Diploma Engineering

Laboratory Manual

Web Development using PHP
4341604

Information Technology 4th Semester

Enrolment No	
Name	
Branch	
Academic Term	
Institute	

Directorate Of Technical Education
Gandhinagar - Gujarat

DTE's Vision:

- To provide globally competitive technical education;
- Remove geographical imbalances and inconsistencies;
- Develop student friendly resources with a special focus on girls' education and support to weaker sections;
- Develop programs relevant to industry and create a vibrant pool of technical professionals.

DTE's Mission:

Institute's Vision:

Institute's Mission:

Department's Vision:

Department's Mission:

Certificate

This is to certify that Mr./Ms …….. Enrollment No. …….. of **4th** Semester of *Diploma in* ***Information Technology*** of Institute …….. (GTU Code: ……) has satisfactorily completed the term work in course "**Web Development using PHP (4341604)"** for the academic year: …….. Term: **Even** prescribed in the GTU curriculum.

Place: ……..

Date: ……..

Signature of Course Faculty **Head of the Department**

Preface

The primary aim of any laboratory/Practical/field work is enhancement of required skills as well as creative ability amongst students to solve real time problems by developing relevant competencies in psychomotor domain. Keeping in view, GTU has designed competency focused outcome-based curriculum -2021 (COGC-2021) for Diploma engineering programmes. In this more time is allotted to practical work than theory. It shows importance of enhancement of skills amongst students and it pays attention to utilize every second of time allotted for practical amongst Students, Instructors and Lecturers to achieve relevant outcomes by performing rather than writing practice in study type. It is essential for effective implementation of competency focused outcome- based Green curriculum-2021. Every practical has been keenly designed to serve as a tool to develop & enhance relevant industry needed competency in each and every student. These psychomotor skills are very difficult to develop through traditional chalk and board content delivery method in the classroom. Accordingly, this lab manual has been designed to focus on the industry defined relevant outcomes, rather than old practice of conducting practical to prove concept and theory.

By using this lab manual, students can read procedure one day in advance to actual performance day of practical experiment which generates interest and also, they can have idea of judgement of magnitude prior to performance. This in turn enhances predetermined outcomes amongst students. Each and every Experiment /Practical in this manual begins by competency, industry relevant skills, course outcomes as well as practical outcomes which serve as a key role for doing the practical. The students will also have a clear idea of safety and necessary precautions to be taken while performing experiment.

This manual also provides guidelines to lecturers to facilitate student-centered lab activities for each practical/experiment by arranging and managing necessary resources in order that the students follow the procedures with required safety and necessary precautions to achieve outcomes. It also gives an idea that how students will be assessed by providing Rubrics.

Website design is a broad term that encompasses a wide variety of tasks, all involved in the formation of web pages. There are essentially two types of web design approaches, which are dynamic and static design. PHP, which stands for "PHP: Hypertext Pre-processor" is a widely-used Open Source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. Its syntax draws upon C, Java, and Perl, and is easy to learn. The main goal of the language is to allow web developers to write dynamically generated web pages quickly, but you can do much more with PHP.

Although we try our level best to design this lab manual, but always there are chances of improvement. We welcome any suggestions for improvement.

Programme Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Engineering practices for society, sustainability and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices.
6. **Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
7. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

Practical Outcome - Course Outcome matrix

Course Outcomes (COs):

a. Create small programs using basic PHP concepts.
b. Create User defined functions in PHP programming.
c. Design and develop a Web site using form controls for presenting web-based content.
d. Debug the Programs by applying state management concepts and error handling techniques of PHP.
e. Create dynamic web pages using PHP and MySQL database.

Sr. No.	Practical Outcome/Title of experiment	CO1	CO2	CO3	CO4	CO5
1.	Environment setup, Configuration and Syntax a. Install and configure PHP, Apache Web Server and database server using XAMPP. b. Write a PHP script to display Welcome message.	√	-	-	-	-
2.	Variables, Constant and Operators a. Write a PHP script to demonstrate use of global, local, static variables and constant. b. Write a PHP script to demonstrate arithmetic operators, comparison operator, and logical operator. c. Write a PHP program to swap two numbers with and without using third variable.	√	-	-	-	-
3.	Conditional Statements a. Write a PHP script to check the given number is odd or even. b. Write a PHP script to print student's grade based on marks of 5 subjects. c. Create a PHP script to show the month of a year using switch statement.	√	-	-	-	-
4.	Looping Structures a. Write PHP Script to print Fibonacci series in html tabular format. b. Write a PHP script to print below number triangle. 1 2 3 4 5 6 7 8 9 10 c. Write a PHP script to create chess board (tabular structure).	√	-	-	-	-

5.	Arrays a. Write PHP Script for addition and multiplication of two 2x2 matrices. b. Write a PHP Script to count the number of elements in an array without using built-in function (use numeric and associative arrays).	√	-	-	-	-
6.	User defined Functions a. Write a PHP script to call by reference and call by value. b. Write a PHP Script for performing function that takes arguments with default argument and returns value. c. Write a PHP Script to show the use of variable length argument.	-	√	-	-	-
7.	Built-In functions a. Write PHP script to demonstrate use of various strings handling function. b. Write a PHP script to Demonstrate Include() and require() function. c. Write PHP script to demonstrate Array functions. d. Write PHP script to demonstrate use of fopen(), fread(), fwrite() and fclose() File functions.	-	√	-	-	-
8.	Form Handling a. Create student registration form using text box, check box, radio button, select, submit button. And display user inserted value in new PHP page using GET or POST Method. b. Write a PHP script to explain the concept of $_REQUEST.	-	-	√	-	-
9.	Email and Validation a. Write PHP script to validate form including name, email using appropriate functions. b. Write PHP script for sending plain text email, HTML email and attachments with email.	-	√	-	-	-
10.	Session and Cookies a. Write a PHP script to demonstrate creating, deleting, updating, retrieving and passing variable cookie data. b. Write PHP script to demonstrate passing information using Session.	-	-	-	√	-

11.	Error Handling a. Write a PHP script to demonstrate Error Handling.	-	-	-	√	-
12.	Database b. Write a PHP script to connect with database server from your webpage. c. Create a database with student table and write a PHP script to insert a record in student table. d. Write a program to read student records from student table and display all these information in table format on output screen. e. Write a PHP script to delete and update a specific record from table. f. Write a PHP script simple login system that allows user to add a new username if user doesn't exist in the database, also create a forgot password link, to redirect user to set up his new password on authentication.	-	-	-	-	√

Industry Relevant Skills

The following industry relevant skills are expected to be developed in the students by performance of experiments of this course.

- Develop Interactive Web application using PHP.
- Apply concept to create webpage using data and form handling.
- Use of state management and error handling.
- Use database operations in web application.
- Follow coding standards.

Guidelines to Course Faculty

1. Couse faculty should demonstrate experiment with all necessary implementation strategies described in curriculum.
2. Couse faculty should explain industrial relevance before starting of each experiment.
3. Course faculty should involve & give opportunity to all students for hands on experience.
4. Course faculty should ensure mentioned skills are developed in the students by asking.
5. Utilise 2 hrs of lab hours effectively and ensure completion of write up with quiz also.
6. Encourage peer to peer learning by doing same experiment through fast learners.

Instructions for Students

1. Organize the work in the group and make record of all observations.
2. Students shall develop maintenance skill as expected by industries.
3. Student shall attempt to develop related hand-on skills and build confidence.
4. Student shall develop the habits of evolving more ideas, innovations, skills etc.
5. Student shall refer technical magazines and data books.
6. Student should develop habit to submit the practical on date and time.
7. Student should well prepare while submitting write-up of exercise.

Continuous Assessment Sheet

Enrollment No:

Name: **Term:**

Sr. No.	Practical Outcome/Title of experiment	Page	Date	Marks (10)	Sign
1.	Environment setup, Configuration and Syntax a. Install and configure PHP, Apache Web Server and database server using XAMPP. b. Write a PHP script to display Welcome message.				
2.	Variables, Constant and Operators a. Write a PHP script to demonstrate use of global, local, static variables and constant. b. Write a PHP script to demonstrate arithmetic operators, comparison operator, and logical operator. c. Write a PHP program to swap two numbers with and without using third variable.				
3.	Conditional Statements a. Write a PHP script to check the given number is odd or even. b. Write a PHP script to print student's grade based on marks of 5 subjects. c. Create a PHP script to show the month of a year using switch statement.				
4.	Looping Structures a. Write PHP Script to print Fibonacci series in html tabular format. b. Write a PHP script to print below number triangle. 1 2 3 4 5 6 7 8 9 10 c. Write a PHP script to create chess board (tabular structure).				
5.	Arrays a. Write PHP Script for addition and multiplication of two 2x2 matrices. b. Write a PHP Script to count the number of elements in an array without using built-in function (use numeric and associative arrays).				

6.	User defined Functions a. Write a PHP script to call by reference and call by value. b. Write a PHP Script for performing function that takes arguments with default argument and returns value. c. Write a PHP Script to show the use of variable length argument.				
7.	Built-In functions a. Write PHP script to demonstrate use of various strings handling function. b. Write a PHP script to Demonstrate Include () and require() function. c. Write PHP script to demonstrate Array functions. d. Write PHP script to demonstrate use of fopen(), fread(), fwrite() and fclose() File functions.				
8.	Form Handling a. Create student registration form using text box, check box, radio button, select, submit button. And display user inserted value in new PHP page using GET or POST Method. b. Write a PHP script to explain the concept of $_REQUEST.				
9.	Email and Validation a. Write PHP script to validate form including name, email using appropriate functions. b. Write PHP script for sending plain text email, HTML email and attachments with email.				
10.	Session and Cookies a. Write a PHP script to demonstrate creating, deleting, updating, retrieving and passing variable cookie data. b. Write PHP script to demonstrate passing information using Session.				
11.	Error Handling a. Write a PHP script to demonstrate Error Handling.				

12.	Database a. Write a PHP script to connect with database server from your webpage. b. Create a database with student table and write a PHP script to insert a record in student table. c. Write a program to read student records from student table and display all these information in table format on output screen. d. Write a PHP script to delete and update a specific record from table. e. Write a PHP script simple login system that allows user to add a new username if user doesn't exist in the database, also create a forgot password link, to redirect user to set up his new password on authentication.				

Date: _____

Practical No.1: Environment setup, Configuration and Syntax

a. Install and configure PHP, Apache Web Server and database server using XAMPP.
b. Write a PHP script to display Welcome message.

A. Objective:

PHP is a popular scripting language, which is used to develop various web applications. PHP environment for executing PHP program using WAMP or XAMPP server. This will help you to write, debug and run the PHP program.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Engineering Tools, Experimentation and Testing: Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"
This practical is expected to develop the following skills.

1. Setup PHP environment for executing PHP program.
2. Write a PHP script and execute it.

D. Expected Course Outcomes(COs)

CO1: Create small programs using basic PHP concepts.

E. Practical Outcome(PrO)

Students will be able install/configure development environment and also write a PHP script to display Welcome message.

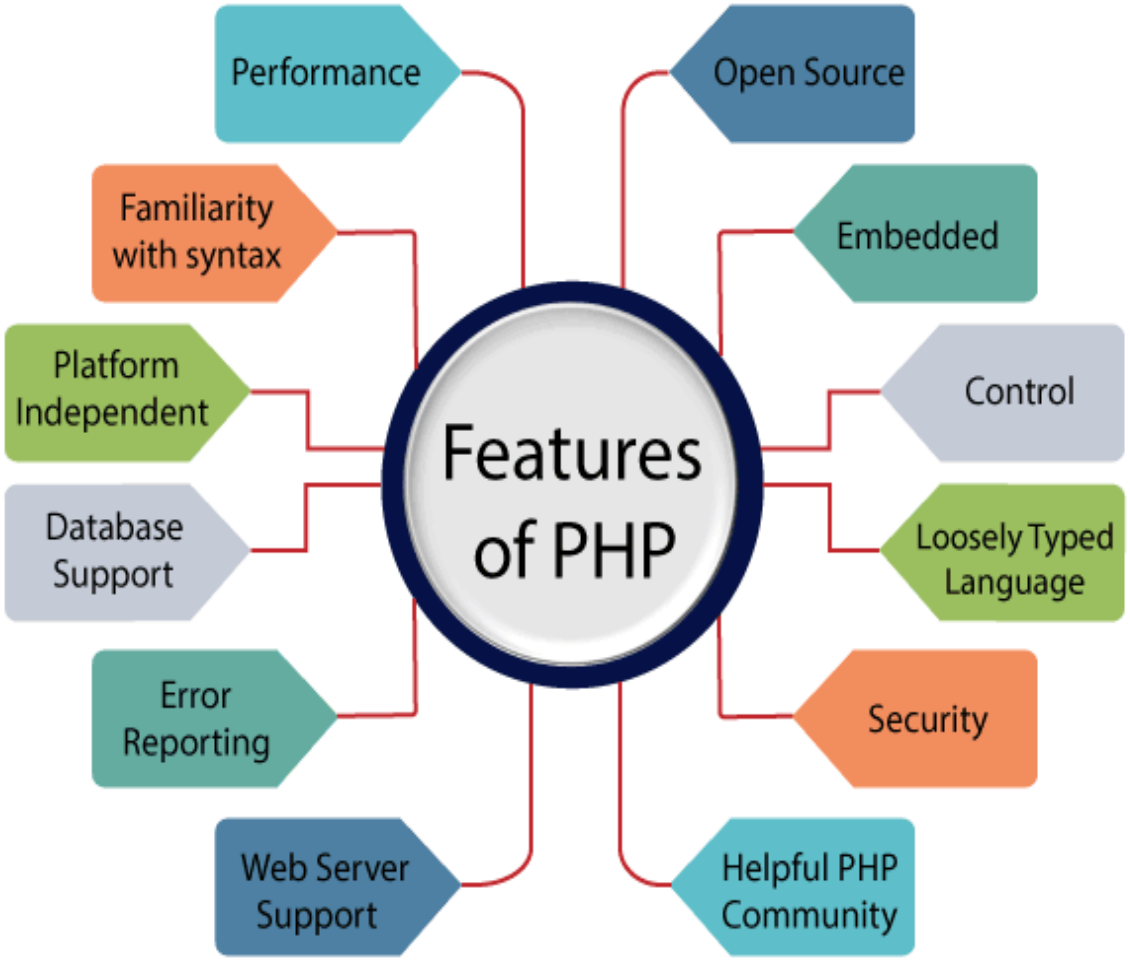
F. Expected Affective domain Outcome(ADos)

1. Follow safety practices.
2. Follow Coding standards and practices.
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.
5. Maintain tools and equipment.

G. Prerequisite Theory:

What is PHP?

- PHP was created by Rasmus Lerdorf in 1994 but appeared in the market in 1995.
- PHP stands for Hypertext Pre-processor.
- PHP is a server scripting language.
- PHP scripts are executed on the server.
- PHP is a widely used open-source scripting language, so PHP is free to download and use.
- PHP is a powerful tool for making dynamic and interactive Web pages.
- PHP is an interpreted language.
- PHP can be embedded into HTML.
- PHP is an object-oriented language.
- PHP is simple and easy to learn language.



[Fig. Ref: https://www.javatpoint.com/php-tutorial]

Environment Setup:

XAMPP is one of the most popular software pack to set up web application development environment for PHP with all required software components. XAMPP is an Open Source AMP stack which stands for Cross platform, Apache, MariaDB, PHP and Perl. Apache is cross platform web server, MariaDB is the most widely used database developed by MySQL, PHP is a backend scripting language and Perl is a programming used for web development. X denotes Cross-platform, which means that it can work on different platforms such as Windows, Linux, and macOS.

Install XAMPP server on Windows 7 or above.

1. Open the XAMPP website. Go to https://www.apachefriends.org/index.html in your computer's web browser.

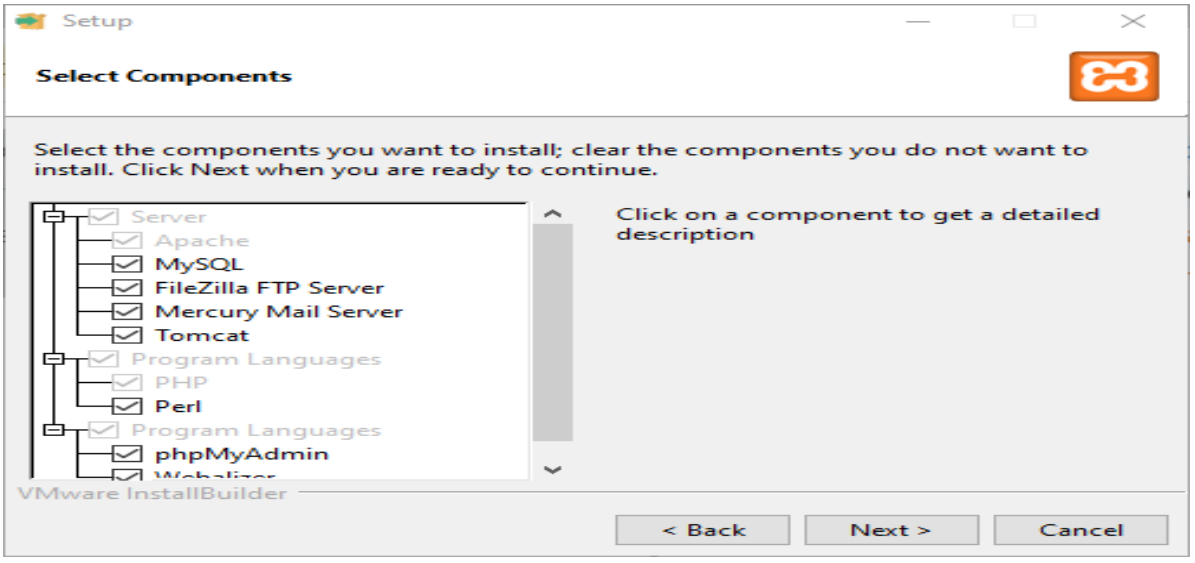
Download latest XAMPP for Windows operating system.

2. Once the XAMPP setup has been downloaded, you can start the installation by double clicking on the .exe file.
3. An active antivirus program can interfere with the installation process, so it is best to temporarily disable any antivirus software until all XAMPP components have been successfully installed.

