Criterion C

Methods & Algorithms:

A) Nested If-Statements to Redirect User

B) Writing (Inserting) into MySQL database

C) Reading from MySQL database through while loop nested into If/Else-statement

D) Varied Array in dependance of If/Else-statement

Word Count: 996

Technique #1 - Nested If-Statements to Redirect User

Source Code

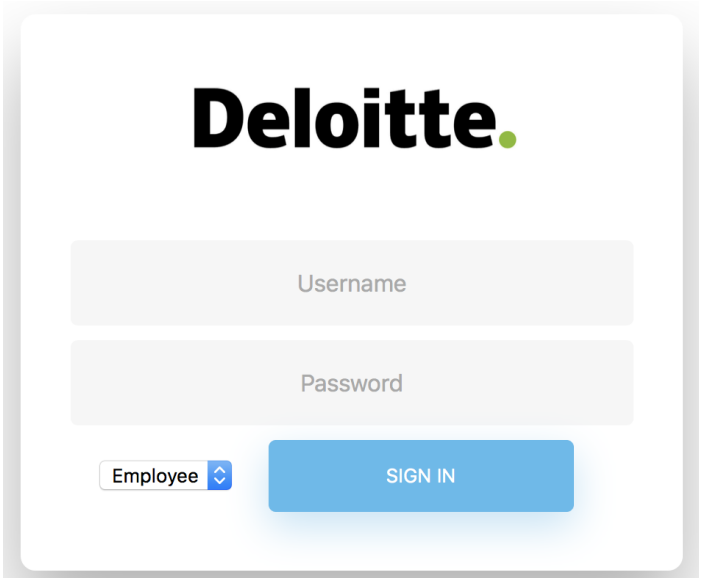
(Technique shown in the last screen capture. The other screen captures are only for context.)

```
    <select name="db">
        <option value="manager_details">Manager</option>
        <option value="date">Employee</option>
    </select><br>
```

```
$dbn = $_GET['db'];

// Connection to database
$servername = "localhost";
$username = $_GET['username'];
$password = $_GET['password'];
$dbname = $dbn;

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

if ($conn -> connect_error) {
    die("Incorrect Username or Password");}
```

```
// Redirect && Nested if statements (1)
if (isset($_GET['Login'])){
    if(!empty($dbn)){
        if($dbn == "date"){
?>

<script type="text/javascript">
window.location = "
    http://localhost:8888/IA2.php?name=Kevin&OL=Waleed+Farooq&month=March&day=10&year=2021&CH=15&NCH=30"
</script>

<?php
} elseif ($dbn == "manager_details") {
?>

<script type="text/javascript">
window.location = "http://localhost:8888/IA3.php?month=march&day=10&year=2021"
</script>

<?php
}}}
?>
```

User Interface



Explanation	The first screen capture essentially uses HTML to create a dropdown for the user. However, the values given to the drop down options are the title of databases within phpmyadmin. The second screen capture displays how the value of the selected option from the drop down is stored within a variable. Which is then used to connect to the desired database if both the password and username are identified correctly. The final screen capture displays the use of php nested if-statements as well as Java to redirect the user to the associated page in accordance to their decision on the drop down. It uses if-statements to identify which value is in accordance to the user's decision and uses Java's "window.location" function to redirect users to the page parallel to the value.
Justification	This code was produced because my client wanted to redirect the user to a specific page in accordance to their position at the company. The use of the drop down essentially helps identify which position the user is and then redirects them to the set pages associated with that position.

