Week:6.

How to create admin login page using PHP

In this article, we will see how we can create a login page for admin, connected with the database, or whose information to log in to the page is already stored in our database.

Follow the steps to create an admin login page using PHP:

Approach: Make sure you have XAMPP or WAMP installed on your windows machine. In case you're using Linux OS then install the LAMP server. In this article, we will be using the XAMPP server.

process of connecting to a localhost database server using PHP

1. **Set up your database server**: Ensure that you have a database server installed and running on your local machine. For example, if you're using MySQL, you might have installed XAMPP, WAMP, MAMP, or a similar stack that includes MySQL as the database server.
2. **Create a database**: Use a database management tool (e.g., phpMyAdmin, MySQL Workbench) to create a database if you haven't already done so.
3. **Write PHP code for database connection**: Write PHP code to connect to the database server from your PHP script. This involves specifying the necessary connection parameters such as the server hostname, database username, password, and database name.
4. **Establish the connection**: Use PHP's database extension (e.g., MySQLi, PDO) to establish a connection to the database server. This typically involves creating a new instance of the database connection object and passing the connection parameters to its constructor.
5. **Check for connection success or failure**: After attempting to establish the connection, check whether the connection was successful or not. If the connection fails, handle the error appropriately (e.g., display an error message).
6. **Perform database operations**: Once the connection is successfully established, you can execute SQL queries and perform various database

operations such as inserting, updating, deleting, or retrieving data from the database.

Localhost:

$servername = "localhost";** is assigning the value "localhost" to the variable **$servername.

This variable is typically used to specify the hostname or IP address of the database server to which you want to connect. In this case, "localhost" refers to the local machine where the PHP script is running.

When connecting to a database server, you need to specify its location so that PHP knows where to send the database queries. If the database server is hosted on the same machine as the PHP script, you can use "localhost" to indicate that the server is on the local system.

Root:

the PHP code snippet provided, **$username = "root";** is assigning the value "root" to the variable **$username**.

This variable is typically used to store the username required to authenticate and access a database. In the context of database connection, it represents the username used to log in to the database server.

In many database management systems (DBMS) like MySQL, PostgreSQL, etc., there's a concept of user accounts. Each user account has a username associated with it, which is used to authenticate and grant access to specific databases and their resources.

localhost used for

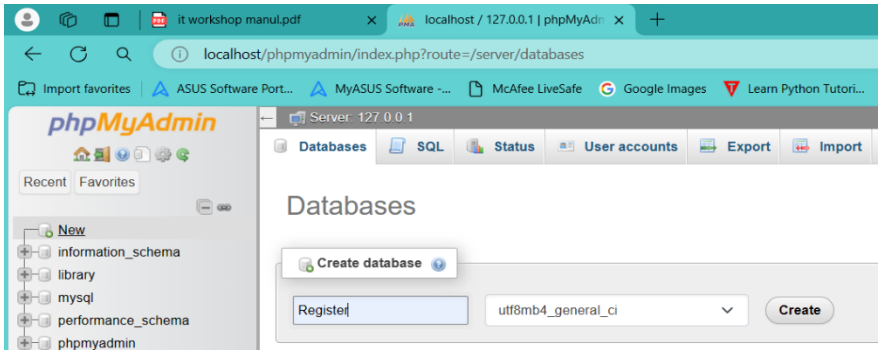
Developers use the local host to test web applications and programs. Network administrators use the loopback to test network connections. Another use for

the localhost is the host's file, where you can use the loopback to block malicious websites.

1. **Create Database**:

register:

First, we will create a database named 'register' (you can give any name to your database). You can also use your existing database or create a new one.



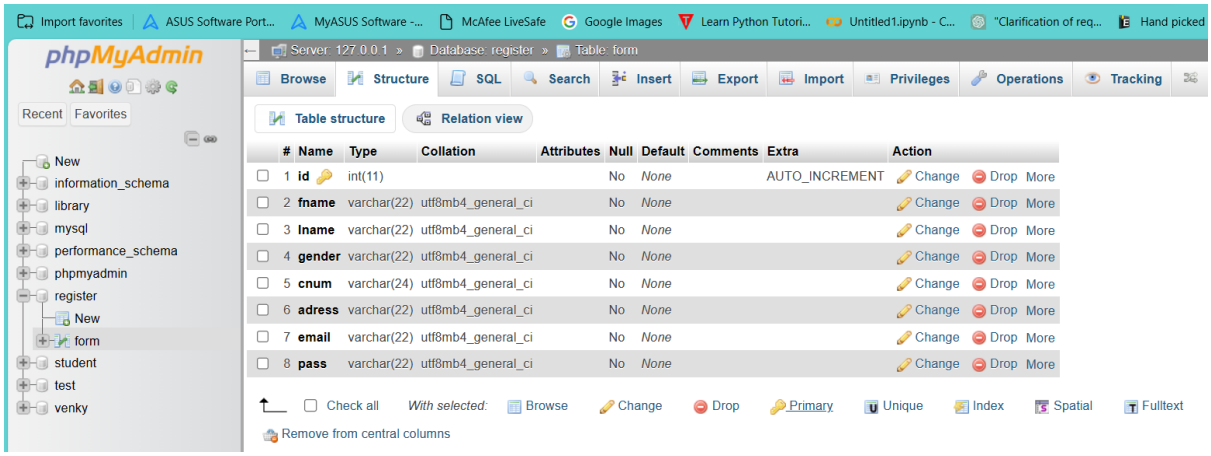
Create Table Structure: The table "adminlogin" contains number of fields.

- id – primary key – auto increment
- fname – varchar(22)
- lname – varchar(22)
- gender– varchar(22)
- cnum – varchar(22)
- adress – varchar(22)
- email – varchar(22)
- pass – varchar(22)

The datatype for username and password is varchar. The size can be altered as per the requirement. However, 22 is sufficient, and the datatype for "id" is int and it is a primary key.

A primary key also called a primary keyword is a key in a relational database that is unique for each record. It is a unique identifier, such as a driver's license number, telephone number (including area code), or vehicle identification number (VIN).

Your table structure should look like this.



Create a folder: that **includes** the **following files:** The folder should be in **"C:\xampp\htdocs\"** (or where your XAMPP is installed). For the WAMP server, it should be in **"C:\wamp64\www\"** and on Linux **"/opt/lampp/htdocs"**.

- **Filename: signup.php**

```
<DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,intial-scale">
    <title>form  Login and register</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div class="signup">
      <h1>sign up</h1>
      <h4>it's free and only taske a minute</h4>
      <form action="verify.php" method="post">
```

```
            <label>First Name</label>
            <input type="text" name="fname" required>
            <label>Last Name</label>
            <input type="text" name="lname" required>
            <label>Gender</label>
            <input type="text" name="gender" required>
            <label>Contact Adress</label>
            <input type="text" name="number" required>
            <label>Adress</label>
            <input type="text" name="adress" required>
            <label>Email</label>
            <input type="email" name="email" required>
            <label>Password</label>
            <input type="password" name="pass" required>
            <input type="submit" name="" value="submit">


        </form>
        <p>by clicking the sign up button,you agree to our<br>
        <a href=""> Term and condition</a> and <a href="login.php">policy
privecy</a></p>
    </div>
  </body>
</html>
```

File Name :login.php

```
<DOCTYPE html>
  <html>
    <head>
      <meta charset="utf-8">
      <meta name="viewport" content="width=device-width,intial-scale">
      <title>form  Login and register</title>
      <link rel="stylesheet" href="style.css">
    </head>
    <body>
      <div class="login">
        <h1>Login</h1>
        <form action="pass_verify.php" method="POST">
        <h4>it's free and only a minute</h4>
```

```
            <form>
               <label>Email</label>
               <input type="email" name="mail" required>
               <label>Password</label>
               <input type="password" name="pass" required>
               <input type="submit" name="submit" value="submit">


            </form>
            <p> Not have an account? <a href="signup.php">sign up
here</a></p>
         </body>
         </html>
```

File name:style.css

```
<body{
        background-color: rgba(41, 81, 184, 0.498);
   width:100%;
   height:100vh;
   font-family:sans-serif;
}
.signup{
   width:360px;
   height:800px;
   margin:auto;
   background:rgb(199, 199, 214);
   border-radius:3px;
}
.login{
        width:360px;
   height:320px;
   margin:auto;
   background:white;
   border-radius:3px;
}


h1{
   text-align:center;
}
h4{
```

```
    text-align:center;
    padding-top:15px;
}
form{
    width:300px;
    margin-left:200px;
    text-align:center;
}
form label{
    display:flex;
    margin-top:20px;
    font-size:18px;
}
form input{
    width:100%;
    padding:7px;
    border:none;
    border:1px solid gray;
    border-radius:6px;
    outline:none;
}

input[type="submit"]{
    width:320px;
    height:35px;
    margin-top:20px;
    border:none;
    background-color:#ff7200;
    color:white;
    font-size:18px;
    cursor:pointer;


}
input[type="submit"]:hover{
    color:white;
    background:rgb(13,211,247);
}

p{
    text-align:center;
    padding-top:20px;
    font-size:15px;
}
```

File name: Verify.php

```
<?php
session_start();

include("db_conn.php"); // Assuming db.php contains your database
connection details

if ($_SERVER['REQUEST_METHOD'] == "POST") {
   $firstname = $_POST['fname'];
   $lastname = $_POST['lname'];
   $gender = $_POST['gender']; // Correct variable name
   $num = $_POST['number'];
   $adress = $_POST['adress']; // Correct variable name
   $gmail = $_POST['email']; // Correct variable name
   $password = $_POST['pass']; // Correct variable name

   if (!empty($gmail) && !empty($password) && is_numeric($num)) { //
Corrected empty check and variable name
      // Assuming db.php contains database connection code
      $query = "INSERT INTO form (fname, lname, gender, cnum, adress,
email, pass) VALUES ('$firstname', '$lastname', '$gender', '$num', '$adress',
'$gmail', '$password')";
      mysqli_query($con, $query);

      // echo "<script type='text/javascript'> alert('Successfully
Registered')</script>";
   header("Location:login.php");

   } else {
      echo "<script type='text/javascript'> alert('Please Enter valid
information')</script>";
   }
}
?>
```

File name: Db_connection.php

This code establishes a connection to the MySQL database named "register" on localhost using the provided username and an empty password. If the connection is successful, it will print "Connected successfully". Otherwise, it will display an error message indicating the reason for the connection failure.
Top of Form

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "register";

// Create connection
$con = mysqli_connect($servername, $username, $password, $dbname);

// Check connection
if (!$con) {
   die("Connection failed: " . mysqli_connect_error());
}
?>
```

File name:Signup_complete.php

```
<html>
   <body>
<?php
session_start();

include("db_conn.php"); // Assuming db.php contains your database
connection details

if ($_SERVER['REQUEST_METHOD'] == "POST") {
   $firstname = $_POST['fname'];
   $lastname = $_POST['lname'];
   $gender = $_POST['gender']; // Correct variable name
   $num = $_POST['number'];
   $adress = $_POST['adress']; // Correct variable name
   $gmail = $_POST['email']; // Correct variable name
```

```
    $password = $_POST['pass']; // Correct variable name

    if (!empty($gmail) && !empty($password) && is_numeric($num)) { // 
Corrected empty check and variable name
        // Assuming db.php contains database connection code
        $query = "INSERT INTO form (fname, lname, gender, cnum, adress, 
email, pass) VALUES ('$firstname', '$lastname', '$gender', '$num', '$adress', 
'$gmail', '$password')";
        mysqli_query($con, $query);

        // echo "<script type='text/javascript'> alert('Successfully 
Registered')</script>";
    header("Location:login.php");

    } else {
        echo "<script type='text/javascript'> alert('Please Enter valid 
information')</script>";
    }
}
?>
        <br><a href="signup.php">Signup</a>
        <br><a href="login.php">Login</a>

    </body>
</html>
```

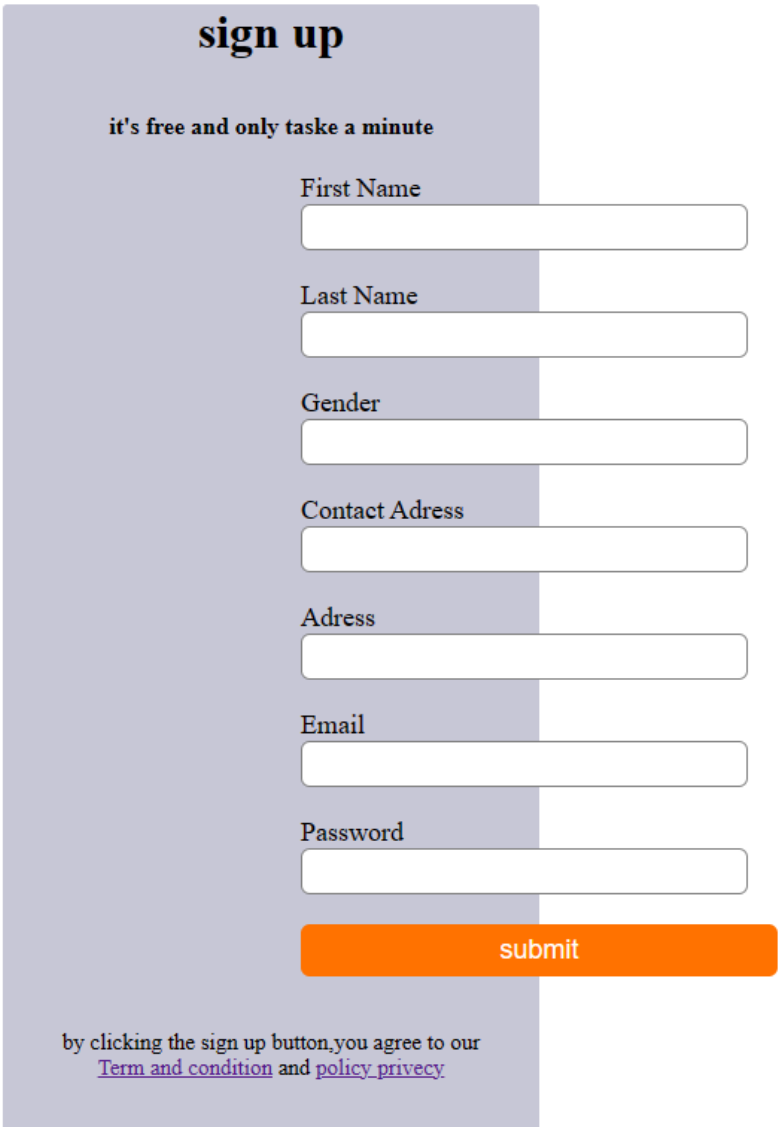
File name: pass_verify.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Result</title>
    <link rel="stylesheet" href="style.css">

</head>
<body>
    <div class="container">
        <?php
```

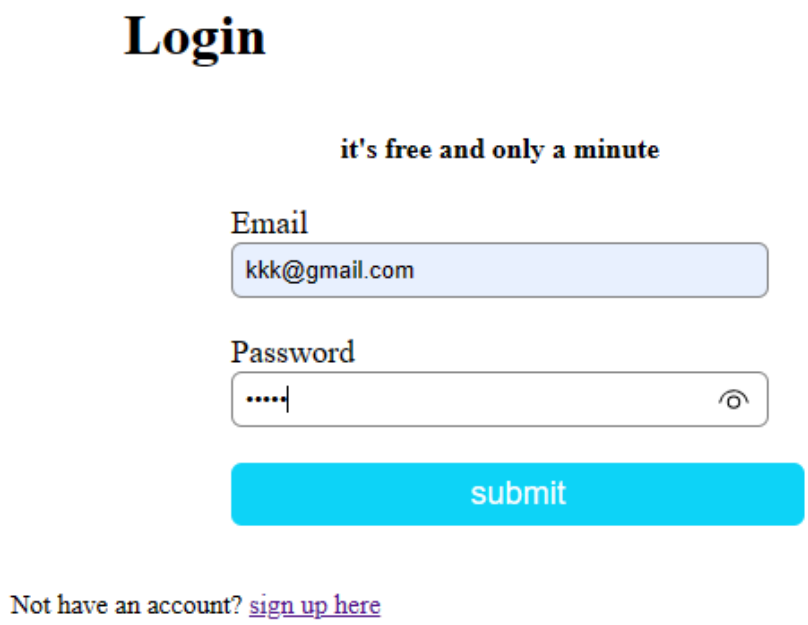
```
        session_start();

        include("db_conn.php"); // Assuming db.php contains your database connection details

        if ($_SERVER['REQUEST_METHOD'] == "POST") {
          $email = $_POST['mail'];
          $password = $_POST['pass'];

          // Query to check if the email and password match
          $query = "SELECT * FROM form WHERE email = '$email' AND pass = '$password'"; // Directly comparing passwords

          $result = mysqli_query($con, $query);

          if (mysqli_num_rows($result) == 1) {
             // Email and password match
             $_SESSION['email'] = $email; // Store email in session for future use
             echo '<div class="output-container">';
             echo '<p class="success-message">Valid email or password</p>';
             foreach ($result as $row){
                echo "<div class='user-details'>";
                echo "<p><strong>First Name:</strong> ".$row['fname']."</p>";
                echo "<p><strong>Last Name:</strong> ".$row['lname']."</p>";
                echo "<p><strong>Gender:</strong> ".$row['gender']."</p>";
                echo "<p><strong>Phone Number:</strong> ".$row['cnum']."</p>";
                echo "<p><strong>Address:</strong> ".$row['adress']."</p>";
                echo "<p><strong>Email:</strong> ".$row['email']."</p>";
                echo "<p><strong>Password:</strong> ".$row['pass']."</p>";
                echo "</div>";
                }
             echo '</div>';
          } else {
             // Email and/or password don't match
             echo "<p class='error-message'>Invalid email or password</p>";
          }
       }
     ?>
   </div>
</body>
</html>
```

