

SOP
&
Lab Manual
for
PHP PROGRAMMING

DEPARTMENT OF COMPUTER SCIENCE (AIDED)

Supported by

DBT Star College Scheme



Published by

KONGUNADU ARTS AND SCIENCE COLLEGE (Autonomous)

Re-accredited by NAAC with A⁺ Grade – 4th Cycle
52nd Rank among Colleges in NIRF 2024
College of Excellence (UGC)
Coimbatore -641 029.
2024-2025

STANDARD OPERATING PROCEDURE

FOR

PHP PROGRAMMING

PREFACE

Welcome to the php Programming Lab Manual! This manual is designed to accompany your journey into the world of php programming, providing practical guidance, exercises, and insights to reinforce your learning experience.

php is a powerful and versatile programming language widely used in various domains, from building web applications to mobile apps and enterprise systems. This manual is crafted to help you grasp the fundamental concepts of php programming and develop your skills through hands-on practice. Each exercise and example included in this manual has been carefully designed to reinforce key concepts and encourage practical application. It is our hope that by actively engaging with the exercises and challenges presented here, you will not only grasp the fundamentals of php but also develop problem-solving skills essential for any program

Additionally, this manual is intended to complement your learning experience in the lab sessions, providing you with a structured resource that you can refer to at any stage of your Java programming journey. We encourage you to experiment, explore, and delve deeper into the world of php programming beyond the scope of this manual. We would like to thank DBT star college scheme for providing support in bringing out this Lab manual.

Standard Operating Procedure

Standard Operating Procedures (SOPs) in java programming refer to a set of guidelines and documented processes that outline the best practices, workflows, and steps involved in various aspects of developing java applications and some common components typically covered in SOPs for java programming

1. Requirements Gathering:

- Define the process for gathering and documenting client requirements for the mobile application.
- Specify methods for conducting interviews, surveys, or workshops to identify and prioritize features and functionalities.
- Outline the documentation format for capturing and validating requirements.

2. Design and Wireframing:

- Describe the process for creating the visual design and user interface (UI) of the mobile application.
- Explain how wireframing and prototyping tools are utilized to develop mockups and obtain feedback .
- Define guidelines for ensuring consistency in UI elements, navigation, and usability across different screens and devices.

3. Development Environment Setup:

- Specify the required development tools, software, and platforms for building the mobile application.
- Provide instructions for configuring integrated development environments (IDEs), emulators, and simulators for testing and debugging.
- Outline the process for setting up version control systems and collaborative tools for code management and team communication.

4. Coding Standards and Guidelines:

- Establish coding standards, naming conventions, and best practices for writing clean, maintainable, and efficient code.
- Define guidelines for organizing project files, folders, and code repositories.
- Specify documentation requirements, such as code comments or inline documentation.

5. Development Process:

- Outline the overall development process, such as Agile or waterfall methodologies, and how it applies to mobile app development.
- Define the roles and responsibilities of team members involved in the development process, including developers, testers, and project managers.

6. Testing and Quality Assurance:

- Specify the testing approaches and methodologies to be followed, including unit testing, integration testing, and user acceptance testing.
- Describe the process for identifying and reporting bugs or issues, including bug tracking tools and workflows.
- Outline the criteria for ensuring the quality and performance of the mobile application before release.

7. Deployment and Release:

- Define the process for packaging and deploying the mobile application to app stores (e.g., Apple App Store, Google Play Store).
- Specify the requirements and guidelines for creating app store listings, including app descriptions, screenshots, and promotional materials.
- Explain the process for obtaining necessary certificates, signing the app, and handling updates or version releases.

8. Maintenance and Support:

- Provide guidelines for post-release maintenance and support, including bug fixes, performance optimizations, and feature enhancements.
- Outline the process for gathering user feedback, conducting user surveys, and incorporating user suggestions into future updates.
- Describe the mechanisms for monitoring and analyzing the app's performance, crash reports, and analytics data.

These guidelines can serve as a starting point, but it's recommended to adapt and customize them to align with your organization's unique requirements and practices in php programming.

Index

Ex. No	Topic	Page No
1.	Control and Function	9
2.	Multiplication table	12
3.	Message passing and Array	15
4.	String function and Array	19
5.	Age computation	23
6.	Parsing Fuction	26
7.	GCD of two numbers	29
8.	File system function	32
9.	Down a file from server	35
10.	Mark Statement	38
11.	Mail ID Creation	43
12.	College application form using MYSQL table	49
13.	Authentication webpage	54
14.	Views count in SESSION	62
15.	Display last visited date-time on webpage using COOKIE	65
16.	php script handle form submission and display the entered data	68
17.	Check Whether A Given Number Is Armstrong Or Not.	72
18.	Find Largest Values Of Two Numbers Using Nesting Of Function.	75

PHP PROGRAMMING LAB EXPERIMENTS

1.DEVELOP A PHP PROGRAMMING USING CONTROLS AND FUNCTION

AIM:

The aim is to develop a PHP Programming using controls and functions.

ALGORITHM:

Step 1: Begin the php coding in a new file from the text editor.

Step 2: Open the php tag and write coding to get from the html file using post method.

Step 3: Create a html file with two text boxes to get the numbers.

Step 4: Create a button named Add.

Step 5: When the button is being pressed after entering the data values the data is past to the php file.

Step 6: Open the browser and run the html file.

CODING:**HTML CODE**

```
<html>

<head>

<title>ADDING TWO NUMBERS</title>

</head>

<body bgcolor = "black" text="red">

<form method="post" action="add.php">
```

Enter A :

```
<input type="text" name="num1" size="10"><br>
```

Enter b :

```
<input type="text" name="num2" size="10"><br>
```

```
<input type="submit" value="ADD">
```

```
</form>
```

```
</body>
```

```
</html>
```

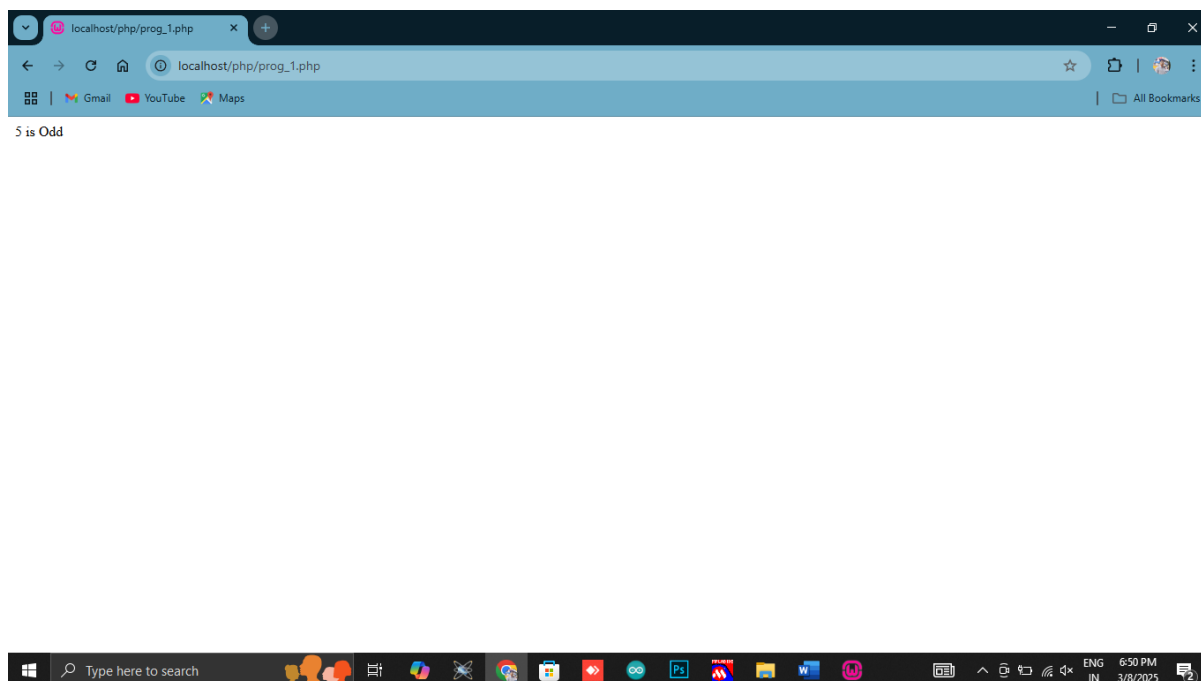
PHP CODE

```
<?php

echo$_POST['num 1']+$_POST['num 2'];

?>
```

OUTPUT:



2. DESIGN A SIMPLE WEBPAGE TO GENERATE MULTIPLICATION TABLE FOR GIVEN NUMBERS

AIM:

The aim is to design a simple webpage to generate multiplication table for given numbers.

ALGORITHM:

Step 1: Open a file in any text editor.

Step 2: Create a html file using post method and action on a php file.

Step 3: Save the file with .html extension in the wamp server.

Step 4: In the same way create a new file in a text editor.

Step 5: Open any browser and in the local host run the html file.

CODING:**HTML CODE**

```
<html>

<head>

<title>Multiplication Table</title>

</head>

<body bgcolor="black" text="white">

<form method="post" action="Gettable.php">

<strong>Enter No:</strong>

<input type="text" name="num" size="10">

<input type="submit" value="Get Table">

</form>

</body>

</html>
```

PHP CODE

```
<?php

    $num=$_POST['num'];

    for($i=1; $i<=10; $i++)

    {

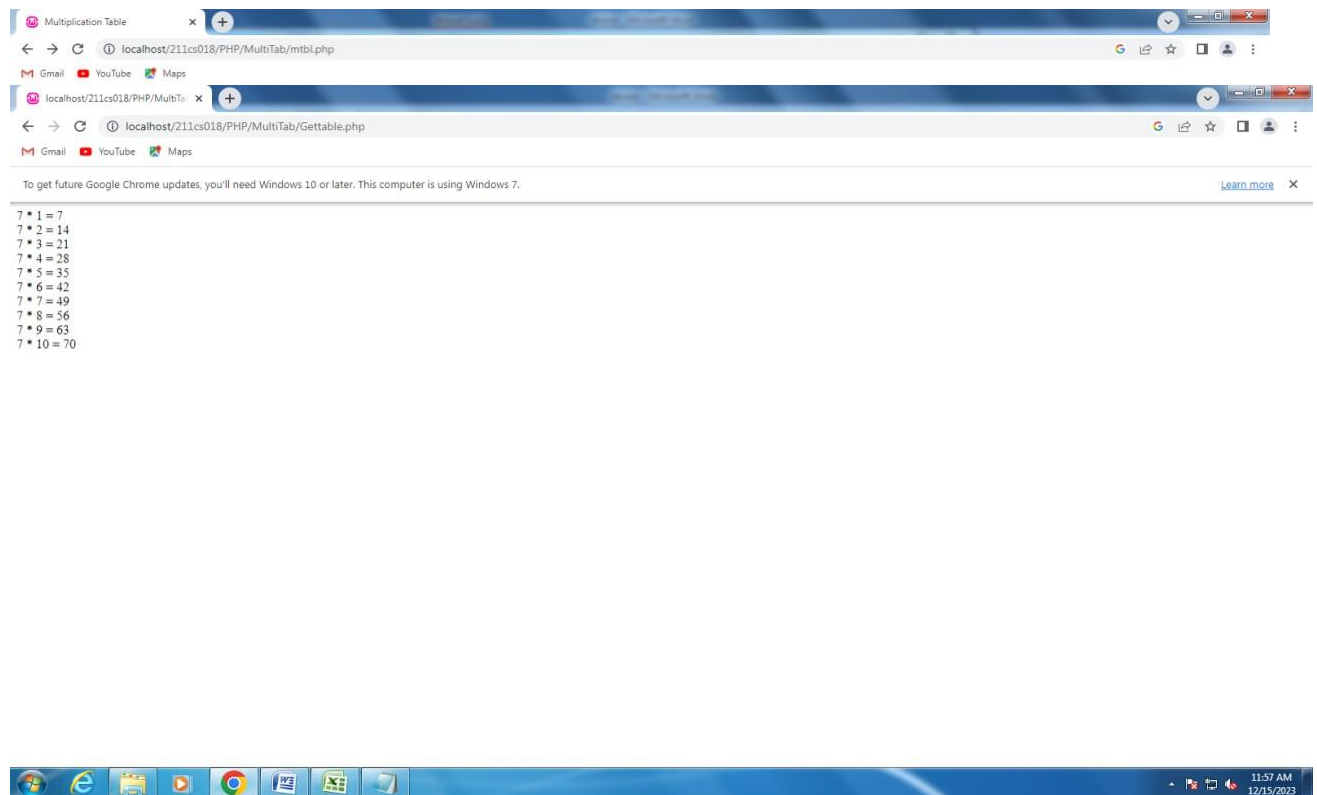
        $mul = $num * $i;

        echo "$num * $i = $mul<br>";

    }

?>
```

OUTPUT:



3. DEVELOP A PHP PROGRAM AND CHECK MESSAGE PASSING MECHANISM BETWEEN PAGES

AIM:

The aim is to develop a php program and check messages passing mechanism between pages .

ALGORITHM:

Step 1: Open a file in any text editor.

Step 2: Create a html file using get method and action on another php file.

Step 3: Save the file with . html extension in wamp server.

Step 4: Save the file in the wamp server with . php extension.

Step 5: Open any browser and in the local host run the html file.

Step 6: Program run successfully.

CODING:**HTML CODE**

```
<html>

<head>

<title>COLOR SAMPLER </title>

</head>

<body>

<h2> AN INTERACTIVE COLOR SAMPLER </h2>

<form method="get" action="display.php">

R:<input type="text" name="r" size="3"/.><p />

G:<input type="text" name="g" size="3"/><p />

B:<input type="submit" value="show me />

</form>

</body>

</html>
```

PHP CODE

```
<html>

<head>

<title> COLOR SAMPLER </TITLE>

</head>

<body>

<h2> interactive color sampler </h2>

<?php

$r=$_GET['r'];
```

```
$g=$_GET['g'];
```

```
$b=$_GET['b'];
```

```
$rgb=$r.','.$g.','.$b;
```

```
?>
```

```
R:<?php echo $r; ?>
```

```
G:<?php echo $d; ?>
```

```
B:<?php echo $b; ?>
```

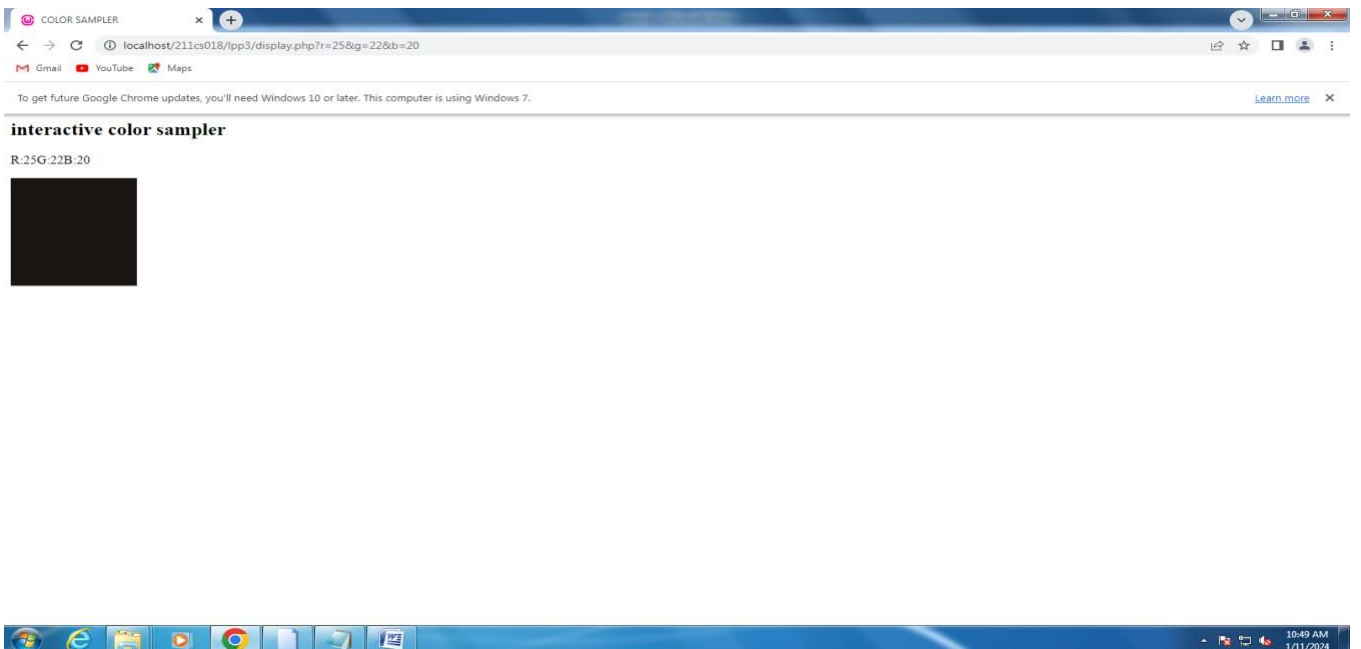
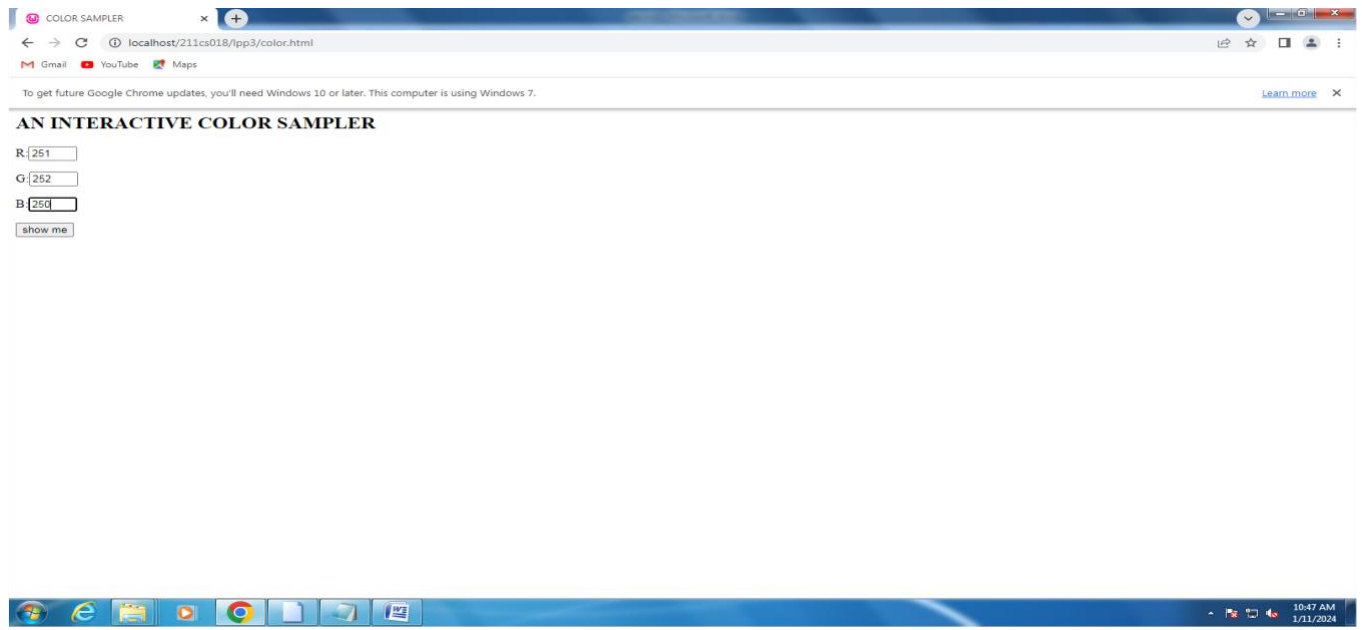
```
<p />
```

```
<div style="width :150px ; height : 150px ; background-color : rgb(<?php echo  
$rgb ; ?>)" />
```

```
</body>
```

```
</html>
```

OUTPUT:



4. DEVELOP A PHP PROGRAM USING STRING FUNCTIONS AND ARRAYS

AIM:

The aim is to develop a php program using string functions and arrays.

ALGORITHM:

Step 1: Open a file in any text editor.

Step 2: Create a text text box to accept a string from the server.

Step 3: Save the file with .html extension in the wamp server.

Step 4: Similarly create a php file in a text editor.

Step 5: Save the file with .php extension in the wamp server.

Step 6: Open any browser and run this .html file

Step 7: Program run successfully.

CODING:**HTML CODE**

```
<html>

<head>

<title>STRING FUNCTION</title>

</head>

<body bgcolor="black" text="white">

<form method="post" action="Disp.php">

<strong>Enter the String:</strong>

<input type="text" name="txt" size="10">

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

</form>

</body>

</html>
```

PHP CODE

```
<?php

$txt = $_POST['txt'];

echo 'String Lenth : '.strlen($txt).'\n';

echo 'String Reverse : '.strrev($txt).'\n';

echo 'To Upper : '.strtoupper($txt).'\n';

echo 'To Lower : '.strtolower($txt).'\n';

$array = array("Aakash", "Ravi", "Prashanth", "49", "50");

print_r(count($array));
```

```
print_r(Sort($array));  
print_r(rSort($array));
```

OUTPUT:



STRING AND ARRAY FUNCTION

The given string is	Good morning
String length	12
Reversing String	gninrom dooG
Repeating string two times	Good morningGood morning
Substring from 2 to 5	od mo
Trimming string	Good morning
Lower case	good morning
Upper case	GOOD MORNING

The given string is	Hello world. Its beautiful day
After explode function	Array ([0] => Hello [1] => world. [2] => Its [3] => beautiful [4] => day)
After implode function	hello world beautiful day
Range function	Generating array from 1 to 5Array ([0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5)
Maximum function	The given array isArray ([0] => 10 [1] => 20 [2] => 30 [3] => 40 [4] => 50) After performing maximum function50
Minimum function	The given array isArray ([0] => 100 [1] => 200 [2] => 300 [3] => 400 [4] => 500) After performing minimum function100
Slicing function	The given array isArray ([0] => 3 [1] => 6 [2] => 9 [3] => 12 [4] => 15) After SlicingArray ([0] => 6 [1] => 9 [2] => 12)



5. DESIGN A WEBPAGE THAT SHOULD COMPUTE ONE'S AGE ON A GIVEN DATA

AIM:

The aim is to design a webpage that should compute one's age on a given data.

ALGORITHM:

Step 1: Create a new file in a text editor.

Step 2: Using new data time function find the difference between a particular date and the current date.

Step 3: Then using echo statement point the answer in the screen respectively.

Step 4: Save the file with .php extension in the wamp server.

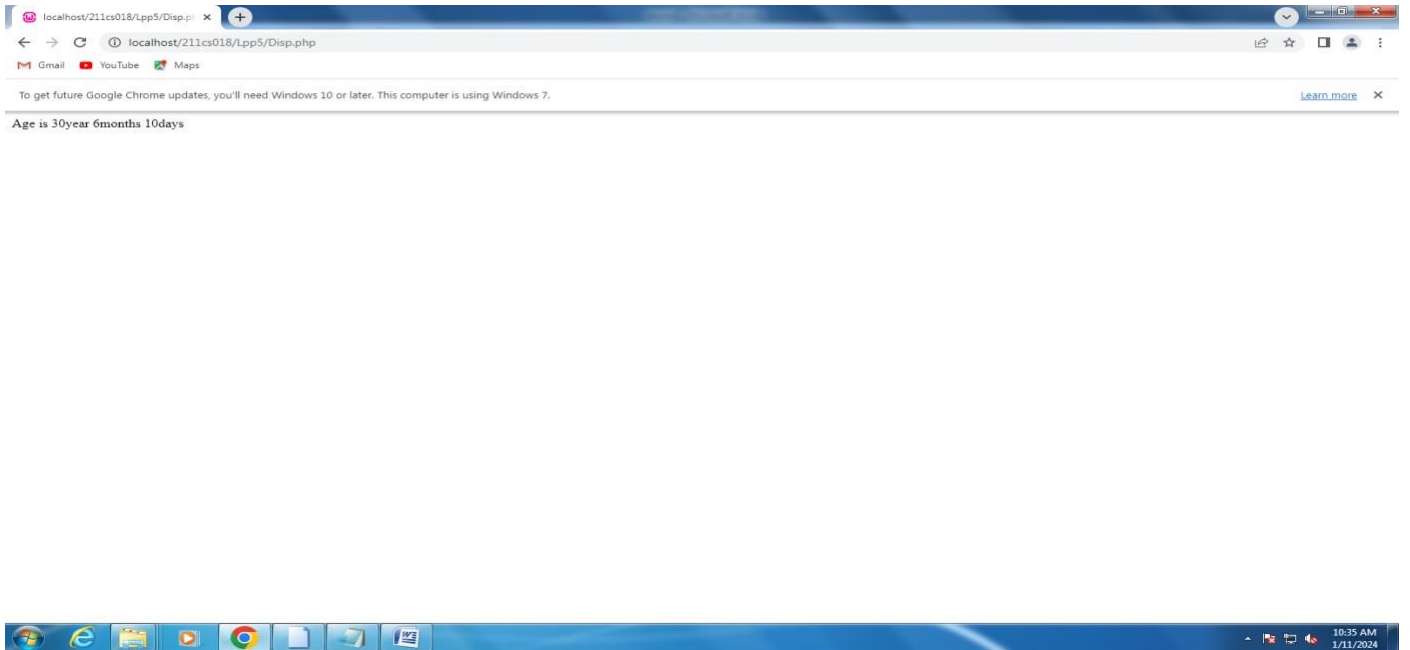
Step 5: Open any browser and run this file.

Step 6: Program run successfully.

CODING:

```
<?php $dob = new DateTime('1993-07-01');  
  
$today = new DateTime('today');  
  
$year = $dob->diff($today)->y;  
  
$month = $dob->diff($today)->m;  
  
$day = $dob->diff($today)->d;  
  
echo "Age is"." ".$year"." ",$month."months"." ".$day."days";  
  
?>
```

OUTPUT:



6. DEVELOP A PHP PROGRAM USING PARSING FUNCTIONS (USE TOKENIZING)

AIM:

The aim is to develop a php program using parsing function.

ALGORITHM:

Step 1: Open a file in any text editor.

Step 2: Create a html file using port method and action on same php file.

Step 3: Create text box to accept a string from the user.

Step 4: Save the file with .php extension.

Step 5: Use function for tokenizing.

Step 6: Enter the data in the text box and click submit.

Step 7: Program run successfully.

CODING:

```
<html>

<head>

<title>String Tokenizing </title>

</head>

<body bgcolor="yellow">

<form action="" " method="post">

Enter the String: </br>

<input type="text" name="t1" id="t1"><br>

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

</form>

<?php

    $t=strtok($_POST['t1'], "");

While($t!==false)

{

echo "$t<br>";

$t=strtok("");

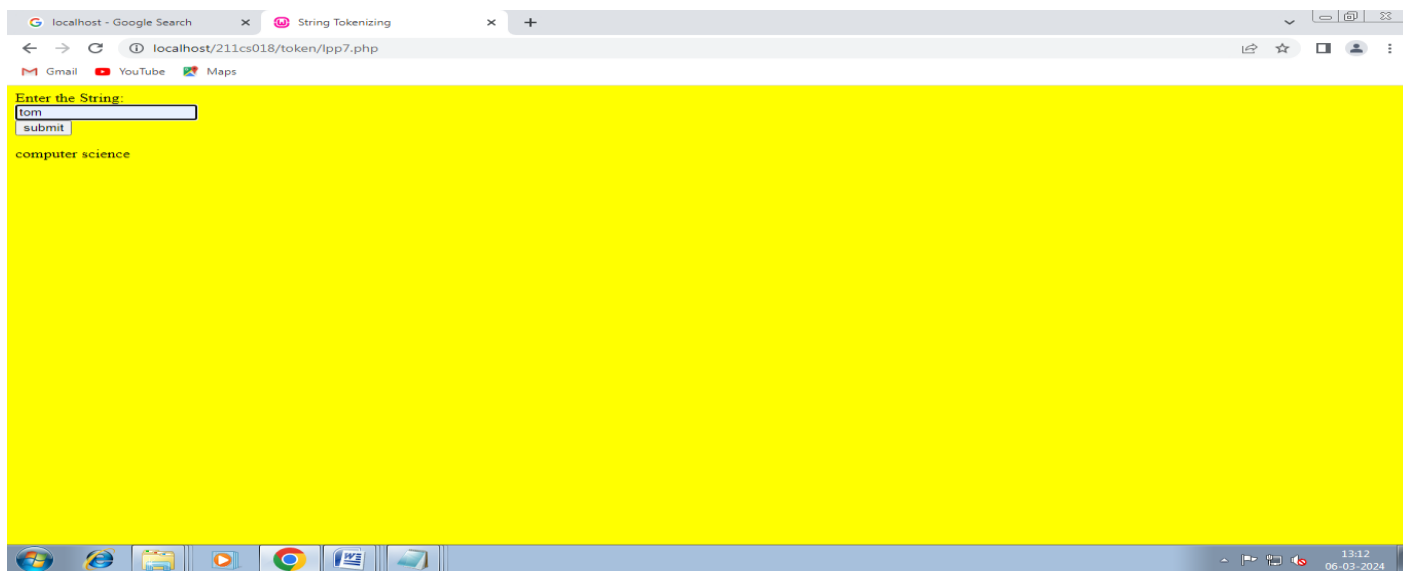
}

?>

</body>

</html>
```

OUTPUT:



7. DEVELOP A PHP PROGRAM TO FIND THE GCD OF TWO NUMBERS USING DEFINED FUNCTIONS

AIM:

The aim is to develop a php program to find the gcd of two numbers using defined functions.

ALGORITHM:

Step 1: Open a file in any text editor.

Step 2: Create a html file using post method and action on other php file.

Step 3: Create two text boxes to accept the values for finding the GCD.

Step 4: Save the file in wamp server with .html extension.

Step 5: Save the program with .php extension in the wamp server.

Step 6: Run the program to get the desired output.

CODING:**HTML CODE**

```
<html>

<head>

<title>GCD of a number</title>

</head>

<body bgcolor="black" text="white">

<form method="post" action="home.php">

<strong>Enter first No:</strong>

<input type="text" name="num1" size=10"><br>

<strong>Enter second No:</strong>

<input type="text" name="num2" size=10">

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

</form>

</body>

</html>
```

PHP CODE

```
<?php

$a = $_POST['num'];

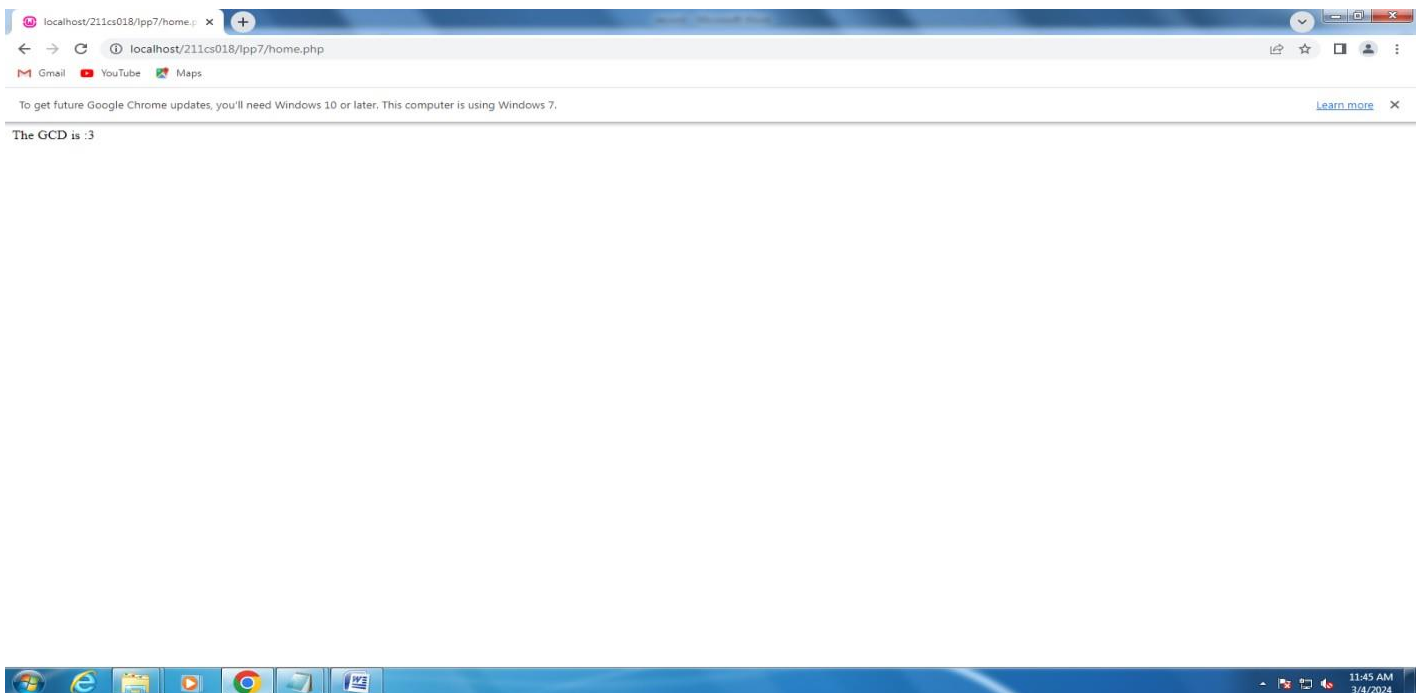
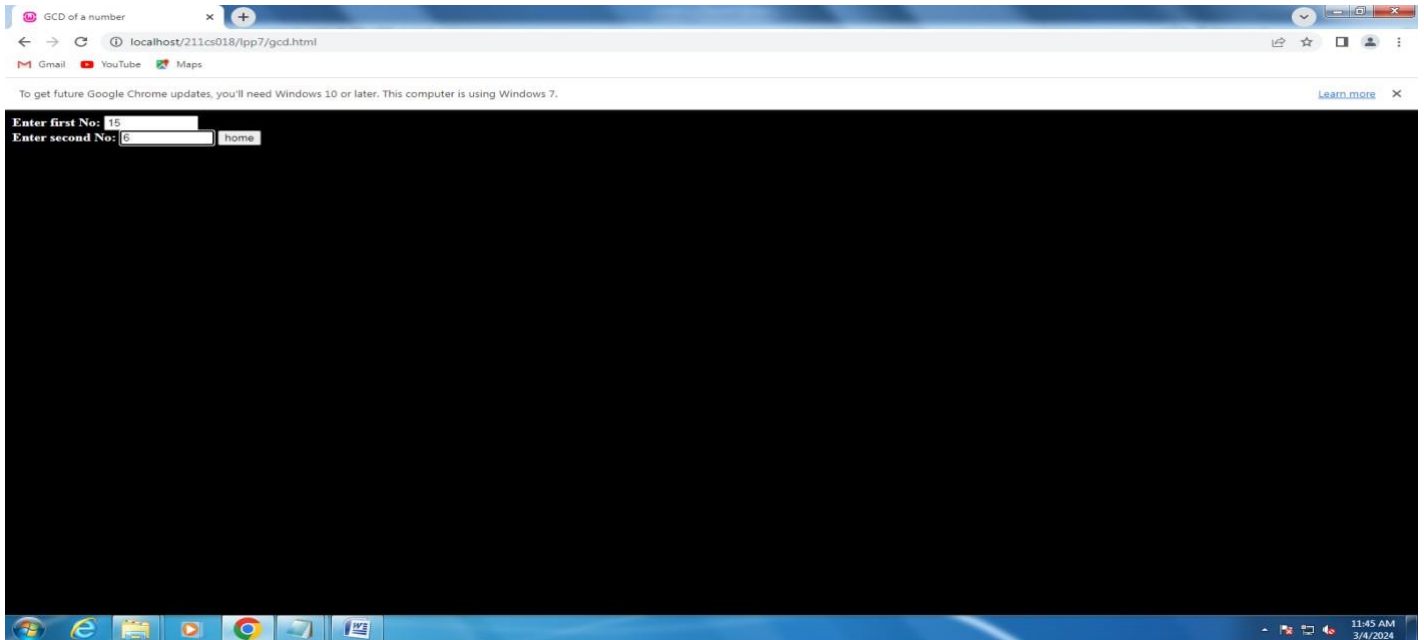
$b = $_POST['num2'];

$gcd=gmp_gcd($a,$b);

echo "The GCD is :". $gcd;

?>
```

OUTPUT:



8. DEVELOP A PHP PROGRAM TO CHECK FILE SYSTEM FUNCTIONS

AIM:

The aim is to develop a php program to check file system functions.

ALGORITHM:

Step 1: Open a new file in the text editor.

Step 2: Save the text file in the wamp server.

Step 3: Create a new file in the text editor.

Step 4: Open the php tag `<? php` .

Step 5: Save the file with .php extension in the wamp server.

Step 6: Open any browser and run the file.

Step 7: Program run successfully.

CODING:

```
<?php
```

```
$file = "my file.txt";
```

```
//opening a file in reading mode
```

```
$h = fopen($file,'r');
```

```
//reading file upto file length and storing it in a variable "fcontents"
```

```
$fContents = fread($h,filesize($file));
```

```
//closing file
```

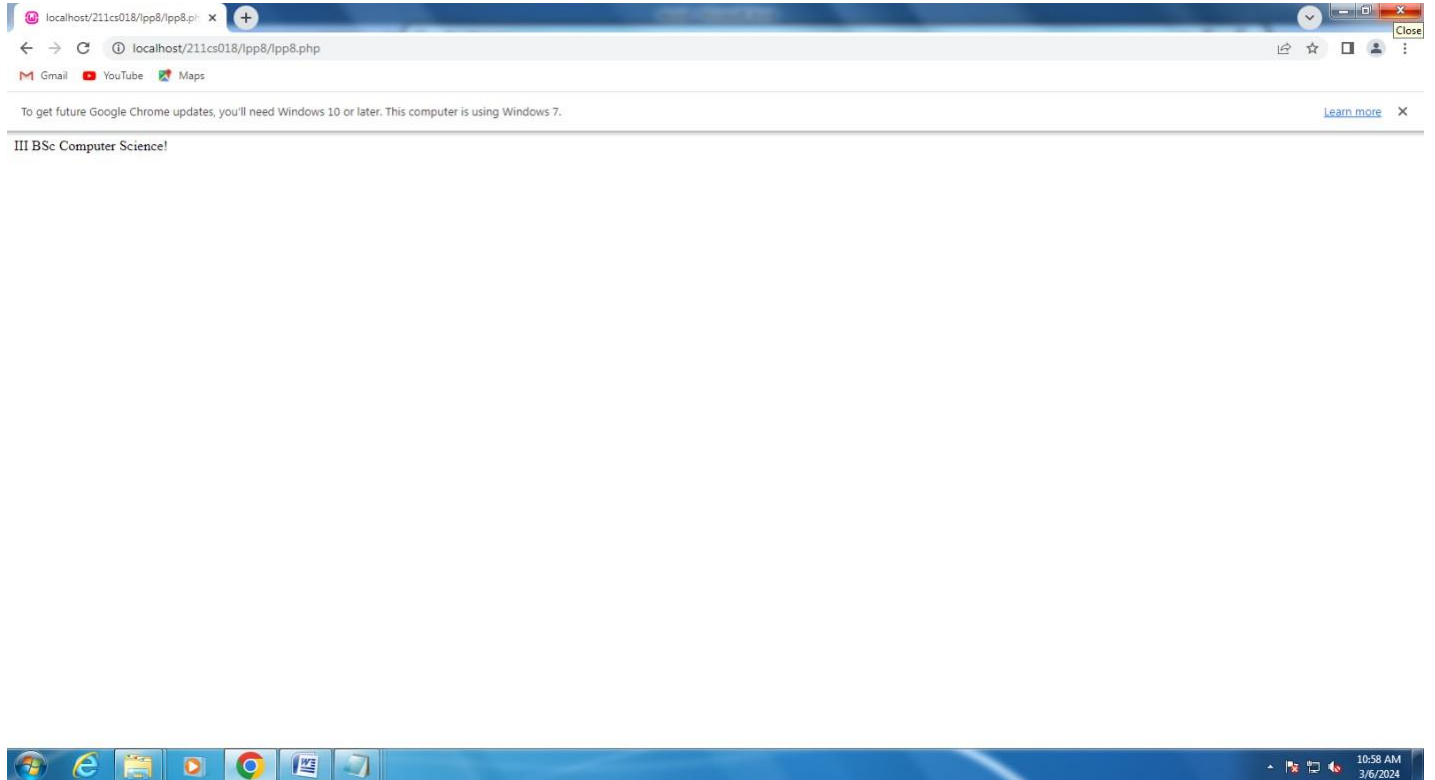
```
fclose($h);
```

```
//displaying the content
```

```
echo $fContents;
```

```
?>
```

OUTPUT:



9. WRITE A PROGRAM TO DOWNLOAD A FILE FROM THE SERVER

AIM:

The aim is to write a program to download a file from the server.

ALGORITHM:

Step 1: Open a new file in the text editor.

Step 2: Design a html file by adding the textboxes . tpn get the details from the users.

Step 3: Save the file with .html extension in the wamp server.

Step 4: Open a new file text editor.

Step 5: Begin the php coding by opening the tag <? php.

Step 6: Save the file .php extension in the wamp server.

Step 7: Program run successfully.

CODING:

```
<?php
```

```
    $URL = 'file:///c:/Users/211cs008/Desktop/ZA-wp1.jpg';
```

```
    $file = basename($URL);
```

```
$success = file_put_contents($file, file_get_contents($URL));
```

```
If ($success) {
```

```
    Echo "File downloaded successfully from the server ";
```

```
}
```

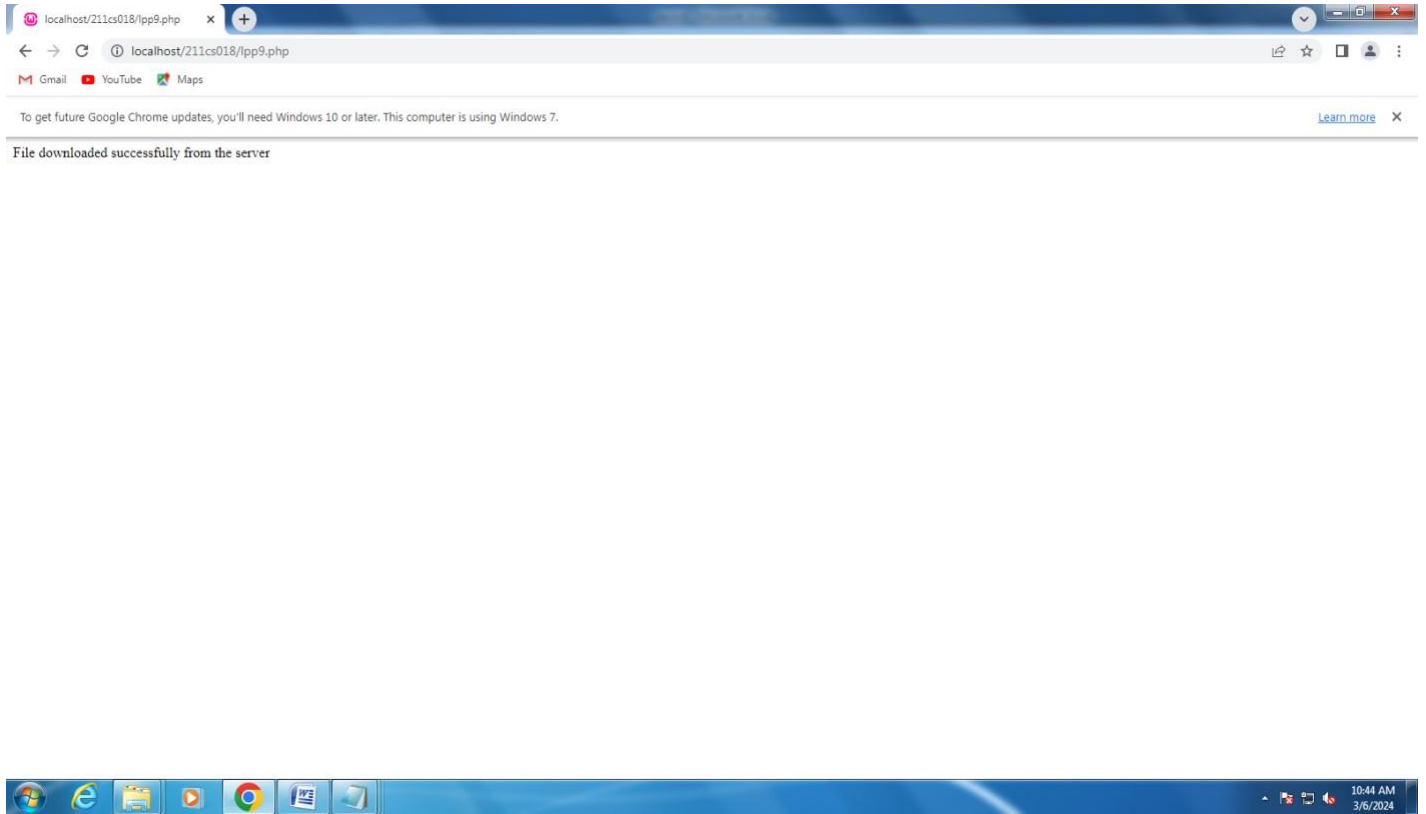
```
else {
```

```
    echo "file downloaded failed.";
```

```
}
```

```
?>
```

OUTPUT:



10. DEVELOP A PHP PROGRAM TO GENERATE MARK STATEMENT

AIM:

The aim is to develop a php program to generate mark statement

ALGORITHM:

Step 1: Open a new file in the text editor.

Step 2: Design an html file by adding the textboxes to get the details from the user.

Step 3: Save the file with html extension in the wamp server.

Step 4: Open a new file text editor.

Step 5: Begin the php coding by opening the tag <? Php.

Step 6: Open any browser and run the html file from the local host of wamp.

Step 7: Program run successfully.

CODING:**HTML CODE**

```
<html>

<body>

<h2>Mark Statement</h2>

<form action="back.php" method="post">

Name:<br>

<input type="text" name="name" required>

<br>

Roll No:<br>

<input type="text" name="rollno" required>

<br>

mark1:<br>

<input type="text" name="m1" required>

<br>

mark2:<br>

<input type="text" name="m2" required>

<br>

mark3:<br>

<input type="text" name="m3" required>

<br>

<br>

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

</form>
```

```
</body>
```

```
</html>
```

PHP CODE

```
<form method="POST" action="">

Name: <input type="text" name="txtname" /><br>

Regno: <input type="text" name="txtreg" /><br>

mark 1: <input type="text" name="txtm1" /><br>

mark 2: <input type="text" name="txtm2" /><br>

mark 3: <input type="text" name="txtm3" /><br>

<input type="submit" value="Get Result">

</form>

<?php
$sub_1 = $_POST['txtm 1'];
$sub_2 = $_POST['txtm 2'];
$sub_3 = $_POST['txtm 3'];
$total=$sub 1 + $sub 2 + $sub 3;
$average $total / 3.0;
$percentage ($total/300.0) 100;
if ($average >=90)
$grade = "A";
else {
if ($average >= 80 && $average <90)
$grade = "B";
else {
```

```
if ($average >= 70 && $average < 80)
$grade = "C";
else {
    if ($average >= 60 && $average < 70)
$grade = "D";
    Else
$grade = "E";
        }
    }
}

// It will print the final output
echo "The Total marks =" . $total. "/300\n";
echo "The Average marks = ". $average. "\n";
echo "The Percentage =" . $percentage. "%\n";
echo "The Grade =" . $grade. ""\n";

?>
```

OUTPUT:

localhost/211cs018/Marksheet/

← → ↻ localhost/211cs018/Marksheet/

Gmail YouTube Maps

To get future Google Chrome updates, you'll need Windows 10 or later. This computer is using Windows 7. [Learn more](#) ×

Mark Statement

Name:

Roll No:

mark1:

mark2:

mark3:

Windows taskbar: 11:27 AM 3/21/2024

localhost/211cs018/Marksheet/

← → ↻ localhost/211cs018/Marksheet/back.php

Gmail YouTube Maps

To get future Google Chrome updates, you'll need Windows 10 or later. This computer is using Windows 7. [Learn more](#) ×

Name:

Regno:

mark1:

mark2:

mark3:

The Total marks = 240/300 The Average marks = 80 The Percentage = 80% The Grade = 'B'

Windows taskbar: 11:31 AM 3/21/2024

11. DEVELOP A PHP PROGRAM FOR MAIL ID CREATION

AIM:

The aim is to develop a php program for mail id creation.

ALGORITHM:

Step 1: Open a new file in the text editor.

Step 2: Here create text boxes to get the values from the user for mail DD creation.

Step 3: Save the file with .html extension.

Step 4: Similarly open a text editor file.

Step 5: Save the file with .php extension in the wamp server.

Step 6: Now the program runs successfully.

CODING:**HTML CODE**

```
<html>

<form action="dis.php">

<table>

<caption>Email ID Creation</caption>

    <tr>

        <td>Name</td>

        <td><input type=text name= na size=30></td>

    </tr>

    <tr>

        <td>Father Name</td>

        <td><input type=text name =fn size=30></td>

    </tr>

    <tr>

        <td>Date of Birth</td>

        <td><input type=text name= dob size=30></td>

    </tr>

    <tr>

        <td>Mobile Number</td>

        <td><input type=text name=mno size=30></td>

    </tr>

    <tr>
```

```

        <td>Email ID</td>

        <td><input type=text name= email size=30></td>

    </tr>

    <tr>

        <td>Gender</td>

        <td><input type=text name=gen size=30></td>

    </tr>

    <tr>

        <td><input type=submit></td>

        <td><input type=reset></td>

    </tr>

</table>

</form>

</html>

```

PHP CODE

```

<html>

<?php

    $pna=$_GET['na'];

    $pfa=$_GET['fn'];

    $pdob=$_GET['dob'];


    $pmno=$_GET['mno'];

    $pemail=$_GET['email'];

    $pgen=$_GET['gen'];

```

?>

<center>

<table border=1>

<tr>

<td>

<table width=100%>

<tr>

<td>

Email Refistration

</td>

</tr>

</table>

</td>

</tr>

<tr>

<td>

<table border=1 width=100%>

</td>

</tr>

<tr> <td>Name</td> <td><?php echo "\$pna"; ?></td> </tr>

<tr> <td>Father Name</td> <td><?php echo "\$pfa"; ?></td> </tr>

<tr> <td>Date of Birth</td> <td><?php echo "\$pdob"; ?></td> </tr>

<tr>

<td>Mobile Number</td> <td><?php echo "\$pmno"; ?></td> </tr>

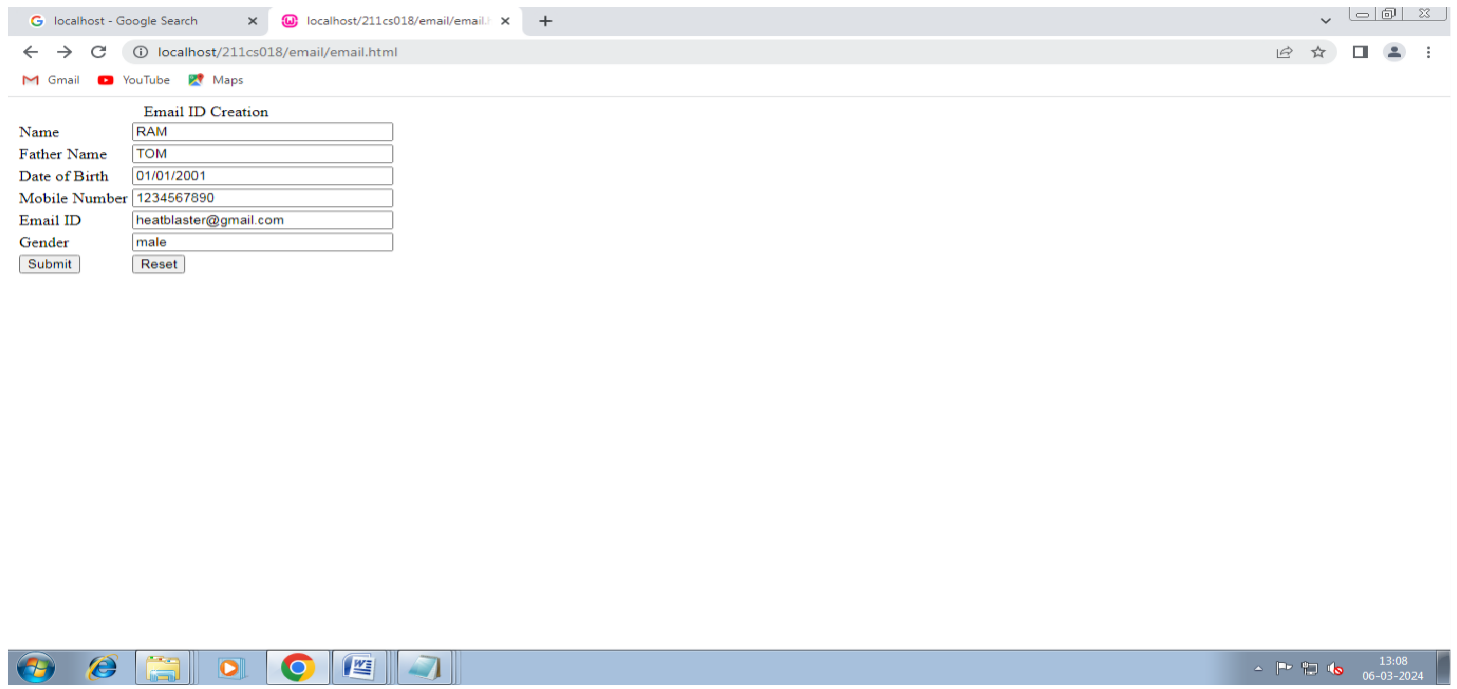
<tr> <td>Email ID</td> <td><?php echo "\$pemail"; ?></td> </tr>

<tr> <td>Gender</td> <td><?php echo "\$pgen"; ?></td> </tr>

</table>

```
</td>
</tr>
</table>
</center>
</html>
```

OUTPUT:



localhost - Google Search x localhost/211cs018/email/email.html x +

localhost/211cs018/email/email.html

Gmail YouTube Maps

Email ID Creation

Name

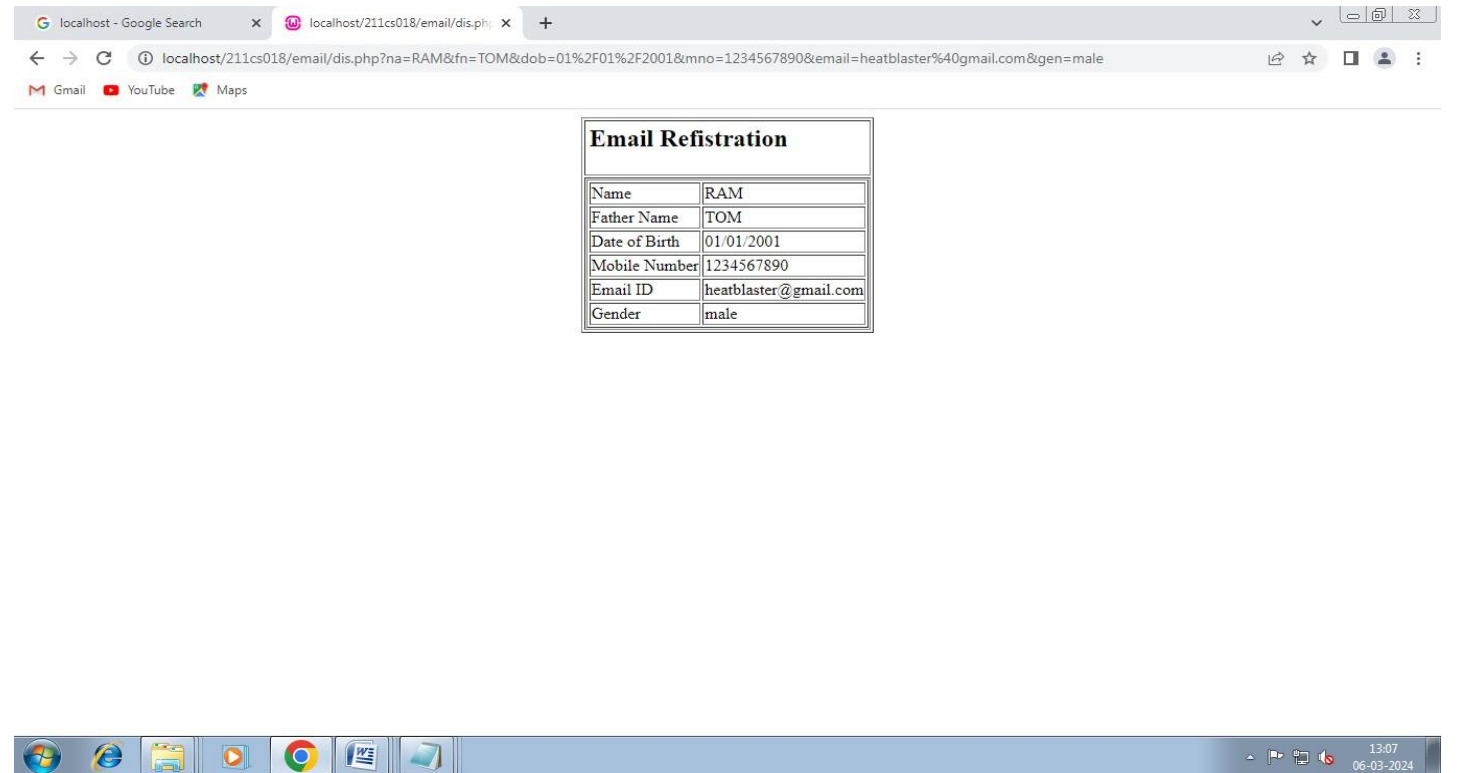
Father Name

Date of Birth

Mobile Number

Email ID

Gender



localhost - Google Search x localhost/211cs018/email/dis.php x +

localhost/211cs018/email/dis.php?na=RAM&fn=TOM&dob=01%2F01%2F2001&mno=1234567890&email=heatblaster%40gmail.com&gen=male

Gmail YouTube Maps

Email Refistration

Name	RAM
Father Name	TOM
Date of Birth	01/01/2001
Mobile Number	1234567890
Email ID	heatblaster@gmail.com
Gender	male

12. DEVELOP A PHP PROGRAM TO DESIGN A COLLEGE APPLICATION FORM USING MYSQL TABLE

AIM:

The aim is to develop a php program to design a college application form using MYSQL table.

ALGORITHM:

Step 1: Open a new file in text editor.

Step 2: Create a database in php myadmin and connect the file php file with it.

Step 3: Within the database create a table.

Step 4: The details of the student will be stored in that database.

Step 5: Program run successfully.

CODING:
HTML CODE:

```
<html>

<body>

<h2>Registration Form</h2>

<form action="backend.php" method="post">

Email:<br>

<input type="email" name="email" required>

<br>

Roll No:<br>

<input type="text" name="rollno" required>

<br>

Phone Number:<br>

<input type="tel" name="phone" required>

<br>

Department:<br>

<input type="text" name="department" required>

<br><br>

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

</form>

</body>

</html>
```

PHP CODE:

```
<?php
```

```
// Database connection

$servername = "localhost"; // Change this to your database server

$username = "root"; // Change this to your database username

$password = ""; // Change this to your database password

$dbname = "registration"; // Change this to your database name

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname).

// Check connection

if ($conn->connect_error) {

    die("Connection failed:" $conn->connect_error);

}

// Create table if not exists

$sql = "CREATE TABLE IF NOT EXISTS registration (

    id INT AUTO_ INCREMENT PRIMARY KEY,

    email VARCHAR(255) NOT NULL,

    rollno VARCHAR(50) NOT NULL,

    phone VARCHAR(15) NOT NULL,

    department VARCHAR(100) NOT NULL

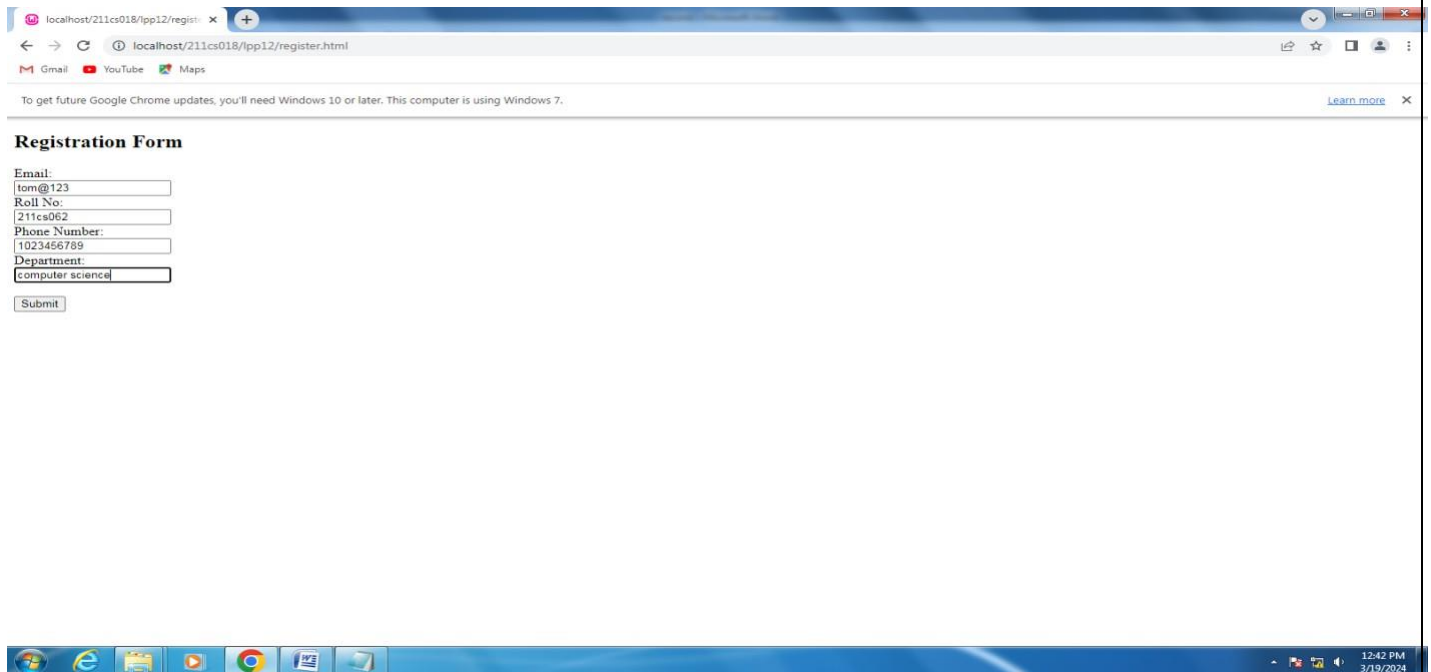
)";

if ($conn->query($sql) === TRUE) {

    echo "Table created successfully<br>";
```

```
} else {  
  
echo "Error creating table:" $conn error, "<br>";  
  
}  
  
//Insert data into database  
  
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
$email=$_POST['email'];  
  
$rolln=$_POST['rollno'];  
  
$phone=$_POST['phone'];  
  
$department = $_POST['department'];  
  
$sql = "INSERT INTO registration (email, rollno, phone, department)  
VALUES ('$email', '$rollno', '$phone', '$department')";  
  
if ($conn->query($sql) === TRUE) {  
echo "New record created successfully";  
} else {  
  
echo "Error: ". $sql. "<br>".$conn->error;  
  
}  
  
}  
  
$conn->close();  
  
?>
```

OUTPUT:



The screenshot displays a web browser window with the address bar showing 'localhost/211cs018/lpp12/register.html'. The page title is 'Registration Form'. The form contains the following fields and values:

Field	Value
Email:	tom@123
Roll No:	211cs062
Phone Number:	1023456789
Department:	computer science

A 'Submit' button is located at the bottom of the form. The browser's taskbar at the bottom shows the Windows logo, Internet Explorer, File Explorer, Google Chrome, and a document icon. The system clock indicates 12:42 PM on 3/19/2024.

13. DESIGN AN AUTHENTICATION WEB PAGE IN PHP WITH MYSQL TO CHECK USER NAME AND PASSWORD

AIM:

The aim is to design an authentication web page in php with MYSQL to check user name and password.

ALGORITHM:

Step 1: Open a new file in the text editor.

Step 2: Open php my admin and create a database with a table for storing the login information.

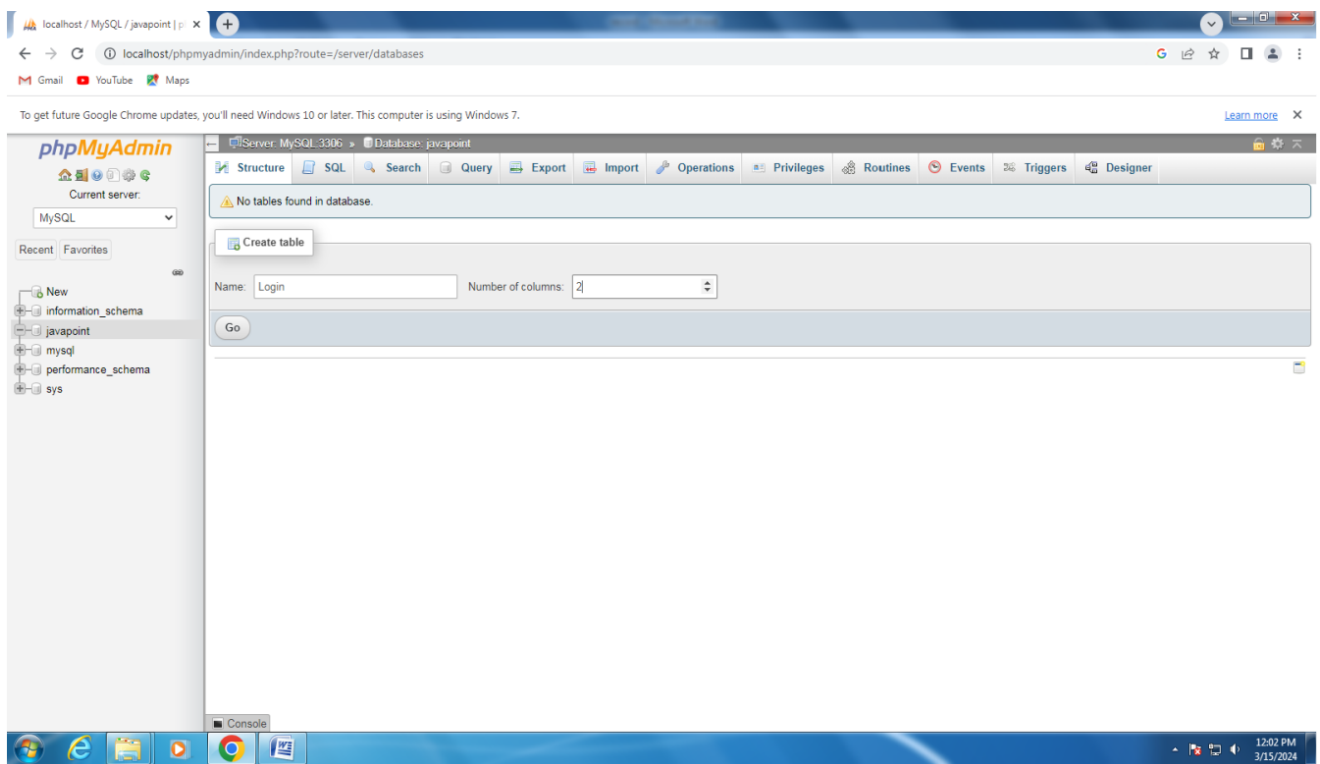
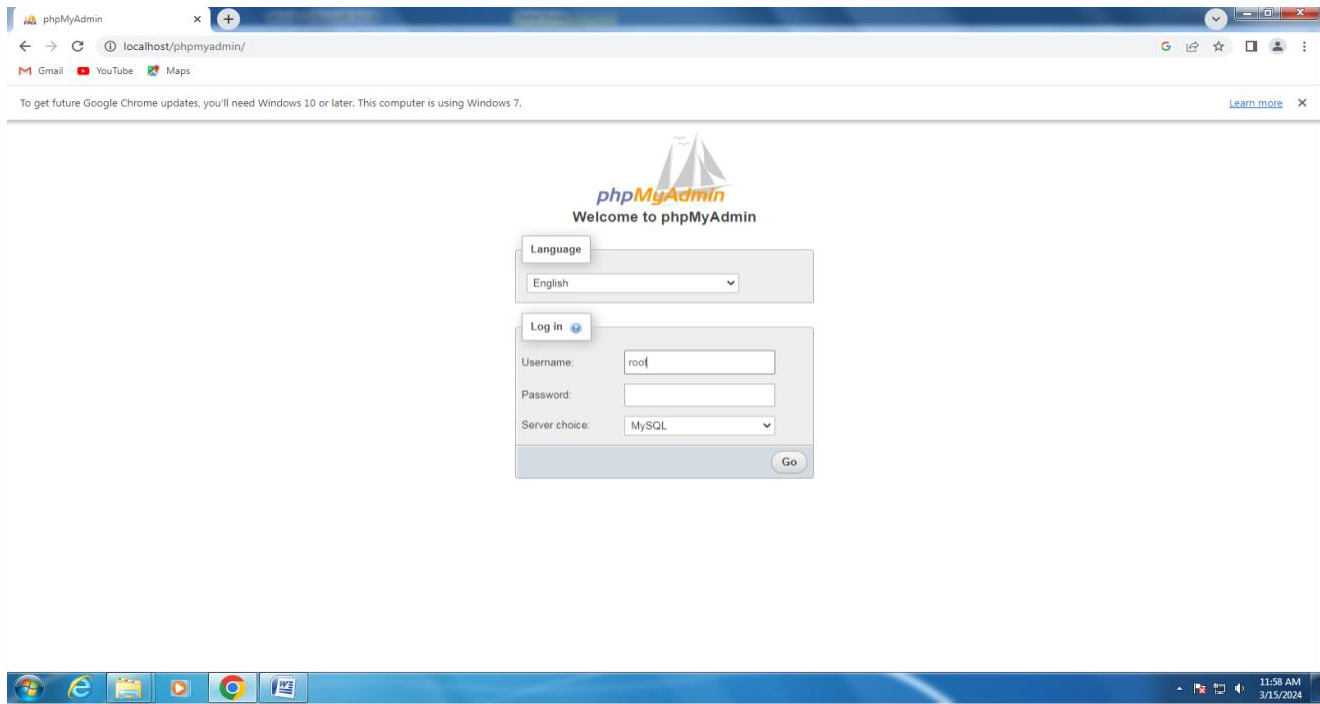
Step 3: Save this file in .php .

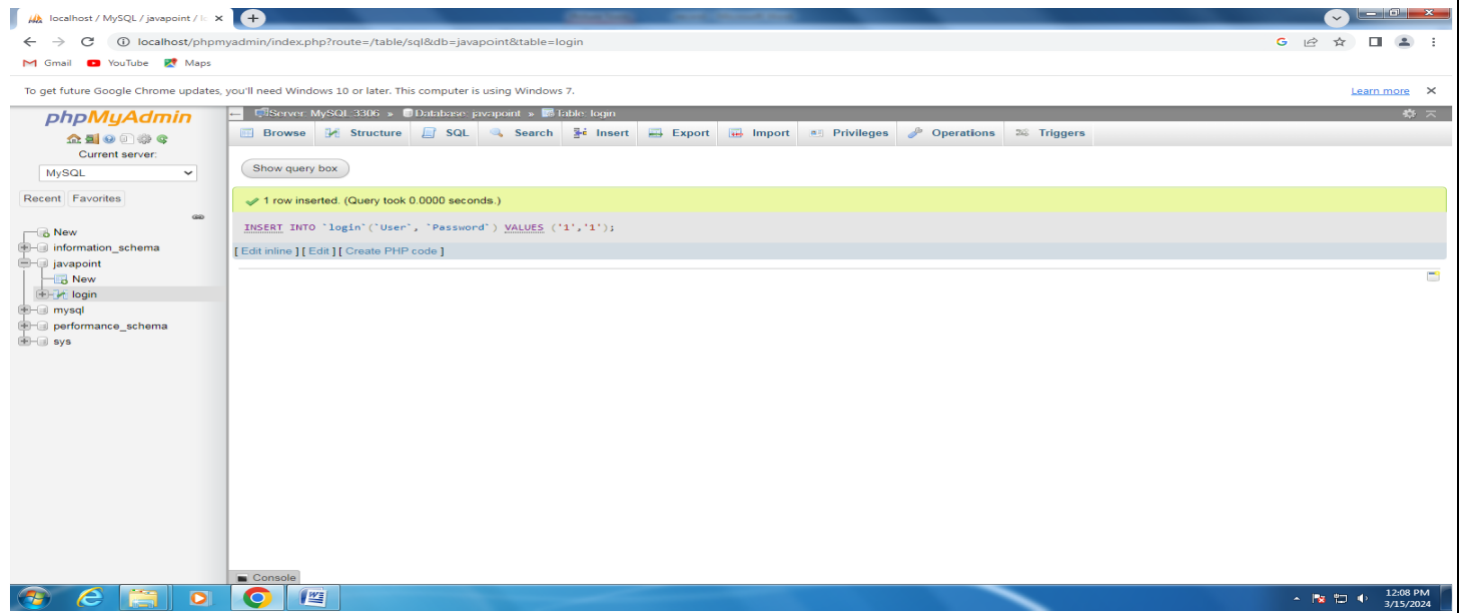
Step 4: Finally create a new file with html which acts as a front end.

Step 5: Save the above file in wamp server.

Step 6: Open any browser and run the html file.

DATABASE CREATION:





PHP CODE:**Connection page:**

```
?php

$host "localhost";

$user "root";

$password = "";

$db_name= "javatpoint";

$con = mysqli_connect($host, $user, $password, $db_name);

if(mysqli_connect_errno()) {

    die("Failed to connect with MySQL: ". mysqli_connect_error());

}

?>
```

Authentication page:

```
<?php

include('connection.php');

$username =$_POST['user'];

$password =$_POST['pass'];

//to prevent from mysqli injection

$username =stripcslashes($username);

$password =stripcslashes($password);

$username= mysqli_real_escape_string($con, $username);

$password= mysqli_real_escape_string($con, password);
```

```
$sql="select *from login where User= '$username' and Password =  
'$password':
```

```
$result =mysqli _query($con, $sql);
```

```
$row =mysqli_fetch_array($result, MYSQLI_ASSOC);
```

```
$count=mysqli_num_rows($result);
```

```
if($count = 1){
```

```
echo "<h1><center> Login successful </center></h1>";
```

```
}
```

```
else {
```

```
echo "<h1> Login failed. Invalid username or password.</h1>";
```

```
}
```

```
?>
```

HTML CODE:

```
<html>
```

```
<head>
```

```
<title>PHP login system</title>
```

```
//insert style.css file inside index.html
```

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

```
</head>
```

```
<body>
```

```
    <div id="frm">
```

```
        <h1>Login</h1>
```

```
<form name="fl" action= "authentication.php" onsubmit= "return  
validation()" method = "POST">
```

```
<p>
```

```
<label> UserName: </label>
```

```
<input type="text" id="user" name = "user" />
```

```
</p>
```

```
<p>
```

```
<label> Password: </label>
```

```
<input type="password" id="pass" name = "pass" />
```

```
</p>
```

```
<p>
```

```
<input type="submit" id="btn" value = "Login" />
```

```
</p>
```

```
</form>
```

```
</div>
```

```
// validation for empty field
```

```
<script>
```

```
function validation()
```

```
{
```

```
var id=document.fl.user.value;
```

```
var ps=document.fl.pass.value;
```

```
if(id.length=="" && ps.length=="") {
```

```
alert("User Name and Password fields are empty");
```

```
return false;
```

```
}
```

```
else
```

```
{
```

```
if(id.length=="") {
```

```
alert("User Name is empty");
```

```
return false;
```

```
}
```

```
if (ps.length=="") {
```

```
alert("Password field is empty");
```

```
return false;
```

```
}
```

```
}
```

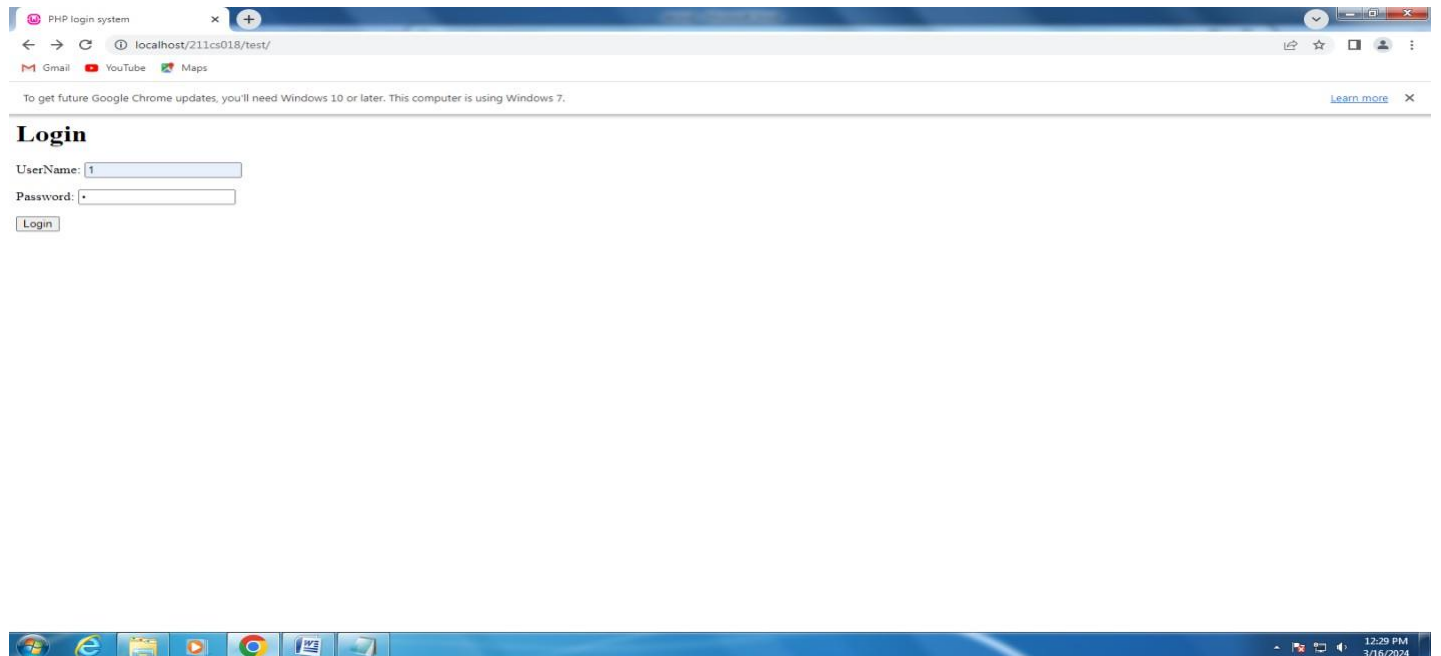
```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

OUTPUT:



14. WRITE A PROGRAM TO STORE PAGE VIEWS COUNT IN SESSION, TO INCREMENT THE COUNT ON EACH REFRESH AND TO SHOW THE COUNT ON WEBPAGE.

AIM:

The aim is to write a program to store page views count in SESSION, to increment the count on each refresh and to show the count on webpage.

ALGORITHM:

Step 1: Open a new file in the text editor.

Step 2: Using session _start () function we are going to count the session.

Step 3: Begin the php coding by opening the tag <? php.

Step 4: Add incremental conditions.

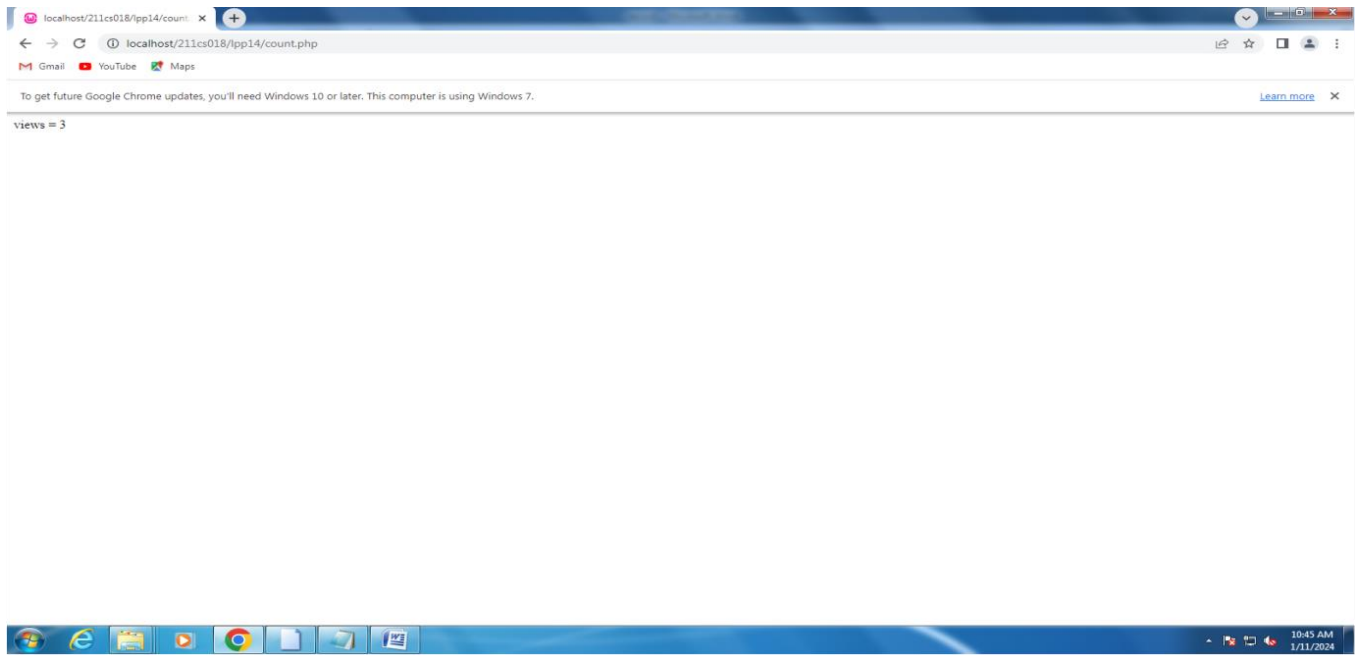
Step 5: Save the file with .php extension in the wamp server.

Step 6: The program is being executed.

CODING:

```
<?php  
  
session_start();  
  
if(isset($_SESSION['views']))  
  
$_SESSION['views'] = $_SESSION['views']+1;  
  
else  
  
$_SESSION['views']=1;  
  
echo"views=".$_SESSION['views'];  
  
?>
```

OUTPUT:



15.WRITE A PHP PROGRAM TO STORE CURRENT DATA TIME IN COOKIE AND DISPLAY THE LAST VISITED ON DATA TIME ON THE WEBPAGE UPON REOPENING OF THE SAME PAGE

AIM:

The aim is to write a php program to store current data time on cookie and display the last visited on data time on the webpage upon reopening of the same page.

ALGORITHM:

Step 1: Open a new file in the text editor.

Step 2: Here the php coding will be written within the `<html>` tag.

Step 3: Add necessary if conditions and save the file with .php extension in the wamp server.

Step 4: Open any browser and run the file.

Step 5: Now the file is being executed successfully.

CODING:

```
<html>

<body bgcolor="skyblue">

<center><h2> Last visited time on the web page</h2></center>

<br>

<?php

setcookie('last Visit', date("m/d/y"));

if(isset($_COOKIE['last Visit']))

{

$visit = $_COOKIE['last Visit'];

echo "Your last visit was- ".$visit;

}

else

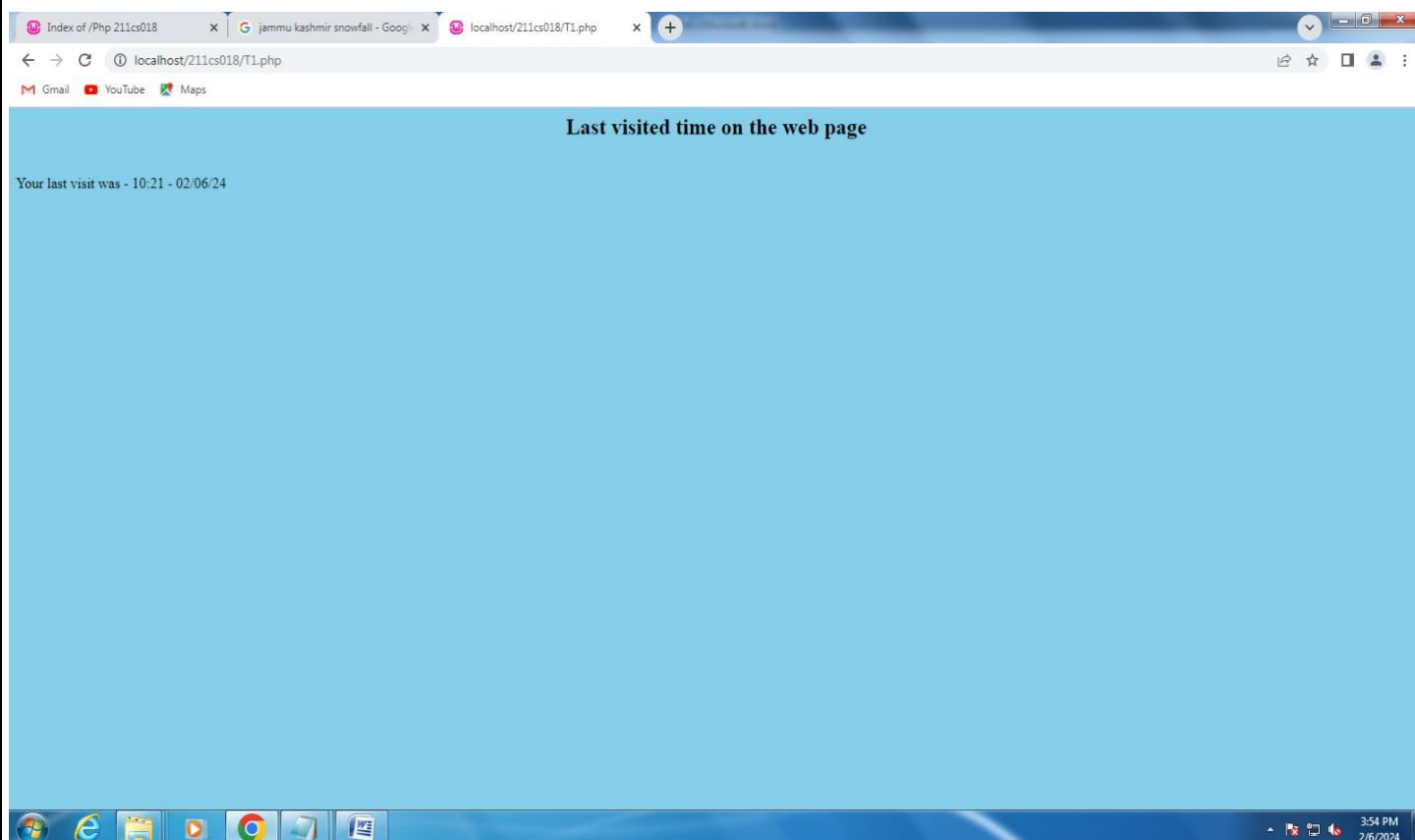
echo "You've got some state cookies!";

?>

</body>

</html>
```

OUTPUT:



16.CREATE A SIMPLE HTML FORM WITH FIELDS FOR NAME AND EMAIL.WRITE PHP SCRIPT HANDLE FORM SUBMISSION AND DISPLAY THE ENTERED DATA

Aim:

To create a simple HTML form with fields for Name and Email and a PHP script to handle form submission, process the input data, and display it on the web page.

Algorithm:

Step1: start the program

Step 2: Create an HTML form with:

1. A text input field for the Name
2. An email input field for the Email
3. A submit button

Step 3 : Set the form method to POST and action to process.php.

Step 4: Create a PHP script (process.php) to:

1. Check if the request method is POST.
2. Retrieve the Name and Email values from the form.
3. Sanitize the inputs to prevent security risks.
4. Display the entered data on the web page.

Step 5 :Run the script using a local PHP server.

Step 6 :Submit the form and verify the output.

HTML CODE :

```
<!DOCTYPE html>

<html>

<head>

    <title>Simple Form</title>

</head>

<body>

    <h2>Simple Form</h2>

    <form method="post" action="process.php">

        <label for="name">Name:</label>

        <input type="text" id="name" name="name" required><br><br>

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

        <input type="email" id="email" name="email" required><br><br>

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

    </form>

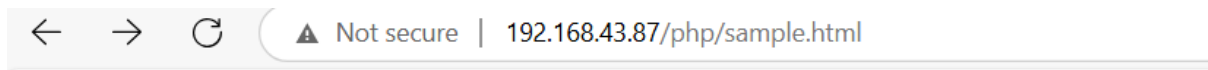
</body>

</html>
```

PHP CODE :

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = htmlspecialchars($_POST["name"]);
    $email = htmlspecialchars($_POST["email"]);

    echo "<h2>Form Data Submitted</h2>";
    echo "Name: " . $name . "<br>";
    echo "Email: " . $email . "<br>";
}
?>
```

OUTPUT:**Simple Form**Name: Email: **Form Data Submitted**Name: SAKTHIPOOJA R M
Email: ps9286322@gmail.com

17.WRITE A PHP PROGRAM TO CHECK WHETHER A GIVEN NUMBER IS ARMSTRONG OR NOT.

Aim:

To write a PHP program that checks whether a given number is an Armstrong number or not.

Algorithm:

Step1 :Start the program

Step 2:Take an integer number as input.

Step 3:Count the number of digits in the given number.

Step 4:Extract each digit from the number.

Step 5:Raise each digit to the power of the number of digits and sum them.

Step 6:Compare the calculated sum with the original number:

Step 7:Display the result.

PHP CODE:

```
<?php

function isArmstrong($number) {

    $sum = 0;

    $temp = $number;

    $numDigits = strlen((string)$number);

    while ($temp > 0) {

        $digit = $temp % 10;

        $sum += pow($digit, $numDigits);

        $temp = (int)($temp / 10);

    }

    return $sum == $number;

}

// Example usage

$number = 153; // Change this to test different numbers

if (isArmstrong($number)) {

    echo "$number is an Armstrong number.";

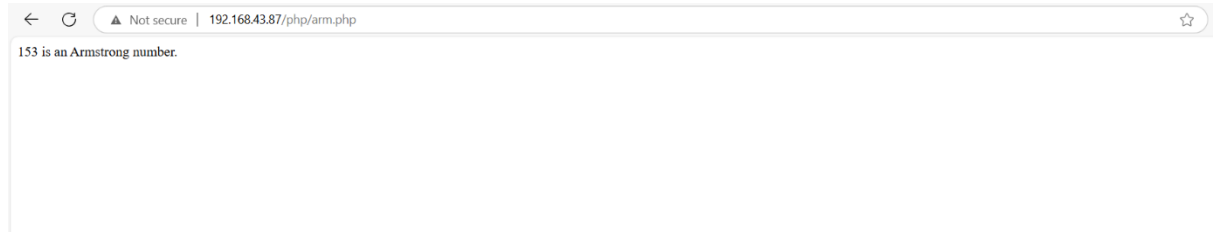
} else {

    echo "$number is not an Armstrong number.";

}

?>
```

OUTPUT:



18.WRITE A PHP PROGRAM TO FIND LARGEST VALUES OF TWO NUMBERS USING NESTING OF FUNCTION.

Aim:

To write a PHP program that finds the largest of two numbers using function nesting.

Algorithm:

Step1 :Define an outer function findLargest(num1, num2).

Step 2 :Inside this function, define a nested function compare(a, b) that:

- Compares a and b.
- Returns the larger value.

Step 3:Call the nested function inside findLargest() to determine the largest number.

Step 4:Return the largest number from findLargest().

Step 5:Take two numbers as input.

Step 6:Call findLargest() with the input numbers.

Step 7:Display the largest number.

PHP CODE :

```
<?php

function findLargest($num1, $num2) {

    // Nested function to compare two numbers

    function compare($a, $b) {

        return ($a > $b) ? $a : $b;

    }

    return compare($num1, $num2);

}

// Example usage

$number1 = 45;

$number2 = 78;

$largest = findLargest($number1, $number2);

echo "The largest number between $number1 and $number2 is: $largest";

?>
```

OUTPUT:

