PHP PDO CRUD Tutorial Using OOP with Bootstrap

Want to create a form where you can perform Create, Read, Update, and Delete operations to your database?
Here, we will look into database transactions in PHP using PDO on a Bootstrap website, for this, we will be using Bootstrap libraries as well as some third party libraries. Below is the folder structure showing the files used.

Overview

In this tutorial, we will be creating 6 main files to complete our task. Save the files with the respective codes and place all the files in the same directory.

- Add.php
- Delete.php
- Index.php
- Update.php
- Config.php
- Data.inc.php

Database Schema

Let us start by creating a database to store our values, I have named my database "biodata",
Then add a table to your database.

```
CREATE TABLE IF NOT EXISTS `tbl_users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `first_name` varchar(25) NOT NULL,
  `last_name` varchar(25) NOT NULL,
  `email_id` varchar(50) NOT NULL,
  `contact_no` bigint(10) NOT NULL,
  PRIMARY KEY (`id`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=27 ;
```

```
CREATE TABLE IF NOT EXISTS `users` (
`username` varchar(50) NOT NULL,
`email` varchar(50) NOT NULL,
`password` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Add.php

Add.php file contains the code which renders the Add data HTML form as well as the Server-side code which is used to perform the Create function. As soon as the user press the submit button, the form will send data to a server using POST method. That data is captured then assigned into variables which will be further stored in the database using PDO statements.

```
<?php
include_once 'includes/config.php';
$database = new Config();
$db = $database->getConnection();
include_once 'includes/data.inc.php';
$product = new Data($db);
```

```
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Tutorial-06</title>

<!-- Bootstrap -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

</head>
<body>
<p><br/></p>
<div class="container">
<p>
<a class="btn btn-primary" href="index.php" role="button">Back</a>
</p><br/>
<?php
if ($_POST) {

$product->name = $_POST['name'];
$product->gender = $_POST['gender'];
$product->contactNum = $_POST['contact'];
$product->address = $_POST['address'];

if ($product->create()) {
?>
<div class="alert alert-success alert-dismissible" role="alert">
<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span
aria-hidden="true">&times;</span></button>
<strong>Success!</strong> <a href="index.php">View Data</a>.
</div>
<?php
} else {
?>
<div class="alert alert-danger alert-dismissible" role="alert">
<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span
aria-hidden="true">&times;</span></button>
<strong>Fail!</strong>
</div>
<?php
}
}
?>
<form method="post">
<div class="form-group">
<label for="nm">Name</label>
```

```
<input type="text" class="form-control" id="fn" name="first_name">
</div>
<div class="form-group">
<label for="gd">Gender</label>
<input type="text" class="form-control" id="ln" name="last_name">
</div>
<div class="form-group">
<label for="tl">Phone</label>
<input type="text" class="form-control" id="em" name="email_id">
</div>
<div class="form-group">
<label for="ar">Address</label>
<textarea class="form-control" rows="3" id="ar" name="address"></textarea>
</div>
<button type="submit" class="btn btn-success">Submit</button>
</form>
</div>

<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Delete.php

Delete file contains the code which performs delete function. For the delete function to execute, we need to make a GET call embedding the relevant ID of the field needed to delete. Then, the PDO function delete will remove the specific entry from the database.

```
<?php
// check if value was posted
// include database and object file
include_once 'includes/config.php';
include_once 'includes/data.inc.php';

// get database connection
$database = new Config();
$db = $database->getConnection();

// prepare product object
$product = new Data($db);

// set product id to be deleted
$product->id = isset($_GET['id']) ? $_GET['id'] : die('Need Product ID');

// delete the product
if($product->delete()){
echo "<script>location.href='index.php'</script>";
}

// if unable to delete the product
else{
echo "<script>alert('Failed to Deleted Data')</script>";

}
```

```
?>
```