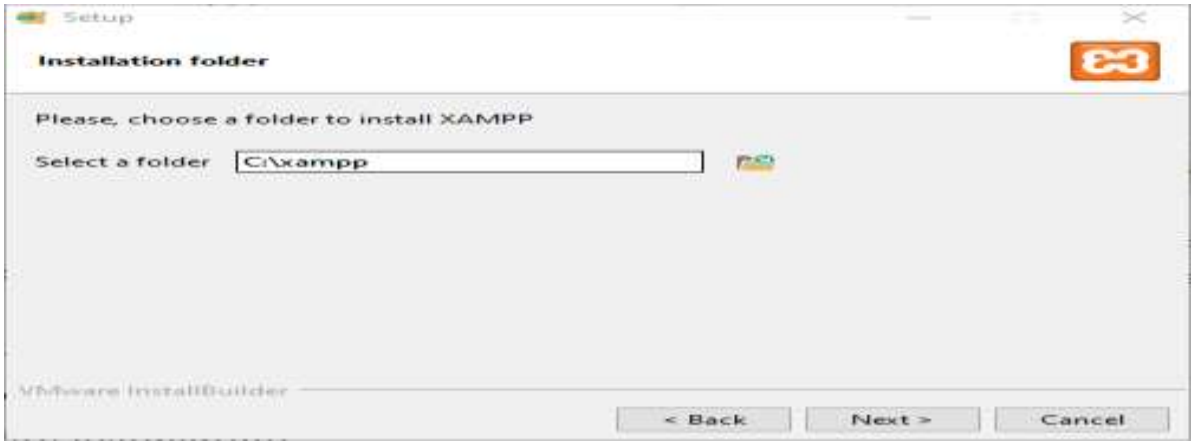
4. Because User Account Control (UAC) restricts writing access to the C: drive and can interfere with the XAMPP installation, it is recommended that this be disabled for the duration of the installation.

5. After that the start screen of the XAMPP setup wizard should appear automatically. Click on 'Next' to configure the installation settings.

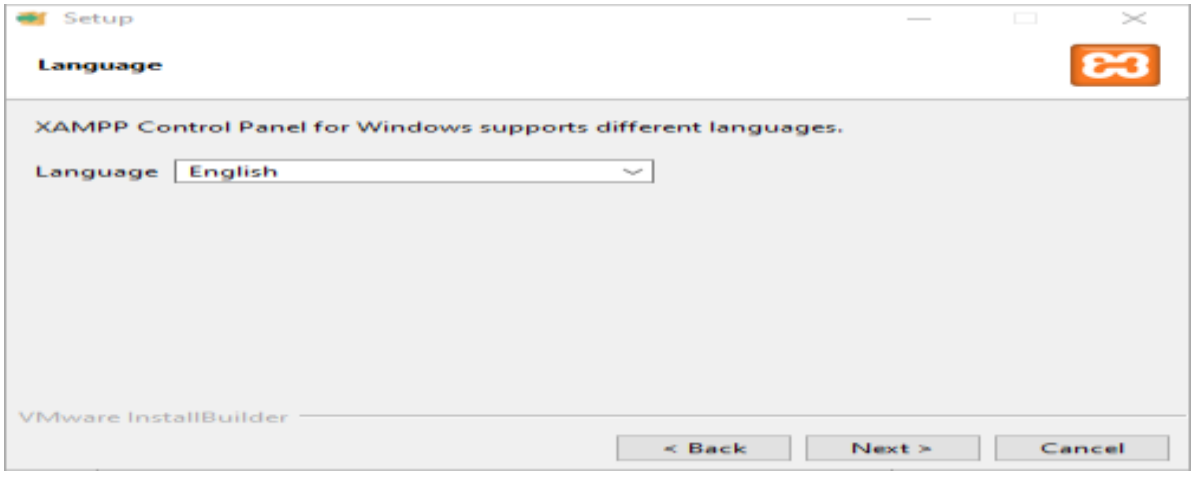
6. Under 'Select Components', you have the option to exclude individual components of the XAMPP software bundle from the installation. But for a full local test server, we recommend you install using the standard setup and all available components. After making your choice, click 'Next'.



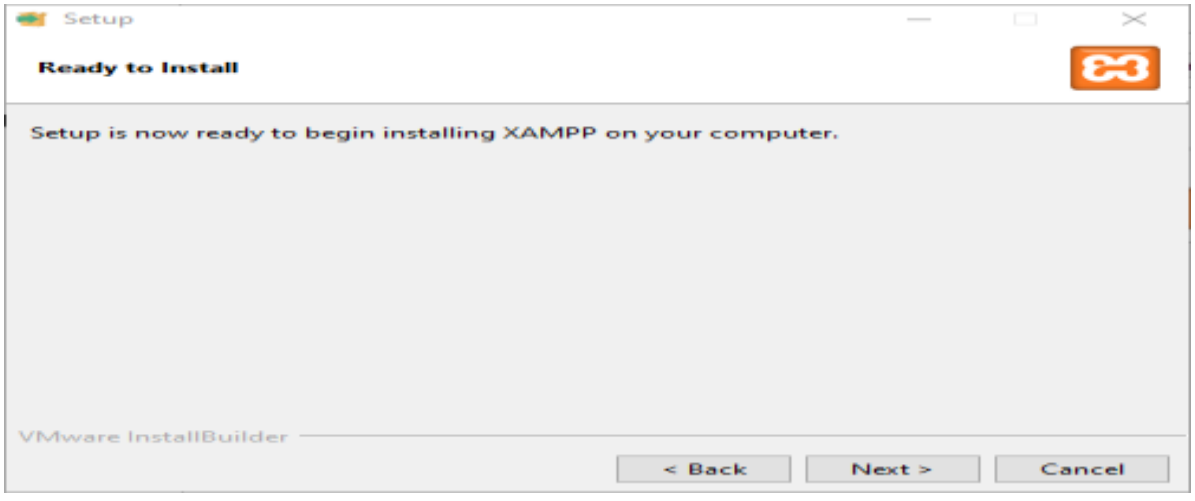
7. In this next step, you have the chance to choose where you'd like the XAMPP software packet to be installed. If you opt for the standard setup, then a folder with the name *xampp* will be created under *C:* for you. After you've chosen a location, click 'Next'.



