PHP last class mysql

Config.php

<?php  
// connecting database  
$servername = "localhost";  
$username = "root";  
$password = "";  
// create connection  
$connection = mysqli\_connect($servername, $username, $password);  
if($connection){  
 echo "Congratulations! Connection Successful";  
}  
// else{  
// die("Sorry! Failed to connection : ". mysqli\_connect\_error());  
// }  
// create DB  
$mysql = "CREATE DATABASE labclass";  
$result = mysqli\_query($connection, $mysql);  
// if the same name db then result=false, otherwise true  
echo "The result is : ";  
echo var\_dump($result);  
echo "<br>";  
?>

1. The script sets up the necessary information to connect to a database. It specifies that the database is located on the local machine (localhost), and the username and password for accessing it are both empty. You would typically replace the empty strings with the appropriate username and password for your database.
2. The script attempts to establish a connection to the database using the provided server name, username, and password. If the connection is successful, it displays the message "Congratulations! Connection Successful."
3. Next, the script tries to create a new database named "labclass" using the **mysqli\_query()** function. This function sends a query to the database server, requesting it to create a new database. The result of this operation is stored in the **$result** variable.
4. Finally, the script uses the **var\_dump()** function to display the result of the **mysqli\_query()** function. It will show whether the database creation was successful or not. If the result is **false**, it means that a database with the same name already exists. Otherwise, if the result is **true**, it means that the database was successfully created.

Remember to uncomment the **die()** statement if you want to display any connection errors that may occur during the process.

Lastclass.php

<!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=1.0">  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">  
 <title>Log In / Sign Up page</title>  
</head>  
<body>  
 <div class="w-25 bg-primary text-white mx-auto text-center m-5 p-5 border">  
 <h2>Log In/Sign Up</h2>  
 <form action="" method="post">  
 <input type="text" name="user" class="form-control mb-4" placeholder="Username/E-mail">  
 <input type="password" class="form-control mb-4" name="pass" placeholder="Password">  
 <input type="submit" value="Log in" name="login" class="btn btn-danger">  
 <a class="btn btn-dark" href="signup.php">Sign Up</a>  
 </form>  
 </div>   
</body>  
</html>

1. **<div class="w-25 bg-primary text-white mx-auto text-center m-5 p-5 border">**: This **<div>** element represents a container with specific CSS classes applied to it. The classes **w-25** set the width to 25% of its parent container, **bg-primary** sets the background color to blue, **text-white** sets the text color to white, **mx-auto** centers the element horizontally, **text-center** aligns the text content in the center, **m-5** adds margin around the element, **p-5** adds padding inside the element, and **border** adds a border to the element.
2. **<h2>Log In/Sign Up</h2>**: This heading element displays the text "Log In/Sign Up" as the main heading for the section.
3. **<form action="" method="post">**: This **<form>** element is used to define a form that contains input fields and buttons. The **action** attribute is currently empty, which means the form will submit to the same page (it will be handled by the same PHP script that generates this page). The **method** attribute is set to "post", indicating that the form data will be sent using the HTTP POST method.
4. **<input type="text" name="user" class="form-control mb-4" placeholder="Username/E-mail">**: This **<input>** element creates a text input field. The **type** attribute is set to "text", indicating it is a text input field. The **name** attribute is set to "user", which will be used to identify this input field when the form is submitted. The **class** attribute is set to "form-control mb-4", which applies CSS classes for styling. The **placeholder** attribute provides a placeholder text that appears in the input field before the user enters any value.
5. **<input type="password" class="form-control mb-4" name="pass" placeholder="Password">**: This **<input>** element creates a password input field. The **type** attribute is set to "password", indicating it is a password input field. The **name** attribute is set to "pass", which will be used to identify this input field when the form is submitted. The **class** attribute is set to "form-control mb-4", applying CSS classes for styling. The **placeholder** attribute provides a placeholder text for the input field.
6. **<input type="submit" value="Log in" name="login" class="btn btn-danger">**: This **<input>** element creates a submit button. The **type** attribute is set to "submit", indicating that this button will submit the form. The **value** attribute is set to "Log in", which sets the text displayed on the button. The **name** attribute is set to "login", which will be used to identify this button when the form is submitted. The **class** attribute is set to "btn btn-danger", applying CSS classes for styling. In this case, it sets the button's color to red.
7. **<a class="btn btn-dark" href="signup.php">Sign Up</a>**: This **<a>** element creates a link. The **class** attribute is set to "btn btn-dark", applying CSS classes for styling. In this case, it sets the link's color to dark. The **href** attribute is set to "signup.php", which specifies the destination URL when the link is clicked. The link text is "Sign Up".