Technique #2 - Writing (Inserting) into MySQL Database

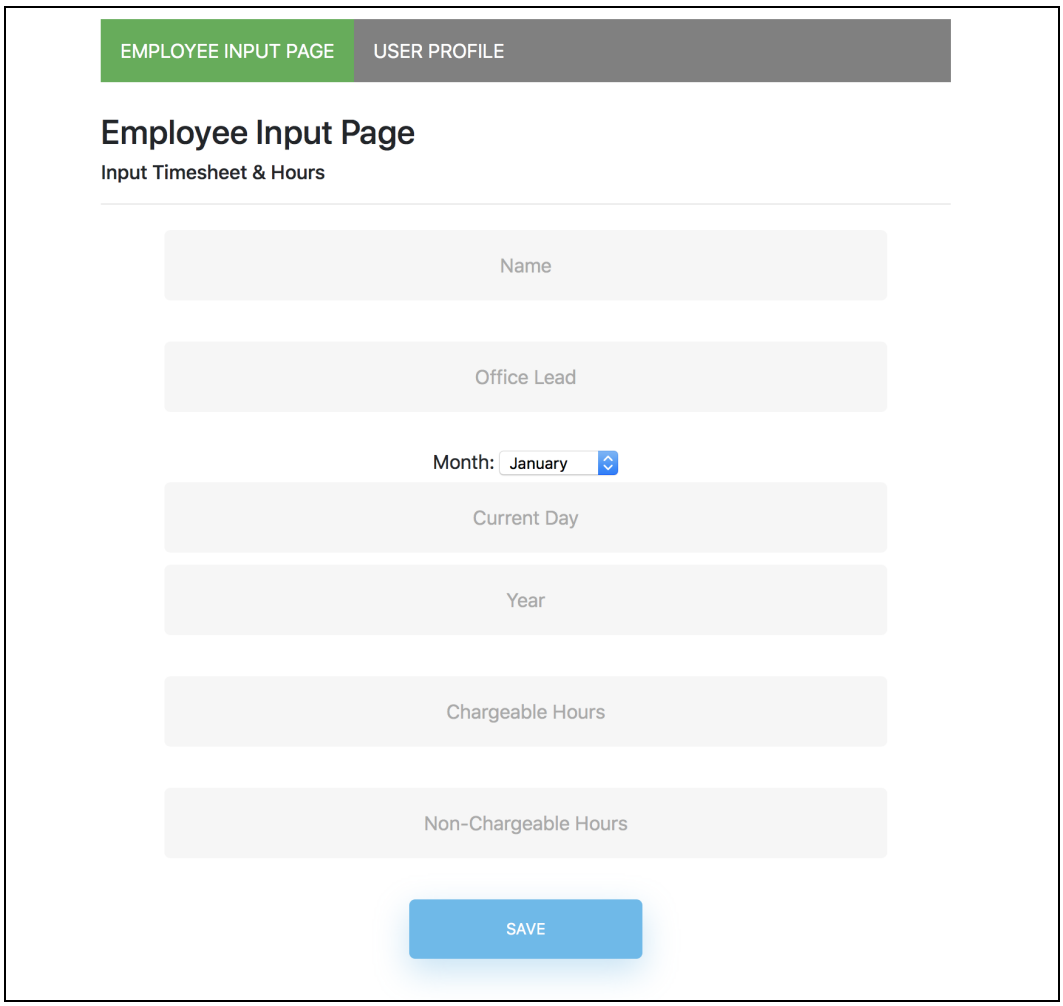
Source Code

(Technique shown in the second screen capture. The other screen captures are only for context.)

```
<form action = "IA2.php" method = "get">

Name:<input type="text" name="name"><br>
Office Lead:<input type="text" name="OL"><br>
Date:<select name="month"> ... </select><input type="text" name="day"><input type="text" name="year"><br>
Chargeable Hours:<input type="text" name="CH"><br>
Non_Chargeable Hours:<input type="text" name="NCH"><br>
<input type="Submit" Value="Save">

</form>
```

```
// Connectiton to Database
$servername = "localhost";
$username = "Izum_1";
$password = "messi2015";
$dbname = "date";

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

if ($conn -> connect_error) {
    die("Connection Failed: " . $conn -> connect_error);}

//Insert Query(2)
$name = $_GET['name'];
$OL= $_GET['OL'];
$d = $_GET['month'].$_GET['day']. $_GET['year'];
$ch= $_GET['CH'];
$nch= $_GET['NCH'];
$th = $ch + $nch;
$c= $ch/$th*100;

$sql = "INSERT INTO `$d`(`Name`, `office_lead`, `c_hours`, `nc_hours`, `t_hours`, `charge`) VALUES ('$name', '
    $OL', '$ch', '$nch', '$th', '$c')";


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

if($result){

    echo "data inserted";
} else {

    echo "not inserted";
}
```