8. Select the language in the next dialog box.

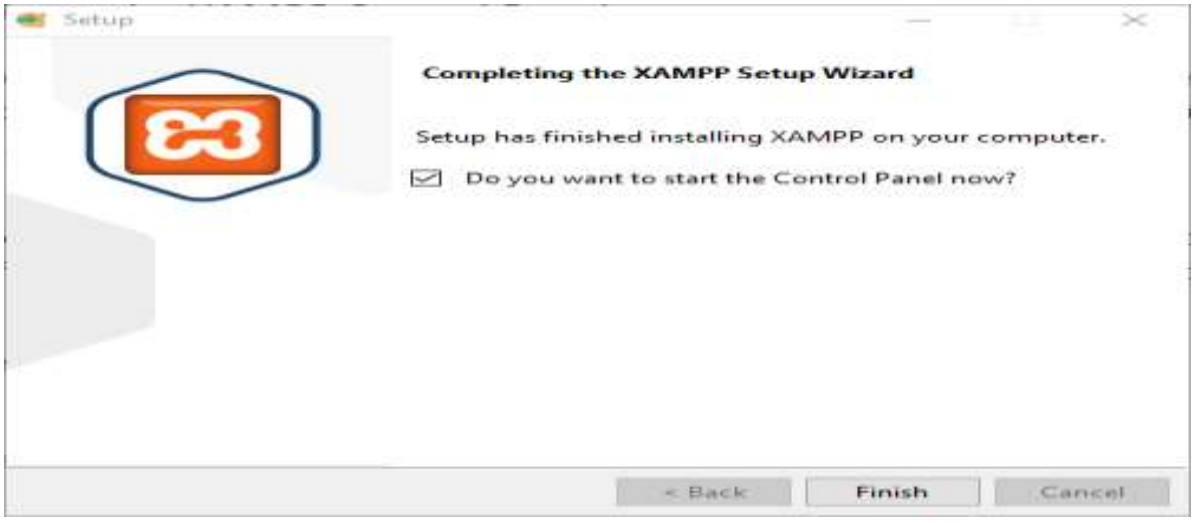


9. On Ready *to* Install screen click on "Next".



10. Once all the preferences have been decided, click to start the installation. The setup wizard will unpack and install the selected components and save them to the designated directory. This process may take few minutes.

11. Your Firewall may interrupt the installation process to block the some components of the XAMPP. Use the corresponding check box to enable communication between the Apache server and your private network or work network. Remember that making your XAMPP server available for public networks isn't recommended.
12. Once all the components are unpacked and installed, you can close the setup wizard by clicking on 'Finish'. Click to tick the corresponding check box and open the XAMPP Control Panel once the installation process is finished.



13. XAMPP Control Panel provides controls for the individual components of your xampp test server. The control panel user interface allows you to start or stop individual modules: Apache, MySQL, FileZilla, Mercury and Tomcat. The XAMPP Control Panel also offers you various other buttons, including:

    Config: allows you to configure the XAMPP as well as the individual components

Netstat: shows all running processes on the local computer

Shell: opens a UNIX shell

Explorer: opens the XAMPP folder in Windows Explorer

Services: shows all services currently running in the background

Help: offers links to user forums

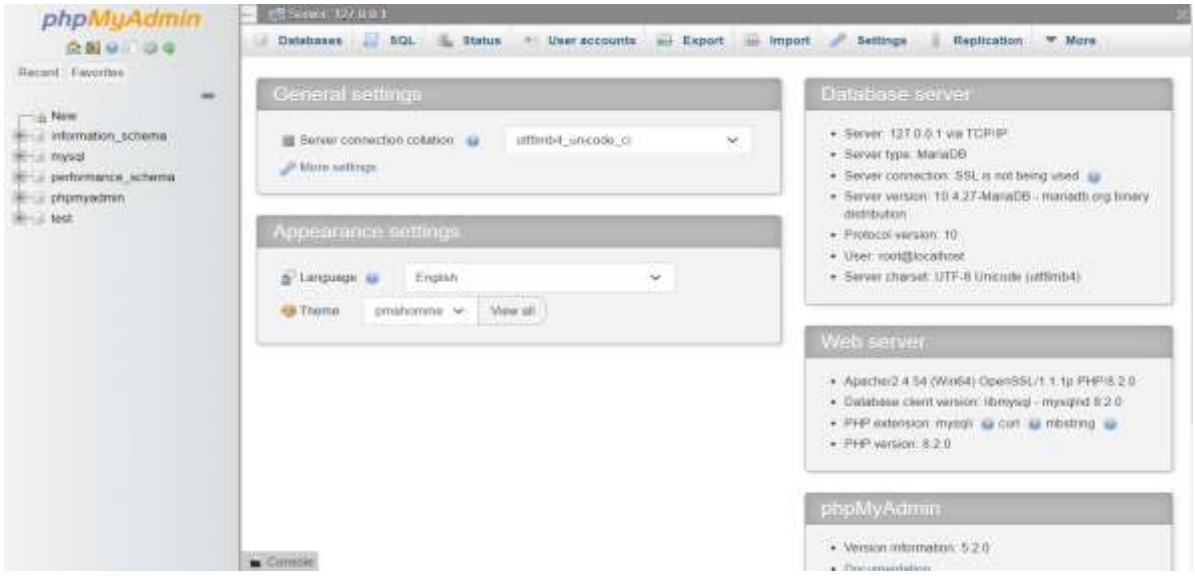
Quit: closes the XAMPP Control Panel

14. Individual modules can be started or stopped on the XAMPP Control Panel through the corresponding buttons under 'Actions'. You can see which modules have been started because their names are highlighted green under the 'Module' title.

If a module can't be started as a result of an error, you'll be informed of this straight away in red font. A detailed error report can help you identify the cause of the issue.

15. You have an 'Admin' option located on the Control Panel for every module in your XAMPP.
16. Click on the Admin button of your Apache server to go to the web address of your web server. The Control Panel will now start in your standard browser, and you'll be led to the dashboard of your XAMPP's local host. The dashboard features numerous links to websites for useful information as well as the open source project BitNami, which offers you many different applications for your XAMPP, like WordPress or other content management systems. Alternatively, you can reach the dashboard through localhost/dashboard/.

17. You can use the Admin button of your database module to open phpMyAdmin. Here, you can manage the databases of your web projects that you're testing on your XAMPP. Alternatively, you can reach the administration section of your MySQL database via localhost/phpmyadmin/



Testing XAMPP installation

To check whether your test server is installed and configured correctly, you have the option to create a PHP test page, store them on your XAMPP's local host, and retrieve them via the web browser.

