

Step-by-Step Guide to Create a Simple CRUD Example in PHP

This guide will walk you through creating a basic CRUD (Create, Read, Update, Delete) application using PHP and MySQL. You will learn how to set up a database connection, create an HTML form to populate the database, and perform CRUD operations.

Step 1: Set Up the Environment

1. Install XAMPP/WAMP/MAMP:
 - Download and install [XAMPP](https://www.apachefriends.org/index.html) or a similar tool to run Apache, MySQL, and PHP.
2. Start the Apache and MySQL Servers:
 - Open the XAMPP Control Panel.
 - Start `Apache` and `MySQL`.
3. Create a New Database:
 - Go to `http://localhost/phpmyadmin`.
 - Click on the "Databases" tab and enter a database name (e.g., `crud_example`).
 - Click "Create."
4. Create a New Table:
 - Click on the `crud_example` database.
 - Create a table named `users` with the following structure:

Field	Type	Length	Attributes	Index	A_I
id	INT	11	Unsigned	PRIMARY	Yes
name	VARCHAR	100			
email	VARCHAR	100			
age	INT	3	Unsigned		

5. Save the Table:
 - Click "Save" to create the table structure.

Step 2: Create Database Connection in PHP

1. Create a Folder for Your Project:
 - Go to your `htdocs` directory (inside the XAMPP installation folder).
 - Create a new folder (e.g., `crud_app`).
2. Create a `config.php` File for Database Connection:
 - Inside your project folder, create a file named `config.php`.
 - Add the following code:

```
<?php
```

```

$servername = "localhost";
$username = "root"; // Default username for XAMPP
$password = ""; // Default password for XAMPP is empty
$dbname = "crud_example";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>

```

3. Create an `index.php` File:

- This file will handle the CRUD operations.
- Create an `index.php` file in the same folder and start by including the database connection:

```

<?php
include 'config.php';
?>

```

Step 3: Create the HTML Form for Data Input

1. Add HTML Form in `index.php`:

- Below the PHP include statement, add the following HTML form to `index.php`:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple CRUD Application</title>
</head>
<body>
    <h2>Simple CRUD Application</h2>
    <form action="index.php" method="POST">
        <input type="text" name="name" placeholder="Name" required>
        <input type="email" name="email" placeholder="Email" required>
        <input type="number" name="age" placeholder="Age" required>
        <button type="submit" name="submit">Add User</button>
    </form>
</body>

```

```
</html>
```

Step 4: Insert Data into the Database

1. Handle Form Submission in `index.php`:

- Below the form, add PHP code to handle form submission and insert data into the database:

```
<?php
if (isset($_POST['submit'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $age = $_POST['age'];

    // Insert query
    $sql = "INSERT INTO users (name, email, age) VALUES ('$name', '$email', $age)";

    if ($conn->query($sql) === TRUE) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}
?>
```

2. Test the Insert Operation:

- Open `http://localhost/crud_app/index.php` in your browser.
- Fill out the form and submit to see if data is inserted into the database.

Step 5: Display Data from the Database

1. Display Users in `index.php`:

- Below the form handling code, add the following PHP code to fetch and display data:

```
<h3>Users List</h3>
<table border="1">
    <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Email</th>
        <th>Age</th>
        <th>Actions</th>
    </tr>
```

```

<?php
$sql = "SELECT * FROM users";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "<tr>
            <td>" . $row['id'] . "</td>
            <td>" . $row['name'] . "</td>
            <td>" . $row['email'] . "</td>
            <td>" . $row['age'] . "</td>
            <td>
                <a href='edit.php?id=" . $row['id'] . "'>Edit</a>
                <a href='delete.php?id=" . $row['id'] . "'>Delete</a>
            </td>
        </tr>";
    }
} else {
    echo "<tr><td colspan='5'>No users found</td></tr>";
}
?>
</table>

```

Step 6: Update Data in the Database

1. Create `edit.php` File:
 - Create an `edit.php` file in your project folder.
 - Add the following code:

```

<?php
include 'config.php';

if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $sql = "SELECT * FROM users WHERE id = $id";
    $result = $conn->query($sql);
    $row = $result->fetch_assoc();
}

if (isset($_POST['update'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $age = $_POST['age'];

    $sql = "UPDATE users SET name='$name', email='$email', age=$age WHERE id=$id";
}

```

```

if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
    header("Location: index.php");
} else {
    echo "Error updating record: " . $conn->error;
}
}
?>

```

