National University of Computer & Emerging Sciences,
Karachi Computer Science Department

Course Code: SE-3003	Course : Web Engineering Lab

Spring 2024, Lab Manual – 08

LLO 02: Web application using PHP and MySQL

Contents:

Introduction to Web Engineering

Web Engineering is the application of systematic and quantifiable approaches (concepts methods, techniques tools) to cost - effective requirements analysis, design, implementation, testing, operation, and maintenance of **high quality Web applications.**

Technologies to be studied

- HTML
- CSS
- JavaScript
- Bootstrap
- JQuery
- PHP
- MySQL [Database]
- Laravel [PHP FRAMEWORK]

Tools – IDEs

- Visual Studio Code
- Adobe Dreamweaver
- Visual Studio

- XAMPP

9.1 Introduction to MySQL:

SQL stands for Structured Query Language. SQL is a standard programming language specifically designed for storing, retrieving, managing or manipulating the data inside a relational database management system (RDBMS). SQL became an ISO standard in 1987.

SQL is the most widely-implemented database language and supported by the popular relational database systems, like MySQL, SQL Server, and Oracle. However, some features of the SQL standard are implemented differently in different database systems.

MySQL Capabilities:

- You can create a database.
- You can create tables in a database.
- You can query or request information from a database.
- You can insert records in a database.
- You can update or modify records in a database.
- You can delete records from the database.
- You can set permissions or access control within the database for data security.
- You can create views to avoid typing frequently used complex queries.

The list does not end here, you can perform many other database-related tasks with SQL.

MySQL Structure:

MySQL database stores data into tables like other relational database. A table is a collection of related data, and it is divided into rows and columns.

Each row in a table represents a data record that are inherently connected to each other such as information related to a particular person, whereas each column represents a specific field such as *id*, *first_name*, *last_name*, *email*, etc. The structure of a simple MySQL table that contains person's general information may look something like this:

```
+----+------------+-----------+----------------------+

| id | first_name | last_name | email                |

+----+------------+-----------+----------------------+

|  1 | Peter      | Parker    | peterparker@mail.com |

|  2 | John       | Rambo     | johnrambo@mail.com   |

|  3 | Clark      | Kent      | clarkkent@mail.com   |
|  4 | John        | Carter    | johncarter@mail.com  |
```

With SQL you can perform any database-related task, such as creating databases and tables, saving data in database tables, query a database for specific records, deleting and updating data in databases.

Look at the following standard SQL query that returns the email address of a person whose first name is equal to 'Peter' in the *persons* table:

```
SELECT email FROM persons WHERE first_name="Peter"
```

If you execute the SQL query above it will return the following record:

peterparker@mail.com

9.2 PHP & MySQL Connection:

In order to store or access the data inside a MySQL database, you first need to connect to the MySQL database server. PHP offers two different ways to connect to MySQL server: **MySQLi** (Improved MySQL) and **PDO** (PHP Data Objects) extensions.

While the PDO extension is more portable and supports more than twelve different databases, MySQLi extension as the name suggests supports MySQL database only. MySQLi extension however provides an easier way to connect to, and execute queries on, a MySQL database server. Both PDO and MySQLi offer an object-oriented API, but MySQLi also offers a procedural API which is relatively easy for beginners to understand.

Connecting to MySQL Database Server:

In PHP you can easily do this using the mysqli_connect() function. All communication between PHP and the MySQL database server takes place through this connection. Here're the basic syntaxes for connecting to MySQL using MySQLi extensions:

Syntax: MySQLi, Procedural way

```
$link = mysqli_connect("hostname", "username", "password",
"database");
```

The *hostname* parameter in the above syntax specify the host name (e.g. localhost), or IP address of the MySQL server, whereas the *username* and *password* parameters specifies the credentials to access MySQL server, and the *database* parameter, if provided will specify the default MySQL database to be used when performing queries.

The following example shows how to connect to MySQL database server using MySQLi in *procedural* way.

