

## Criterion C: Development

In order to make a website with PHP we need to create the webpages in HTML and CSS. After this step we will have static pages that will include places for the data to be shown. I created a list of pages such as: Home page which includes the restaurant logo and links to other pages, The Order page which includes pictures of the menu and a form that accepts user data for a new order, A check your order status page which is a simple page that you input your order ID which redirects you to the Order status page which gives you info about your order including the total amount due. After making the order related pages I created a few pages for the restaurant owner to see a list of orders and manipulate them by confirming, entering the total amount of the receipt and letting the customers know when their order is ready.

After making all those pages as static pages that will show the data with the PHP code implanted, I created my database.

Code to connect to the MYSQL database:


```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "PASS";
5
6  // connecting to database
7  $connect = new mysqli($servername, $username, $password);
8
9  // test the connection
10 if ($connect->connect_error) {
11     die("Connection failed: " . $connect->connect_error);
12 }
13 ?>
```

After making a connection to the MYSQL database I created my tables accordingly. I created two tables, orders and users. The orders table will include parameters such as: Varchar for customer's name, Integer for customer's phone number, Text for the body of the order, Boolean to check if the order is confirmed, Boolean to check if the order is ready, Double to hold the total amount for the order. The PHP and SQL code to create the table and the final created table as shown below for the orders:

```

1  <?php
2  // new table
3  $sql = "CREATE TABLE orders (
4  id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
5  name VARCHAR(255) NOT NULL,
6  phone INT NOT NULL,
7  order TEXT NOT NULL,
8  confirmed TINYINT(1) NOT NULL DEFAULT 0,
9  is_ready TINYINT(1) NOT NULL DEFAULT 0,
10 total DOUBLE,
11 reg_date TIMESTAMP
12 );";
13
14 if (!$connection->query($sql) === TRUE) {
15     echo "Error creating table: " . $connection->error;
16 }
17
18 $connection->close();
19 ?>

```

id 	int(10)		UNSIGNED	No	None	AUTO_INCREMENT
name	varchar(191)	utf8mb4_unicode_ci		No	None	
phone	int(11)			No	None	
order	text	utf8mb4_unicode_ci		No	None	
confirmed	tinyint(1)			No	None	
is_ready	tinyint(1)			No	None	
total	double			Yes	NULL	
created_at	timestamp			Yes	NULL	
updated_at	timestamp			Yes	NULL	

The users table will have the parameters: integer for ID, Varchar for user's name, Varchar for user's email and a Varchar for user's password. The table for users is shown below:

1	<b>id</b> 📱	int(10)		UNSIGNED	No	<i>None</i>	AUTO_INCREMENT
2	<b>name</b>	varchar(191)	utf8mb4_unicode_ci		No	<i>None</i>	
3	<b>email</b>	varchar(191)	utf8mb4_unicode_ci		No	<i>None</i>	
4	<b>password</b>	varchar(191)	utf8mb4_unicode_ci		No	<i>None</i>	
6	<b>created_at</b>	timestamp			Yes	<i>NULL</i>	
7	<b>updated_at</b>	timestamp			Yes	<i>NULL</i>	

After creating the database and the tables within the database it was time to make the HTML forms store data into the tables using PHP and SQL. The form (code included below) will call for the file ordersubmit.php and will send the data input from the user to the PHP code. The data will then be processed by the PHP method and be stored in the database.

HTML form code:

```

1      <form id="forms" class="needs-validation" action="/ordersubmit.php" method="POST">
2          <div class="mb-3">
3              <label for="name">Name</label>
4              <input type="text" class="form-control" id="name" name="name" placeholder="Name" required>
5          </div>
6
7          <div class="mb-3">
8              <label for="phone">Phone Number</label>
9              <input type="number" class="form-control" id="phone" name="phone" placeholder="phone" required>
10         </div>
11         <hr class="mb-4">
12         <h4 class="mb-3">Order info</h4>
13         <div class="row">
14             <div class="col-md-12 mb-3">
15                 <label for="order">Order</label>
16                 <textarea type="text" form_id="forms" name="order" class="form-control" id="order" placeholder="" required> </textarea>
17                 <div class="invalid-feedback">
18                     Order is required
19                 </div>
20             </div>
21         </div>
22         <hr class="mb-4">
23         <button class="btn btn-primary btn-lg btn-block" type="submit">Submit Order</button>
24     </form>