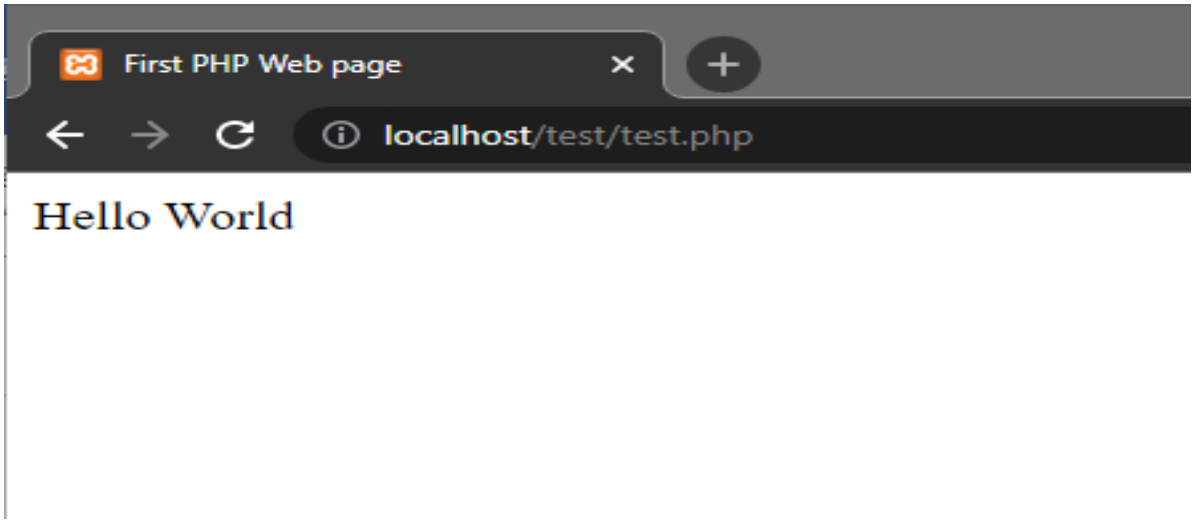
18. Open the XAMPP directory through the 'Explorer' button in the Control Panel and choose the folder *htdocs* (*C:\xampp\htdocs* for standard installations). This directory should store all the web pages that you want to test on your XAMPP server. The *htdocs* folder should already contain data to help configuration of the web server. But you should store your own projects in a new folder (for example '*test*' folder).
19. You can create a new PHP file with below code in your editor and storing it as *test.php* in your '*test*' folder (*C:\xampp\htdocs\test*):

test.php

```
<html>
    <head>
        <title>First PHP Web page</title>
    </head>
    <body>
        <?php
            echo '<p>Hello World</p>';
        ?>
    </body>
</html>

```

20. Now open a web browser and load your PHP page via localhost/test/test.php. If your browser window displays the words 'Hello World', then you've successfully installed and configured your XAMPP.



H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

Write a PHP script to display Welcome message.

K. Input-Output:

L. Practical related Quiz.

1) PHP stands for _____.
2) _____ is the default file extension of PHP.
3) PHP is _____ scripting type of language.
4) A PHP script starts with _____ and ends with _____.

M. References / Suggestions

1. https://www.w3schools.com/php/default.asp
2. https://www.tutorialspoint.com/php/
3. https://tutorialehtml.com/en/php-tutorial-introduction/
4. www.tizag.com/phpT/
5. https://books.goalkicker.com/PHPBook/
6. https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
7. https://codecourse.com/watch/php-basics
8. https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

N. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _____

Practical No.2: Variables, Constant and Operators

a. Write a PHP script to demonstrate use of global, local, static variables and constant.
b. Write a PHP script to demonstrate arithmetic operators, comparison operator, and logical operator.
c. Write a PHP program to swap two numbers with and without using third variable.

A. Objective:

Variable, Constant and Operators plays an important role in many programming languages. In PHP also it's important to learn all of them. Understand the difference between different scope of variables, the use of constant and how different operators work in PHP.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"

This practical is expected to develop the following skills.

1. Understand the use of different scopes of variable in PHP.
2. Write a PHP script for given problems and execute it.
3. Write appropriate code for different operators.
4. Debug code and solve the errors.

D. Expected Course Outcomes(Cos)

CO1: Create small programs using basic PHP concepts.

E. Practical Outcome(PRo)

Students will be able to write a PHP script for different scopes of variable and constant.

Students will be able to write a PHP script for different kind of operators.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Variable in PHP

Variables in a program are used to store some values, so they are the containers for storing information. These information can be numeric values, characters, character strings, or memory addresses.

In PHP, a variable does not need to be declared before adding a value to it. PHP automatically converts the variable to the correct data type, depending on its value, so it is called loosely typed variables. After declaring a variable it can be reused throughout the code. Below rules should be followed if you are dealing with variable in PHP.

- Any variables declared in PHP must **begin with a dollar sign ($)**, followed by the variable name.
- A variable name must start with a letter or the underscore character.
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters and underscores (A-Z, a-z, 0-9, and _).
- Variable names are case-sensitive ($avg and $AVG are two different variables).
- The assignment operator (=) used to assign value to a variable.

PHP has a total of eight data types which we use to construct our variables –

- Integers – are whole numbers, without a decimal point, like 25, 9876.
- Doubles – are floating-point numbers, like 3.14 or 49.19876.

- Booleans – have only two possible values either true or false.
- NULL – is a special type that only has one value: NULL.
- Strings – are sequences of characters, like 'Web Development using PHP'
- Arrays – are named and indexed collections of other values.
- Objects – are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- Resources – are special variables that hold references to resources external to PHP (such as database connections).

Example-1

```
<?php
        // Declaring variables
        $txt = "Hello World";
        $num = 5;


        // Displaying variables value
        echo $txt;  // Output: Hello World!
        echo $num; // Output: 5
?>
```

Scope of Variable

Variable can be declared anywhere in the script in PHP. According to where the variable can be used/accessed in the script it is divided into three scopes:

- Local
- Global
- Static

Local Scope

The variables declared within a function are called local variables to that function and have their scope only in that particular function. In simple words, it cannot be accessed outside that function. Any declaration of a variable outside the function with the same name as that of the one within the function is a completely different variable.

```
<?php
        function test()
        {
          $a = 10; // local scope
          echo "Variable a inside function is: $a <br/>";
        }
        test();

        echo "Variable a outside function is: $a";
?>
```

O/P:

```
Variable a inside function is: 10
Variable a outside function is:
```

Global Scope

In PHP, A variable declared outside a function has a global scope and can only be accessed outside a function.

```
<?php
        $a = 10; // global scope
        function test()
        {
          echo "Variable a inside function is: $a <br/>";
        }
        test();

        echo "Variable a outside function is: $a";
?>
```

O/P:

```
Variable a inside function is:
Variable a outside function is: 10
```

To get access within a function we need to use the “global” keyword before the variable to refer to the global variable.

```
<?php
        $a = 10; // global scope
        function test()
        {
                global $a;
                echo "Variable a inside function is: $a <br/>";
        }
        test();

        echo "Variable a outside function is: $a";
?>
```

O/P:

```
Variable a inside function is: 10
Variable a outside function is: 10
```

PHP also stores all global variables in an array called $GLOBALS[*index*]. The *index* holds the name of the variable. This array is also accessible from within functions and can be used to update global variables directly.

```
<?php
        $a = 10; // global scope
        function test()
        {
                global $a;
                echo "Variable a inside function is:". $GLOBALS['a'] ."<br/>";
        }
        test();

        echo "Variable a outside function is: $a";
?>
```

O/P:

```
Variable a inside function is: 10
Variable a outside function is: 10
```

Static Variable

It is the characteristic of PHP to delete the variable, once it completes its execution and the memory is freed. But sometimes we need to store the variables even after the completion of function execution. To do this we use the static keywords and the variables are then called static variables.

```
<?php
        function test()
        {
                static $a = 10
                echo $a .”<br/>”;
                $a++;
        }
        test();
        test();
        test();
?>
O/P:

        10
        11
        12
```

Constant

Constants are opposite to variables in terms of that once they are defined they cannot be changed or undefined. A constant is an identifier (name) for a simple value. The value cannot be changed during the script. A valid constant name starts with a letter or underscore (no $ sign before the constant name).

To create a constant, **define()** function is used.

Syntax:

define(*name*, *value*, *case-insensitive*)

Here,
name: Specifies the name of the constant
value: Specifies the value of the constant
case-insensitive: Specifies whether the constant name should be case-insensitive. Default is false

```
<?php
        // case-sensitive constant
        define("TEST", "This is constant.");
        echo TEST ."<br/>";

        // case-insensitive constant
        define("TEST", " This is constant.", true);
        echo test;
?>

O/P:
        This is constant.
        This is constant.
```

Operators

Operator is a symbol used to perform operations on operands (variables or values). For example:

$a = $b + 10;

Above code uses arithmetic operator (+) to add 10 to variable *$b* and assign it to variable *$a.*

PHP operators can be categorized into following types:

Arithmetic Operators:

Operator	Name	Example	Description
+	Addition	$a + $b	Sum of two operands
-	Subtraction	$a - $b	Difference of two operands
*	Multiplication	$a * $b	Multiply tow operands
/	Division	$a / $b	Quotient of operands
%	Modulo	$a % $b	Reminder of operands

++	Increment	$a++	Same as $a = $a + 1
--	Decrement	$a--	Same as $a = $a - 1

Assignment Operators:

Operator	Name	Example	Description
=	Assign	$a = $b	Value of right operand is assigned to left operand
+=	Add then assign	$a += $b	Same as $a = $a + $b
-=	Subtract then assign	$a -= $b	Same as $a = $a - $b
*=	Multiply then assign	$a *= $b	Same as $a = $a * $b
/=	Divide then assign (Quotient)	$a /= $b	Same as $a = $a / $b
%=	Divide then assign (Reminder)	$a %= $b	Same as $a = $a % $b

Bitwise Operators:

Operator	Name	Example	Description
&	Bitwise AND	$a & $b	Bitwise AND operation between $a and $b
|	Bitwise OR	$a | $b	Bitwise OR operation between $a and $b
^	Bitwise XOR	$a ^ $b	Bitwise XOR operation between $a and $b
~	Bitwise NOT	~ $a	Bitwise NOT operation on $a
<<	Left shift	$a << $b	Left shift bits of $a by $b steps
>>	Right shift	$a >> $b	Right shift bits of $a by $b steps

Comparison Operators:

Operator	Name	Example	Description
==	Equal	$a == $b	Returns TRUE if $a is equal to $b
!=	Not equal	$a != $b	Returns TRUE if $a is not

			equal to $b
<>	Not equal	$a <> $b	Returns TRUE if $a is not equal to $b
===	Identical	$a === $b	Returns TRUE if $a and $b are equal and of same data type
!==	Not identical	$a !== $b	Returns TRUE if $a and $b are not equal or of different data type
<	Less than	$a < $b	Returns TRUE if $a is less than $b
>	Greater than	$a > $b	Returns TRUE if $a is greater than $b
<=	Less than or equal to	$a <= $b	Returns TRUE if $a is less than or equal to $b
>=	Greater than or equal to	$a >= $b	Returns TRUE if $a is greater than or equal to $b
<=>	Spaceship	$a <=> $b	Return -1 if $a is less than $b Return 0 if $a is equal $b Return 1 if $a is greater than $b

Logical Operators:

Operator	Name	Example	Description
and	Logical AND	$a and $b	Returns TRUE if both $a and $b are true
or	Logical OR	$a or $b	Returns TRUE if either $a or $b is true
xor	Logical XOR	$a xor $b	Returns TRUE if either $a or $b is true, but not both are TRUE
!	Logical NOT	! $a	Returns TRUE if $a is FALSE
&&	Logical AND	$a && $b	Returns TRUE if both $a and $b are true
||	Logical OR	$a || $b	Returns TRUE if either $a or $b is true

String Operators:

Operator	Name	Example	Description
.	Concatenation	$a . $b	Concatenate both $a and $b
.=	Concatenation and assign	$a .= $b	Same as $a = $a . $b