Sign up:

<!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=1.0">  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">  
 <title>Sign Up page</title>  
</head>  
<body>  
 <div class="w-25 bg-primary text-white mx-auto text-center m-5 p-5 border">  
 <h2>Sign Up</h2>  
 <form action="" method="post">  
 <input type="text" name="name" class="form-control mb-4" placeholder="Enter your name">  
 <input type="text" name="user" class="form-control mb-4" placeholder="Enter your E-mail">  
 <input type="password" class="form-control mb-4" name="pass" placeholder="Enter your Password">  
 <!-- <a class="btn btn-dark" href="signup.php">Sign Up</a> -->  
 <input type="submit" class="btn btn-dark" value="Sign Up" name="signup">  
 </form>  
  
 <?php  
 include "config.php";  
 if(isset($\_POST["signup"])){  
 // echo "You have clicked..!";  
 // $sql = "INSERT INTO `user` (`email`, `password`, `name`) VALUES ('tushar15-3549@diu.edu.bd', 'tushar3549', 'Tushar Ahmed')";  
 $sql = "INSERT INTO `user` (`email`, `password`, `name`) VALUES ('".$\_POST["user"]."', '".$\_POST["pass"]."', '".$\_POST["name"]."')";  
 $temp = mysqli\_query($connection, $sql);  
 if($temp){  
 echo "<h5>Sign Up Success</h5>";  
 }  
 }  
 ?>  
  
 </div>  
</body>  
</html>

1. **include "config.php";**: This line includes the contents of the "config.php" file, which typically contains the necessary configuration settings and the database connection details. It allows the PHP code to establish a connection to the database by accessing the variables and functions defined in the "config.php" file.
2. **if(isset($\_POST["signup"])){ ... }**: This conditional statement checks if the form has been submitted and the "signup" button has been clicked. It uses the **isset()** function to check if the "signup" key exists in the **$\_POST** superglobal array, which contains the form data submitted via HTTP POST method.
3. The code block within the if statement is executed only if the condition is true, i.e., the "signup" button has been clicked.
4. **$sql = "INSERT INTO** user **(**email**,** password**,** name**) VALUES ('".$\_POST["user"]."', '".$\_POST["pass"]."', '".$\_POST["name"]."')";**: This line constructs an SQL query to insert the form data into the "user" table of the database. It uses the **$\_POST** superglobal array to access the values submitted via the form. The values are concatenated into the query string using concatenation (**.**) operator.
5. **$temp = mysqli\_query($connection, $sql);**: This line executes the SQL query using the **mysqli\_query()** function. It takes two arguments: the database connection variable **$connection** and the query string **$sql**. The function returns a result object or **false** if the query execution fails. The result object (**$temp** in this case) can be used to check if the query was executed successfully.
6. **if($temp){ ... }**: This conditional statement checks if the query execution was successful by evaluating the result object (**$temp**). If the query was executed successfully, the code block within the if statement is executed.
7. **echo "<h5>Sign Up Success</h5>";**: This line displays the "Sign Up Success" message as a heading (**<h5>**) on the webpage. It confirms that the sign-up process was successful.

