Please note: -

I have not used XAMPP before and so have not experimented with transferring local servers onto other computers. Please make contact via Lidia at EDT should you require my system in a different format.

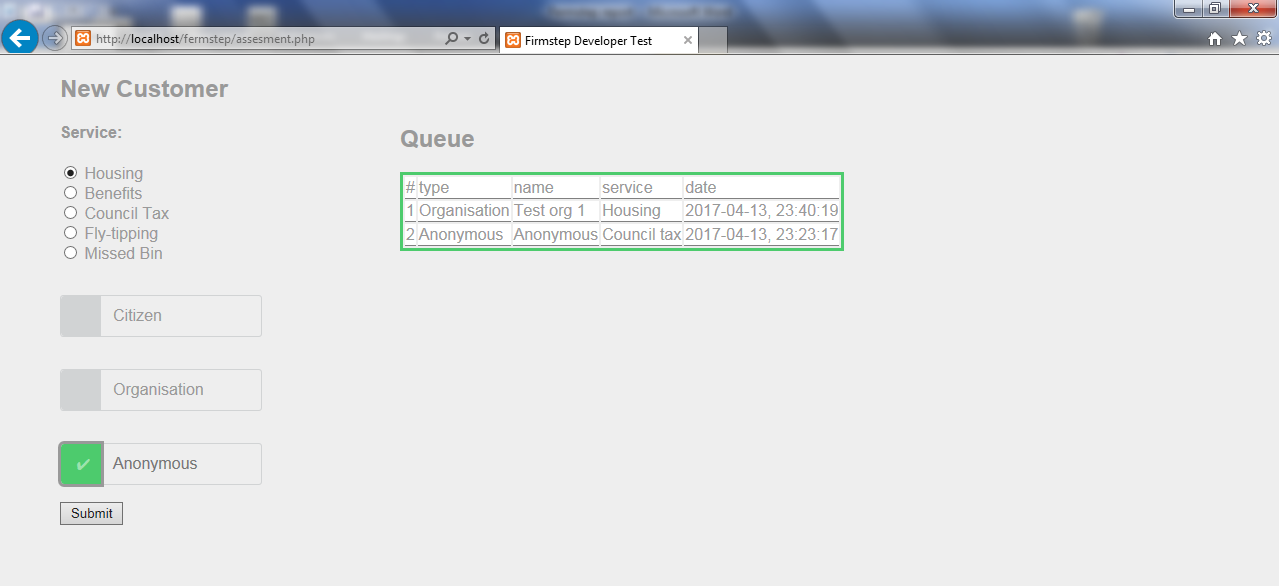
**Summary of languages and frameworks used**

I used a combination of PHP, HTML, CSS, and JavaScript to tackle this problem. HTML was used to construct the skeleton of the webpage which was then styled by CSS to make it more presentable. JavaScript was used to hide various sections of the form that were only appropriate if the user selected certain radio buttons. PHP was used as an interface between the database and the user input form and made it possible to save the queue over multiple sessions. Furthermore, I used a MySQL database to save the details that the user inputs into the system and to display them again in a table queue.

I installed XAMPP for this task, which provided me with a localhost server that allowed me to use PHP on my laptop. It was interesting re-familiarising myself with this server side language which I haven’t used since GCSE.

**Testing**

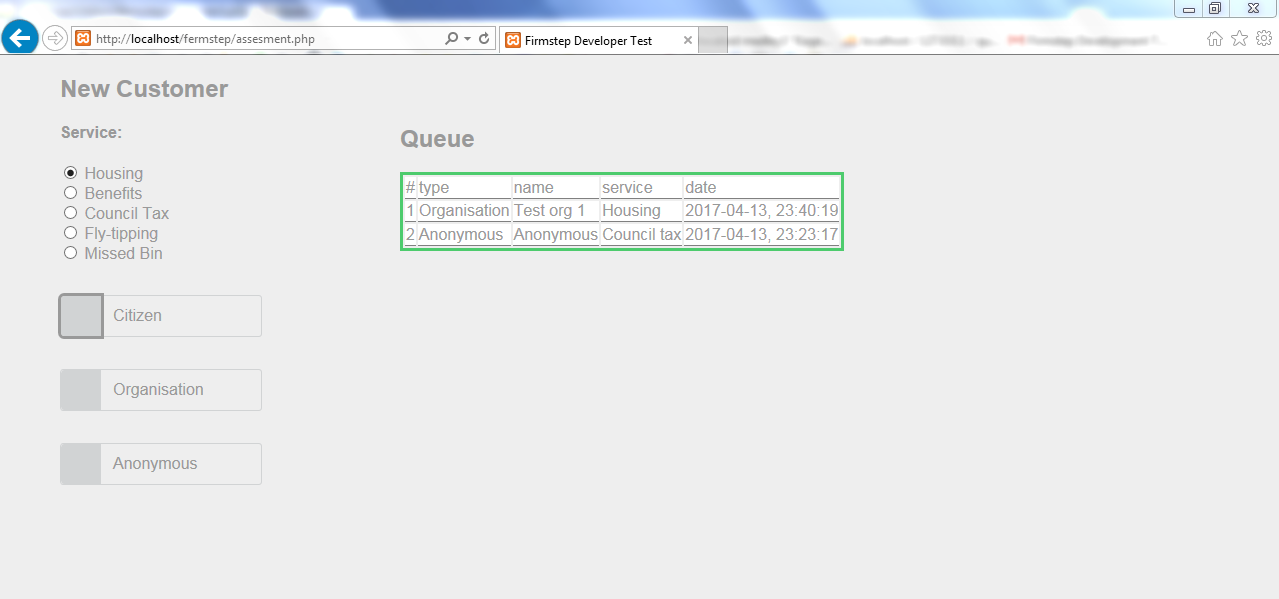
For all tests, the webpage initially looked like this before it was performed.



**Test #1.**

**Table displays data from previous session:**

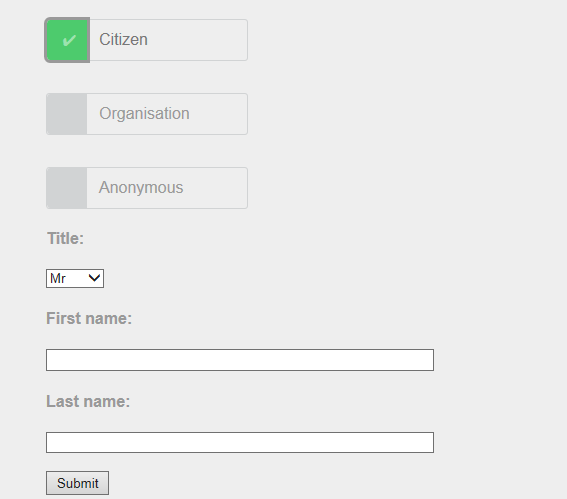
Each time the webpage is loaded, the table with the user details that have been previous entered, are displayed.



**Test #2.**

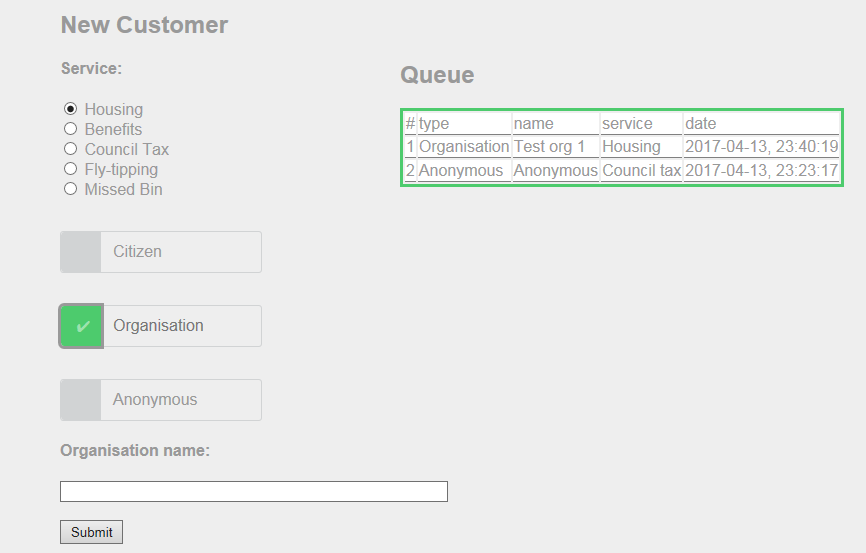
**Sub menus are displayed when the user selects appropriate sub form.**