Syntax:

```
<?php
```

```
/* Attempt MySQL server connection. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
$link = mysqli_connect("localhost", "root", "");

// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}

// Print host information
echo "Connect Successfully. Host info: " .
mysqli_get_host_info($link);
?>
```

Note: The default username for MySQL database server is root and there is no password. However to prevent your databases from intrusion and unauthorized access you should set password for MySQL accounts.

Closing the MySQL Database Server Connection

The connection to the MySQL database server will be closed automatically as soon as the execution of the script ends. However, if you want to close it earlier you can do this by simply calling the PHP mysqli_close() function.

SYNTAX:

```
<?php
/* Attempt MySQL server connection. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
$link = mysqli_connect("localhost", "root", "");

// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}

// Print host information
echo "Connect Successfully. Host info: " .
mysqli_get_host_info($link);

// Close connection
mysqli_close($link);
?>
```

9.3 PHP CRUD Operations:

CRUD is an acronym, it stands for:

C – Create OR Insert data to MySQL Database.
R – Read Database Records.
U – Update Selected MySQL Records
D – Delete Selected Record From MySQL Database.

Step 1: Setup the Environment & Create Database:

1. Install Apache and MySQL server on your local machine.
2. Create a new database in MySQL using the following SQL command:

```
CREATE DATABASE demo;

USE demo;
```

Step 2: Creating the Database Table:

Execute the following SQL query to create a table named *employees* inside your MySQL database. We will use this table for all of our future operations.

```
CREATE TABLE employees (
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    address VARCHAR(255) NOT NULL,
    salary INT(10) NOT NULL
);
```

Step 3: Creating the Config File:

After creating the table, we need to create a PHP script in order to connect to the MySQL database server. Let's create a file named "config.php" and put the following code inside it.

We'll later include this config file in other pages using the PHP require_once() function.

```
<?php
/* Database credentials. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '');
define('DB_NAME', 'demo');

/* Attempt to connect to MySQL database */
$link = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);
```

```
// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
?>
```

Step 4: Creating the Landing Page:

First, we will create a landing page for our CRUD application that contains a data grid showing the records from the *employees* database table. It also has action icons for each record displayed in the grid, that you may choose to view its details, update it, or delete it.

We'll also add a create button on the top of the data grid that can be used for creating new records in the *employees* table. Create a file named "index.php" and put the following code in it:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Dashboard</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
.min.css">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
    <script src="https://code.jquery.com/jquery-
3.5.1.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min
.js"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.m
in.js"></script>
    <style>
        .wrapper{
            width: 600px;
            margin: 0 auto;
        }
        table tr td:last-child{
            width: 120px;
        }
    </style>
    <script>
        $(document).ready(function(){
            $('[data-toggle="tooltip"]').tooltip();
        });
    </script>
```

```
</head>
<body>
    <div class="wrapper">
        <div class="container-fluid">
            <div class="row">
                <div class="col-md-12">
                    <div class="mt-5 mb-3 clearfix">
                        <h2 class="pull-left">Employees Details</h2>
                        <a href="create.php" class="btn btn-success
pull-right"><i class="fa fa-plus"></i> Add New Employee</a>
                    </div>
                    <?php
                    // Include config file
                    require_once "config.php";

                    // Attempt select query execution
                    $sql = "SELECT * FROM employees";
                    if($result = mysqli_query($link, $sql)){
                        if(mysqli_num_rows($result) > 0){
                            echo '<table class="table table-bordered
table-striped">';
                                echo "<thead>";
                                    echo "<tr>";
                                        echo "<th>#</th>";
                                        echo "<th>Name</th>";
                                        echo "<th>Address</th>";
                                        echo "<th>Salary</th>";
                                        echo "<th>Action</th>";
                                    echo "</tr>";
                                echo "</thead>";
                                echo "<tbody>";
                                while($row =
mysqli_fetch_array($result)){
                                    echo "<tr>";
                                        echo "<td>" . $row['id'] .
"</td>";
                                        echo "<td>" . $row['name'] .
"</td>";
                                        echo "<td>" . $row['address']
. "</td>";
                                        echo "<td>" . $row['salary'] .
"</td>";
                                        echo "<td>";
                                            echo '<a
href="read.php?id='. $row['id'] .'" class="mr-3" title="View Record"
data-toggle="tooltip"><span class="fa fa-eye"></span></a>';
```

```
                                            echo '<a
href="update.php?id='. $row['id'] .'" class="mr-3" title="Update
Record" data-toggle="tooltip"><span class="fa fa-pencil"></span></a>';
                                            echo '<a
href="delete.php?id='. $row['id'] .'" title="Delete Record" data-
toggle="tooltip"><span class="fa fa-trash"></span></a>';
                                        echo "</td>";
                                    echo "</tr>";
                                }
                                echo "</tbody>";
                            echo "</table>";
                            // Free result set
                            mysqli_free_result($result);
                        } else{
                            echo '<div class="alert alert-
danger"><em>No records were found.</em></div>';
                        }
                    } else{
                        echo "Oops! Something went wrong. Please try
again later.";
                    }

                    // Close connection
                    mysqli_close($link);
                    ?>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

Step 5: Creating the Create Page:

In this section we'll build the **C**reate functionality of our CRUD application.

Let's create a file named "create.php" and put the following code inside it. It will generate a web form that can be used to insert records in the *employees* table.

```
<?php
// Include config file
require_once "config.php";

// Define variables and initialize with empty values
$name = $address = $salary = "";
$name_err = $address_err = $salary_err = "";
```

```
// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){
    // Validate name
    $input_name = trim($_POST["name"]);
    if(empty($input_name)){
        $name_err = "Please enter a name.";
    } elseif(!filter_var($input_name, FILTER_VALIDATE_REGEXP,
array("options"=>array("regexp"=>"/^[a-zA-Z\s]+$/")))){
        $name_err = "Please enter a valid name.";
    } else{
        $name = $input_name;
    }

    // Validate address
    $input_address = trim($_POST["address"]);
    if(empty($input_address)){
        $address_err = "Please enter an address.";
    } else{
        $address = $input_address;
    }

    // Validate salary
    $input_salary = trim($_POST["salary"]);
    if(empty($input_salary)){
        $salary_err = "Please enter the salary amount.";
    } elseif(!ctype_digit($input_salary)){
        $salary_err = "Please enter a positive integer value.";
    } else{
        $salary = $input_salary;
    }

    // Check input errors before inserting in database
    if(empty($name_err) && empty($address_err) && empty($salary_err)){
        // Prepare an insert statement
        $sql = "INSERT INTO employees (name, address, salary) VALUES
(?, ?, ?)";

        if($stmt = mysqli_prepare($link, $sql)){
            // Bind variables to the prepared statement as parameters
            mysqli_stmt_bind_param($stmt, "sss", $param_name,
$param_address, $param_salary);

            // Set parameters
            $param_name = $name;
            $param_address = $address;
            $param_salary = $salary;

            // Attempt to execute the prepared statement
            if(mysqli_stmt_execute($stmt)){
                // Records created successfully. Redirect to landing
page
                header("location: index.php");
                exit();
            } else{
                echo "Oops! Something went wrong. Please try again
later.";

            }
        }

        // Close statement
        mysqli_stmt_close($stmt);
    }

    // Close connection
    mysqli_close($link);
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Create Record</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
.min.css">
    <style>
        .wrapper{
            width: 600px;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <div class="wrapper">
        <div class="container-fluid">
            <div class="row">
                <div class="col-md-12">
                    <h2 class="mt-5">Create Record</h2>
                    <p>Please fill this form and submit to add
```

```
employee record to the database.</p>
```

```
                    <form action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
                        <div class="form-group">
                            <label>Name</label>
                            <input type="text" name="name"
class="form-control <?php echo (!empty($name_err)) ? 'is-invalid' :
''; ?>" value="<?php echo $name; ?>">
                            <span class="invalid-feedback"><?php echo
$name_err;?></span>
                        </div>
                        <div class="form-group">
                            <label>Address</label>
                            <textarea name="address" class="form-
control <?php echo (!empty($address_err)) ? 'is-invalid' : '';
?>"><?php echo $address; ?></textarea>
                            <span class="invalid-feedback"><?php echo
$address_err;?></span>
                        </div>
                        <div class="form-group">
                            <label>Salary</label>
                            <input type="text" name="salary"
class="form-control <?php echo (!empty($salary_err)) ? 'is-invalid' :
''; ?>" value="<?php echo $salary; ?>">
                            <span class="invalid-feedback"><?php echo
$salary_err;?></span>
                        </div>
                        <input type="submit" class="btn btn-primary"
value="Submit">
                        <a href="index.php" class="btn btn-secondary
ml-2">Cancel</a>

                    </form>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

The same "create.php" file will display the HTML form and process the submitted form data. It will also perform basic validation on user inputs (*line no-11 to 37*) before saving the data.

Step 6: Creating the Read Page:

Now it's time to build the **R**ead functionality of our CRUD application.

Let's create a file named "read.php" and put the following code inside it. It will simply retrieve the records from the *employees* table based the id attribute of the employee.