	Name	office_lead	c_hours	nc_hours	t_hours	charge
Edit Copy Delete	Heejae Park	Waleed Farooq	30	50	80	37.5
Edit Copy Delete	Izum Adnan	Waleed Farooq	15	30	45	5
Edit Copy Delete	Kevin	Waleed Farooq	15	30	45	33.333333333333
Edit Copy Delete	Mousa	Waleed Farooq	15	200	215	0.069767441860465
Edit Copy Delete	Youssef Tarek	Waleed Farooq	15	30	45	33.333333333333

User Interface	
Explanation	The first screen capture solely shows the values associated with the different types of HTML inputs such as; text boxes, drop downs and the submit button.(The folded code is just the options for the month dropdown. I chose to do a drop down instead of a text field because it makes it easier for the user. The folded code is the same throughout the document.) The second screen capture first shows how a connection was made to the "date" database. It then shows how the values associated with the different types of inputs were then stored within php variables. This was done so that it is easier for the manipulation of the Insert SQL query afterwards. It also shows two variables ("$th", "$c") which incorporate simple mathematical functions to meet the client's need of pre-calculating the total hours and chargeability of the employee's. Finally, it shows the use of the Insert SQL query. However, certain positions such as the position in which the table from the database is mentioned as well as the position of all the values -- they are replaced by the variables which store the user's inputs.
Justification	This code was produced as my client wanted a simple system in which all the employee time sheets would be located all organized and pre-calculated. I chose to manipulate the data and inputs through php as it easily communicates with MySQL databases. Furthermore, the reason for choosing to store all the data within a MySQL database is because it is a very straightforward and routine system to understand and operate -- it does not require any programming knowhow.

Technique #3 - Reading from MySQL database through while loop nested into If/Else-statement

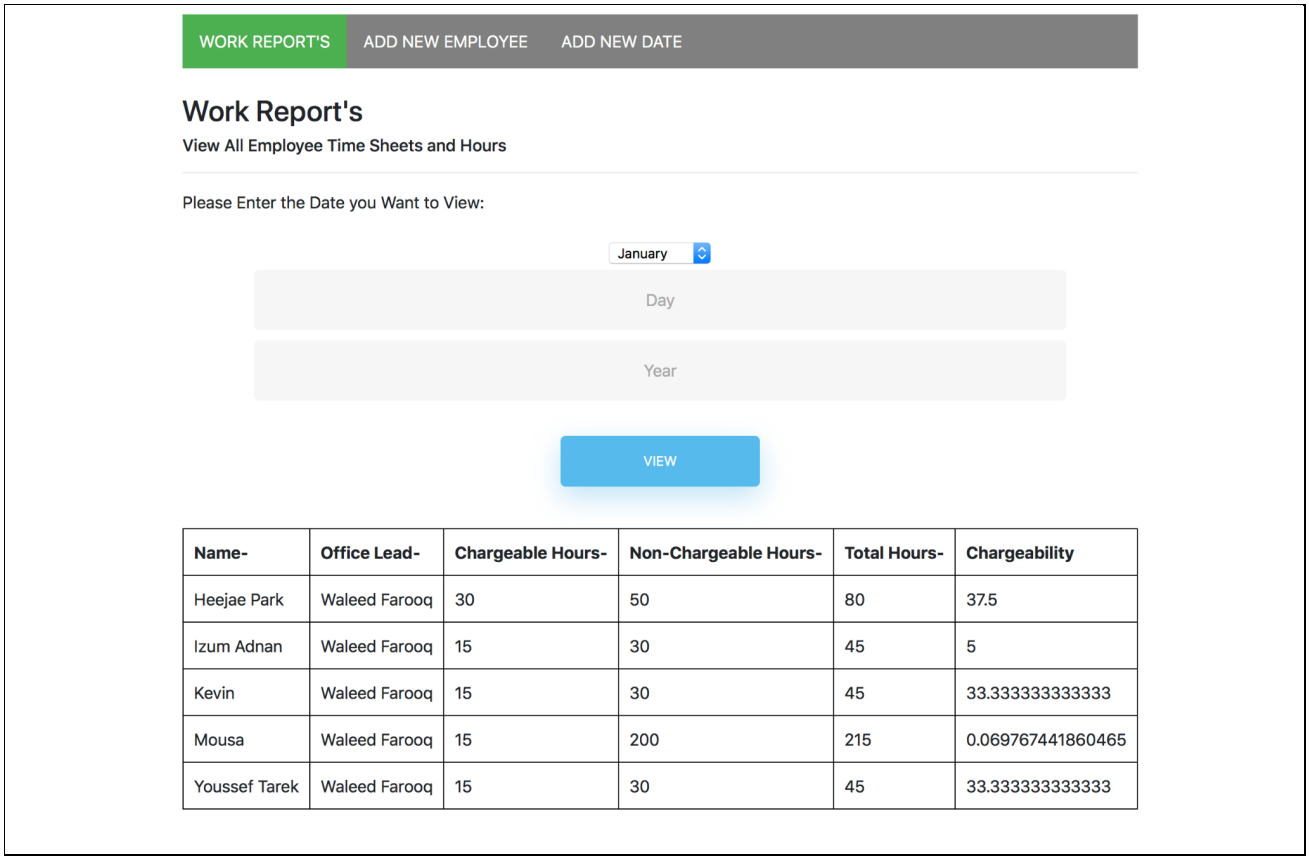
Source Code

(Technique shown in the last screen capture. The other screen captures are only for context.)

```
<form>
Date:<select name="month"> ... <input type="text" name="day"><input type="text" name="year"><br>
    <input type="submit" value="View">
</form>
```

```
<?php
// Connection to MySQL Database
$servername = "localhost";
$username = "Izum_1";
$password = "messi2015";
$dbname = "date";

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

if ($conn -> connect_error) {
    die("Connection Failed: " . $conn -> connect_error);}

$month = $_GET['month'];
$day = $_GET['day'];
$year = $_GET['year'];
$dob = $month.$day.$year;
?>
```

```
<table>
    <tr>
        <th>Name</th>
        <th>Office Lead</th>
        <th>Chargeable Hours</th>
        <th>Non-Chargeable Hours</th>
        <th>Total Hours</th>
        <th>Chargeability</th>
    </tr>

<?php

// Reading from MySQL database through while loop nested into If Statement (3)
$sql = "SELECT * FROM $dob";
$result = $conn->query($sql);


if ($result->num_rows>0){
    while($row = $result-> fetch_assoc()){
        echo "<tr><td>". $row["Name"]. "</td><td>". $row["office_lead"]. "</td><td>". $row["c_hours"]. "
            </td><td>". $row["nc_hours"]. "</td><td>". $row["t_hours"]. "</td><td>". $row["charge"]. "
            </td></tr>";
    } echo "</table>";
} else {

    echo mysqli_error($conn);
}


?>
```

User Interface	 **Name-** / **Office Lead-** / **Chargeable Hours-** / **Non-Chargeable Hours-** / **Total Hours-** / **Chargeability** Heejae Park / Waleed Farooq / 30 / 50 / 80 / 37.5 Izum Adnan / Waleed Farooq / 15 / 30 / 45 / 5 Kevin / Waleed Farooq / 15 / 30 / 45 / 33.333333333333 Mousa / Waleed Farooq / 15 / 200 / 215 / 0.069767441860465 Youssef Tarek / Waleed Farooq / 15 / 30 / 45 / 33.333333333333
Explanation	The first screen capture is solely just 3 inputs which work parallel to each other as they all produce different sections of a date. The inputs are then stored into variables as shown in the second screen capture as it makes programming a lot more efficient and streamline. The variable is then placed within the Select SQL query so whatever date the user inputs - the data from that day would be displayed. The last screen capture shows the use of both HTML and php alongside each other. HTML is used to create the format of a table whereas php is used to communicate with the databases through the Select SQL query. The technique displayed to read and retrieve data is primarily through a while loop which continues to loop through the length of the table. The contents within the while loop consists of HTML which continues formatting the table whilst the php code within the loop retrieves the data. The If/else-statement in which the loop is nested in is primarily there to make certain that there is a table within the database.
Justification	This code was produced to further ease the client's understanding of the product. This code essentially displays the table in the MySQL database in a more simple form in which they only have to input a date. This way the client does not have to undergo the hassle of trying to understand phpmyadmin and the MySQL database instead just receives the data once they input the desired date.