```

<form action="" method="POST">
    <input type="text" name="name" value="<?php echo $row['name']; ?>" required>
    <input type="email" name="email" value="<?php echo $row['email']; ?>" required>
    <input type="number" name="age" value="<?php echo $row['age']; ?>" required>
    <button type="submit" name="update">Update User</button>
</form>

```

2. Test the Update Operation:

- Click the "Edit" link in `index.php` next to a user.
- Update the user information and submit the form.

Step 7: Delete Data from the Database

1. Create `delete.php` File:

- Create a `delete.php` file in your project folder.
- Add the following code:

```

<?php
include 'config.php';

if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $sql = "DELETE FROM users WHERE id = $id";

    if ($conn->query($sql) === TRUE) {
        echo "Record deleted successfully";
        header("Location: index.php");
    } else {
        echo "Error deleting record: " . $conn->error;
    }
}
?>

```

2. Test the Delete Operation:

- Click the “Delete” link next to a user in `index.php`.
- The user should be deleted from the database.

Step 8: Final Touches

1. Close the Database Connection:

- At the bottom of `index.php`, add the following line to close the connection:

```
$conn->close();
```

2. Styling and Enhancements:

- Add some CSS to style your form and table.
- Enhance security by using prepared statements and data validation.

This guide provides a complete walkthrough for creating a basic CRUD application in PHP. Each step includes detailed code and explanations to help you build a functional and interactive web application.

config.php

```
<?php
$servername = "localhost";
$username = "root"; // Default username for XAMPP/WAMP/MAMP
$password = ""; // Default password is empty
$dbname = "crud_example"; // Your database name

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

index.php

```
<?php
include 'config.php'; // Include database connection

// Handle form submission to add a new user
if (isset($_POST['submit'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $age = $_POST['age'];

    // Insert query
    $sql = "INSERT INTO users (name, email, age) VALUES ('$name', '$email', $age)";

    if ($conn->query($sql) === TRUE) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple CRUD Application</title>
    <style>
        form, table {
            margin: 20px;
```

```

    }
    table, th, td {
        border: 1px solid black;
        border-collapse: collapse;
        padding: 8px;
    }
</style>
</head>
<body>
    <h2>Simple CRUD Application</h2>

    <!-- Form to add new user -->
    <form action="index.php" method="POST">
        <input type="text" name="name" placeholder="Name" required>
        <input type="email" name="email" placeholder="Email" required>
        <input type="number" name="age" placeholder="Age" required>
        <button type="submit" name="submit">Add User</button>
    </form>

    <h3>Users List</h3>
    <table>
        <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Email</th>
            <th>Age</th>
            <th>Actions</th>
        </tr>
        <?php
            // Fetch and display users from the database
            $sql = "SELECT * FROM users";
            $result = $conn->query($sql);

            if ($result->num_rows > 0) {
                while($row = $result->fetch_assoc()) {
                    echo "<tr>
                        <td>" . $row['id'] . "</td>
                        <td>" . $row['name'] . "</td>
                        <td>" . $row['email'] . "</td>
                        <td>" . $row['age'] . "</td>
                        <td>
                            <a href='edit.php?id=" . $row['id'] . "'>Edit</a>
                            <a href='delete.php?id=" . $row['id'] . "'>Delete</a>
                        </td>
                    </tr>";
                }
            } else {
                echo "<tr><td colspan='5'>No users found</td></tr>";
            }
        </?php>
    </table>

```



```

    }
    ?>
</table>

```

```

<?php
$conn->close(); // Close the database connection
?>
</body>
</html>

```

edit.php

```

<?php
include 'config.php'; // Include database connection

// Fetch user details to edit
if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $sql = "SELECT * FROM users WHERE id = $id";
    $result = $conn->query($sql);
    $row = $result->fetch_assoc();
}

// Update user details in the database
if (isset($_POST['update'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $age = $_POST['age'];

    $sql = "UPDATE users SET name='$name', email='$email', age=$age WHERE id=$id";

    if ($conn->query($sql) === TRUE) {
        echo "Record updated successfully";
        header("Location: index.php"); // Redirect to main page after update
    } else {
        echo "Error updating record: " . $conn->error;
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit User</title>
</head>
<body>

```

```

<h2>Edit User</h2>
<form action="" method="POST">
    <input type="text" name="name" value="<?php echo $row['name']; ?>" required>
    <input type="email" name="email" value="<?php echo $row['email']; ?>" required>
    <input type="number" name="age" value="<?php echo $row['age']; ?>" required>
    <button type="submit" name="update">Update User</button>
</form>
</body>
</html>

```

delete.php

```

<?php
include 'config.php'; // Include database connection

// Delete user from the database
if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $sql = "DELETE FROM users WHERE id = $id";

    if ($conn->query($sql) === TRUE) {
        echo "Record deleted successfully";
        header("Location: index.php"); // Redirect to main page after deletion
    } else {
        echo "Error deleting record: " . $conn->error;
    }
}
?>

```