```
<?php
// Check existence of id parameter before processing further
if(isset($_GET["id"]) && !empty(trim($_GET["id"]))){
    // Include config file
    require_once "config.php";

    // Prepare a select statement
    $sql = "SELECT * FROM employees WHERE id = ?";

    if($stmt = mysqli_prepare($link, $sql)){
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "i", $param_id);

        // Set parameters
        $param_id = trim($_GET["id"]);

        // Attempt to execute the prepared statement
        if(mysqli_stmt_execute($stmt)){
            $result = mysqli_stmt_get_result($stmt);

            if(mysqli_num_rows($result) == 1){
                /* Fetch result row as an associative array. Since the
result set
                contains only one row, we don't need to use while loop
*/
                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);

                // Retrieve individual field value
                $name = $row["name"];
                $address = $row["address"];
                $salary = $row["salary"];
            } else{
                // URL doesn't contain valid id parameter. Redirect to
error page
                header("location: error.php");
                exit();
            }

        } else{
            echo "Oops! Something went wrong. Please try again
later.";
```

```
        }
    }

    // Close statement
    mysqli_stmt_close($stmt);

    // Close connection
    mysqli_close($link);
} else{
    // URL doesn't contain id parameter. Redirect to error page
    header("location: error.php");
    exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>View Record</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
.min.css">
    <style>
        .wrapper{
            width: 600px;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <div class="wrapper">
        <div class="container-fluid">
            <div class="row">
                <div class="col-md-12">
                    <h1 class="mt-5 mb-3">View Record</h1>
                    <div class="form-group">
                        <label>Name</label>
                        <p><b><?php echo $row["name"]; ?></b></p>
                    </div>
                    <div class="form-group">
                        <label>Address</label>
                        <p><b><?php echo $row["address"]; ?></b></p>
                    </div>
                    <div class="form-group">
                        <label>Salary</label>
                        <p><b><?php echo $row["salary"]; ?></b></p>
                    </div>
                    <p><a href="index.php" class="btn btn-
primary">Back</a></p>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

Step 7: Creating the Update Page:

Similarly, we can build the **U**pdate functionality of our CRUD application.

Let's create a file named "update.php" and put the following code inside it. It will update the existing records in the *employees* table based the id attribute of the employee.

```
<?php
// Include config file
require_once "config.php";

// Define variables and initialize with empty values
$name = $address = $salary = "";
$name_err = $address_err = $salary_err = "";

// Processing form data when form is submitted
if(isset($_POST["id"]) && !empty($_POST["id"])){
    // Get hidden input value
    $id = $_POST["id"];

    // Validate name
    $input_name = trim($_POST["name"]);
    if(empty($input_name)){
        $name_err = "Please enter a name.";
    } elseif(!filter_var($input_name, FILTER_VALIDATE_REGEXP,
array("options"=>array("regexp"=>"/^[a-zA-Z\s]+$/")))){
        $name_err = "Please enter a valid name.";
    } else{
        $name = $input_name;
    }

    // Validate address address
    $input_address = trim($_POST["address"]);
    if(empty($input_address)){
        $address_err = "Please enter an address.";
    } else{
        $address = $input_address;
    }

    // Validate salary
    $input_salary = trim($_POST["salary"]);
    if(empty($input_salary)){
        $salary_err = "Please enter the salary amount.";
    } elseif(!ctype_digit($input_salary)){
        $salary_err = "Please enter a positive integer value.";
    } else{
        $salary = $input_salary;
    }

    // Check input errors before inserting in database
    if(empty($name_err) && empty($address_err) && empty($salary_err)){
        // Prepare an update statement
        $sql = "UPDATE employees SET name=?, address=?, salary=? WHERE
id=?";

        if($stmt = mysqli_prepare($link, $sql)){
            // Bind variables to the prepared statement as parameters
            mysqli_stmt_bind_param($stmt, "sssi", $param_name,
$param_address, $param_salary, $param_id);

            // Set parameters
            $param_name = $name;
            $param_address = $address;
            $param_salary = $salary;
            $param_id = $id;

            // Attempt to execute the prepared statement
            if(mysqli_stmt_execute($stmt)){
                // Records updated successfully. Redirect to landing
page
                header("location: index.php");
                exit();
            } else{
                echo "Oops! Something went wrong. Please try again
later.";

            }
        }

        // Close statement
        mysqli_stmt_close($stmt);
    }
    // Close connection
    mysqli_close($link);
} else{
    // Check existence of id parameter before processing further
    if(isset($_GET["id"]) && !empty(trim($_GET["id"]))){
        // Get URL parameter
        $id =  trim($_GET["id"]);

        // Prepare a select statement
        $sql = "SELECT * FROM employees WHERE id = ?";
        if($stmt = mysqli_prepare($link, $sql)){
            // Bind variables to the prepared statement as parameters
            mysqli_stmt_bind_param($stmt, "i", $param_id);

            // Set parameters
            $param_id = $id;

            // Attempt to execute the prepared statement
            if(mysqli_stmt_execute($stmt)){
                $result = mysqli_stmt_get_result($stmt);

                if(mysqli_num_rows($result) == 1){
                    /* Fetch result row as an associative array. Since 
the result set
                    contains only one row, we don't need to use while 
loop */
                    $row = mysqli_fetch_array($result, MYSQLI_ASSOC);

                    // Retrieve individual field value
                    $name = $row["name"];
                    $address = $row["address"];
                    $salary = $row["salary"];
                } else{
                    // URL doesn't contain valid id. Redirect to error 
page
                    header("location: error.php");
                    exit();
                }

            } else{
                echo "Oops! Something went wrong. Please try again 
later.";

            }
        }
        // Close statement
        mysqli_stmt_close($stmt);

        // Close connection
        mysqli_close($link);
    }  else{
        // URL doesn't contain id parameter. Redirect to error page
        header("location: error.php");
        exit();
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Update Record</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
.min.css">
    <style>
        .wrapper{
            width: 600px;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <div class="wrapper">
        <div class="container-fluid">
            <div class="row">
                <div class="col-md-12">
                    <h2 class="mt-5">Update Record</h2>
                    <p>Please edit the input values and submit to
update the employee record.</p>
                    <form action="<?php echo
htmlspecialchars(basename($_SERVER['REQUEST_URI'])); ?>"
method="post">
                        <div class="form-group">
                            <label>Name</label>
                            <input type="text" name="name"
class="form-control <?php echo (!empty($name_err)) ? 'is-invalid' :
''; ?>" value="<?php echo $name; ?>">
                            <span class="invalid-feedback"><?php echo
$name_err;?></span>
                        </div>
                        <div class="form-group">
                            <label>Address</label>
                            <textarea name="address" class="form-
control <?php echo (!empty($address_err)) ? 'is-invalid' : '';
?>"><?php echo $address; ?></textarea>
                            <span class="invalid-feedback"><?php echo
$address_err;?></span>
                        </div>
                        <div class="form-group">
                            <label>Salary</label>
                            <input type="text" name="salary"
class="form-control <?php echo (!empty($salary_err)) ? 'is-invalid' :
''; ?>" value="<?php echo $salary; ?>">
                            <span class="invalid-feedback"><?php echo
$salary_err;?></span>
                        </div>
                        <input type="hidden" name="id" value="<?php
echo $id; ?>"/>
                        <input type="submit" class="btn btn-primary"
value="Submit">
                        <a href="index.php" class="btn btn-secondary
ml-2">Cancel</a>

                    </form>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

Step 8: Creating the Delete Page:

Finally, we will build the **D**elete functionality of our CRUD application.

Let's create a file named "delete.php" and put the following code inside it. It will delete the existing records from the *employees* table based the id attribute of the employee.