```

PHP storing the data in the database in ordersubmit.php:

```

1  <?php
2
3  $name = $_Post['name'];
4  $phone = $_Post['phone'];
5  $order = $_Post['order'];
6
7
8  try {
9      $host = "localhost";
10     $username = "root";
11     $password = "PASS";
12     $database = "my_db";
13     $dsn = "mysql:host=$host;dbname=$database";
14
15     $conn = new PDO( $dsn, $username, $password );
16     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
17
18     $sql = "INSERT INTO orders("
19         . "name, phone, order"
20         . " ) VALUES ("
21         . "'" . $name . "',"
22         . "'" . $phone . "',"
23         . "'" . $order . "'";
24
25     $conn->query($sql);
26 }
27 catch (PDOException $e) {
28     $conn = null;
29     exit("Connection failed: " . $e->getMessage());
30 }
31 ?>

```

After coding the PHP code to insert the data from the HTML form to the database I tested the code by making some sample orders and checked to ensure that data will save into the database. Now that storing the data was working, the next step was to retrieve the data from the database to show it to both the website administrator and the customer. Admin also needs to manipulate the data in the database to confirm the order, change the total amount and confirm that the order is ready for pickup.

Next step: Retrieving the data from the database to show to admin. Admin needs to see all the orders listed which needs the PHP and SQL code to retrieve all the rows within the orders table and show the info.

```
1  <?php
2
3  try {
4      $sql = "SELECT * FROM orders";
5      $userdata = $conn->query($sql);
6      echo '<table>';
7      echo '<tr>';
8      echo '<th>Name</th>';
9      echo '<th>Phone</th>';
10     echo '<th>Order</th>';
11     echo '<tr>';
12     foreach ($userdata as $userdata) {
13         echo '<tr>';
14         echo '    <td>' . $userdata['name'] . '</td>';
15         echo '    <td>' . $userdata['phone'] . '</td>';
16         echo '    <td>' . $userdata['order'] . '</td>';
17         echo '    <td>' .
18             '<a href="confirm.php">confirm</a>'
19             '<a href="ready.php">ready</a>'
20             '<form action="total.php">'
21             '<input type="text" name="total">'
22             '<button type="submit">submit</button>'
23             '</form>' . '</td>';
24         echo '    </tr>';
25     }
26
27     echo '</table>';
28
29     $conn = null;
30 }
31 catch (PDOException $e) {
32     $conn = null;
33     exit("Connection failed: " . $e->getMessage());
34 }
35 ?>
```

The code above will run a loop for each row in orders table which prints out the name and number for the customer and two buttons to confirm and change the status to ready plus a form that you

can input the total into. This loop will run for each row and print out the data as many times as needed until there's no data left in the database to show.

The next codes that were required is the methods to change the data for a specific row in the table. The website admin needs the ability to change the confirmed parameter from 0 to 1, change the is\_ready parameter from 0 to 1, and change the total parameter to any given amount. The PHP and SQL code the update the mentioned info in the database is as below (example updating confirmed to 1):

```
1  <?php
2
3  $sql=mysql_query("SELECT * FROM orders WHERE id='$id'");
4  if($data=mysql_fetch_array($sql))
5  {
6      $sqla="UPDATE orders SET confirmed=1";
7  }
8
9
10 <?php
11 $servername = "localhost";
12 $username = "root";
13 $password = "PASS";
14 $dbname = "myDB";
15
16 $conn = mysqli_connect($servername, $username, $password, $dbname);
17 if (!$conn) {
18     die("Connection failed: " . mysqli_connect_error());
19 }
20
21 $sql = "UPDATE orders SET confirmed=1 WHERE id='$id'";
22
23 if (!mysqli_query($conn, $sql)) {
24     echo "Error updating record: " . mysqli_error($conn);
25 }
26 mysqli_close($conn);
27 ?>
```

Similar codes were used to update the is\_ready to 1 and total to the given amount. With this step coded and successfully running, the admin area was completed and was given all the required abilities. The last step was to create a form that includes a form getting the order id from the customer and retrieving the order data for them.

The form is simple and includes one input: order ID. Then the PHP and SQL code searches throughout the database to find a record with the same ID and shows the data from that row to the customer.

HTML form:

```
1  enter your order ID
2  <form action="/status" method="POST">
3      <input type="number" name="order">
4      <button type="submit">submit</button>
5  </form>
```

PHP Code to retrieve the data and show it:

```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "PASS";
5  $dbname = "myDB";
6
7
8  $db = new mysqli($servername, $username, $password, $dbname);
9
10 $id = $_POST['id'];
11
12 $query = "SELECT * FROM orders WHERE `id`='" . $id . "'";
13 $result = mysqli_query($db, $query);
14
15 echo $row['id']; while($row = mysqli_fetch_array( $result )) {
16
17     echo "<br><br>";
18     echo $row['name'];
19     echo $row['order'];
20     echo $row['phone'];
21     echo $row['confirmed'];
22     echo $row['is_ready'];
23     echo $row['total'];
24 }
25 ?>
```

Word Count: 773