**Form handler**

**Formhandler.php**

<?php  
if($\_SERVER["REQUEST\_METHOD"] == "POST"){  
 $firstname = htmlspecialchars($\_POST["firstname"]);  
 $lastname = htmlspecialchars($\_POST["lastname"]);  
 $favouritepet = htmlspecialchars($\_POST["favouritepet"]);  
  
 if(empty($firstname)){  
 exit();  
 // header("Location: ../index.php");  
 }  
  
 echo "These are the data that the user submitted:";  
 echo "<br>";  
 echo $firstname;  
  
 echo "<br>";  
 echo $lastname;  
  
 echo "<br>";  
 echo $favouritepet;  
  
 // header("Location: ../index.php");  
}  
else{  
 header("Location: ../index.php");  
}  
  
  
?>

1. **if($\_SERVER["REQUEST\_METHOD"] == "POST")**: This line checks if the current HTTP request method is POST. It determines if the form has been submitted and the data is being sent to the server.
2. **$firstname = htmlspecialchars($\_POST["firstname"]);**: This line retrieves the value of the "firstname" field from the submitted form data using the **$\_POST** superglobal array. The **htmlspecialchars()** function is used to convert any special characters in the input to their HTML entities, which helps prevent cross-site scripting (XSS) attacks.
3. **$lastname = htmlspecialchars($\_POST["lastname"]);**: This line retrieves the value of the "lastname" field from the submitted form data in a similar manner as the previous line.
4. **$favouritepet = htmlspecialchars($\_POST["favouritepet"]);**: This line retrieves the value of the "favouritepet" field from the submitted form data in a similar manner as the previous lines.
5. **if(empty($firstname)){ exit(); }**: This line checks if the "firstname" field is empty. If it is empty, the script exits and no further code is executed. This could be used for validation purposes, where you might want to make sure that a required field is not empty before processing the data.
6. The following lines of code **echo** the submitted form data on the webpage, displaying the values entered by the user for "firstname", "lastname", and "favouritepet".
7. The line **header("Location: ../index.php");** redirects the user back to the "index.php" page. This line is commented out in the code you provided, but it can be used to redirect the user to another page after processing the form data.

Index.php

<!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=1.0">  
 <title>PHP Lab Task</title>  
 <style>  
 form{  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 justify-content: center;  
 height: 100vh;  
 }  
 body{  
 background-color: black;  
 color: white;  
 font-size: 27px;  
 }  
 button{  
 padding: 15px;  
 cursor: pointer;  
 background-color: aqua;  
 color: black;  
 border-radius: 15px;  
 text-align: center;  
 }  
 input{  
 border-radius: 10px;  
 font-size: 22px;  
 }  
 </style>  
</head>  
<body>  
 <main>  
 <form action="formhandler.php" method="post">  
 <label for="firstname">First Name</label>  
 <input required type="text" id="firstname" name="firstname" placeholder="Enter your firstname..">  
 <br>  
 <label for="lastname">Last Name</label>  
 <input required type="text" id="lastname" name="lastname" placeholder="Enter your lastname..">  
 <br>  
 <label for="favouritepet">Which is your favourite pet?</label>  
 <select name="favouritepet" id="favouritepet">  
 <option value="none">NONE</option>  
 <option value="cat">CAT</option>  
 <option value="dog">DOG</option>  
 <option value="bird">BIRD</option>  
 </select>  
 <br>  
 <button type="submit">Submit</button>  
 </form>  
 </main>  
</body>  
</html>

The **<main>** element is used to define the main content area of the webpage.

The **<form>** element is used to create an HTML form. It has two attributes:

* **action**: Specifies the URL or file where the form data will be submitted for processing. In this case, the form data will be sent to "formhandler.php" for processing.
* **method**: Specifies the HTTP method to be used when submitting the form. Here, it is set to "post," which means the form data will be sent in the HTTP request body.

Within the form, there are three form fields:

* The **<label>** elements provide a text description for each input field.
* The **<input>** elements allow users to enter their first name and last name. They have the attributes **type** set to "text," **id** for uniquely identifying each input field, **name** for defining the name of the form field, and **placeholder** for displaying a placeholder text inside the input field. The **required** attribute ensures that these fields must be filled out before the form can be submitted.
* The **<select>** element creates a dropdown menu for selecting the user's favorite pet. It has the **name** attribute set to "favouritepet" and the **id** attribute for identification. The available options are defined using **<option>** elements, each with a corresponding value.

Finally, there is a **<button>** element with **type** set to "submit." It represents a submit button that users can click to submit the form.

When the form is submitted, the form data will be sent to "formhandler.php" for processing based on the **action** attribute.