Index.php

The index file contains the main HTML form as well as the Server side code which performs the Read function in order to populate a form with data. In order to populate with data, first, server issues a PDO statement to query and return all the data from the database. Then according to the count of entries, the User interface will generate rows and columns and display them.

```
<?php
$page = isset($_GET['page']) ? $_GET['page'] : 1;

$records_per_page = 5;

$from_record_num = ($records_per_page * $page) - $records_per_page;

include_once 'includes/config.php';
include_once 'includes/data.inc.php';

$database = new Config();
$db = $database->getConnection();

$product = new Data($db);

$stmt = $product->readAll($page, $from_record_num, $records_per_page);
$num = $stmt->rowCount();

?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Tutorial-06</title>

<link href="css/bootstrap.min.css" rel="stylesheet">

<script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

</head>
<body>
<p><br/></p>
<div class="container">
<p>
<a class="btn btn-primary" href="add.php" role="button">Add Data</a>
</p>
<?php
if($num>0){
?>
<table class="table table-bordered table-hover table-striped">
<caption>Personal Data Table</caption>
<thead>
<tr>
<th>#</th>
<th>First Name</th>
<th>Last Name</th>
<th>Email</th>
<th>Contact</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<?php
while ($row = $stmt->fetch(PDO::FETCH_ASSOC)){
extract($row);
?>
<tr>
<?php echo "<td>{$id_pdo}</td>" ?>
<?php echo "<td>{$nm_pdo}</td>" ?>
<?php echo "<td>{$gd_pdo}</td>" ?>
<?php echo "<td>{$tl_pdo}</td>" ?>
<?php echo "<td>{$ar_pdo}</td>" ?>
<?php echo "<td width='100px'>
<a class='btn btn-warning btn-sm' href='update.php?id={$id_pdo}'
role='button'><span class='glyphicon glyphicon-pencil' aria-
hidden='true'></span></a>
<a class='btn btn-danger btn-sm' href='delete.php?id={$id_pdo}'
role='button'><span class='glyphicon glyphicon-trash' aria-
hidden='true'></span></a>
</td>" ?>
</tr>
<?php
}
?>
</tbody>
</table>
<?php
$page_dom = "index.php";
include_once 'includes/pagination.inc.php';
}
else{
?>
<div class="alert alert-warning alert-dismissible" role="alert">
<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
<strong>Warning!</strong> Data is still empty
</div>
<?php
}
?>
</div>
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Update.php

Update file contains the code which renders the Update form as well as the Server-side code which enables the update function. For the update form to be loaded we need to pass the relevant ID via a GET method when loading the page. Then the PDO will issue a update query in order to finish update of data.

```
<?php
include_once 'includes/config.php';

$id = isset($_GET['id']) ? $_GET['id'] : die('Need Product ID');

$database = new Config();
$db = $database->getConnection();

include_once 'includes/data.inc.php';
$product = new Data($db);

$product->id = $id;
$product->readOne();
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Tutorial-06</title>
<link href="css/bootstrap.min.css" rel="stylesheet">

<script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

</head>
<body>
<p><br/></p>
<div class="container">
<p>
<a class="btn btn-primary" href="index.php" role="button">Back</a>
</p><br/>
<?php
if($_POST){

$product->name = $_POST['name'];
$product->gender = $_POST['gender'];
$product->contactNum = $_POST['contact'];
$product->address = $_POST['address'];

if($product->update()){
?>
<script>window.location.href='index.php'</script>
<?php
}else{
?>
<div class="alert alert-danger alert-dismissible" role="alert">
<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
<strong>Fail!</strong>
</div>
<?php
}
}
?>
<form method="post">
<div class="form-group">
<label for="nm">Name</label>
<input type="text" class="form-control" id="nm" name="name" value='<?php echo
$product->name; ?>'>
</div>
<div class="form-group">
<label for="gd">Gender</label>
<input type="text" class="form-control" id="gd" name="gender" value='<?php
echo $product->gender; ?>'>
</div>
<div class="form-group">
<label for="tl">Phone</label>
<input type="text" class="form-control" id="tl" name="contact" value='<?php
echo $product->contactNum; ?>'>
</div>
<div class="form-group">
<label for="ar">Address</label>
<textarea class="form-control" rows="3" id="ar" name="address"><?php echo
$product->address; ?></textarea>
</div>
<button type="submit" class="btn btn-success">Submit</button>
</form>
</div>

<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>

</body>
</html>
```

Config.php

The config file contains the code which keeps the connection with the database, in every PHP file where there is a need to interact with the database we need to include this configuration file.

