**PHP ROUGH RECORD**

**Submitted To : Reena Miss**

**Submitted by : IHSAN K**

**SECTION I**

**SIMPLE PROGRAMS USING PHP**

**FACTORIAL**

**Experiment No: 1 Date: 18/01/22**

**AIM**

To find the Factorial of a number.

**ALGORITHM**

Step1 : Start

Step2 : Read the number ,num

Step3 : Set fact=1 , i=1

Step4 : check i<=num if its true do

Step5 : fact=fact\*i

Step6 : Print the factorial of the number, fact

Step7 : increment i by one , i=i+1

Step8 : Stop

**SOURCE CODE**

<!DOCTYPE html>

<html>

<body>

<center>

<div class="outer-box">

<div style="margin-top: 178px;" class="boxed">

<div class="inner-box">

<form method="post">

<h2 style="font-family: monospace;color: white;">Factorial</h2><br>

<input type="text" name="number" placeholder="Enter the No.">

<input type="submit" name="submit" value="submit">

</form>

<?php

if($\_POST)

{

$fact = 1;

$num=$\_POST['number'];

for($i=1;$i<=$num;$i++){

$fact=$fact\*$i;

}

echo "<br>";

echo "<i style='color: white;'>factorial of $num is :$fact </i>";

}

echo "<br><br>";

?>

</center></div></div></div>

<style>

.boxed {

margin-left: 500px;

margin-right: 500px;

background-color: #00695c; }

</style></body></html>

**OUTPUT**

****

**SUM OF DIGITS**

**Experiment No: 2 Date: 18/01/22**

**AIM**

To find the sum of digits.

**ALGORITHM**

Step1 : Start

Step2 : Read the number ,num

Step3 : Print the given number,num and Set rem=0,sum=0

Step4 : check the condition num ! = 0 if its true do

Step5 : rem=num%10, sum=sum+rem, num=num/10

Step6 : Print the sum of digits, sum

Step7 : Stop

**SOURCE CODE**

<!DOCTYPE html>

<html>

<body>

<center>

<div class="outer-box">

<div style="margin-top: 178px;" class="boxed">

<div class="inner-box">

<form method="post">

<h2>Sum of digits</h2><br>

<input name="number" placeholder="Enter a number ">

<input type="submit" name="submit" value="submit">

</form>

<?php

if ($\_POST) {

$num = $\_POST['number'];

$sum = 0;

$rem = 0;

for ($i = 0; $i <= strlen($num); $i++) {

$rem = $num % 10;

$sum = $sum + $rem;

$num = $num / 10;

}

echo "<br><br>Sum of Digits is :$sum";

echo "<br><br>";

}

?>

</center></div></div></div>

<style>

.boxed {

border: 1px solid green;

margin-left: 500px;

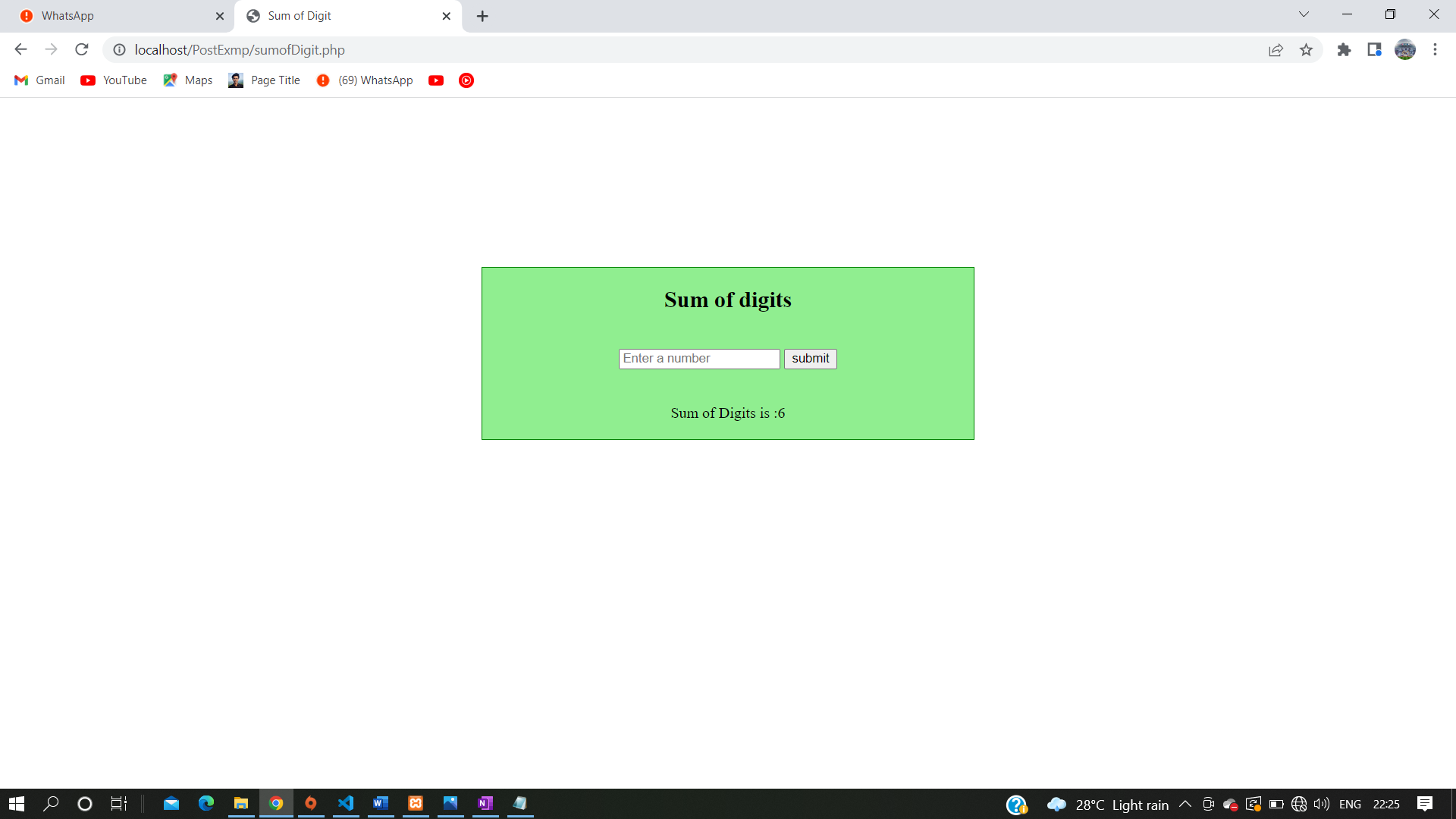
margin-right: 500px;

background-color: lightgreen;

}

</style></body></html>

**OUTPUT**

****

Here the input is 123

**FIBONACCI SEQUENCE**

**Experiment No: 3 Date: 18/01/22**

**AIM**

To print Fibonacci series.

**ALGORITHM**

Step1 : Start

Step2 : Set n1=0 ,n2=1

Step3 : Read the limit of the Fibonacci sequence, n

Step4 : Set i =1

Step5 : Check condition i<n if its true

Step6 : Do n3 = n1 + n2,n1 = n2,n2 = n3

Step7: Print the sequence n1,n2,n3

Step8 : increment i by one , i=i+1 upto n fails

Step9 : Stop

**SOURCE CODE**

<!DOCTYPE html>

<html>

<body>

<center>

<div class="outer-box">

<div style="margin-top: 178px;" class="boxed">

<div class="inner-box">

<form method="post">

<h2 style="font-family: monospace;color: white;">Fibanocci Series</h2><br>

<input type="text" name="number" placeholder="Fibanocci series upto..">

<input type="submit" name="submit" value="submit">

</form>

<?php

if ($\_POST) {

$num = $\_POST['number'];

$n1 = 0;

$n2 = 1;

echo "<br><br><i style='color: white;'>",$n1 . ' ' . $n2 . " </i>";

for ($i = 0; $i < $num; $i++) {

$n3 = $n2 + $n1;

echo "<i style='color: white;'>",$n3 . " </i>";

$n1 = $n2;

$n2 = $n3;

}

}

echo "<br><br>";

?>

</center></div></div> </div>

<style>

.boxed {

margin-left: 500px;

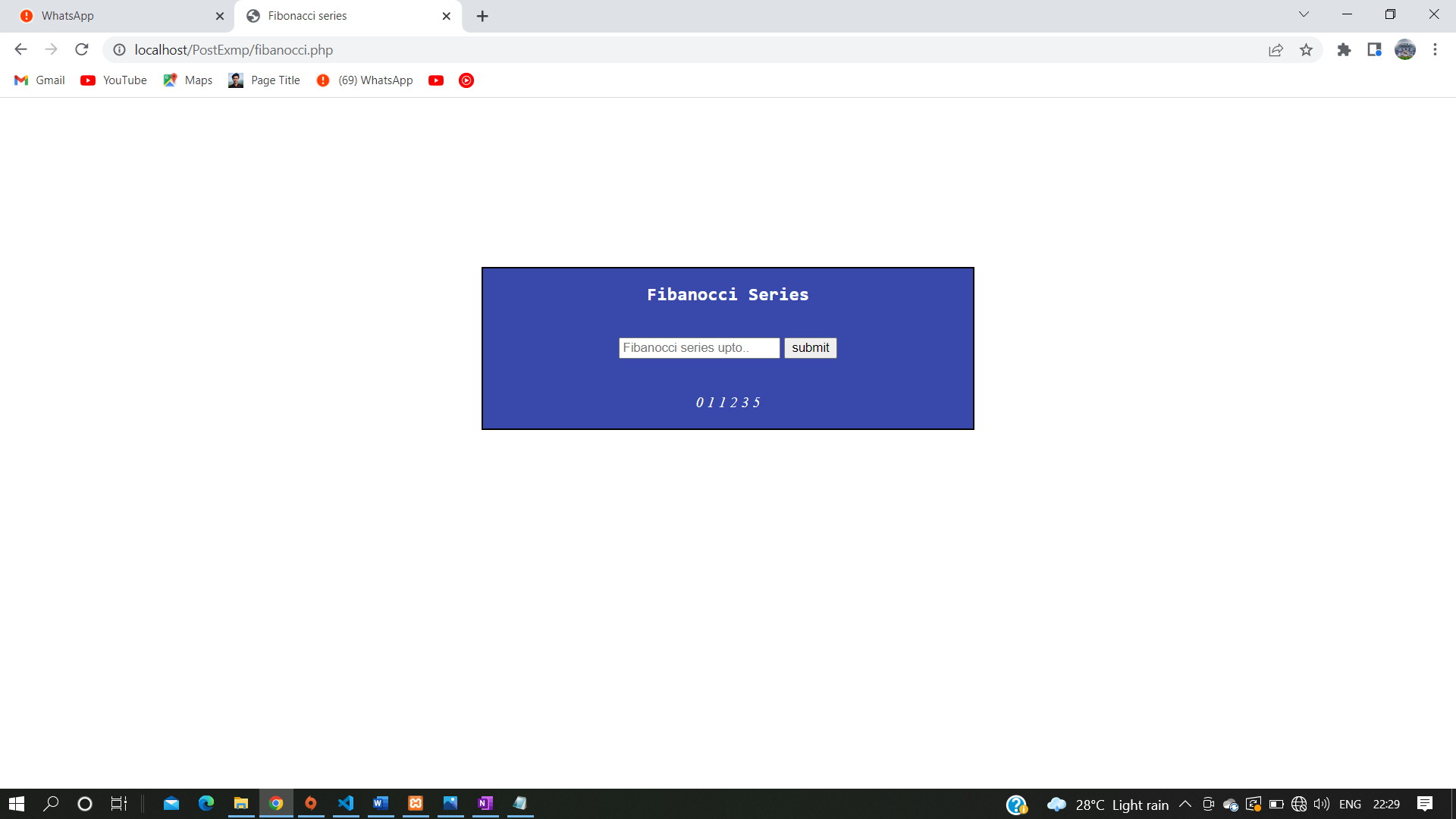
margin-right: 500px;

background-color: #3949ab;

}

</style></body></html>

**OUTPUT**

****

Here the input is 4

**ARMSTRONG NUMBER**

**Experiment No: 4 Date: 18/01/22**

**AIM**

To check whether the given number is an Armstrong or not.

**ALGORITHM**

Step1 : Start

Step2 : Read the number ,num

Step3 : Set sum=0 , a=num

Step4 : check condition a!=0 if its true

Step5 : Do rem=a%10,sum=sum+rem\*rem\*rem,a=a/10

Step6 : if num=sum

Step7 : Print the num is an Armstrong number

Step8 : else

Step7 : Print the num is not an Armstrong number

Step8 : Stop

**SOURCE CODE**

<!DOCTYPE html>

<html>

<body>

<center>

<div class="outer-box">

<div style="margin-top: 178px;" class="boxed">

<div class="inner-box">

<form method="post">

<h2 style="font-family: monospace;color: white;">Armstrong</h2><br>

<input type="text" name="number" placeholder="Enter the No.">

<input type="submit" name="submit" value="submit"><br><br>

</form>

<?php

if ($\_POST) {

$num = $\_POST['number'];

$a = $num;

$sum = 0;

while ($a != 0) {

$rem = $a % 10;

$sum = $sum + ($rem \* $rem \* $rem);

$a = $a / 10;

}

if ($num == $sum) {

echo "<i style='color: white;'>","$num is a Armstrong Number </i>";

} else {

echo "<i style='color: white;'>","$num is not a Armstrong Number </i>";

}

}

echo "<br><br>";

?>

</center></div></div></div>

<style>

.boxed {

border: 2px solid black;

margin-left: 500px;

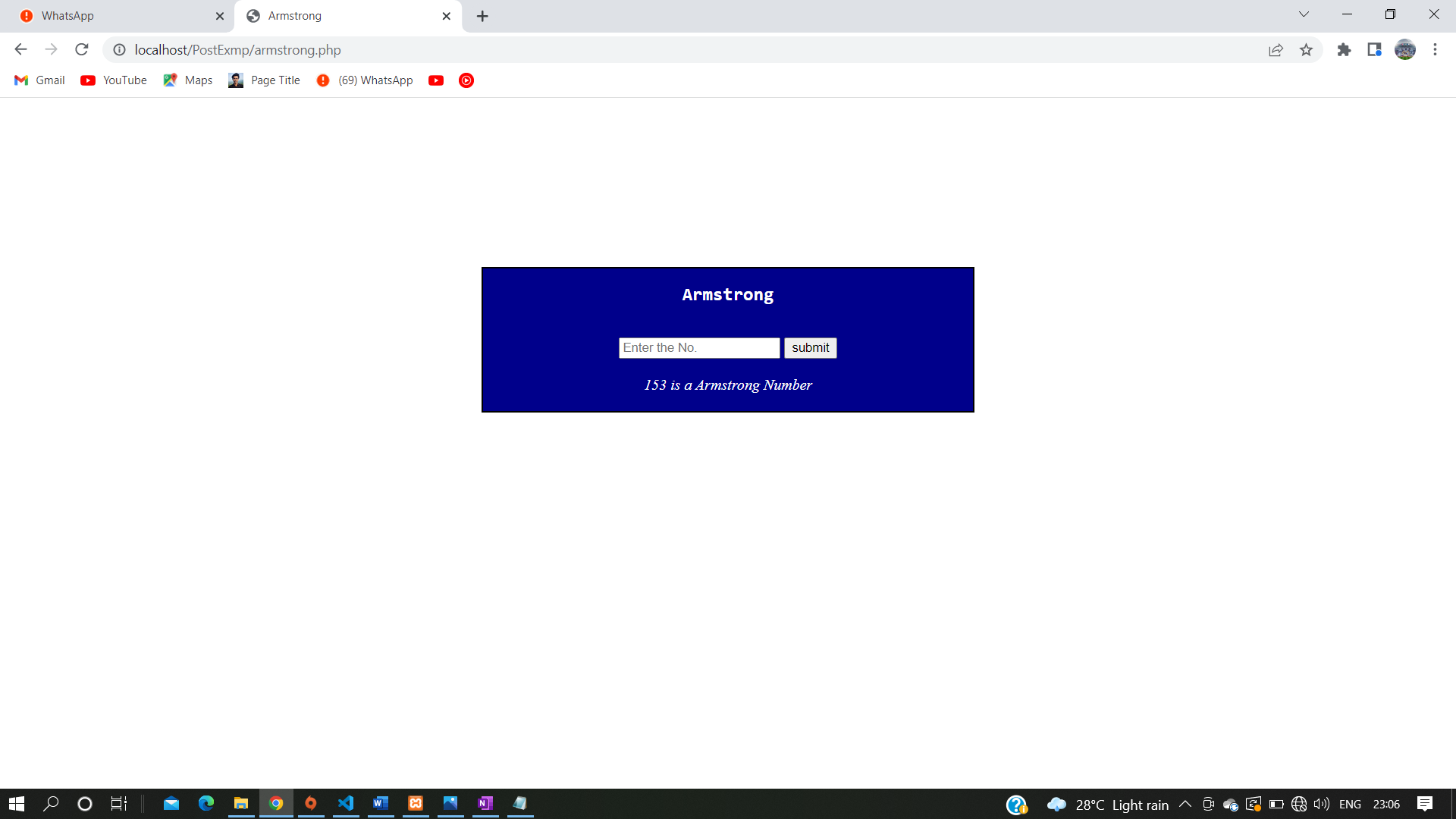
margin-right: 500px;

background-color: darkblue;

}

</style></body></html>

**OUTPUT**

****

Here the input is 153

**SORT**

**Experiment No: 5 Date: 19/01/22**

**AIM**

To sort an array of numbers ,strings etc.

**ALGORITHM**

Step1 : Start

Step2 : Create an array of numbers and assign the values into variable a

Step3 : Print the Array elements

Step4 : Sort the array elements using sort(a) function

Step5 : Print the Sorted array elements

Step6 : Create an array of strings and assign the values

Step7 : Print the Array elements

Step8 : Sort the array elements using sort( ) function

Step9 : Print the Sorted array elements and unsorted elements

Step10: Stop

**SOURCE CODE**

<!DOCTYPE html>

<html>

<head><title>Sort and unsort</title></head>

<body>

<center>

<div class="outer-box">

<div style="margin-top: 178px;" class="boxed">

<div class="inner-box">

<h2 style="font-family: monospace;color: white;">

Sorting Array of Strings</h2><br>

<?php

$cars = array("One", "Two", "Three","Four");

$ucars = array("One", "Two", "Three","Four");

sort($cars);

$clength = count($cars);

echo "<i style='color: white;'>Sorted strings:<br><br></i>";

foreach ($cars as $sorted\_names) {

echo "<i style='color: white;'>$sorted\_names<br></i>";

}

echo "<i style='color: white;'>Unsorted strings:<br><br></i>";

foreach ($ucars as $Unsorted\_cars) {

echo "<i style='color: white;'> $Unsorted\_cars<br></i>";

}

echo "<br><br>";

?>

</div></div>

<div class="boxed2" style="margin-left: 500px;border: 2px solid black;

background-color: #860d4d;

margin-left: 764px;

margin-right: 181px;

margin-top: -340px;

height: 336px;">

<h2 style= "font-family: monospace;color: white;">

Sorting of Array of Numbers</h2><br>

<?php

$numbers = array(42, 14, 15, 74, 45);

$unumbers = array(42, 14, 15, 74, 45);

sort($numbers);

$clength = count($cars);

echo "<i style='color: white;'>Sorted Numbers<br><br></i>";

foreach ($numbers as $sorted\_numbers) {

echo "<i style='color: white;'>$sorted\_numbers<br></i>";

}

echo "<i style='color: white;'>Unsorted Numbers:<br><br></i>";

foreach ($unumbers as $Unsorted\_numbers) {

echo "<i style='color: white;'> $Unsorted\_numbers<br></i>";

}

echo "<br>";

?></div> </center></div>

<style>

.boxed {

border: 2px solid black;

margin-left: 200px;

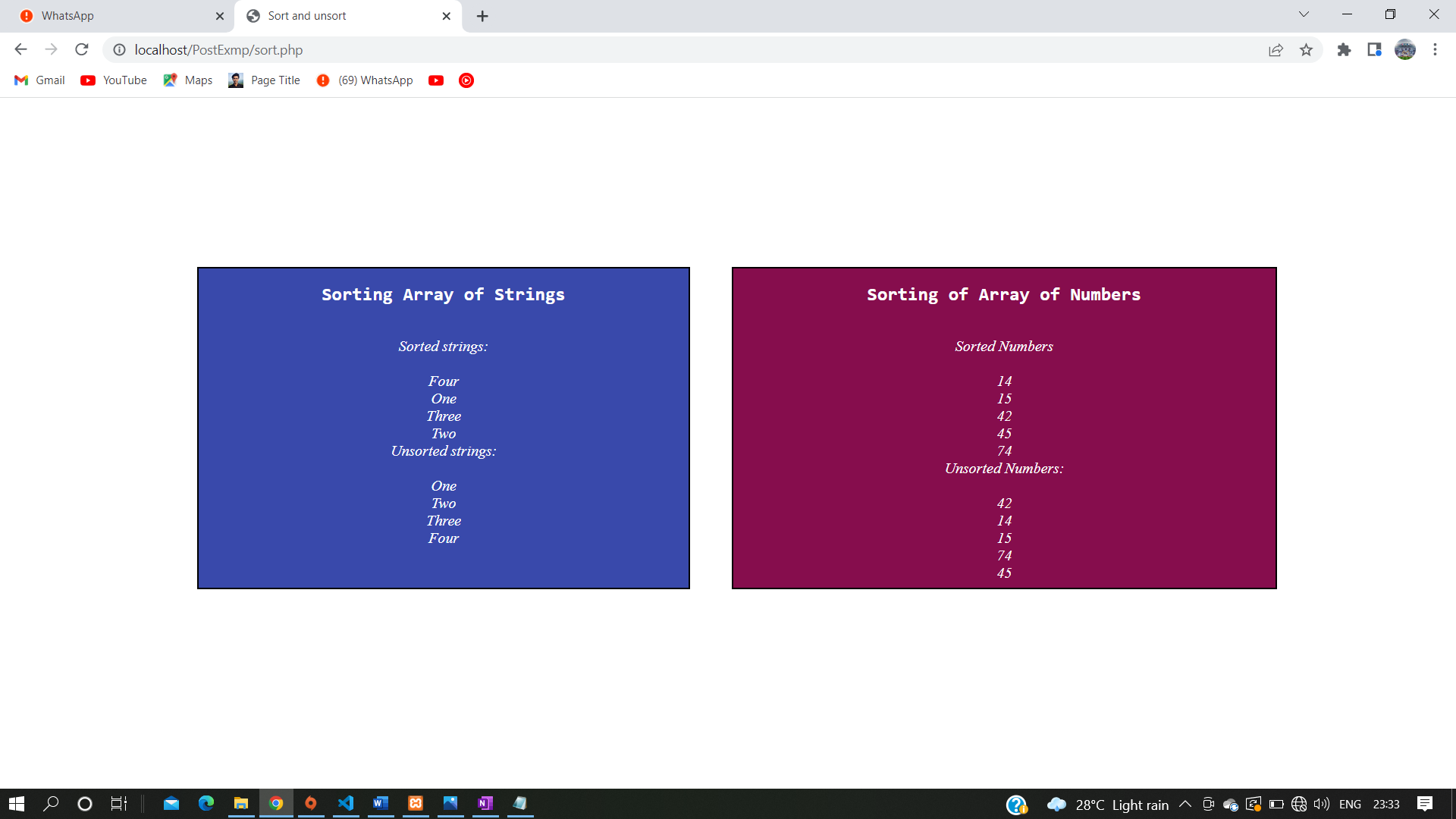
margin-right: 800px;

background-color: #3949ab;

height: 336px;

}

</style></body></html>

**OUTPUT**

**SECTION II**

**REGISTRATION FORM**

**Experiment No: 6 Date: 03/02/22**

**AIM**

Create a registration form and insert the data into database using the form through PHP post method.

**ALGORITHM**

Step1 : Start

Step2 : Create a form to enter name, address ,phone number ,email,

username and password

Step3 : Take the values from the user and submit the data’s using post

method

Step4 : The database stores the values

Step5 : And then display the data’s in the Web browser

Step6 : Stop

**CREATION OF DATABASE AND TABLE**

Step1 : Start

Step2 : Open XAMP and start Apache and MySQL

Step3 : To create a DATABASE, take the admin panel of MySQL and take new

Then create a New database called ‘reg’.

Step4 : Open the database ‘reg’

Step5 : Create TABLE called ‘profil’ with fields like id, ename, addr,

phone,email,u\_name and p\_name

Step6 : Stop

**SOURCE CODE**

<html>

<head>

<title>

Profile

</title>

<link rel="stylesheet" href="css/style.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

</head>

<?php

session\_start();

?>

<body style="background-color: #b5b5bc;">

<form method="POST" action="fileUploadScript.php" enctype="multipart/form-data" >

<section class="vh-100 gradient-custom">

<div class="container py-5 h-100">

<div class="row d-flex justify-content-center align-items-center h-100">

<div class="col-12 col-md-8 col-lg-6 col-xl-5">

<div class="card bg-dark text-white" style="border-radius: 1rem;">

<div class="card-body p-5 text-center" style="height: 1020px;margin-top: -49;">

<div class="mb-md-5 mt-md-4 pb-5">

<h2 class="fw-bold mb-2 text-uppercase">Registration Form</h2>

<p class="text-white-50 mb-5">Please enter your details here</p>

<div class="form-outline form-white mb-4">

<label class="form-label">Name</label>

<input type="text" name="name" class="form-control form-control-lg" placeholder="Enter Your name.." />

</div>

<div class="form-outline form-white mb-4">

<label class="form-label">Address</label>

<input type="text" name="address" class="form-control form-control-lg" placeholder="Enter Your Address.." />

</div>

<div class="form-outline form-white mb-4">

<label class="form-label">Phone Number</label>

<input type="text" name="phone" class="form-control form-control-lg" placeholder="Enter Your Phone Number" />

</div>

<div class="form-outline form-white mb-4">

<label class="form-label">Mail</label>

<input type="mail" name="mail" class="form-control form-control-lg" placeholder="Enter Your Email.." />

</div>

<div class="form-outline form-white mb-4">

<label class="form-label">UserName</label>

<input type="text" name="username" class="form-control form-control-lg" placeholder="Enter Your name.." />

</div>

<div class="form-outline form-white mb-4">

<label class="form-label">Password</label>

<input type="password" name="pass" class="form-control form-control-lg" placeholder="Enter strong password" />

</div>

<button class="btn btn-outline-light btn-lg px-5" type="submit" name="submit" value="submit" formaction="Reg-form.php">Submit</button>

<button class="btn btn-outline-light btn-lg px-5" type="submit" name="submit" value="login" formaction="login.php">Login</button>

</div>

</div>

</div>

</div>

</div>

</div>

</section><br><br><br><br><br><br><br><br><br><br><br><br>

<?php

$name = "";

$address = "";

$phone = "";

$email = "";

$password = "";

$uname = "";

if (isset($\_POST['submit'])) {

$name = $\_POST['name'];

$address = $\_POST['address'];

$phone = $\_POST['phone'];

$email = $\_POST['mail'];

$password = $\_POST['pass'];

$uname = $\_POST['username'];

$db = mysqli\_connect('localhost', 'root', "", 'reg');

mysqli\_query($db, "INSERT INTO regform(name,address,phone,mail,password,username)VALUES('$name','$address','$phone','$email','$password','$uname')");

}

?>

<?php

$db = mysqli\_connect('localhost', 'root', '', 'reg');

$query = mysqli\_query($db, "SELECT \*FROM regform");

if (isset($\_POST['submit'])) {

echo"<table>

<tr>

<th>Name</th>

<th>Address</th>

<th>Phone</th>

<th>Mail</th>

<th>Password</th>

<th>Username</th>

</tr>";

while($row=mysqli\_fetch\_array($query)){

echo"<tr>

<td>".$row['name']."</td>

<td>".$row['address']."</td>

<td>".$row['phone']."</td>

<td>".$row['mail']."</td>

<td>".$row['password']."</td>

<td>".$row['username']."</td>

</tr>";

}

echo"</table>";

}

?>

<style>

table, th, td {

border:1px solid black;

}

</style>

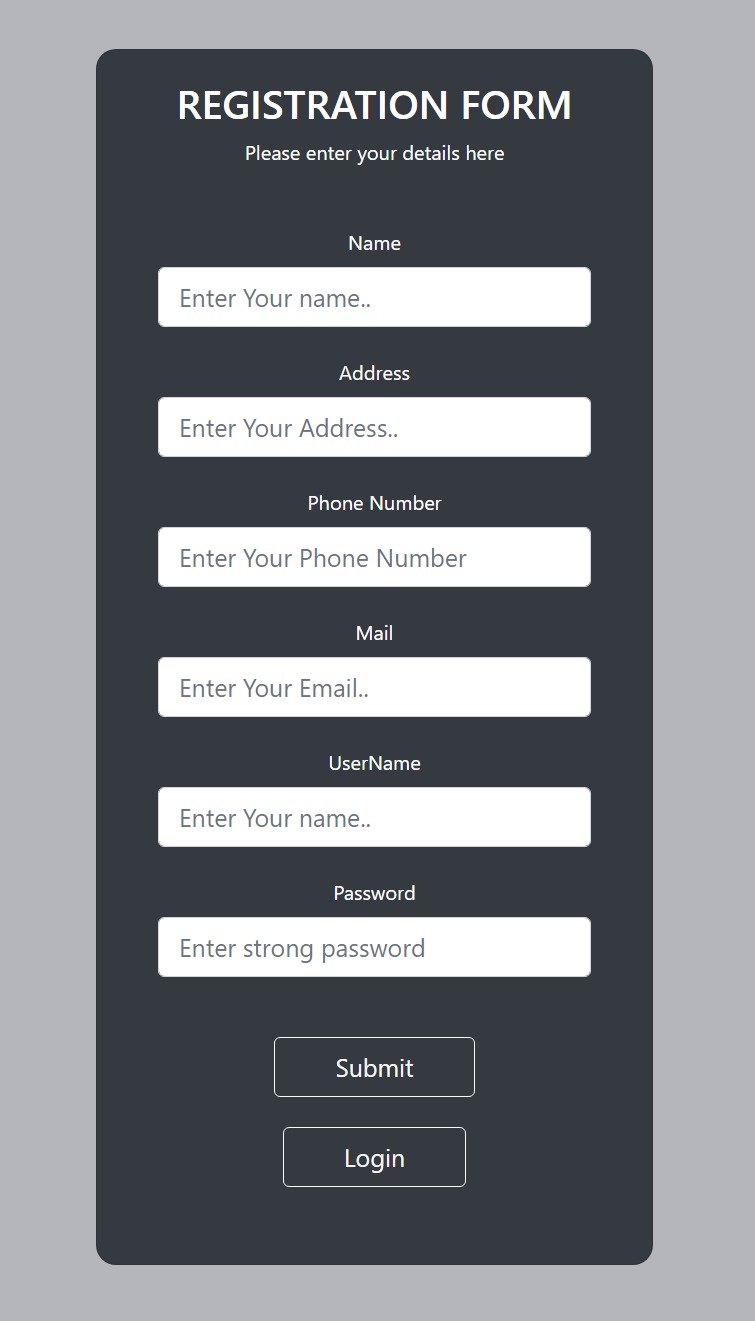
</table>

</form>

</body>

</html>

**OUTPUT**

****

****After submitting we got our values in a table

**Create a Login Form**

**Experiment No: 2 Date:3/2/22**

**AIM**

Write a PHP program to Create a login form in PHP to sign into your registered

account (using PHP session ).And also add a logout button to exit from the account.

**ALGORITHM**

Step 1: Start

Step 2: Enter into the login web page

Step 3: Sign in to the Registered Account

Step 4: Enter username and password

Step 5: Username and password valid then logged into the home page

Step 6: Otherwise it couldn’t be login (Invalid or No registered user) and new user can register first

Step 7: And also add the logout button on the home page

to exit from the account

Step 8: Stop

**SOURCE CODE**

<html>

<head>

    <title>

        Profile

    </title>

    <link *rel*="stylesheet" *href*="css/style.css">

    <link *rel*="stylesheet" *href*="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" *integrity*="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" *crossorigin*="anonymous">

</head>

<body *style*="background-color: #b5b5bc;">

    <form *method*="post">

        <section *class*="vh-100 gradient-custom">

            <div *class*="container py-5 h-100">

                <div *class*="row d-flex justify-content-center align-items-center h-100">

                    <div *class*="col-12 col-md-8 col-lg-6 col-xl-5">

                        <div *class*="card bg-dark text-white" *style*="border-radius: 1rem;">

                            <div *class*="card-body p-5 text-center" *style*="height: 545px;margin-top: -49;">

                                <div *class*="mb-md-5 mt-md-4 pb-5">

                                    <h2 *class*="fw-bold mb-2 text-uppercase">Login</h2>

                                    <p *class*="text-white-50 mb-5">Please enter your username and password here</p>

                                    <div *class*="form-outline form-white mb-4">

                                        <label *class*="form-label">Enter UserName</label>

                                        <input *type*="text" *name*="u\_name" *class*="form-control form-control-lg" *placeholder*="Enter Username" />

                                    </div>

                                    <div *class*="form-outline form-white mb-4">

                                        <label *class*="form-label">Password</label>

                                        <input *type*="password" *name*="p\_word" *class*="form-control form-control-lg" *placeholder*="Enter Password" />

                                    </div>

                                    <input *type*="submit" *name*="submit" *value*="submit" *class*="submit" *style*="

    padding-left: 10px;

    width: 116px;

    height: 45px;

"></a><br>If you dont have account click here<br><br>

                                    <a *href*="Reg-form.php"><input *type*="button" *name*="register" *value*="REGISTER" *class*="register" *style*="

    padding-left: 10px;

    width: 90px;

    height: 40px;

    padding-top: 6px;

"></a>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </section>

    </form>

        <?php

        session\_start();

        $con = mysqli\_connect('localhost','root', '', 'reg');

        $submitted\_uname;

        $submitted\_pass;

        if (count($\_POST) > 0) {

            $submitted\_uname = $\_POST["u\_name"];

            $submitted\_pass = $\_POST["p\_word"];

            $query = mysqli\_query($con, "SELECT \* FROM regform WHERE username ='$submitted\_uname'");

            if (mysqli\_num\_rows($query) == 0) {

                echo "NO REGISTERED USER";

            } else {

                $row = mysqli\_fetch\_array($query);

                $saved\_uname = $row['username'];

                $saved\_pass = $row['password'];

                if ($submitted\_pass == $saved\_pass) {

*//SUCCESSFUL LOGIN*

                    $\_SESSION["name"] = $saved\_uname;

                    header("Location:welcome.php");

                } else {

                    echo "Invalid password";

                }

            }

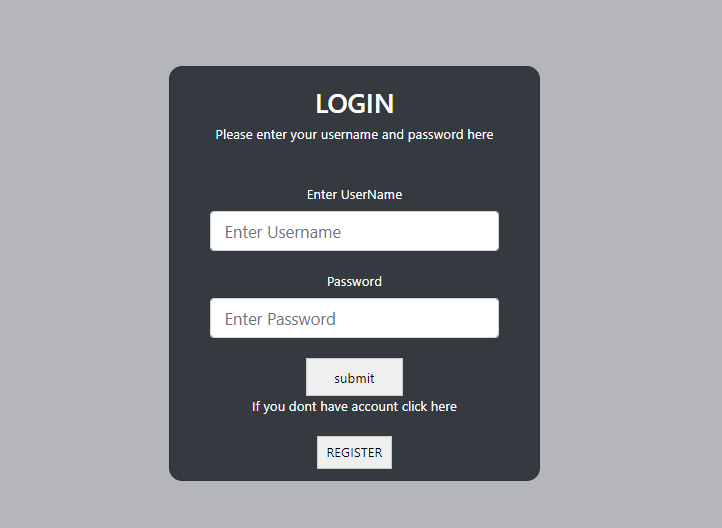
        }

        ?>

</body>

</html>

**Output**

****

**Welcome.php**

<html>

<head>

    <title>

        Profile

    </title>

    <link *rel*="stylesheet" *href*="css/style.css">

</head>

    <body>

        <?php

            session\_start();

            echo "<center>";

            echo"<br><br><h2>welcome ". $\_SESSION["name"]."</h2><br>";

            echo "You Are successfully registered!<br>";

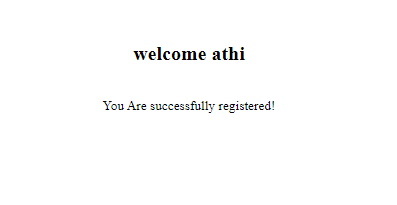
            echo "</center>";

        ?>

    </body>

</html>

**Output**

****

**Logout.php**

<?php

session\_start();

if(session\_destroy())

{

header("Location: login.php");

}

?>

**Create a Form to Upload Images**

**Experiment No: 3**

**AIM**

Write a PHP program to create a form to upload images into a database. And also create a PHP page to view all resources uploaded.

**ALGORITHM**

Step 1:Start

Step 2:Enter into the web page

Step 3:Upload single image on the web page

Step 4:View all resources uploaded

Step 5:Stop

**SOURCE CODE**

<?php

$db = mysqli\_connect('localhost', 'root','', 'reg');

$msg = " ";

if (isset($\_POST["upload"])) {

    $image = $\_FILES["image"]["name"];

    $target = "image/" . basename($image);

    $sql = "INSERT INTO img(file) VALUES('$image')";

    mysqli\_query($db, $sql);

    if (move\_uploaded\_file($\_FILES['image']['tmp\_name'], $target)) {

        $msg = "image uploaded successfully";

    } else {

        $msg = "failed to upload image";

    }

}

$result = mysqli\_query($db, "SELECT \* FROM img");

if (isset($\_POST["delete"]))

{

    mysqli\_query($db,"TRUNCATE TABLE img");

}

?>

<html *lang*="en">

<body>

    <form *enctype*="multipart/form-data" *method*="post">

        <input *type*="file" *name*="image"><br><br>

        <input *type*="submit" *class*="btn-btn-primary" *name*="upload" *value*="SUBMIT"><br><br>

        <input *type*="submit" *class*="btn-btn-primary" *name*="delete" *value*="DELETE ALL">

    </form>

    <centre>

        <?php

        $db = mysqli\_connect('localhost', 'root', '', 'reg');

        $result = mysqli\_query($db, "SELECT \* FROM img");

        while ($row = mysqli\_fetch\_array($result)) {

            echo "<img src='image/" . $row[0] . "'width='130px' height='100px'>";

        }

        mysqli\_close($db);

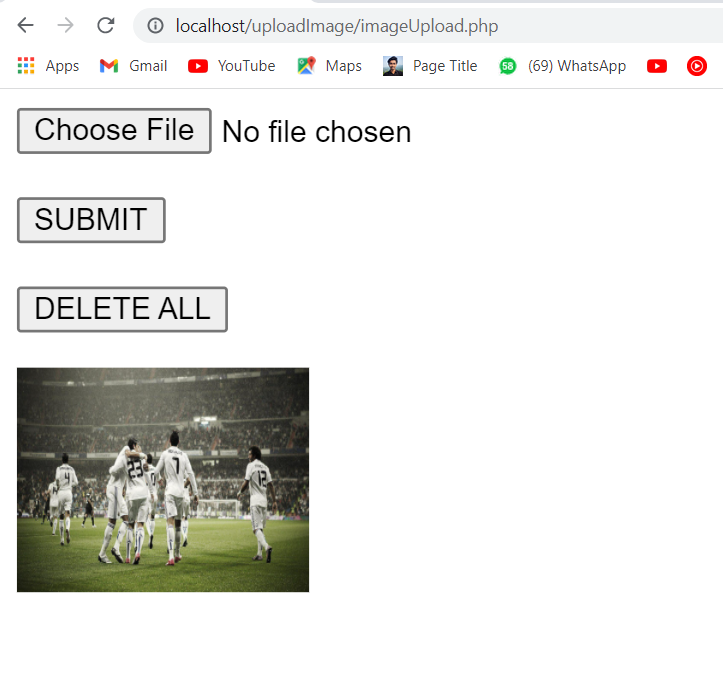
        ?>

    </centre>

</body>

</html>

**Output**

****

**Create a PHP page to upload & delete multiple images**

**Experiment No: 4 Date:**

**AIM**

Create a PHP page to upload multiple images into a database and view it & delete it.

**ALGORITHM**

Step 1:Start

Step 2:Enter into the web page

Step 3:Read number from form as limit

Step 4:Upload multiple images on the web page

Step 5:View all resources uploaded

Step 6:Delete the images from the database

Step 7:Stop

**SOURCE CODE**

**Gallery.php**

<html>

<head>

<title>Gallery</title>

</head>

<link rel="stylesheet" href="css/gallery.css">

<body style="background-image:url('resources/images/menubg.gif');

background-size:cover">

<?php

session\_start();

$uname = $\_SESSION["username"];

$con=mysqli\_connect('localhost','root','','registration');

$query=mysqli\_query($con,"SELECT \* FROM userdata WHERE username ='$uname'");

$row=mysqli\_fetch\_array($query);

$split = explode(",",$row['img\_collect']);

$image\_files=array("jpg","png");

//Making window size dynamic

$x=596;

$x2=416;

$size=count($split);

$incr=168;

for($j=5;$j<=$size;$j++)

{

if($j%2 == 1)

{

$x=$x+$incr;

$x2=$x2+$incr;

}

}

?>

<div id="gallery" style=<?php echo "height:".$x."px";?>>

<br><br>

<x class="title">GALLERY</x><br><br>

<div id="bg\_box" style=<?php echo "height:".$x2."px";?>>

<form method="post" name="uploading" enctype="multipart/form-data" class="up\_form"><br>

<input type="file" name="files[]" id="select\_file" class="files" accept=".jpg,.png,.mp4,.mkv,.flv" multiple>

<label for="select\_file" class="upload\_label">SELECT FILES</label>&nbsp;

<input type="submit" name="upload" value="UPLOAD" class="upload">

<a href="user\_home.php"><input type="button" name="back" value="BACK" class="back"></a>

</form>

<div class="del\_btn"><form method="post" name="deleting" class="delete\_all">

<input type="submit" name="delete" value="DELETE ALL" class="delete"></div>

</form>

<div class="line1"></div>

<div class="images">

<?php

$directory= "uploads/";

$count=0;

foreach ($split as $img)

{

if ($img != "")

{

$rec\_path = $directory.$img;

$filetype=pathinfo($rec\_path, PATHINFO\_EXTENSION);

if (in\_array($filetype,$image\_files, TRUE))

{

echo "<img width='200' height='151' src='$rec\_path'>&nbsp";

}

else

{

echo "<video width='200' height='151' controls='controls'><source src='$rec\_path'></video>&nbsp";

}

if ($count >= 1)

{

echo "<br>";

$count=0;

continue;

}

}

$count=$count+1;

}

if(isset($\_POST['upload']))

{

$fullpath="";

$sqlpath = $row['img\_collect'];

$fileNames=array\_filter($\_FILES['files']['name']);

$pathnames=array\_filter($\_FILES['files']['tmp\_name']);

$num = count($fileNames);

for ($i=0;$i<$num;$i++)

{

if(($fileNames[$i]) != end($fileNames))

{

$fullpath = $fullpath.$fileNames[$i].",";

}

else

{

$fullpath = $fullpath.$fileNames[$i];

}

$target\_file = $directory.$fileNames[$i];

move\_uploaded\_file($pathnames[$i],$target\_file);

}

$newpath = $sqlpath;

if ($fullpath != '')

{

if (($sqlpath == ''))

{

$newpath = $fullpath;

}

else

{

$newpath = $newpath.",".$fullpath;

}

}

mysqli\_query($con, "UPDATE userdata SET img\_collect='$newpath' WHERE username='$uname'");

header("Location:gallery.php");

}

if(isset($\_POST['delete']))

{

mysqli\_query($con, "UPDATE userdata SET img\_collect='' WHERE username='$uname'");

foreach ($split as $img)

{

$f\_path = $directory.$img;

unlink($f\_path);

}

header("Location:gallery.php");

}

?>

</div>

</div>

</div>

</body>

</html>

**Gallery.css**

#gallery{

width: 620px;

margin: auto auto;

border-radius: 15px;

border: 5px solid rgb(3, 10, 102);

font-family: 'comic sans ms';

font-size: 15px;

color: rgb(4, 13, 135);

background-color: rgba(173, 218, 232, 0.7);

}

.title{

margin: auto 244px;

font-weight: bolder;

font-size: 30px;

font-family: 'gisha';

text-shadow: 0 0 2px rgb(1, 63, 156), 0 0 5px rgb(143, 210, 236), 0 0 8px rgb(1, 63, 156), 0 0 12px rgb(143, 210, 236);

color: rgb(7, 26, 131);

}

#bg\_box{

width: 450px;

margin: auto auto;

border: none;

font-family: 'gisha';

font-size: 12px;

background-color: rgba(255, 255, 255, 0.55);

box-shadow: 0 0 5px rgb(21, 1, 1);

color: black;

}

.up\_form{

padding-top: 14px;

margin: -4px -235px;

color: red;

}

.files{

visibility: hidden;

}

.upload\_label{

margin: auto 3px;

padding-top: 3px;

padding-bottom: 4px;

padding-left: 10px;

padding-right: 10px;

box-shadow: 0 0 2px rgb(4, 15, 114);

color: rgb(2, 24, 122);

background-color: rgb(79, 175, 220);

font-size: 11px;

font-weight: bold;

}

.upload{

margin: auto -2px;

padding-top: 3px;

padding-bottom: 4px;

padding-left: 10px;

padding-right: 10px;

box-shadow: 0 0 2px rgb(21, 1, 1);

border: none;

color: white;

background-color: green;

font-size: 11px;

}

.back{

margin: auto 190px;

padding-top: 3px;

padding-bottom: 4px;

padding-left: 10px;

padding-right: 10px;

box-shadow: 0 0 2px rgb(21, 1, 1);

border: none;

color: rgb(79, 175, 220);

background-color: rgb(4, 15, 114);

font-size: 11px;

font-weight: bold;

}

.delete{

margin: -16px 282px;

padding-top: 3px;

padding-bottom: 4px;

padding-left: 10px;

padding-right: 10px;

box-shadow: 0 0 2px rgb(21, 1, 1);

border: none;

color: rgb(224, 233, 237);

background-color: rgb(179, 3, 3);

font-size: 11px;

font-weight: bold;

}

.line1{

height: 1px;

background: rgb(99, 93, 93);

box-shadow: 0 0 3px rgb(21, 1, 1);

margin: 10px 23px;

}

.images{

margin: auto 23px;

}

**Output**



**Edit your Profile**

**Experiment No:5 Date:17/02/2021**

**AIM**

Write a PHP program to Create a PHP page to edit your profile and update your details in the database.

**ALGORITHM**

Step 1: Start

Step 2: Enter a webpage of registration that we already registered

Step 3: Edit the profile details

Step 4: Stop

**SOURCE CODE**

**Reg-edit.php**

**Reg.php**

<html>

<head>

<title>Registration Form</title>

</head>

<style>

table{

width:"50%";

margin-left: auto;

}

table, th, td {

border: 1px solid black;

}

body {

background-image: url("reg.jpg");

background-repeat: no-repeat;

background-size: cover;

align-content: center;

}

div {

text-align: end;

margin-left: auto ;

}

</style>

<div>

<h1>REGISTRATION FORM</h1>

<br>

<th>

<form method="post" >

Enter Name:

<input type="text" name="name" placeholder="Enter Name"><br><br>

Enter Address:

<input type="text" name="addr" placeholder="Enter Address"><br><br>

Enter Phone Number:

<input type="number" name="phone" placeholder="Enter Phone Number"><br><br>

Enter Email Id:

<input type="email" name="email" placeholder="Enter Email Id"><br><br>

Enter Username:

<input type="text" name="user" placeholder="Enter Username"><br><br>

Enter Password:

<input type="password" name="pass" placeholder="Enter Password"><br><br>

<input type="submit" name="clear" value="clear">

<input type="submit" name="submit" value="submit">

<input type="submit" name="back" value="back">

</form>

</th>

</div>

<?php

if(isset($\_POST['back']))

{

header("Location:login.php");

}

?>

<?php

$db = mysqli\_connect('localhost','root',"",'registration form');

$name="";

$addr="";

$phone="";

$email="";

$user="";

$pass="";

if(isset($\_POST['submit']))

{

$name=$\_POST['name'];

$addr=$\_POST['addr'];

$phone=$\_POST['phone'];

$email=$\_POST['email'];

$user=$\_POST['user'];

$pass=$\_POST['pass'];

mysqli\_query($db,"INSERT INTO reg\_table(name,address,phone,email,username,password) VALUES ('$name','$addr','$phone','$email','$user','$pass')");

}

?>

<?php

$db=mysqli\_connect('localhost','root',"",'registration form');

$query=mysqli\_query($db,"SELECT \* FROM reg\_table");

?>

<table>

<thead>

<tr>

<th>Name</th>

<th>Address</th>

<th>Phone Number</th>

<th>Email</th>

<th>Username</th>

<th>Password</th>

<th>Edit</th>

<th>Delete</th>

</tr>

</thead>

<?php

while($row=mysqli\_fetch\_array($query))

{

?>

<tr>

<td>

<?php echo $row['name'] ;?>

</td>

<td>

<?php echo $row['address'];?>

</td>

<td>

<?php echo $row['phone'];?>

</td>

<td>

<?php echo $row['email'];?>

</td>

<td>

<?php echo $row['username'];?>

</td>

<td>

<?php echo $row['password'];?>

</td>

<td>

<a href="edit.php?edit=<?php echo $row['id']; ?>">Edit</a>

</td>

<td>

<a href="delete.php?del=<?php echo $row['id']; ?>">Delete</a>

</td>

</tr>

<?php

if(isset($\_POST['clear']))

{

mysqli\_query($db,"TRUNCATE TABLE reg\_table");

}

?>

<?php

}

?>

</table>

</body>

</html>

**EDIT.PHP**

<html>

<head>

<title>Update</title>

</head>

<style>

table{

width:"50%";

margin-left: auto;

}

table, th, td {

border: 1px solid black;

}

body {

background-image: url("reg.jpg");

background-repeat: no-repeat;

background-size: cover;

align-content: center;

}

div {

text-align: end;

margin-left: auto ;

}

</style>

<form method="post">

Enter Name:

<input type="hidden" name="id" value="id">

<input type="text" name="name" value="<?php echo $name; ?>"><br><br>

Enter Address:

<input type="text" name="addr" value="<?php echo $addr; ?>"><br><br>

Enter Phone Number:

<input type="number" name="phone" value="<?php echo $phone; ?>"><br><br>

Enter Email Id:

<input type="email" name="email" value="<?php echo $email; ?>"><br><br>

Enter Username:

<input type="text" name="user" value="<?php echo $user; ?>"><br><br>

Enter Password:

<input type="password" name="pass" value="<?php echo $pass; ?>"><br><br>

<button type="submit" name="update">Update</button>

</form>

<?php

$db=mysqli\_connect('localhost','root','','registration form');

if(isset($\_GET['edit']))

{

$id=$\_GET['edit'];

$update=true;

$record=mysqli\_query($db,"SELECT \* FROM reg\_table WHERE id='$id' ");

if($record)

{

$n=mysqli\_fetch\_array($record);

$name=$n['name'];

$addr=$n['address'];

$phone=$n['phone'];

$email=$n['email'];

$user=$n['username'];

$pass=$n['password'];

}

}

?>

<?php

if(isset($\_POST['update']))

{

$id=$\_POST['id'];

$name=$\_POST['name'];

$addr=$\_POST['addr'];

$phone=$\_POST['phone'];

$email=$\_POST['email'];

$user=$\_POST['user'];

$pass=$\_POST['pass'];

mysqli\_query($db,"UPDATE reg\_table SET name='$name',address='$addr',phone='$phone',email='$email',username='$user',password='$pass' WHERE id=$id");

header('location:reg.php');

}

?>

</html>

**Delete.php**

<?php

$db=mysqli\_connect('localhost','root','','registration form');

if(isset($\_GET['del']))

{

$id=$\_GET['del'];

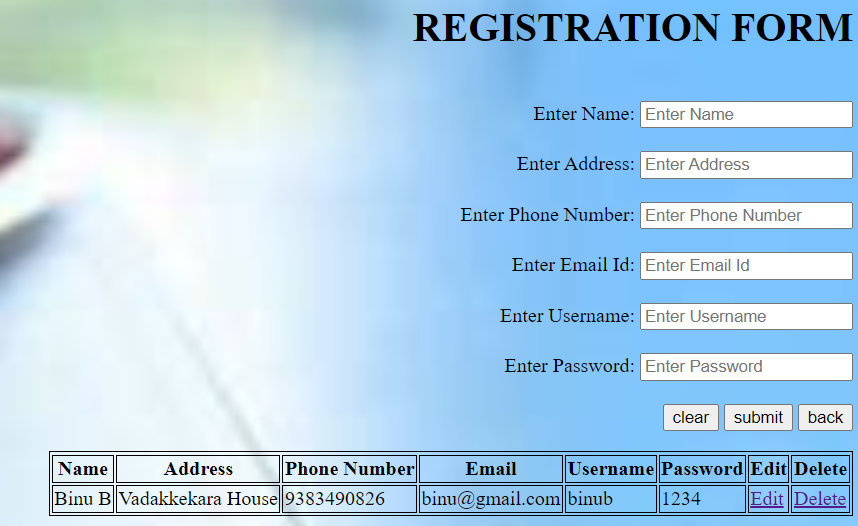
mysqli\_query($db,"DELETE \* FROM reg\_table WHERE id=$id");

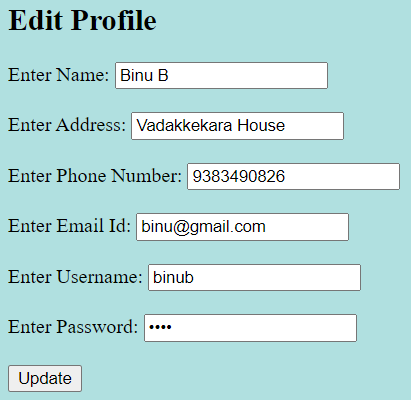
header('location:reg.php');

}

?>

**Output**









**Change in your account password**

**Experiment No: 6 Date:**

**AIM**

Write a PHP program to change your account password by specifying your correct password, new password and confirm password.

**ALGORITHM**

Step 1:Start

Step 2 : Enter into the web page

Step 3 : Enter current password

Step 4 : Enter the new password and confirm password

Step 5 : If everything is correct, change the password into the new password

Step 6 : Else print error message

Step 7 : Stop

**SOURCE CODE**

**Edit\_password.php**

<?php

$db=mysqli\_connect('localhost','root','','registration form');

if(isset($\_GET['pedit']))

{

$id=$\_GET['pedit'];

$update=true;

$record=mysqli\_query($db,"SELECT \*FROM reg\_table WHERE id=$id");

if($record)

{

$n=mysqli\_fetch\_array($record);

$name=$n['name'];

$addr=$n['address'];

$phone=$n['phone'];

$email=$n['email'];

$user=$n['username'];

$pass=$n['password'];

}}

?>

<html>

<head>

<title>Update</title>

</head>

<body bgcolor="biege">

<form method="post">

<h2>Edit & Confirm Password</h2>

<input type="hidden" name="id" value="<?php echo $id; ?>">

Password:

<input type="password" name="pass" value="<?php echo $pass; ?>"><br><br>

New Password:

<input type="password" name="pass"><br><br>

Confirm NewPassword:

<input type="password" name="cpass"><br><br>

<button type="submit" name="update">Change Password</button>

</form>

<?php

if(isset($\_POST['update']))

{

$id=$\_POST['id'];

$pass=$\_POST['pass'];

mysqli\_query($db,"UPDATE reg\_table SET password='$pass' WHERE id=$id");

header('location:reg.php');

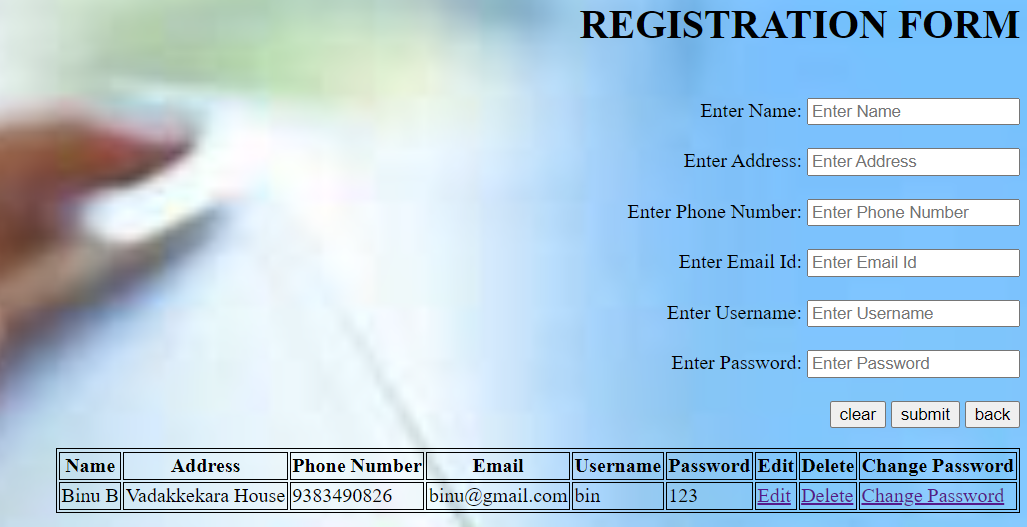
}

?>

</body>

</html>

**Output**







**Chat Box**

**Experiment No: 7**

**AIM**

Create a chat box in your web page using php.

**ALGORITHM**

Step 1:Start

Step 2:Enter into the web page

Step 3:Using the chat box

Step 4:Client and server communication each other.

Step 5:Stop

**SOURCE CODE**

**Sample\_chat.php**

<html>

    <head>

    <link *rel*="stylesheet" *href*="css/chat\_new.css">

    </head>

    <body>

        <?php

            session\_start();

            $con=mysqli\_connect('localhost','root','','stud') or die('connection error');

            $query=mysqli\_query($con,"SELECT \* FROM chat");

            $query2=mysqli\_query($con,"SELECT \* FROM chat");

            $left\_user=$\_SESSION['l\_user'];

            $right\_user=$\_SESSION['r\_user'];

        ?>

        <div *id*="left\_box">

            <form *method*="POST">

                <input *type*="text" *name*="uname\_left">

                <input *type*="submit" *name*="l\_user" *value*="Set Username">

                <br>

                Logged in as :<?php echo $left\_user ?><br>

            </form>

            <div *id*="l\_chat">

                <?php

                    while($row=mysqli\_fetch\_array($query)){

                        if($row['uname']==$left\_user){

                            echo "<x style='float:left'>";

                            echo $row['msg'].'</x><br>';

                        }

                        else{

                            echo "<x style='float:right'>";

                            echo $row['uname'].":". $row['msg']."</x><br>";

                        }

                    }

                ?>

            </div>

            <form *method*="POST">

                <input *type*="textarea" *name*="l\_msg">

                <input *type*="submit" *name*="l\_submit" *value*="SEND">

            </form>

        </div>

        <div *id*="right\_box">

            <form *method*="POST">

                <input *type*="text" *name*="uname\_right">

                <input *type*="submit" *name*="r\_user" *value*="Set Username">

                <br>

                Logged in as<?php echo $right\_user ?><br>

            </form>

            <div *id*="r\_chat">

                <?php

                    while($row=mysqli\_fetch\_array($query2)){

                        if($row['uname']==$right\_user){

                            echo "<x style='float:left'>";

                            echo $row['msg'].'</x><br>';

                        }

                        else{

                            echo "<x style='float:right'>";

                            echo $row['uname'].":". $row['msg']."</x><br>";

                        }

                    }

                ?>

            </div>

            <form *method*="POST">

                <input *type*="textarea" *name*="r\_msg">

                <input *type*="submit" *name*="r\_submit" *value*="SEND">

            </form>

        </div>

        <?php

            if(isset($\_POST['l\_user'])){

                $\_SESSION['l\_user']=$\_POST['uname\_left'];

                header("Location:simplechat.php");

            }

            if(isset($\_POST['r\_user'])){

                $\_SESSION['r\_user']=$\_POST['uname\_right'];

                header("Location:simplechat.php");

            }

            if(isset($\_POST['l\_submit'])){

                $msg\_l=$\_POST['l\_msg'];

                mysqli\_query($con,"INSERT INTO chat(uname,msg)VALUES('$left\_user','$msg\_l')");

                header("Location:simplechat.php");

            }

            if(isset($\_POST['r\_submit'])){

                $msg\_r=$\_POST['r\_msg'];

                mysqli\_query($con,"INSERT INTO chat(uname,msg)VALUES('$right\_user','$msg\_r')");

                header("Location:simplechat.php");

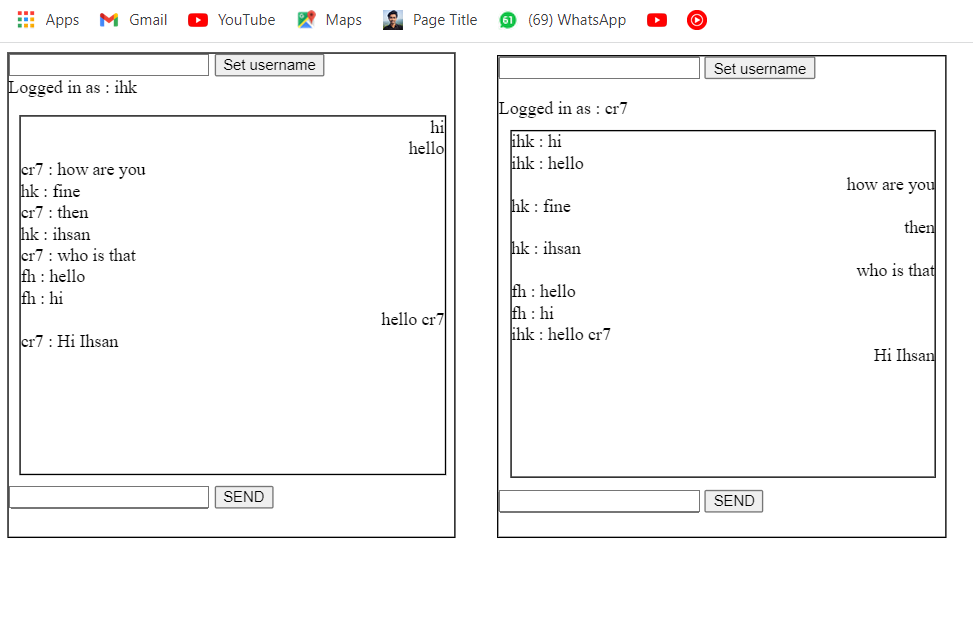
            }

        ?>

    </body>

</html>

**Output**

****

**DATABASE**

**INSERT & SELECT**

**Experiment No:1 Date:**

**AIM**

Create a webpage with laravel framework to learn controller ,view and model

concept or sufficient forms to insert data into and select data from the

database.

**PROCEDURE**

**Step 1:** Composer create-project --prefer-dist laravel/laravel crud

**Step 2:** Create a database in phpmyadmin , database name mydb and give this name in the .env file of crud

**Step 3:**Create the model Post,php artisan make:model Post -m

& it will create a create\_posts\_table in the migrations

**Step 4:**Then modify the create\_posts\_table.php & give the table name in post.php file in models

**Step 5:**After running the migrations, create a controller,php artisan make:controller PostController

**Step 6:**Create function addPost(),savePost() inPostController & create addpost.blade.php file in views

**Step 7:**Create another function postList() inPostController & create postlist.blade.php file in views

**Step 8:**After this give the routes for addpost & postlist in web.php

**Step 9:**Then run ,php artisan serve

**SOURCE CODE**

**Crud/.env**

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=mydb

DB\_USERNAME=root

DB\_PASSWORD=

**Crud/app/Models/Post.php** :

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Post extends Model

{

use HasFactory;

protected $table="posts";

}

**Crud/app/database/migrations/create\_posts\_table.php:**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('posts', function (Blueprint $table) {

$table->id();

$table->string('name');

$table->text('description');

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('posts');

}

};

**Crud/app/Http/Controllers/PostController.php:**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use DB;

use Illuminate\Support\Facades\DB as FacadesDB;

use Symfony\Contracts\Service\Attribute\Required;

class PostController extends Controller

{

public function addPost(){

return view('addpost');

}

public function savePost(Request $request){

FacadesDB::table('posts')->insert([

'name'=>$request->name,

'description'=>$request->description

]);

return back()->with('post\_add','post added successfully');

}

public function postList(){

$posts=FacadesDB::table('posts')->get();

return view('postlist',compact('posts'));

}

}

**Crud/resources/views/addpost.blade.php:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>CustomAuthController</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDP vg6uqKI2xXr2" crossorigin="anonymous">

<title>Add Post</title>

</head>

<body>

@if(Session::has('post\_add'))

<span>{{Session::get('post\_add')}}</span>

@endif

<form method="post" action="{{route('save.post')}}">

@csrf

Post: <br><input type="text" name="name" value="" ><br>

Description : <br><textarea name="description"></textarea><br>

<input type="submit" value="Submit">

</form>

</body>

</html>

**Crud/resources/views/postlist.blade.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>post list</title>

<style>

table{

border-collapse: collapse;

width: 100%;

}

td,th{

padding: 5px;

border: 1px solid;

}

</style>

</head>

<body>

@if(Session::has('post\_delete'))

<span>{{Session::get('post\_delete')}}</span>

@endif

<a style ="float:right" href="{{route('post.add')}}">Add Post</a>

<table>

<tr>

<th>ID</th>

<th>Name</th>

<th>Description</th>

</tr>

@foreach($posts as $post)

<tr>

<td>{{$post->id}}</td>

<td>{{$post->name}}</td>

<td>{{$post->description}}</td>

</tr>

@endforeach

</table>

</body>

</html>

**Crud/routes/web.php:**

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\PostController;

/\*

|--------------------------------------------------------------------------

| Web Routes

|--------------------------------------------------------------------------

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('/', function () {

return view('welcome');

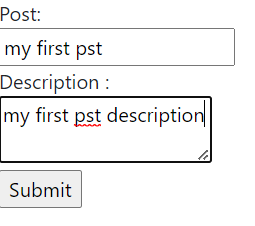
});

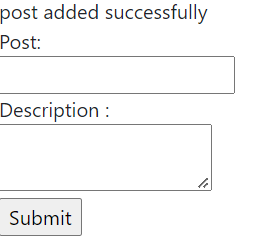
Route::get('/addpost',[PostController::class,'addPost'])->name('post.add');

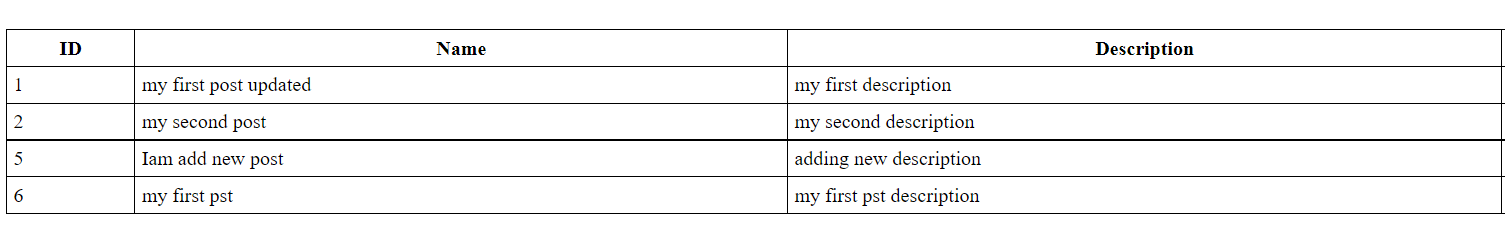
Route::post('/addpost',[PostController::class,'savePost'])->name('save.post');

Route::get('/postlist',[PostController::class,'postList'])->name('post.list');

**Output**







**EDIT & DELETE**

**Experiment No:2 Date:**

**AIM**

Develop a webpage to edit and delete the details you are added into your database.

**PROCEDURE**

**Step 1:** Composer create-project --prefer-dist laravel/laravel crud

**Step 2:** Create a database in phpmyadmin , database name mydb and give this name in the .env file of crud

**Step 3:**Create the model Post,php artisan make:model Post -m & it will create a create\_posts\_table in the migrations

**Step 4:**Then modify the create\_posts\_table.php & give the table name in Post.php file in models

**Step 5:**After running the migrations, create a controller,php artisan make:controller PostController

**Step 6:**Create function addPost(),savePost() inPostController & create addpost.blade.php file in views

**Step 7:**Create function editPost(),updatePost() inPostController & create editpost.blade.php file in views

**Step 8:**Create next function deletePost() inPostController

**Step 9:**Create another function postList() inPostController & create postlist.blade.php file in views

**Step 10:**After this give the routes for addpost,postlist,editpost,deletepost,updatepost in web.php

**Step 11:**Then run ,php artisan serve

**SOURCE CODE**

**Crud/.env**

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=mydb

DB\_USERNAME=root

DB\_PASSWORD=

**Crud/app/Models/Post.php** :

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Post extends Model

{

use HasFactory;

protected $table="posts";

}

**Crud/app/database/migrations/create\_posts\_table.php:**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('posts', function (Blueprint $table) {

$table->id();

$table->string('name');

$table->text('description');

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('posts');

}

};

**Crud/app/Http/Controllers/PostController.php:**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use DB;

use Illuminate\Support\Facades\DB as FacadesDB;

use Symfony\Contracts\Service\Attribute\Required;

class PostController extends Controller

{

public function addPost(){

return view('addpost');

}

public function savePost(Request $request){

FacadesDB::table('posts')->insert([

'name'=>$request->name,

'description'=>$request->description

]);

return back()->with('post\_add','post added successfully');

}

public function postList(){

$posts=FacadesDB::table('posts')->get();

return view('postlist',compact('posts'));

}

public function editPost($id){

$post=FacadesDB::table('posts')->where('id',$id)->first();

return view('editpost',compact('post'));

}

public function updatePost(Request $request){

FacadesDB::table('posts')->where('id', $request->id)->update([

'name'=>$request->name,

'description'=>$request->description

]);

return back()->with('post\_update','post updated successfully');

}

public function deletePost($id){

FacadesDB::table('posts')->where('id', $id)->delete();

return back()->with('post\_delete','post deleted successfully');

}

}

**Crud/resources/views/addpost.blade.php:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>CustomAuthController</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDP vg6uqKI2xXr2" crossorigin="anonymous">

<title>Add Post</title>

</head>

<body>

@if(Session::has('post\_add'))

<span>{{Session::get('post\_add')}}</span>

@endif

<form method="post" action="{{route('save.post')}}">

@csrf

Post: <br><input type="text" name="name" value="" ><br>

Description : <br><textarea name="description"></textarea><br>

<input type="submit" value="Submit">

</form>

</body>

</html>

**Crud/resources/views/editpost.blade.php:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>CustomAuthController</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

<title>Edit Post</title>

</head>

<body>

@if(Session::has('post\_update'))

<span>{{Session::get('post\_update')}}</span>

@endif

<form method="post" action="{{route('update.post')}}">

@csrf

<input type="hidden" name="id" value="{{$post->id}}">

Post: <br><input type="text" name="name" value="{{$post->name}}" ><br>

Description : <br><textarea name="description">{{$post>description}}</textarea><br>

<input type="submit" value="Submit">

</form>

</body>

</html>

**Crud/resources/views/postlist.blade.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>post list</title>

<style>

table{

border-collapse: collapse;

width: 100%;

}

td,th{

padding: 5px;

border: 1px solid;

}

</style>

</head>

<body>

@if(Session::has('post\_delete'))

<span>{{Session::get('post\_delete')}}</span>

@endif

<a style ="float:right" href="{{route('post.add')}}">Add Post</a>

<table>

<tr>

<th>ID</th>

<th>Name</th>

<th>Description</th>

<th>Action</th>

</tr>

@foreach($posts as $post)

<tr>

<td>{{$post->id}}</td>

<td>{{$post->name}}</td>

<td>{{$post->description}}</td>

<td>

<a href="/editpost/{{$post->id}}">Edit</a> |

<a href="/deletepost/{{$post->id}}">Delete</a>

</tr>

@endforeach

</table>

</body>

</html>

**Crud/routes/web.php:**

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\PostController;

/\*

|--------------------------------------------------------------------------

| Web Routes

|--------------------------------------------------------------------------

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('/', function () {

return view('welcome');

});

Route::get('/addpost',[PostController::class,'addPost'])->name('post.add');

Route::post('/addpost',[PostController::class,'savePost'])->name('save.post');

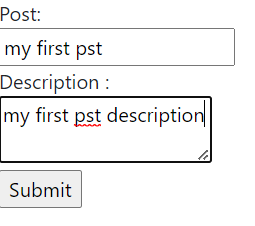
Route::get('/postlist',[PostController::class,'postList'])->name('post.list');

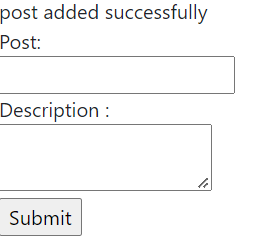
Route::get('/editpost/{id}',[PostController::class,'editPost'])->name('post.edit');

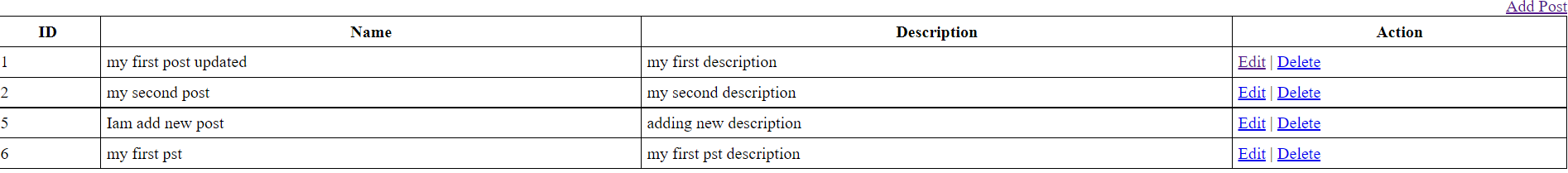
Route::get('/deletepost/{id}',[PostController::class,'deletePost'])->name('post.delete');

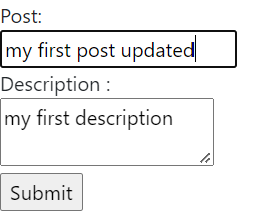
Route::post('/updatepost',[PostController::class,'updatePost'])->name('update.post');

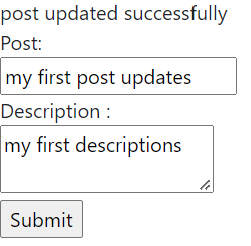
**Output**

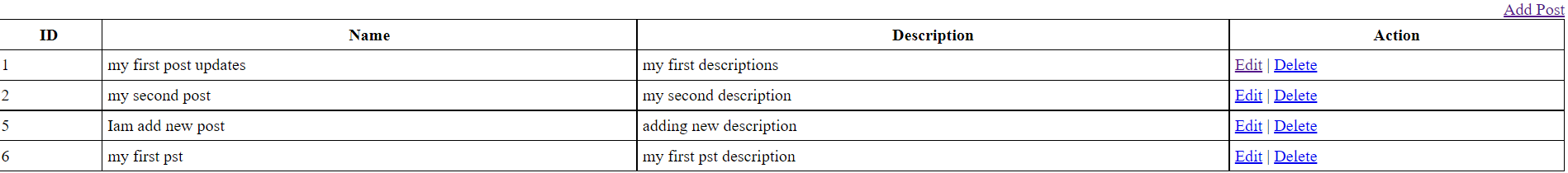


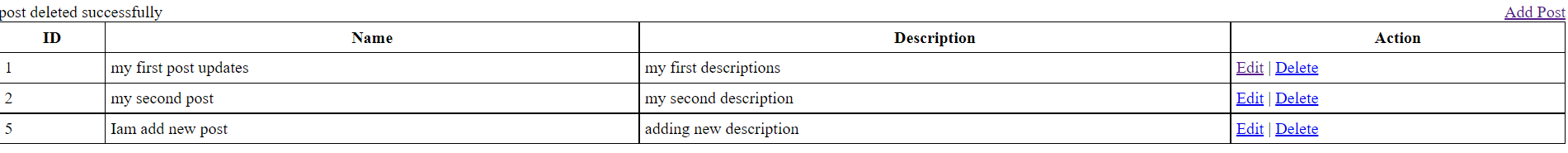












**UPLOAD IMAGE**

**Experiment No:3 Date:**

**AIM**

Develop a webpage to upload & view image into & from the database.

**PROCEDURE**

**Step 1:** Composer create-project --prefer-dist laravel/laravel imgupload

**Step 2:**Create a controller for imageupload,php artisan make:controller ImageController

**Step 3:**Create a routes for image-upload in web.php

**Step 4:**Create imageupload.blade.php file in views

**Step 5:**Then run,php artisan serve

**SOURCE CODE**

**imgupload/app/Controllers/ImageController**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ImageController extends Controller

{

public function index(){

return view('imageupload');

}

public function store(Request $request){

$request->validate([

'image'=>'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',

]);

$imageName=time();

$request->image->extension();

$request->image->move(public\_path('images'),$imageName);

return back()->with('success','You have successfully upload image.')- >with('image',$imageName);

}

}

**imgupload/routes/web.php:**

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\ImageController;

/\*

|--------------------------------------------------------------------------

| Web Routes

|--------------------------------------------------------------------------

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('/', function () {

return view('welcome');

});

Route::controller(ImageController::class)->group(function()

{

Route::get('image-upload','index');

Route::post('image-upload','store')->name('image.store');

});

**imgupload/resources/views/imageUpload.blade.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>Image Upload</title>

</head>

<body bgcolor="cyan">

<div class="container">

<div class="panel panel-primary">

<div class="panel-heading">

<h2>Image Upload</h2>

</div>

<div class="panel-boby">

@if ($message = Session::get('success'))

<div class="alert alert-success alert-block">

<button type="button" class="close" data-dismiss="alert">×</button>

<strong>{{ $message }}</strong>

</div>

<img src="images/{{ Session::get('image') }}">

@endif

<form action="{{ route('image.store') }}" method="POST" enctype="multipart/form-data">

@csrf

<div class="mb-3">

<label class="form-label" for="inputImage">Image:</label>

<input

type="file"

name="image"

id="inputImage"

class="form-control @error('image') is-invalid @enderror">

@error('image')

<span class="text-danger">{{ $message }}</span>

@enderror

</div>

<div class="mb-3">

<button type="submit" class="btn btn-success">Upload</button>

</div>

</form>

</div>

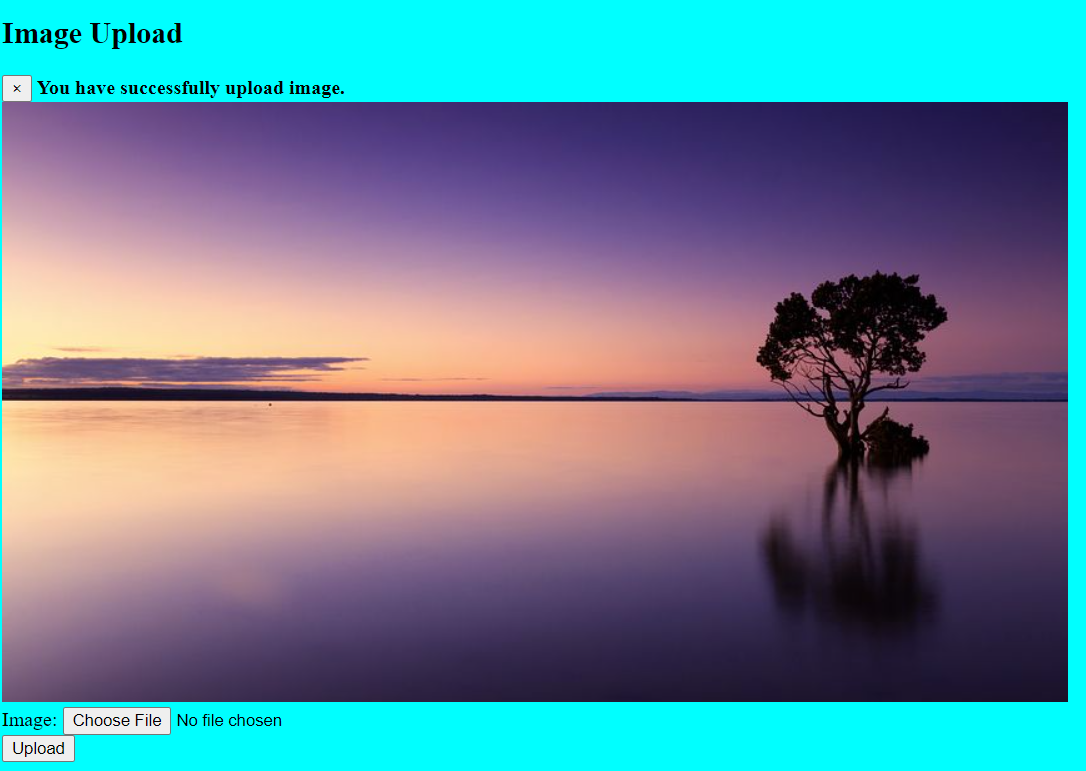
</div>

</div>

</body>

</html>

**Output**



**LOGIN & LOGOUT**

**Experiment No:4 Date:**

**AIM**

Create a webpage to learn account login & logout.

**PROCEDURE**

**Step 1:** Composer create-project --prefer-dist laravel/laravel mylaravel-app

**Step 2:** Create a database in phpmyadmin , database name phplaravel and give this name in the .env file of mylaravel-app

**Step 3:**Create the model User & it will create a create\_users\_table in the migrations

**Step 4:**Create a controller for login,registration,login-users,registration-users, dashboard,logout, php artisan make:controller CustomAuthController

**Step 5:**Create a routes for login,registration,login-users,registration-users,dashboard,logout, in web.php

**Step 6:**Create blade for dashboard.blade.php file in views and login.blade.php,registration.blade.php files in auth folder in views

**Step 7:**Then run,php artisan serve

**SOURCE CODE**

**mylaravel-app/.env**

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=phplaravel

DB\_USERNAME=root

DB\_PASSWORD=

**mylaravel-app/database/migrations/create\_users\_table** :

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('users', function (Blueprint $table) {

$table->id();

$table->string('name');

$table->string('email')->unique();

$table->string('password');

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('users');

}

};

**mylaravel-app/app/Controllers/CustomAuthController:**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\User;

use Hash;

use Session;

class CustomAuthController extends Controller

{

public function login(){

return view("auth.login");

}

public function registration(){

return view("auth.registration");

}

public function registerUser(Request $request){

$request->validate([

'name'=>'required',

'email'=>'required|email|unique:users',

'password'=>'required|min:5|max:12'

]);

$user= new User();

$user->name =$request->name;

$user->email =$request->email;

$user->password = Hash::make($request->password);

$res=$user->save();

if($res){

return back()->with('success','You have registered successfully');

}else{

return back()->with('fail','Something wrong');

}

}

public function loginUser(Request $request){

$request->validate([

'email'=>'required|email',

'password'=>'required|min:5|max:12'

]);

$user=User::where('email','=',$request->email)->first();

if($user){

if (Hash::check($request->password,$user->password)){

$request->session()->put('loginId',$user->id);

return redirect('dashboard');

}else{

return back()->with('fail','Password not matches.');

}

}else{

return back()->with('fail','This email is not registered.');

}

}

public function dashboard(){

$data=array();

if (Session::has('loginId')){

$data = User::where('id','=', Session::get('loginId'))->first();

}

return view('dashboard', compact('data'));

}

public function logout(){

if (Session::has('loginId')){

Session::pull('loginId');

return redirect('login');

}

}

}

**mylaravel-app/routes/web.php:**

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\CustomAuthController;

/\*

|--------------------------------------------------------------------------

| Web Routes

|--------------------------------------------------------------------------

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('/', function () {

return view('welcome');

});

Route::get('/login',[CustomAuthController::class,'login'])->middleware('alreadyLoggedIn');

Route::get('/registration',[CustomAuthController::class,'registration'])->middleware('alreadyLoggedIn');

Route::post('/register-user',[CustomAuthController::class,'registerUser'])->name('register-user');

Route::post('/login-user',[CustomAuthController::class,'loginUser'])->name('login-user');

Route::get('/dashboard',[CustomAuthController::class,'dashboard'])->middleware('isLoggedIn');

Route::get('/logout',[CustomAuthController::class,'logout']);

**mylaravel-app/resources/views/dashboard.blade.php:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>CustomAuthController</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-4 col-md-offset-4" >

<h4>Welcome to dashboard</h4>

<hr>

<table class="table">

<thead>

<th>Name</th>

<th>Email</th>

<th>Action</th>

</thead>

<tbody>

<tr>

<td>{{$data->name}}</td>

<td>{{$data->email}}</td>

<td><a href="logout">Logout</a></td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

</body>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

</html>

**mylaravel-app/resources/views/auth/registration.blade.php:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>CustomAuthController</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-4 col-md-offset-4" >

<h4>REGISTRATION</h4>

<hr>

<form action="{{route('register-user')}}" method="post">

@if(Session::has('success'))

<div class="alert alert-success">{{Session::get('success')}}</div>

@endif

@if(Session::has('fail'))

<div class="alert alert-fail">{{Session::get('fail')}}</div>

@endif

@csrf

<div class="form-group">

<label for="name">Full name</label>

<input type="text" class="form-control" placeholder="Enter Full Name" name="name" value="{{old('name')}}">

<span class="text-danger">@error('name') {{$message}} @enderror</span>

</div>

<div class="form-group">

<label for="email">Email</label>

<input type="email" class="form-control" placeholder="Enter Email" name="email" value="{{old('email')}}">

<span class="text-danger">@error('email') {{$message}} @enderror</span>

</div>

<div class="form-group">

<label for="password">Password</label></label>

<input type="password" class="form-control" placeholder="Enter Password" name="password" value="{{old('password')}}">

<span class="text-danger">@error('password') {{$message}} @enderror</span>

</div>

<div class="form-group">

<button class="btn btn-block btn-primary" type="submit">Register</button>

</div>

<a href="login">Already Registered !! Login Here.</a>

</form>

</div>

</div>

</div>

</body>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

</html>

**mylaravel-app/resources/views/auth/login.blade.php:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>CustomAuthController</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-4 col-md-offset-4" >

<h4>LOGIN</h4>

<hr>

<form action="{{route('login-user')}}" method="post">

@if(Session::has('success'))

<div class="alert alert-success">{{Session::get('success')}}</div>

@endif

@if(Session::has('fail'))

<div class="alert alert-fail">{{Session::get('fail')}}</div>

@endif

@csrf

<div class="form-group">

<label for="email">Email</label>

<input type="email" class="form-control" placeholder="Enter Email" name="email" value="{{old('email')}}">

<span class="text-danger">@error('email') {{$message}} @enderror</span>

</div>

<div class="form-group">

<label for="password">Password</label></label>

<input type="password" class="form-control" placeholder="Enter Password" name="password" value="{{old('password')}}">

<span class="text-danger">@error('password') {{$message}} @enderror</span>

</div>

<div class="form-group">

<button class="btn btn-block btn-primary" type="submit">Login</button>

</div>

<br>

<a href="registration">New User!! Register Here.</a>

</form>

</div>

</div>

</div>

</body>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

</html>

**mylaravel-app/app/Http/Middleware/AlreadyLoggedIn.php:**

<?php

namespace App\Http\Middleware;

use Closure;

use Illuminate\Http\Request;

class AlreadyLoggedIn

{

/\*\*

\* Handle an incoming request.

\*

\* @param \Illuminate\Http\Request $request

\* @param \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse) $next

\* @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse

\*/

public function handle(Request $request, Closure $next)

{

if(Session()->has('loginId') && (url('login')==$request->url() || url('registration')==$request->url())){

return back();

}

return $next($request);

}

}

**mylaravel-app/app/Http/Middleware/AuthCheck.php:**

<?php

namespace App\Http\Middleware;

use Closure;

use Illuminate\Http\Request;

class AuthCheck

{

/\*\*

\* Handle an incoming request.

\*

\* @param \Illuminate\Http\Request $request

\* @param \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse) $next

\* @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse

\*/

public function handle(Request $request, Closure $next)

{

if(!Session()->has('loginId')){

return redirect('login')->with('fail','You have to login first.');

}

return $next($request);

}

}

**mylaravel-app/app/Http/Kernel.php:**

<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel

{

/\*\*

\* The application's global HTTP middleware stack.

\*

\* These middleware are run during every request to your application.

\*

\* @var array<int, class-string|string>

\*/

protected $middleware = [

// \App\Http\Middleware\TrustHosts::class,

\App\Http\Middleware\TrustProxies::class,

\Illuminate\Http\Middleware\HandleCors::class,

\App\Http\Middleware\PreventRequestsDuringMaintenance::class,

\Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,

\App\Http\Middleware\TrimStrings::class,

\Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,

];

/\*\*

\* The application's route middleware groups.

\*

\* @var array<string, array<int, class-string|string>>

\*/

protected $middlewareGroups = [

'web' => [

\App\Http\Middleware\EncryptCookies::class,

\Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,

\Illuminate\Session\Middleware\StartSession::class,

\Illuminate\View\Middleware\ShareErrorsFromSession::class,

\App\Http\Middleware\VerifyCsrfToken::class,

\Illuminate\Routing\Middleware\SubstituteBindings::class,

],

'api' => [

// \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,

'throttle:api',

\Illuminate\Routing\Middleware\SubstituteBindings::class,

],

];

/\*\*

\* The application's route middleware.

\*

\* These middleware may be assigned to groups or used individually.

\*

\* @var array<string, class-string|string>

\*/

protected $routeMiddleware = [

'alreadyLoggedIn'=> \App\Http\Middleware\AlreadyLoggedIn::class,

'isLoggedIn' => \App\Http\Middleware\AuthCheck::class,

'auth' => \App\Http\Middleware\Authenticate::class,

'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,

'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,

'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,

'can' => \Illuminate\Auth\Middleware\Authorize::class,

'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,

'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,

'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,

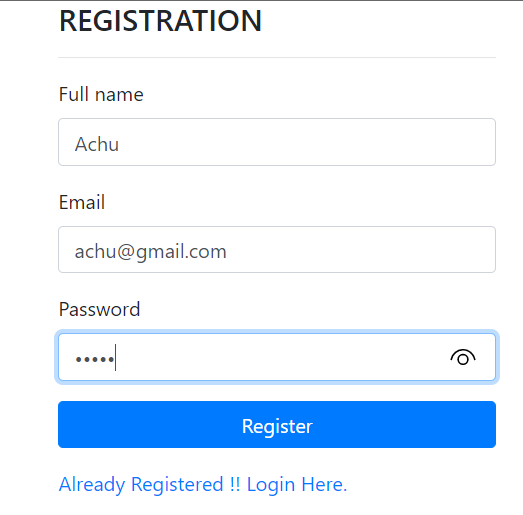
'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,

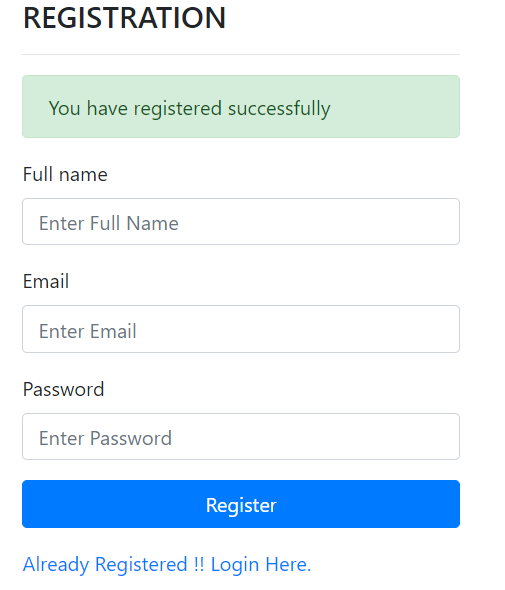
'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,

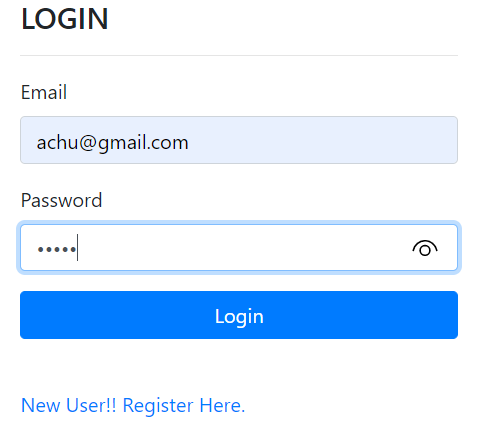
];

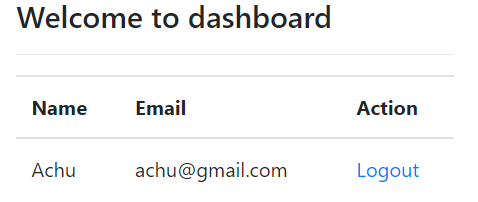
}

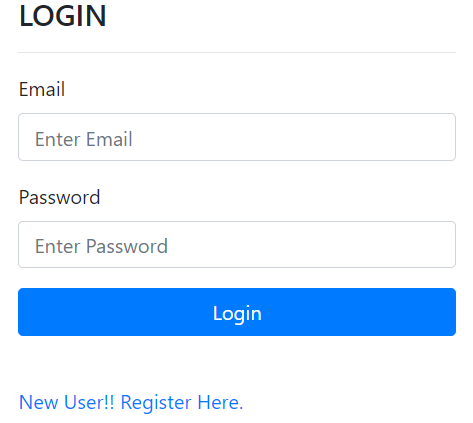
**Output**











**SESSIONS**

**Experiment No:5 Date:**

**AIM**

Create a sessions to access,store,delete session data.

**PROCEDURE**

**Step 1:** Composer create-project --prefer-dist laravel/laravel session

**Step 2:**Create a controller called SessionController,php artisan make:controller SessionController --plain

**Step 3:**Create a route for sessions in web.php

**Step 4:**Then run,php artisan serve

**SOURCE CODE**

**session/app/Http/Controllers/SessionController:**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests;

use App\Http\Controllers\Controller;

class SessionController extends Controller {

public function accessSessionData(Request $request) {

if($request->session()->has('my\_name'))

echo $request->session()->get('my\_name');

else

echo 'No data in the session';

}

public function storeSessionData(Request $request) {

$request->session()->put('my\_name','Virat Gandhi');

echo "Data has been added to session";

}

public function deleteSessionData(Request $request) {

$request->session()->forget('my\_name');

echo "Data has been removed from session.";

}

}

**session/app/routes/web.php:**

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\SessionController;

/\*

|--------------------------------------------------------------------------

| Web Routes

|--------------------------------------------------------------------------

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('/', function () {

return view('welcome');

});

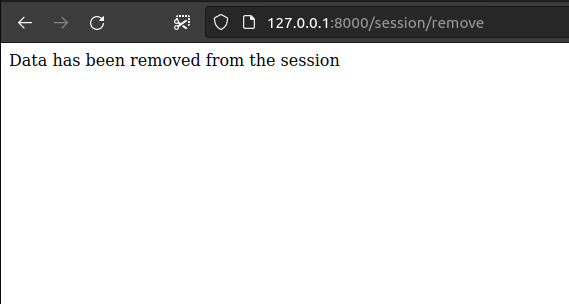
Route::get('session/get','SessionController@accessSessionData');

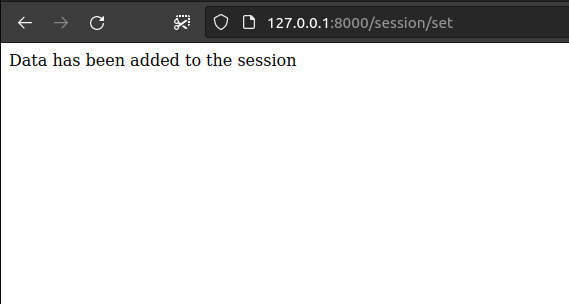
Route::get('session/set','SessionController@storeSessionData');

Route::get('session/remove','SessionController@deleteSessionData');

**Output**

**Session/set Session/remove**

****



**Session/get Session/get**