```
<?php
// Process delete operation after confirmation
if(isset($_POST["id"]) && !empty($_POST["id"])){
    // Include config file
    require_once "config.php";

    // Prepare a delete statement
    $sql = "DELETE FROM employees WHERE id = ?";

    if($stmt = mysqli_prepare($link, $sql)){
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "i", $param_id);

        // Set parameters
        $param_id = trim($_POST["id"]);

        // Attempt to execute the prepared statement
        if(mysqli_stmt_execute($stmt)){
            // Records deleted successfully. Redirect to landing page
            header("location: index.php");
            exit();
        } else{
            echo "Oops! Something went wrong. Please try again
later.";

        }
    }

    // Close statement
    mysqli_stmt_close($stmt);

    // Close connection
    mysqli_close($link);
} else{
    // Check existence of id parameter
    if(empty(trim($_GET["id"]))){
        // URL doesn't contain id parameter. Redirect to error page
        header("location: error.php");
        exit();
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Delete Record</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
.min.css">
    <style>
        .wrapper{
            width: 600px;
            margin: 0 auto;
        }
    </style>
```

```
</head>
<body>
    <div class="wrapper">
        <div class="container-fluid">
            <div class="row">
                <div class="col-md-12">
                    <h2 class="mt-5 mb-3">Delete Record</h2>
                    <form action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
                        <div class="alert alert-danger">
                            <input type="hidden" name="id"
value="<?php echo trim($_GET["id"]); ?>"/>
                            <p>Are you sure you want to delete this
employee record?</p>
                            <p>
                                <input type="submit" value="Yes"
class="btn btn-danger">
                                <a href="index.php" class="btn btn-
secondary">No</a>
                            </p>
                        </div>

                    </form>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

Step 9: Creating the Error Page:

At the end, let's create one more file "error.php". This page will be displayed if request is invalid i.e. if id parameter is missing from the URL query string or it is not valid.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Error</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
.min.css">
    <style>
        .wrapper{
            width: 600px;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <div class="wrapper">
        <div class="container-fluid">
            <div class="row">
                <div class="col-md-12">
                    <h2 class="mt-5 mb-3">Invalid Request</h2>
                    <div class="alert alert-danger">Sorry, you've made
an invalid request. Please <a href="index.php" class="alert-link">go
back</a> and try again.</div>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

After a long journey finally we've finished our CRUD application with PHP and MySQL. You can run it through browser.

Tasks

1. Implement a basic PHP CRUD system using MySQL database and PHP. Test the system by performing CRUD operations on the database.

2. Implement a user registration system using PHP and MySQL. Create a registration form that allows users to enter their details such as name, email, and password. Once the user submits the form, validate the inputs and store them in the database.

3. Create a user profile page that displays the user's information such as name, email, and profile picture. Allow the user to edit their profile details and upload a new profile picture.