Week 6 - PHP & MySQL – Simple CRUD

Application Task Overview

You will build a **Student List** application. The application will store student records in a MySQL database and allow users to add, view, edit, and delete students.

Part 1: Database Setup (MySQL)

1. Create a database named **school_db**.

2. Create a table named **students** with the following

fields: 1 id – INT, Primary Key, Auto Increment

2 name – VARCHAR(100)

3 email – VARCHAR(100)

4 course – VARCHAR(100)

3. Insert at least **5 sample student records** using SQL.

Part 2: Connecting PHP to MySQL

1. Create a file named **db.php** to store your database connection

code. 2. Connect PHP to MySQL using **PDO or MySQLi**.

3. Display a simple message if the connection is successful or fails.

Part 3: CRUD Operations

Create: Create an HTML form to add a new student (name, email,

course). **Read:** Display all students in an HTML table.

Update: Add an Edit button to update student details.

Delete: Add a Delete button to remove a student record.

Part 4: Security

1. Use prepared statements for INSERT, UPDATE, and DELETE

queries. 2. Do not write SQL queries directly using user input.

```php
<?php

require'db.php';

try{

if($_SERVER['REQUEST_METHOD']==="POST" && isset($_POST['add_student'])){
    $name=$_POST['name'];
    $email=$_POST['email'];
    $course=$_POST['course'];

    $sql="INSERT INTO students(name,email,course)VALUES(?,?,?)";
    $stmt = $pdo-> prepare($sql);
    $stmt-> execute([$name,$email,$course]);

    echo"<h3 style='color:green;'>Student added</h3>";

    header("Location:index.php");
}
}
catch(PDOException $e){
    die("Failed to insert student".$e->getMessage());
}
?>

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Student Database</title>
</head>
<body>
    <form method="POST">
        Name: <input type="text" name="name" required><br>
        Email: <input type="email" name="email" required><br>
        Course: <input type="text" name="course" required><br>
        <button type="submit" name="add_student">Add Student</button>
    </form>

</body>
</html>
```

```php
<?php
require'db.php';
try{
$sql="SELECT * FROM students";
$stmt=$pdo->query($sql);
$students = $stmt->fetchAll();
}
catch(PDOException $e){
    die("Unable to f=get data from database".$e->getMessage());
}
?>



<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Studentb Database</title>
</head>
<body>
<h1 style="color: red;">Welcome to Student Database!</h1>

<table>
    <tr>
        <th>Name</th>
        <th>Email</th>
        <th>Course</th>
        <th>Edit</th>
    </tr>
    <?php foreach ($students as $student):?>
    <tr>
        <td><?=$student['name']?></td>
        <td><?=$student['email']?></td>
        <td><?=$student['course']?></td>
        <td><a href="edit.php?id=<?=$student['id']?>">Edit</a>
        <a href="delete.php?id=<?=$student['id']?>">Delete</a></td>
    </tr>
    <?php endforeach ?>
</table>
    <br>
    <a href="create.php">Add New Student</a>
</body>
</html>
```

```php
<?php

require 'db.php';
$id=$_GET['id']??'';

if (!$id) {
    die("Invalid ID");
}
    $sql="SELECT * FROM students WHERE id=?";
    $stmt= $pdo-> prepare($sql);
    $stmt->execute([$id]);
    $student = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$student) {
    die("Student not found");
}

if($_SERVER['REQUEST_METHOD']==="POST" && isset($_POST['add_student'])){
    $name=$_POST['name'];
    $email=$_POST['email'];
    $course=$_POST['course'];

    $updateSql = "UPDATE students SET name = ?, email = ?, course = ? WHERE id = ?";
    $updateStmt = $pdo->prepare($updateSql);
    $updateStmt->execute([$name, $email, $course, $id]);

    header("Location:index.php");
}


?>



<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Student Databa</title>
</head>
<body>
    <form method="POST">
        Name: <input type="text" name="name" value=<?=htmlspecialchars($student['name'])?> required><br>
        Email: <input type="email" name="email" value=<?=htmlspecialchars($student['email'])?> required><br>
        Course: <input type="text" name="course" value=<?=htmlspecialchars($student['course'])?> required><br>
        <button type="submit" name="add_student">Update</button>
    </form>
</body>
</html>
```

```php
<?php

require 'db.php';

$id=$_GET['id']??'';
if(isset($id)){
    $sql="DELETE FROM students WHERE id=?";
    $stmt=$pdo->prepare($sql);
    $stmt->execute([$id]);
        echo "Student deleted successfully!";
} else {
    echo "Invalid ID";
}

header("Location: index.php");
    exit;


?>
```

Database connected successfully

Name: tokki

Email: tokki123@gmail.com

Course: BIT

Add Student

Welcome to Student Database!

Name	Email	Course	Edit	
sapana	sapana@gmail.com	BSc.HONS	Edit	Delete
yunisa	yunisa@gmail.com	BIT	Edit	Delete
tokki	tokki123@gmail.com	BIT	Edit	Delete

Add New Student

Database connected successfully

Name: tokki

Email: tokki123@gmail.com

Course: BA

Update

Database connected successfully

Welcome to Student Database!

Name	Email	Course	Edit	
sapana	sapana@gmail.com	BSc.HONS	Edit	Delete
yunisa	yunisa@gmail.com	BIT	Edit	Delete
tokki	tokki123@gmail.com	BA	Edit	Delete

Add New Student

				id	name	email	course
☐	🖉 Edit	Copy	⊖ Delete	2	yunisa	yunisa@gmail.com	BIT
☐	🖉 Edit	Copy	⊖ Delete	3	tokki	tokki123@gmail.com	BA

↑   ☐ Check all   *With selected:*   🖉 Edit   Copy   ⊖ Delete   Export

☐ Show all | Number of rows: [25 ˅]   Filter rows: [Search this table]   Sort by key: [None]

				id	name	email	course

Database connected successfully

Welcome to Student Database!

Name	Email	Course	Edit	
yunisa	yunisa@gmail.com	BIT	Edit	Delete
tokki	tokki123@gmail.com	BA	Edit	Delete

Add New Student