

Experiment-08

- a. Develop a PHP program (with HTML/CSS) to keep track of the number of visitors visiting the web page and to display this count of visitors, with relevant headings.
- b. Develop a PHP program (with HTML/CSS) to sort the student records which are stored in the database using selection sort.

Program:

Index.php

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Visitor Counter</title>
```

```
    <style>
```

```
        body {
```

```
            font-family: Arial, sans-serif;
```

```
            line-height: 1.6;
```

```
            margin: 0;
```

```
            padding: 20px;
```

```
background-color: #f4f4f4;

}

.container {

    max-width: 600px;

    margin: auto;

    background: white;

    padding: 20px;

    border-radius: 5px;

    box-shadow: 0 0 10px rgba(0,0,0,0.1);

}

h1 {

    color: #333;

    text-align: center;

}

.counter {

    font-size: 24px;

    text-align: center;

    margin-top: 20px;

}
```

</style>

</head>

<body>

<div class="container">

<h1>Welcome to Our Website</h1>

<div class="counter">

<?php

\$counterFile = 'visitor_count.txt';

// Read the current count

if (file_exists(\$counterFile)) {

\$count = (int)file_get_contents(\$counterFile);

} else {

\$count = 0;

}

// Increment the count

\$count++;

```
// Save the new count

file_put_contents($counterFile, $count);


// Display the count

echo "<h2>Visitor Count</h2>";

echo "<p>You are visitor number: $count</p>";

?>

</div>

</div>

</body>

</html>


student_records.php

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Student Record Sorter</title>
```

<style>

body {

font-family: Arial, sans-serif;

line-height: 1.6;

margin: 0;

padding: 20px;

background-color: #f4f4f4;

}

.container {

max-width: 800px;

margin: auto;

background: white;

padding: 20px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0,0,0,0.1);

}

h1 {

color: #333;

text-align: center;

```
}  
  
table {  
  
    width: 100%;  
  
    border-collapse: collapse;  
  
    margin-top: 20px;  
  
}  
  
th, td {  
  
    padding: 10px;  
  
    border: 1px solid #ddd;  
  
    text-align: left;  
  
}  
  
th {  
  
    background-color: #f2f2f2;  
  
}  
  
</style>  
</head>  
<body>  
  
    <div class="container">  
  
        <h1>Student Records</h1>
```

```
<?php

// Database connection details

$host = 'localhost';

$dbname = 'student_records';

$username = 'your_username';

$password = 'your_password';

try {

    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username,
$password);

    $pdo->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);

    // Fetch student records

    $stmt = $pdo->query("SELECT * FROM students");

    $students = $stmt->fetchAll(PDO::FETCH_ASSOC);

    // Selection sort function

    function selectionSort(&$arr, $n) {

        for ($i = 0; $i < $n - 1; $i++) {
```

```
$min_idx = $i;

for ($j = $i + 1; $j < $n; $j++) {

    if ($arr[$j]['gpa'] < $arr[$min_idx]['gpa']) {

        $min_idx = $j;

    }

}

if ($min_idx != $i) {

    $temp = $arr[$i];

    $arr[$i] = $arr[$min_idx];

    $arr[$min_idx] = $temp;

}

}

}
```

```
// Sort students by GPA
```

```
selectionSort($students, count($students));
```

```
// Display sorted student records
```

```
echo "<table>";
```

```
echo "<tr><th>ID</th><th>Name</th><th>GPA</th></tr>";

foreach ($students as $student) {

    echo "<tr>";

    echo "<td>" . htmlspecialchars($student['id']) . "</td>";

    echo "<td>" . htmlspecialchars($student['name']) . "</td>";

    echo "<td>" . htmlspecialchars($student['gpa']) . "</td>";

    echo "</tr>";

}

echo "</table>";

} catch(PDOException $e) {

    echo "Connection failed: " . $e->getMessage();

}

?>

</div>

</body>

</html>
```

Explanation

Visitor Counter:

This is a simple webpage that keeps track of how many visitors have accessed the site.

HTML structure for the page layout

CSS for styling (making it look nice)

PHP code to count and display visitors

How it works:

The page reads a number from a file (visitor_count.txt).

It adds 1 to this number.

It saves the new number back to the file.

It displays the new number on the page.

This gives visitors a sense of how many people have visited the site before them.

Student Record Sorter:

This page displays a sorted list of student records, ordered by their GPAs.

Key components:

HTML structure for the page layout

CSS for styling

PHP code to connect to a database, retrieve student records, sort them, and display them

How it works:

The page connects to a database containing student information.

It retrieves all student records from the database.

It uses a sorting algorithm called Selection Sort to order the students by their GPAs.

It displays the sorted list in a table on the webpage.

The sorting algorithm (Selection Sort):

This method goes through the list multiple times.

Each time, it finds the student with the lowest GPA that hasn't been placed in order yet.

It then puts this student at the beginning of the unsorted part of the list.

This process repeats until all students are in order.

Important learning points:

Database Interaction: The code demonstrates how to connect to and retrieve data from a MySQL database using PHP.

Sorting Algorithms: It implements the Selection Sort algorithm, which is a fundamental concept in computer science.

Web Development: The code shows how to combine HTML, CSS, and PHP to create dynamic web pages.

Security: The code uses `htmlspecialchars()` to prevent XSS (Cross-Site Scripting) attacks when displaying data.