# Experiment: Employee Details Management Using PHP and MySQL

## Aim

To write a PHP program that manages Employee details (EmpID, Name, Designation, Salary, DOJ, etc.) by connecting to a MySQL database and performing operations such as data insertion, retrieval, updating, and generating reports for individual and multiple employees.

---

## Software & Hardware Requirements

- PHP 7 or above
- MySQL / MariaDB
- XAMPP / WAMP Server
- HTML5 & CSS3
- Web Browser
- Code Editor (VS Code / Sublime Text)
- 4GB RAM system

---

## Theory

PHP is a server-side scripting language used for dynamic web development. MySQL is a relational database management system. When combined, they allow CRUD (Create, Read, Update, Delete) operations on structured data.

This experiment demonstrates: - Database connectivity using `mysqli_connect()` - Executing SQL queries using PHP - Displaying single and multiple employee records - Updating employee information

---

## Database Design

### Employee Table Structure (employee)

| Field | Type | Description |
|---|---|---|
| empid | INT (PK) | Employee ID |
| name | VARCHAR(50) | Employee Name |
| designation | VARCHAR(50) | Job Title |
| salary | DECIMAL(10,2) | Salary |

| Field | Type | Description |
|---|---|---|
| doj | DATE | Date of Joining |

### SQL Query to Create Table

```
CREATE TABLE employee (
    empid INT PRIMARY KEY,
    name VARCHAR(50),
    designation VARCHAR(50),
    salary DECIMAL(10,2),
    doj DATE
);
```

---

## Algorithm

1. Create a database and employee table.
2. Design an HTML form to accept employee details.
3. Write a PHP script to connect to the database.
4. Insert form data into the employee table.
5. Create PHP pages to:
    - Retrieve single employee data
    - Retrieve multiple employee records
    - Update employee details
6. Display results in tabular format.

---

## Program

### 1. HTML Form (employee_form.html)

```
<!DOCTYPE html>
<html>
<head>
    <title>Employee Details</title>
</head>
<body>
    <h2>Enter Employee Details</h2>
    <form method="POST" action="insert.php">
        Emp ID: <input type="text" name="empid"><br><br>
        Name: <input type="text" name="name"><br><br>
        Designation: <input type="text" name="designation"><br><br>
        Salary: <input type="text" name="salary"><br><br>
        DOJ: <input type="date" name="doj"><br><br>
        <input type="submit" value="Save">
    </form>
</body>
</html>
```

---

## 2. PHP Insert Script (insert.php)

```php
<?php
$conn = mysqli_connect("localhost", "root", "", "company");

$empid = $_POST['empid'];
$name = $_POST['name'];
$designation = $_POST['designation'];
$salary = $_POST['salary'];
dojo = $_POST['doj'];

$sql = "INSERT INTO employee VALUES ('$empid', '$name', '$designation',
'$salary', '$dojo')";

if (mysqli_query($conn, $sql)) {
    echo "Employee Record Inserted Successfully";
} else {
    echo "Error: " . mysqli_error($conn);
}
?>
```

---

## 3. Retrieve Single Employee (single.php)

```php
<?php
$conn = mysqli_connect("localhost", "root", "", "company");
$empid = $_GET['empid'];

$sql = "SELECT * FROM employee WHERE empid='$empid'";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    $row = mysqli_fetch_assoc($result);
    echo "Emp ID: " . $row['empid'] . "<br>";
    echo "Name: " . $row['name'] . "<br>";
    echo "Designation: " . $row['designation'] . "<br>";
    echo "Salary: " . $row['salary'] . "<br>";
    echo "DOJ: " . $row['doj'] . "<br>";
} else {
    echo "No Employee Found";
}
?>
```

---

## 4. Retrieve Multiple Employees (group.php)

```php
<?php
$conn = mysqli_connect("localhost", "root", "", "company");
```

```php
$sql = "SELECT * FROM employee";
$result = mysqli_query($conn, $sql);

echo "<h2>Employee List</h2>";
echo "<table
border='1'><tr><th>ID</th><th>Name</th><th>Designation</th><th>Salary</th><th
>DOJ</th></tr>";

while ($row = mysqli_fetch_assoc($result)) {
    echo "<tr>
            <td>{$row['empid']}</td>
            <td>{$row['name']}</td>
            <td>{$row['designation']}</td>
            <td>{$row['salary']}</td>
            <td>{$row['doj']}</td>
          </tr>";
}

echo "</table>";
?>
```

---

## 5. Update Employee (update.php)

```php
<?php
$conn = mysqli_connect("localhost", "root", "", "company");

$empid = $_POST['empid'];
salary = $_POST['salary'];

$sql = "UPDATE employee SET salary='$salary' WHERE empid='$empid'";

if (mysqli_query($conn, $sql)) {
    echo "Employee Salary Updated Successfully";
} else {
    echo "Error: " . mysqli_error($conn);
}
?>
```

---

# Output

- Inserts new employee details.
- Shows single employee record.
- Displays table of multiple employees.
- Allows salary update.

---

## Result

Thus, the PHP program for maintaining employee details using MySQL database was successfully developed. It allows insertion, retrieval, updating, and generating reports for single and multiple employees based on user needs.