

LAMP CRUD OPERATION

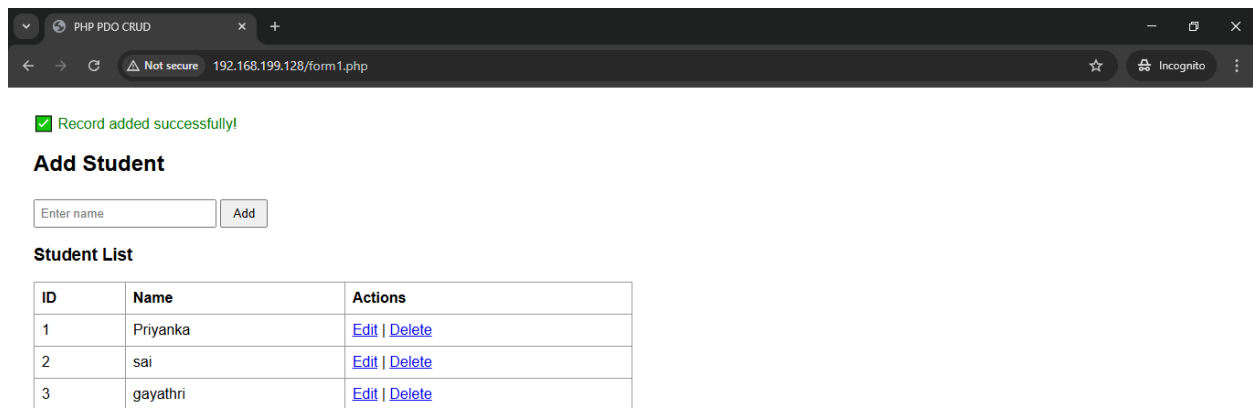
LAMP stands for:

- **Linux** – the operating system (e.g., Ubuntu)
 - **Apache** – the web server
 - **MariaDB** – the database
 - **PHP** – the server-side scripting language
-

Steps to Create a LAMP-Based CRUD App

1. **Install LAMP Stack**
Install Apache, Mariadb, and PHP on your Ubuntu system.
2. **Set Up Database**
Use Mariadb to create a database and a table (e.g., `users` with fields like `id`, `name`).
3. **Create PHP Web Pages for CRUD:**
 - Create – Add new records to the database.
 - Read – Display records from the database.
 - Update – Edit existing records.
 - Delete – Remove records from the database.
4. **Connect PHP to MariaDB**
Use a PHP file to handle database connections securely.
5. **Run and Access the App**
Store files in the Apache web directory (`/var/www/html`) and access via browser using `http://localhost`.

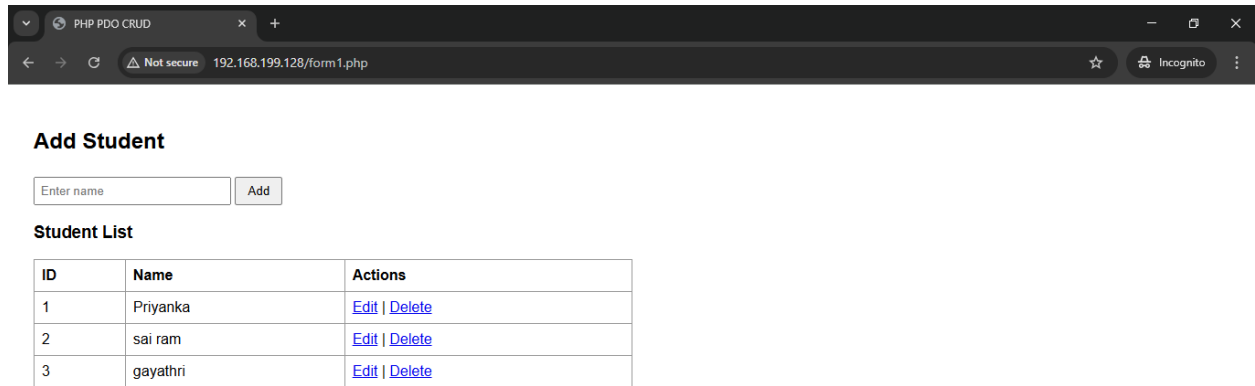
INSERT PAGE



The screenshot shows a web browser window with the address bar displaying '192.168.199.128/form1.php'. A green success message at the top reads 'Record added successfully!'. Below this is the 'Add Student' section, which includes a text input field labeled 'Enter name' and an 'Add' button. Underneath is the 'Student List' section, which contains a table with three columns: 'ID', 'Name', and 'Actions'. The table lists three students: Priyanka (ID 1), sai (ID 2), and gayathri (ID 3). Each row has 'Edit' and 'Delete' links in the 'Actions' column.

ID	Name	Actions
1	Priyanka	Edit Delete
2	sai	Edit Delete
3	gayathri	Edit Delete

UPDATE PAGE



The screenshot shows a web browser window with the title 'PHP PDO CRUD'. The address bar shows '192.168.199.128/form1.php'. The page has two main sections: 'Add Student' and 'Student List'.

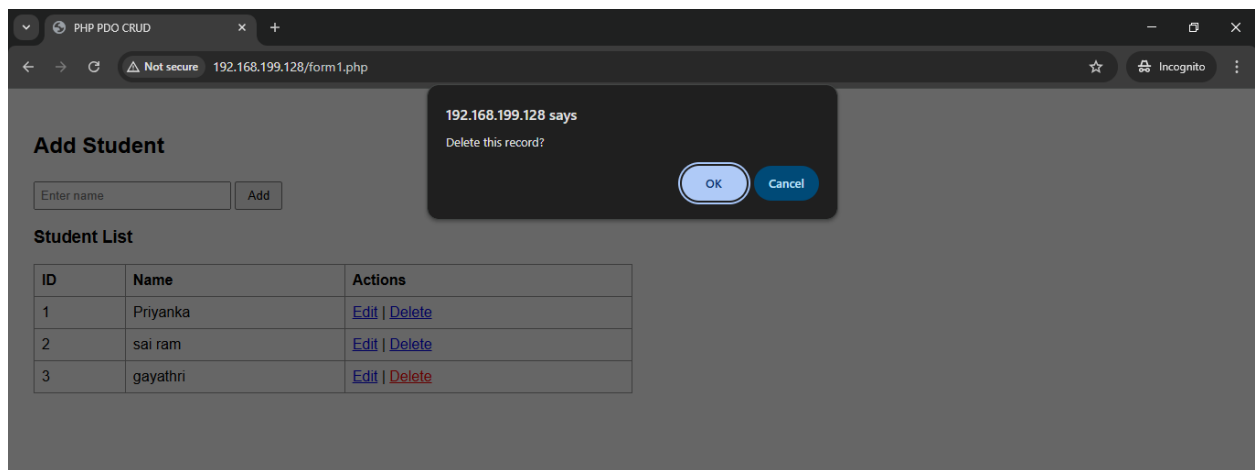
Add Student

Enter name

Student List

ID	Name	Actions
1	Priyanka	Edit Delete
2	sai ram	Edit Delete
3	gayathri	Edit Delete

DELETE PAGE



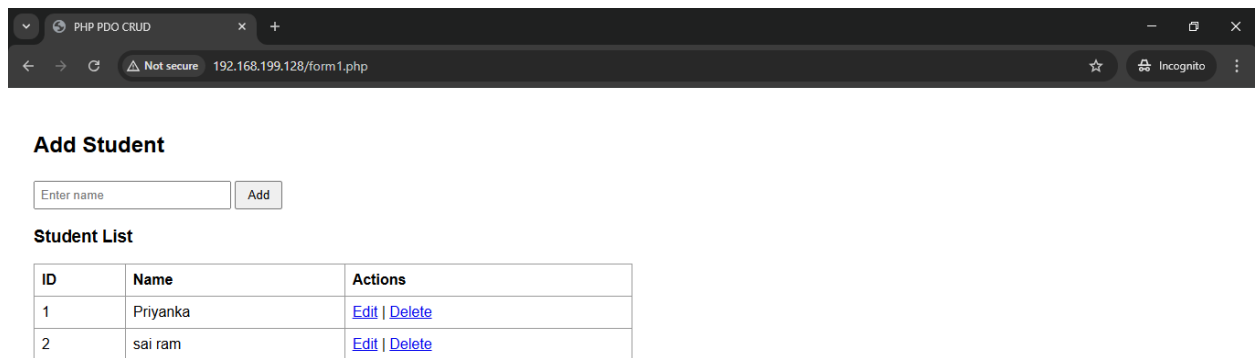
The screenshot shows the same web browser window as the 'UPDATE PAGE', but with a confirmation dialog box overlaid on top. The dialog box has a dark background and contains the text '192.168.199.128 says' and 'Delete this record?'. There are two buttons: 'OK' and 'Cancel'.

Add Student

Enter name

Student List

ID	Name	Actions
1	Priyanka	Edit Delete
2	sai ram	Edit Delete
3	gayathri	Edit Delete



The screenshot shows the same web browser window as the 'DELETE PAGE', but the confirmation dialog box is no longer present. The 'Add Student' form and 'Student List' table are visible. The 'Student List' table now only contains two rows, indicating that the third student has been successfully deleted.

Add Student

Enter name

Student List

ID	Name	Actions
1	Priyanka	Edit Delete
2	sai ram	Edit Delete

Code

```
<?php
```

```
// Database credentials
```

```
$host = "localhost";

$dbname = "db1";

$username = "root";

$password = "123";

// Enable errors

error_reporting(E_ALL);

ini_set('display_errors', 1);

// PDO Connection

try {

    $pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8mb4", $username, $password);

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

} catch (PDOException $e) {

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

}

// Create (Add)

if (isset($_POST['add'])) {

    $name = $_POST['name'];

    try {

        $stmt = $pdo->prepare("INSERT INTO bunny1 (name) VALUES (:name)");

        $stmt->execute(['name' => $name]);

        echo "<p style='color:green;'>✔ Record added successfully!</p>";

    } catch (PDOException $e) {

        echo "<p style='color:red;'>✔ Error adding: " . $e->getMessage() . "</p>";

    }

}
```

```

// Update

if (isset($_POST['update'])) {

    $id = $_POST['id'];

    $name = $_POST['name'];

    $stmt = $pdo->prepare("UPDATE bunny1 SET name = :name WHERE id = :id");

    $stmt->execute(['name' => $name, 'id' => $id]);

    header("Location: " . $_SERVER['PHP_SELF']);

    exit;

}

// Delete

if (isset($_GET['delete'])) {

    $id = $_GET['delete'];

    $stmt = $pdo->prepare("DELETE FROM bunny1 WHERE id = :id");

    $stmt->execute(['id' => $id]);

    header("Location: " . $_SERVER['PHP_SELF']);

    exit;

}

// Read All

$stmt = $pdo->query("SELECT * FROM bunny1 ORDER BY id ASC");

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

// Edit Mode

$edit = false;

$edit_id = "";

$edit_name = "";

if (isset($_GET['edit'])) {

```

```

$edit = true;

$edit_id = $_GET['edit'];

$stmt = $pdo->prepare("SELECT * FROM bunny1 WHERE id = :id");

$stmt->execute(['id' => $edit_id]);

$student = $stmt->fetch(PDO::FETCH_ASSOC);

if ($student) {

    $edit_name = $student['name'];

} }    ?>

<!DOCTYPE html>

<html>

<head>

    <title>PHP PDO CRUD</title>

    <style>

        body { font-family: Arial; padding: 20px; }

        table { border-collapse: collapse; width: 50%; }

        th, td { border: 1px solid #999; padding: 8px; text-align: left; }

        input[type="text"] { padding: 6px; width: 200px; }

        button { padding: 6px 12px; margin-top: 5px; }

    </style>

</head>

<body>

<h2><?php echo $edit ? "Edit Student" : "Add Student"; ?></h2>

<form method="post" action="">

    <input type="text" name="name" required placeholder="Enter name" value="<?=
htmlspecialchars($edit_name) ?>">

    <?php if ($edit): ?>

```

```
<input type="hidden" name="id" value="<?= $edit_id ?>">

<button type="submit" name="update">Update</button>

<a href="<?= $_SERVER['PHP_SELF'] ?>">Cancel</a>

<?php else: ?>

    <button type="submit" name="add">Add</button>

<?php endif; ?>

</form>

<h3>Student List</h3>

<table>

    <tr>

        <th>ID</th><th>Name</th><th>Actions</th>

    </tr>

    <?php foreach ($students as $row): ?>

        <tr>

            <td><?= $row['id'] ?></td>

            <td><?= htmlspecialchars($row['name']) ?></td>

            <td>

                <a href="?edit=<?= $row['id'] ?>">Edit</a> |

                <a href="?delete=<?= $row['id'] ?>" onclick="return confirm('Delete this record?');">Delete</a>

            </td>

        </tr>

    <?php endforeach; ?>

</table>

</body>

</html>
```