1. When the user selects the Citizen Button, the sub-form with the correct entries is displayed.



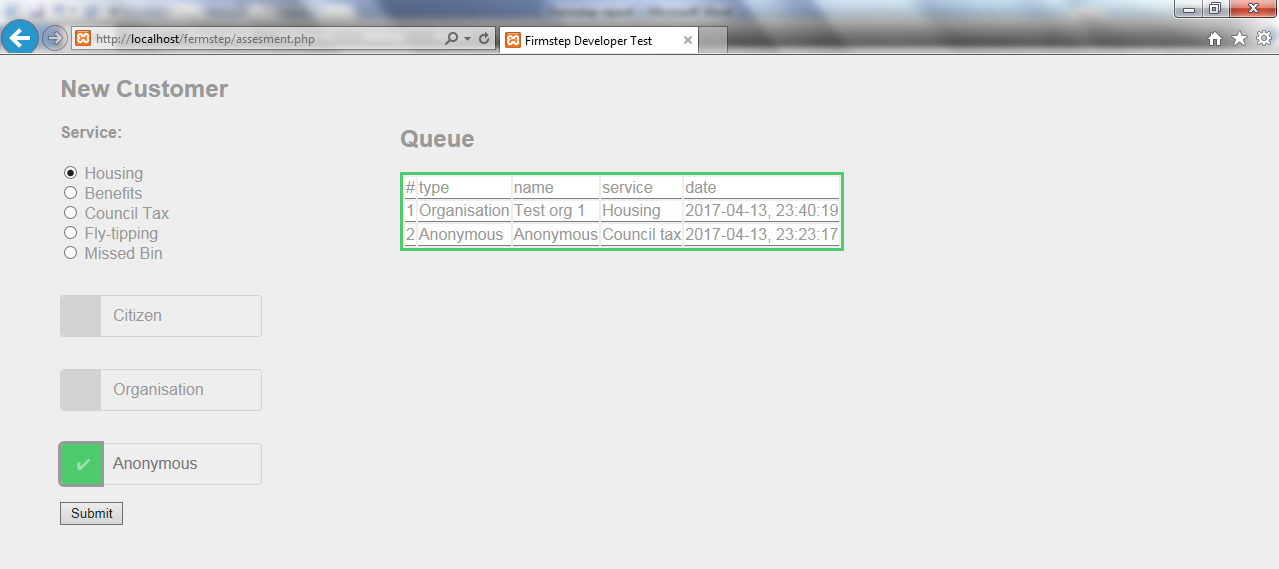
Sub form is displayed as expected.

1. When the user selects the Organisation button the sub-form with the correct entries is displayed.



Appropriate sub-form is displayed as expected.

1. When the user selects the Anonymous button no sub-form is displayed (apart from a submit button).



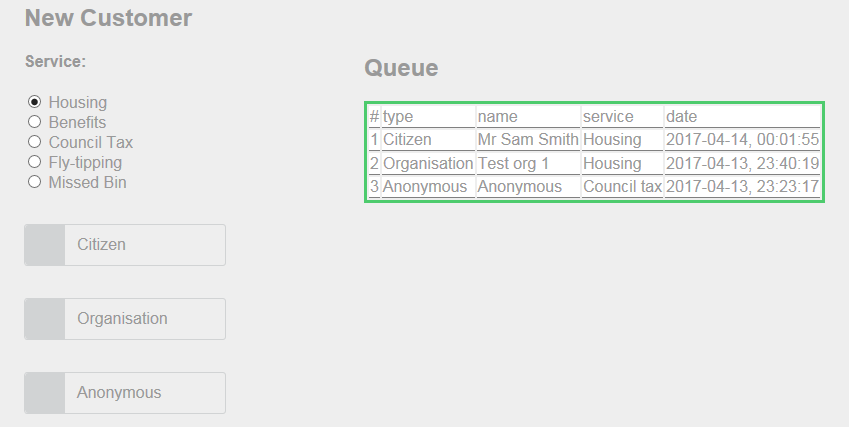
Only the submit button is displayed as expected.

**Test #3.**

**Data is added to the table (and hence the database).**

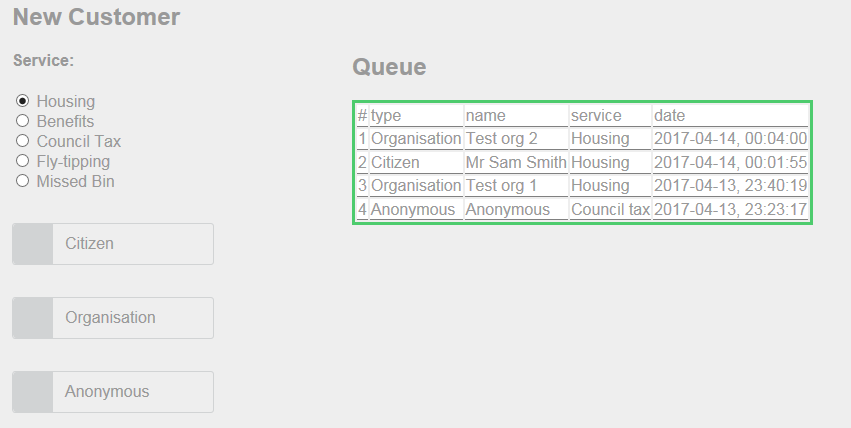
The appropriate data should be added to the Queue when:

1. The citizen form is filled in.



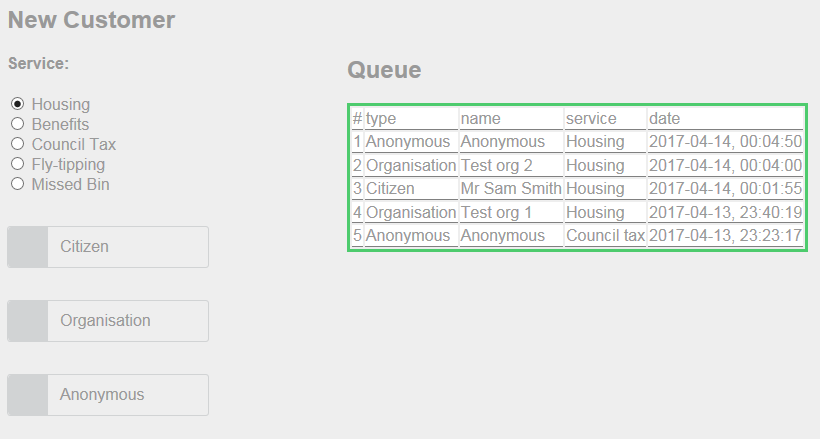
Appropriate data is added to the table as expected.

1. The organisation form is filled in.



Appropriate data is added to the table as expected.

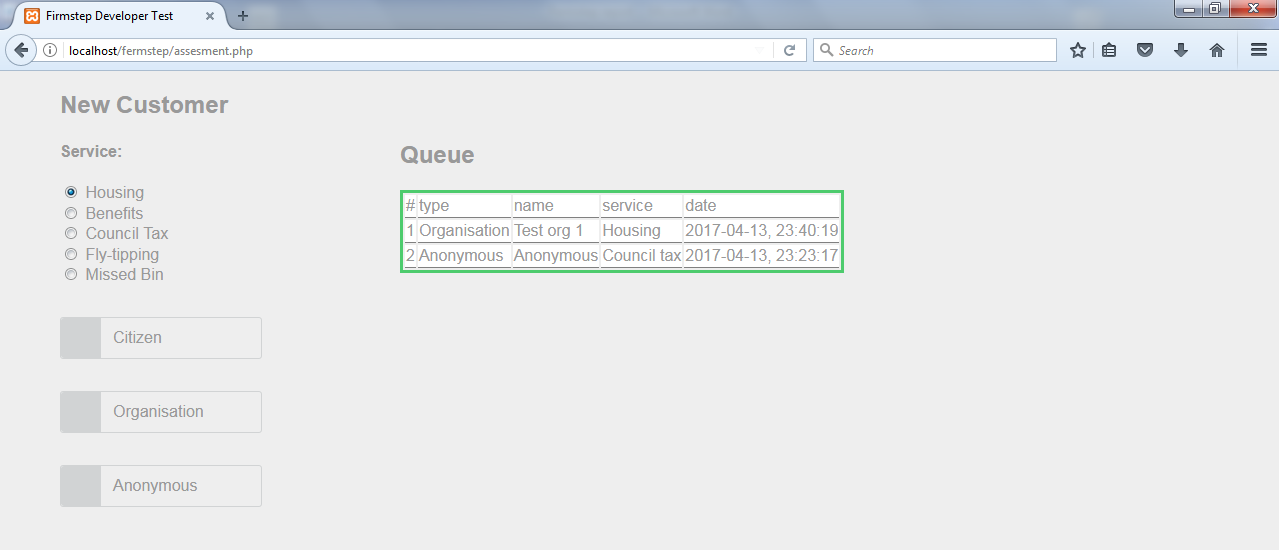
1. The submit button when the anonymous button has been clicked.



Appropriate data is added to the table as expected.

**Test #4**

**Webpage is displayed in the same format in an alternative web browser.**



Webpage displays correctly as expected.

**Possible extensions:**

The most noteworthy extension is input validation. I tend to prioritise this in the programs I write, however, due to time constraints I focused on the main functionality of the system rather than focus on the finer details. I do however, understand the importance of validation as some algorithms only work with a set of inputs of a specific data type and/or format. In this case is was not as important as, for example, the validation of an adjacency matrix which Dijkstra’s shortest path algorithm will use. This is because the user inputs were only stored in a database and displayed back to the user. This would work with data of any format.

It would have been a good idea to limit the number of characters of the inputs for the user’s name or the organisation name to 16 characters.

An addition feature that I would like to add to the system is a de-queue button which will remove items from the table that are displayed to the user (and ultimately removes a record from the database).

Technical solution:

assessment.php:

<!DOCTYPE html>

<html>

<head>

<title>Firmstep Developer Test</title>

<?php

// Setting up variable to connect to the MySQL database

$host = "localhost";

$user = "root";

$pass = "";

$db\_name = "Queue";

// Establishing connection to the database

$connection = mysqli\_connect($host, $user, $pass, $db\_name);

// A small test to check if the connection has failed.

if(mysqli\_connect\_errno()){

die("connection failed: "

. mysqli\_connect\_error()

. " (" . mysqli\_connect\_errno()

. ")");

}

// Get all data from database to display in a table.

$result = mysqli\_query($connection,"SELECT \* FROM Queue ORDER BY `id` DESC");

$all\_property = array(); // Data will be stored in array

// A series of 'div's are set up for formatting purposes.

echo '<div id="float">';

echo '<h2>Queue</h2><div id="smallfloat">';

// Creating table for data to be displayed in.

echo '<table class="data-table">

<tr class="data-heading">';

# While loop is used to get all table column names (and put them into table).

while ($property = mysqli\_fetch\_field($result)) {

if ($property->name == "id"){

echo '<td>' . "#" . '</td>'; //get field name for header

}

else{

echo '<td>' . $property->name . '</td>'; // Obtains field name for header

}

array\_push($all\_property, $property->name); // Save sthese to items to array

}

echo '</tr>'; // End table row tag.

# Indefinite iteration ensures every database table record (row) is inputted into table.

$count = 0;

while ($row = mysqli\_fetch\_array($result)) {

$count +=1;

echo "<tr>";

for ($i = 0; $i<5;$i++){

# Selection is used to put queue number into table.

if ($i == 0){

echo '<td>' . $count . '</td>';

}

else{

echo '<td>' . $row[$i] . '</td>';

}

}

echo '</tr>';

}

echo "</table>";

echo "</div></div>";

?>

<script type="text/javascript">

// Following function is used to display and hide sections of the form.

function displayForm(c){

if(c.value == "1"){

// style.display is used to insead of 'visible' property so that it does not take up space.

document.getElementById("citizen").style.display ='inline';

document.getElementById("organisation").style.display='none';

document.getElementById("anon").style.display='none';

}

else if(c.value =="2"){

document.getElementById("citizen").style.display ='none';

document.getElementById("organisation").style.display='inline';

document.getElementById("anon").style.display='none';

}

else{

document.getElementById("citizen").style.display ='none';

document.getElementById("organisation").style.display='none';

document.getElementById("anon").style.display='inline';

}

}

</script>

<style>

body {

font-family: sans-serif;

font-weight: normal;

margin: 10px;

color: #999;

background-color: #eee;

}

form {

margin: 0 50px;

}

#check{

clear: both;

}

label {

width: 200px;

border-radius: 3px;

border: 1px solid #D1D3D4

}

input.radio:empty {

margin-left: -999px;

}

input.radio:empty ~ label {

position: relative;

float: left;

line-height: 2.5em;

text-indent: 3.25em;

margin-top: 2em;

cursor: pointer;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

input.radio:empty ~ label:before {

position: absolute;

display: block;

top: 0;

bottom: 0;

left: 0;

content: '';

width: 2.5em;

background: #D1D3D4;

border-radius: 3px 0 0 3px;

}

input.radio:hover:not(:checked) ~ label:before {

content:'\2714';

text-indent: .9em;

color: #C2C2C2;

}

input.radio:hover:not(:checked) ~ label {

color: #888;

}

input.radio:checked ~ label:before {

content:'\2714';

text-indent: .9em;

color: #9CE2AE;

background-color: #4DCB6D;

}

input.radio:checked ~ label {

color: #777;

}

input.radio:focus ~ label:before {

box-shadow: 0 0 0 3px #999;

}

#title{

width:380px;

}

#float{

position: absolute;

left: 400px;

top: 50px;

}

#smallfloat{

border: 3px solid #4DCB6D;

}

th, td {

border-bottom: 1px solid grey;

background-color: white;

}

</style>

</head>

<body>

<!-- Initial form that is always displayed to user is below, followed by three sections of

the form which are hidden.

