PHP BASIC

PHP Introduction
PHP Getting Started
PHP Syntax
PHP Variables
PHP Constants
PHP Echo and Print
PHP Data Types
PHP Strings
PHP Operators
PHP If...Else
PHP Switch...Case
PHP Arrays
PHP Sorting Arrays
PHP Loops
PHP Functions
PHP Math Operations
PHP GET and POST

PHP ADVANCED

PHP Date and Time
PHP Include Files
PHP File system
PHP Parsing Directories
PHP File Upload
PHP File Download
PHP Cookies
PHP Sessions
PHP Send Email
PHP Form Handling
PHP Form Validation
PHP Filters
PHP Error Handling
PHP Classes and Objects
PHP Magic Constants
PHP JSON Parsing
PHP Regular Expressions
PHP Exception Handling

PHP & MySQL DATABASE

PHP MySQL Introduction
PHP MySQL Connect
PHP MySQL Create Database
PHP MySQL Create Table
PHP MySQL Insert
PHP MySQL Prepared
PHP MySQL Last Inserted ID
PHP MySQL Select
PHP MySQL Where
PHP MySQL Limit
PHP MySQL Order By
PHP MySQL Update
PHP MySQL Delete
PHP MySQL CRUD Application

PHP MySQL CRUD Application

In this tutorial you'll learn how to build a CRUD application with PHP and MySQL.

What is CRUD

CRUD is an acronym for **C**reate, **R**ead, **U**pdate, and **D**elete. CRUD operations are basic data manipulation for database. We've already learned how to perform create (i.e. insert), read (i.e. select), update and delete operations in previous chapters. In this tutorial we'll create a simple PHP application to perform all these operations on a MySQL database table at one place.

Well, let's start by creating the table which we'll use in all of our example.

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.

Example ⤓ Download

```
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
);
```

Creating the Config File

After creating the table, we need 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.

Example Procedural Object Oriented PDO ⤓ Download

```
<?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);
```

If you've downloaded the Object Oriented or PDO code examples using the download button, please remove the text "-oo-format" or "-pdo-format" from file names before testing the code.

Note: Replace the credentials according to your MySQL server setting before testing this code, for example, replace the database name 'demo' with your own database name, replace

PHP MySQL Ajax Search
PHP MySQL Login System

PHP EXAMPLES

PHP Practice Examples
PHP FAQ's Answers

PHP REFERENCE

PHP String Functions
PHP Array Functions
PHP File System Functions
PHP Date/Time Functions
PHP Calendar Functions
PHP MySQLi Functions
PHP Filters
PHP Error Levels
More references

username 'root' with your own database username, specify database password if there's any.

Creating the Landing Page

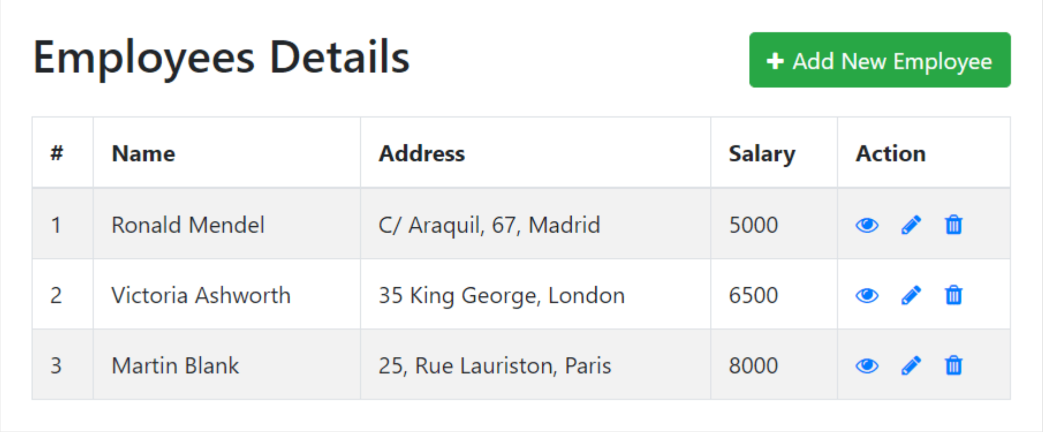
First we will create a landing page for our CRUD application that contains a data grid showing the records from the *employees* database table. It also has action icons for each record displayed in the grid, that you may choose to view its details, update it, or delete it.

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:

Example Procedural Object Oriented PDO ⤓ Download

```
<!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
```

Once *employees* table is populated with some records the landing page i.e. the CRUD data grid may look something like the picture shown below:



Employees Details

+ Add New Employee

#	Name	Address	Salary	Action
1	Ronald Mendel	C/ Araquil, 67, Madrid	5000	👁 ✎ 🗑
2	Victoria Ashworth	35 King George, London	6500	👁 ✎ 🗑
3	Martin Blank	25, Rue Lauriston, Paris	8000	👁 ✎ 🗑

Tip: We've used the Bootstrap framework to make this CRUD application layout quickly and beautifully. Bootstrap is the most popular and powerful front-end framework for faster and easier responsive web development. Please, checkout the Bootstrap tutorial section to learn more about this framework.

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.

Example Procedural Object Oriented PDO ⤓ Download

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

```

```
5  // Define variables and initialize with empty values
6  $name = $address = $salary = "";
7  $name_err = $address_err = $salary_err = "";
8
9  // Processing form data when form is submitted
10 if($_SERVER["REQUEST_METHOD"] == "POST"){
11     // Validate name
12     $input_name = trim($_POST["name"]);
```

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.

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.

Example Procedural Object Oriented PDO ⤓ Download

```
<?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);
```

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.

Example Procedural Object Oriented PDO ⤓ Download

```
<?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"];
```

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.

Example Procedural Object Oriented PDO ⤓ Download

```php
<?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);
```

Creating the Error Page

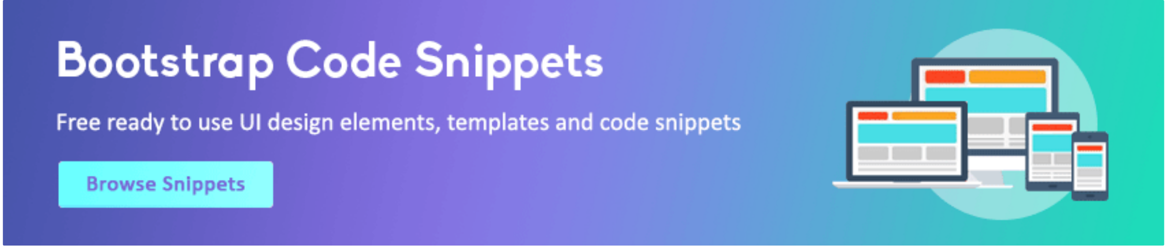
At the end, let's create one more file "error.php". This page will be displayed if request is invalid i.e. if id parameter is missing from the URL query string or it is not valid.

Example ⤓ Download

```html
<!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;
```

After a long journey finally we've finished our CRUD application with PHP and MySQL. We recommend you to check out PHP & MySQL database tutorial section from the beginning, if you haven't already covered, for a better understanding of each and every part of this tutorial.

« PREVIOUS PAGE NEXT PAGE »


Bootstrap Code Snippets
Free ready to use UI design elements, templates and code snippets
Browse Snippets

Is this website helpful to you? Please give us a like, or share your feedback to help us improve. Connect with us on Facebook and Twitter for the latest updates.

ABOUT US

- Our Story
- Terms of Use
- Privacy Policy

CONTACT

- Contact Us
- Report Error
- Advertise

INTERACTIVE TOOLS

- Font Awesome Icon Search Utility
- Title & Meta Length Calculator
- Bootstrap Button Generator
- Bootstrap Icon Finder
- HTML Formatter
- HTML Color Picker
- SQL Playground
- HTML Editor

Share This: