**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**](https://www.tutorialrepublic.com/examples/downloads/employees-table.zip)

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

);CREATE TABLE employees (

**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**](https://www.tutorialrepublic.com/examples/bin/download-source.php?topic=php&file=config)

<?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());

}

?>

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 'domo' with your own database name, replace 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**](https://www.tutorialrepublic.com/examples/bin/download-source.php?topic=php&file=index)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Dashboard</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.js"></script>

<style type="text/css">

.wrapper{

width: 650px;

margin: 0 auto;

}

.page-header h2{

margin-top: 0;

}

table tr td:last-child a{

margin-right: 15px;

}

</style>

<script type="text/javascript">

$(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="page-header clearfix">

<h2 class="pull-left">Employees Details</h2>

<a href="create.php" class="btn btn-success pull-right">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'] ."' title='View Record' data-toggle='tooltip'><span class='glyphicon glyphicon-eye-open'></span></a>";

echo "<a href='update.php?id=". $row['id'] ."' title='Update Record' data-toggle='tooltip'><span class='glyphicon glyphicon-pencil'></span></a>";

echo "<a href='delete.php?id=". $row['id'] ."' title='Delete Record' data-toggle='tooltip'><span class='glyphicon glyphicon-trash'></span></a>";

echo "</td>";

echo "</tr>";

}

echo "</tbody>";

echo "</table>";

// Free result set

mysqli\_free\_result($result);

} else{

echo "<p class='lead'><em>No records were found.</em></p>";

}

} else{

echo "ERROR: Could not able to execute $sql. " . mysqli\_error($link);

}

// Close connection

mysqli\_close($link);

?>

</div>

</div>

</div>

</div>

</body>

</html>

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:

[](https://www.tutorialrepublic.com/php-tutorial/php-mysql-crud-application.php)

**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](https://www.tutorialrepublic.com/twitter-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**](https://www.tutorialrepublic.com/examples/bin/download-source.php?topic=php&file=create)

<?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 "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://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">

<style type="text/css">

.wrapper{

width: 500px;

margin: 0 auto;

}

</style>

</head>

<body>

<div class="wrapper">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<div class="page-header">

<h2>Create Record</h2>

</div>

<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 <?php echo (!empty($name\_err)) ? 'has-error' : ''; ?>">

<label>Name</label>

<input type="text" name="name" class="form-control" value="<?php echo $name; ?>">

<span class="help-block"><?php echo $name\_err;?></span>

</div>

<div class="form-group <?php echo (!empty($address\_err)) ? 'has-error' : ''; ?>">

<label>Address</label>

<textarea name="address" class="form-control"><?php echo $address; ?></textarea>

<span class="help-block"><?php echo $address\_err;?></span>

</div>

<div class="form-group <?php echo (!empty($salary\_err)) ? 'has-error' : ''; ?>">

<label>Salary</label>

<input type="text" name="salary" class="form-control" value="<?php echo $salary; ?>">

<span class="help-block"><?php echo $salary\_err;?></span>

</div>

<input type="submit" class="btn btn-primary" value="Submit">

<a href="index.php" class="btn btn-default">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.

**Creating the Read Page**

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

Let's create a file named "create.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**](https://www.tutorialrepublic.com/examples/bin/download-source.php?topic=php&file=read)

<?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://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">

<style type="text/css">

.wrapper{

width: 500px;

margin: 0 auto;

}

</style>

</head>

<body>

<div class="wrapper">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<div class="page-header">

<h1>View Record</h1>

</div>

<div class="form-group">

<label>Name</label>

<p class="form-control-static"><?php echo $row["name"]; ?></p>

</div>

<div class="form-group">

<label>Address</label>

<p class="form-control-static"><?php echo $row["address"]; ?></p>

</div>

<div class="form-group">

<label>Salary</label>

<p class="form-control-static"><?php echo $row["salary"]; ?></p>

</div>

<p><a href="index.php" class="btn btn-primary">Back</a></p>

</div>

</div>

</div>

</div>

</body>

</html>

**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**](https://www.tutorialrepublic.com/examples/bin/download-source.php?topic=php&file=update)

<?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 "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://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">

<style type="text/css">

.wrapper{

width: 500px;

margin: 0 auto;

}

</style>

</head>

<body>

<div class="wrapper">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<div class="page-header">

<h2>Update Record</h2>

</div>

<p>Please edit the input values and submit to update the record.</p>

<form action="<?php echo htmlspecialchars(basename($\_SERVER['REQUEST\_URI'])); ?>" method="post">

<div class="form-group <?php echo (!empty($name\_err)) ? 'has-error' : ''; ?>">

<label>Name</label>

<input type="text" name="name" class="form-control" value="<?php echo $name; ?>">

<span class="help-block"><?php echo $name\_err;?></span>

</div>

<div class="form-group <?php echo (!empty($address\_err)) ? 'has-error' : ''; ?>">

<label>Address</label>

<textarea name="address" class="form-control"><?php echo $address; ?></textarea>

<span class="help-block"><?php echo $address\_err;?></span>

</div>

<div class="form-group <?php echo (!empty($salary\_err)) ? 'has-error' : ''; ?>">

<label>Salary</label>

<input type="text" name="salary" class="form-control" value="<?php echo $salary; ?>">

<span class="help-block"><?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-default">Cancel</a>

</form>

</div>

</div>

</div>

</div>

</body>

</html>

**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**](https://www.tutorialrepublic.com/examples/bin/download-source.php?topic=php&file=delete)

<?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>View Record</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">

<style type="text/css">

.wrapper{

width: 500px;

margin: 0 auto;

}

</style>

</head>

<body>

<div class="wrapper">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<div class="page-header">

<h1>Delete Record</h1>

</div>

<form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="post">

<div class="alert alert-danger fade in">

<input type="hidden" name="id" value="<?php echo trim($\_GET["id"]); ?>"/>

<p>Are you sure you want to delete this record?</p><br>

<p>

<input type="submit" value="Yes" class="btn btn-danger">

<a href="index.php" class="btn btn-default">No</a>

</p>

</div>

</form>

</div>

</div>

</div>

</div>

</body>

</html>

**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**](https://www.tutorialrepublic.com/examples/bin/download-source.php?topic=php&file=error)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Error</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">

<style type="text/css">

.wrapper{

width: 750px;

margin: 0 auto;

}

</style>

</head>

<body>

<div class="wrapper">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<div class="page-header">

<h1>Invalid Request</h1>

</div>

<div class="alert alert-danger fade in">

<p>Sorry, you've made an invalid request. Please <a href="index.php" class="alert-link">go back</a> and try again.</p>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

After a long journey finally we've finished our CRUD application with PHP and MySQL. We recommend you to check out [PHP & MySQL database](https://www.tutorialrepublic.com/php-tutorial/php-mysql-introduction.php) tutorial section from the beginning, if you haven't already covered, for a better understanding of each and every part of this tutorial.