```
<?php

class Config
{

// specify your own database credentials
private $host = "localhost";
private $db_name = "biodata";
private $username = "root";
private $password = "";
public $conn;

// get the database connection
public function getConnection()
{

$this->conn = null;

try {
$this->conn = new PDO("mysql:host=" . $this->host . ";dbname=" . $this-
>db_name, $this->username, $this->password);
} catch (PDOException $exception) {
echo "Connection error: " . $exception->getMessage();
}

return $this->conn;
}
}

?>
```

Data.inc.php

This file contains the Code for all the CRUD operations written against the database. Every other file uses an object of this class to call the operations. This class act as the base class which is written on top of the PDO data class With this it perform all the Database CRUD operations that are written in a more relevant way to this web service logic.

```
<?php

class Data
{

// database connection and table name
private $conn;
private $table_name = "crudpdo";
// public $name;public $gender;public $contactNum;public $address;
// object properties
public $id;
public $first_name;
public $last_name;
public $email_id;
public $contact_no;

public function __construct($db)
{
$this->conn = $db;
}

// create product
function create()
{

//write query
```

```
$query = "INSERT INTO " . $this->table_name . " values('',?,?,?,?)";

$stmt = $this->conn->prepare($query);

$stmt->bindParam(1, $this->first_name);
$stmt->bindParam(2, $this->last_name);
$stmt->bindParam(3, $this->email_id);
$stmt->bindParam(4, $this->contact_no);

if ($stmt->execute()) {
return true;
} else {
return false;
}

}

// read products
function readAll($page, $from_record_num, $records_per_page)
{

$query = "SELECT * FROM " . $this->table_name . " ORDER BY nm_pdo ASC LIMIT
{$from_record_num}, {$records_per_page}";

$stmt = $this->conn->prepare($query);
$stmt->execute();

return $stmt;
}

// used for paging products
public function countAll()
{

$query = "SELECT id_pdo FROM " . $this->table_name . "";

$stmt = $this->conn->prepare($query);
$stmt->execute();

$num = $stmt->rowCount();

return $num;
}

// used when filling up the update product form
function readOne()
{

$query = "SELECT * FROM " . $this->table_name . " WHERE id_pdo = ? LIMIT
0,1";

$stmt = $this->conn->prepare($query);
$stmt->bindParam(1, $this->id);
$stmt->execute();
```

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

$this->name = $row['nm_pdo'];
$this->gender = $row['gd_pdo'];
$this->contactNum = $row['tl_pdo'];
$this->address = $row['ar_pdo'];
}

// update the product
function update()
{
$query = "UPDATE
" . $this->table_name . "
SET
nm_pdo = :nm,
gd_pdo = :gd,
tl_pdo = :tl,
ar_pdo = :ar
WHERE
id_pdo = :id";

$stmt = $this->conn->prepare($query);

$stmt->bindParam(':nm', $this->name);
$stmt->bindParam(':gd', $this->gender);
$stmt->bindParam(':tl', $this->contactNum);
$stmt->bindParam(':ar', $this->address);
$stmt->bindParam(':id', $this->id);

// execute the query
if ($stmt->execute()) {
return true;
} else {
return false;
}
}

// delete the product
function delete()
{

$query = "DELETE FROM " . $this->table_name . " WHERE id_pdo = ?";

$stmt = $this->conn->prepare($query);
$stmt->bindParam(1, $this->id);

if ($result = $stmt->execute()) {
return true;
} else {
return false;
}
}
}

?><
```