Technique #4 - Varied Array in dependance of If/Else-statement

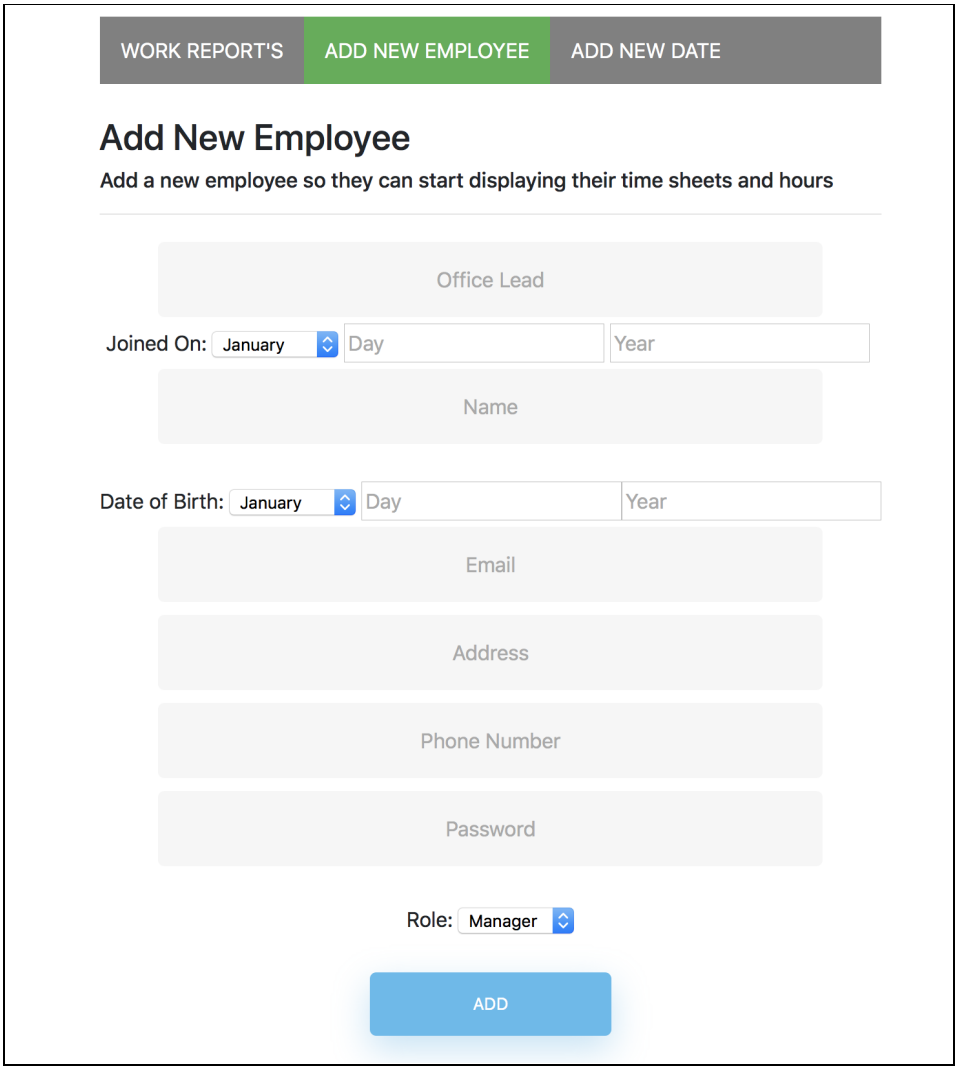
Source Code

(Technique shown in the last screen capture. The other screen captures are only for context.)

```
<form>
    Office Lead:<input type="text" name="office_lead"><br>
    Joined On:<select name="month"> ... </select><input type="text" name="day"> <input type="text" name="year">
    <br>
    Name:<input type="text" name="name"><br>
    Date of Birth:<select name="m"> ... <input type="text" name="d"><input type="text" name="y"><br>
    Email:<input type="text" name="email"><br>
    Address:<input type="text" name="address"><br>
    Phone Number:<input type="text" name="number"><br>
    Password:<input type="text" name="password"><br>
    Role:<select name="db">
            <option value="manager_details">Manager</option>
            <option value="employee_details">Employee</option>
        </select><br>
    <input type="submit" value="Add">
</form>
```

```
$servername = "localhost";
$username = "Izum_1";
$password = "messi2015";
$dbname = $role[0];

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

if ($conn -> connect_error) {
    die("Connection Failed: " . $conn -> connect_error);}

$OF = $_GET['office_lead'];
$joined = $_GET['month'].$_GET['day'].$_GET['year'];
$name = $_GET['name'];
$DOB = $_GET['m'].$_GET['d'].$_GET['y'];
$email = $_GET['email'];
$address = $_GET['address'];
$number = $_GET['number'];
$password = $_GET['password'];
```

```
//Varied Array in dependance of If Statement (4)
$role[0] = $_GET['db'];

if ($role[0] == "manager_details") {
    $role[1] = "MD";
} elseif($role[0] == "employee_details") {
    $role[1] = "ED";
}

$sql = "INSERT INTO $role[1](`name`, `DOB`, `email`, `address`, `office_lead`, `phone_number`) VALUES ('$name'
    ,'$DOB','$email','$address','$OF','$number')";

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

if($result){

    echo "data inserted";
} else {

    echo mysqli_error($conn);
}
?>
```

User Interface	
Explanation	The first screen capture displays multiple input types through HTML - all who have a specific value. These values are then stored within variables as shown in the second screen capture. This was done to retrieve the user's inputs as well as make the code more efficient by storing them into variables. The last screen capture shows an array with two strings. - The first string is essentially the database in accordance to the new employee's position within the company. The database is allocated through the selection of the dropdown options for the "Role" section in which each option contains the value of it's associated database(Eg: Role: Manager - Database: manager_details). The first string of the variable is then placed within the database name variable ($dbname) in the connection shown in the second screen capture which then connects to the desired database. - The second string varies between two options because of an if/else-statement. It holds the only table of the database within the first string (Both databases only have one table). It uses the if/else-statement to essentially change it's sequence of characters when a specific role is chosen. For example, if the option "Manager" is chosen within the role dropdown, the first string would be set to "manager_details" whereas the second string would be set to "MD" (The table's name within the database). This works vice versa.

Justification	This code was produced to also make the client's understanding of the system a bit more easier. This code essentially inputs the information of the new employee into the database associated with their position within the company through a dropdown. This way the client does not have to understand how to input data within different databases through phpmyadmin but only needs to input their data within this page and choose the associated role.