Output:

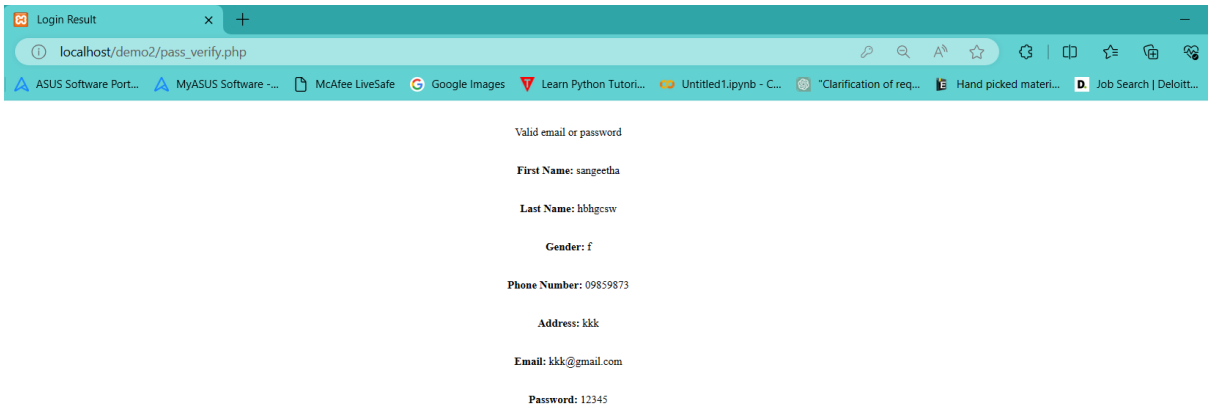
Signup.php:



Signup result:



Login output:



Lab Task Question:

Design a comprehensive web application for exam registration utilizing HTML, PHP, and MySQL. The application should include the following functionalities:

1. *Registration Form:*

Create a registration form with the following fields:

- Year

- Semester

- ID

- Name

- Gender

- Mobile Number

- Photo upload option

- List of subjects (checkboxes)

Additionally, implement JavaScript to dynamically calculate the amount to be paid based on the number of selected subjects.

2. *Data Storage:*

Upon submission of the registration form, ensure that the entered data is securely stored in a MySQL database. Display a success message confirming the successful registration.

3. *Login Page:*

Design a login page where students can authenticate themselves.

4. *Hall Ticket Display:*

Upon successful login, students should be able to view their registered data displayed as a hall ticket. Include a print option for convenience.

Your task is to develop the entire web application as described above, ensuring functionality, security, and user-friendly design. Provide appropriate error handling and validation mechanisms to enhance the robustness of the application.

Note: You are required to implement the frontend using HTML, backend logic using PHP, and database management using MySQL. Ensure that your solution meets the specified requirements and adheres to best practices in web development.