxes	Discount	Total	Details
41	11/21/19 1:37 PM	Drew	Nancy	(233) 232-2323	drew@drew.com	\$62.91	\$3.77	\$0.00	\$66.68	Details
40	11/19/19 5:48 PM	Harper	Nancy	(313) 234-2323	harp@harp.com	\$31.96	\$1.92	\$0.00	\$33.88	Details
29	11/14/19 2:54 PM	Rogers	Paul	(313) 222-2393	rogers@ggg.com	\$71.91	\$4.31	\$0.00	\$76.22	Details
38	11/13/19 5:44 PM	Smith	Hillary	(313) 223-2323	smith@smith.com	\$7.99	\$0.48	\$0.00	\$8.47	Details
34	11/12/19 1:15 PM	George	George	(313) 232-2323	george@george.com	\$31.96	\$1.92	\$0.00	\$33.88	Details
33	11/07/19 2:30 PM	Fred	Fred	(313) 333-2323	fred@fred.com	\$45.94	\$2.76	\$0.00	\$48.70	Details
27	11/04/19 5:31 PM	Johnson	Mike	(313) 277-0000	johnson@johnson.com	\$15.96	\$0.96	\$0.00	\$16.94	Details

Extra Credit #2

Table Creator (4 pts)

Create a page called **table.php** that is a data entry form allowing a user to enter the number of rows they want in a table, the number of columns they want in a table, and the color they want the alternating rows to be.

Screenshot of table.php input page:

Table Creator

Number of rows:

Number of columns:

Alternating row background color:

When the user clicks the Submit button, this page will run the page **table2.php** which will create the HTML table with that many rows and columns and that alternating row color. Use embedded CSS for the alternating row color.

Row 1 Col 1	Row 1 Col 2	Row 1 Col 3	Row 1 Col 4	Row 1 Col 5
Row 2 Col 1	Row 2 Col 2	Row 2 Col 3	Row 2 Col 4	Row 2 Col 5
Row 3 Col 1	Row 3 Col 2	Row 3 Col 3	Row 3 Col 4	Row 3 Col 5
Row 4 Col 1	Row 4 Col 2	Row 4 Col 3	Row 4 Col 4	Row 4 Col 5

Be sure to validate and sanitize the user input being sent over.

```
$rows = substr(filter_input(INPUT_GET, 'rows', FILTER_SANITIZE_NUMBER_INT),0,3);  
$cols = substr(filter_input(INPUT_GET, 'cols', FILTER_SANITIZE_NUMBER_INT),0,3);  
$color = substr(filter_input(INPUT_GET, 'color', FILTER_SANITIZE_STRING),0,10);
```