Each hidden section will be displayed if and only if the user selects the corresponding radio button -->

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

<h2>New Customer</h2>

<h4>Service:</h4>

<input type="radio" name="queue" value="Housing" checked> Housing<br>

<input type="radio" name="queue" value="Benefits"> Benefits<br>

<input type="radio" name="queue" value="Council tax"> Council Tax<br>

<input type="radio" name="queue" value="Fly-tipping"> Fly-tipping<br>

<input type="radio" name="queue" value="Missed bin"> Missed Bin

<div id = "check">

<input type="radio" name="radio" id="radio1" class="radio" value = "1" onClick="displayForm(this)"/>

<label for="radio1">Citizen</label>

</div>

<div id = "check">

<input type="radio" name="radio" id="radio2" class="radio" value = "2" onClick="displayForm(this)"/>

<label for="radio2">Organisation</label>

</div>

<div id = "check">

<input type="radio" name="radio" id="radio3" class="radio" value = "3" onClick="displayForm(this)"/ >

<label for="radio3">Anonymous</label>

</div>

<br>

<p> </p>

<br>

<div id = "citizen" style = "display:none">

<h4>Title:</h3>

<select name = "title">

<option value="Mr">Mr</option>

<option value="Miss">Miss</option>

<option value="Mrs">Mrs</option>

<option value="Ms">Ms</option>

<option value="Other">Other</option>

</select>

<h4>First name:</h3>

<input size = "60" type="text" name="firstname"><br>

<h4>Last name:</h3>

<input size = "60" type="text" name="lastname"><br>

<br>

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

</div>

<div id = "organisation" style = "display:none">

<h4>Organisation name:</h3>

<input size = "60" type="text" name="orgname"><br>

<br>

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

</div>

<div id = "anon" style = "display:none">

<br>

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

</div>

</form>

</body>

</html>

process.php

<?php

// Setting up variable to connect to the MySQL database.

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "Queue";

$date = (new \DateTime())->format('Y-m-d, H:i:s');

// Establishing connection to the database.

$conn = new mysqli($servername, $username, $password,$dbname);

// A small test to check if the connection has failed.

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Following code executes if user filled out the organisation subform.

if ($\_POST["orgname"] != ""){

# Name and service variables will be used in the query to insert user data into database table.

$name = $\_POST["orgname"];

$service = $\_POST["queue"];

$sql = "INSERT INTO Queue (type, name, service, date)

VALUES ('Organisation', '$name', '$service','$date')";

if ($conn->query($sql) === TRUE) {

echo "New record created successfully";

} else {

# Error message is displayed if query is not successfully executed.

echo "Error: " . $sql . "<br>" . $conn->error;

}

}

// Following code executes if user filled out the citizen subform.

else if ($\_POST["firstname"] != ""){

# Name and service variables will be used in the query to insert user data into database table.

$name = $\_POST["title"]." ".$\_POST["firstname"] ." ". $\_POST["lastname"];

$service = $\_POST["queue"];

$sql = "INSERT INTO Queue (type, name, service, date)

VALUES ('Citizen', '$name', '$service','$date')";

if ($conn->query($sql) === TRUE) {

echo "New record created successfully";

} else {

# Error message is displayed if query is not successfully executed.

echo "Error: " . $sql . "<br>" . $conn->error;

}

}

// Following code executes if user filled out the anonymous subform.

else{

# Service variable will be used in the query to insert user data into database table.

$service = $\_POST["queue"];

$sql = "INSERT INTO Queue (type, name, service, date)

VALUES ('Anonymous', 'Anonymous', '$service','$date')";

if ($conn->query($sql) === TRUE) {

echo "New record created successfully";

} else {

# Error message is displayed if query is not successfully executed.

echo "Error: " . $sql . "<br>" . $conn->error;

}

}

$conn->close();

$newURL = "http://localhost/fermstep/assesment.php";

# Redirects browser back to initial page where the new details are added to the table.

header('Location: '.$newURL);

?>