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

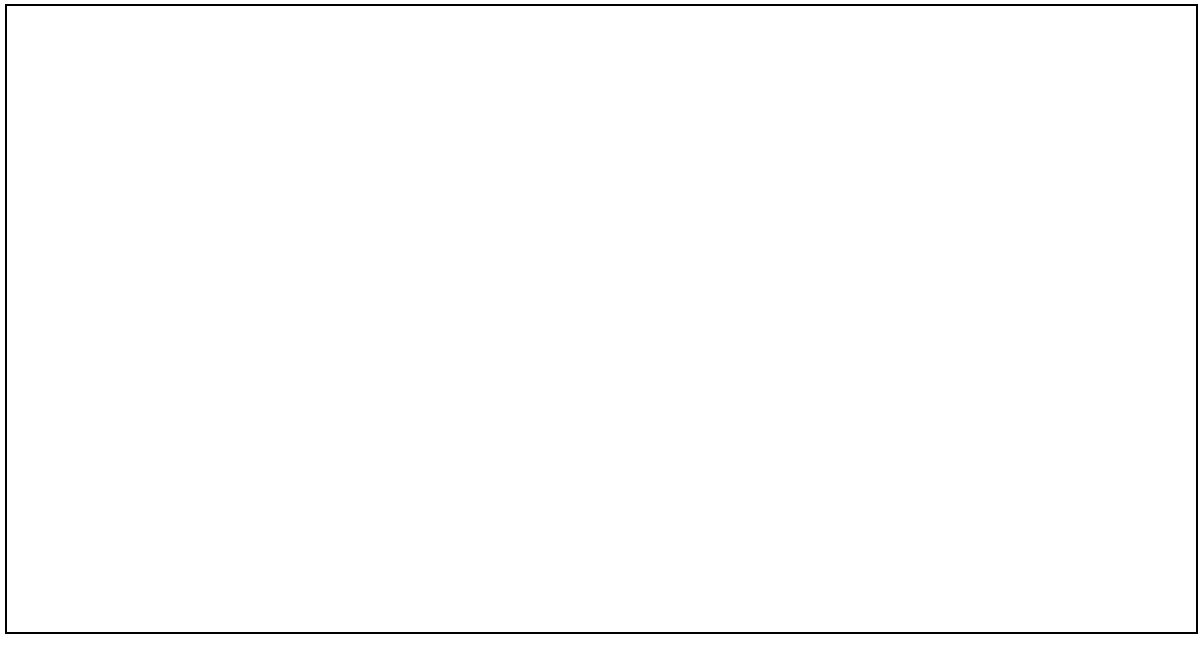
I. Safety and necessary Precautions followed

NA

J. Source code:

A) Write *a PHP script to demonstrate use of global, local, static variables and constant.*

A) Input-Output:



B) *Write a PHP script to demonstrate arithmetic operators, comparison operator, and logical operator.*

B) Input-Output:

C) *Write a PHP program to swap two numbers with and without using third variable.*

C) Input-Output:

K. Practical related Quiz.

1) Which of the following is correct to add a comment in PHP?
   a. & …… &
   b. // ……
   c. /* …… */
   d. Both (b) and (c)

2) Which of the following is used for concatenation in PHP?
   a. + (plus)
   b. * (Asterisk)
   c. . (dot)
   d. append()

3) Which is the right way of declaring a variable in PHP?
   a. $3hello
   b. $_hello
   c. $this
   d. $5_Hello

4) Are variable names case-sensitive?
   a. Yes
   b. No

5) Which is not a valid variable scope in PHP?
   a. local
   b. global
   c. static
   d. external

L. References / Suggestions

1) https://www.w3schools.com/php/default.asp
2) https://www.tutorialspoint.com/php/
3) https://tutorialehtml.com/en/php-tutorial-introduction/
4) www.tizag.com/phpT/
5) https://books.goalkicker.com/PHPBook/
6) https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
7) https://codecourse.com/watch/php-basics
8) https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

M. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _____

Practical No.3: Conditional Statements

a. Write a PHP script to check the given number is odd or even.
b. Write a PHP script to print student's grade based on marks of 5 subjects.
c. Create a PHP script to show the month of a year using switch statement.

A. Objective:

Conditional statements are performed on different computations or actions depending on whether Boolean condition evaluates to true or false. Students will be able to use various forms of if statements and switch…case statement to check the condition.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.

Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.

Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"

This practical is expected to develop the following skills.

1. Understand the use of different conditional structures in PHP.
2. Write a PHP script for given problems to use different form of 'if' and execute it.
3. Debug code and solve the errors.

D. Expected Course Outcomes(Cos)

CO1: Create small programs using basic PHP concepts.

E. Practical Outcome(PRo)

Students will be able to write a PHP script for different conditional structures.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Conditional statements are used to perform different actions based on different conditions. PHP also allows you to write code that perform different actions based on the results of a logical or comparative test conditions at run time.

There are several statements in PHP that you can use to make decisions:

- The if statement
- The if...else statement
- The if...elseif....else statement
- The switch...case statement

The if statement

if statement allow us to run a block of code if certain condition is true. If condition is false it will not execute block of code.

Syntax:

```
if(condition) {
    Block of code  // statements to execute if condition is true
}
```

The if...else statement

if...else executes a block of code if certain condition is true and another block of code if condition is false.

Syntax:

```
if(condition) {
    Block of code  // statements to execute if condition is true
}
else {
    Block of code  // statements to execute if condition is false
}
```

The if...elseif....else statement

It is similar to multiple if...else statements. It executes different blocks of code based on different conditions.

Syntax:

```
if(condition) {
    Block of code  // statements to execute if condition is true
}
elseif(condition) {
    Block of code  // statements to execute if condition is true
}
else {
    Block of code          // statements to execute if all conditions are //false
}
```

The switch...case statement

The switch statement is used to perform different actions based on different conditions. Use the switch statement to select one of many blocks of code to be executed.

Syntax:

```
switch(n) {
  case value1:
    code to be executed if n== value1;
    break;
  case value2:
    code to be executed if n== value2;
    break;
  ......
  default:
    code to be executed if n != any case;
}
```

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) Write *a PHP script to check the given number is odd or even.*

A) Input-Output:

B) *Write a PHP script to print student's grade based on marks of 5 subjects.*

B) Input-Output:

C) *Create a PHP script to show the month of a year using switch statement.*

C) Input-Output:

K. Practical related Quiz.

1) Write the output of the following PHP code?

```
<?php
    $x;
    if ($x)
        print "hi" ;
    else
    print "how are u";
?>
```

Output: _____

2) What will be the output of the following PHP code?

```
<?php
    $a = 1;
    if (print $a)
        print "True";
    else
        print "False";
?>
```

Output: _____

3) What will be the output of the following PHP code?

```
<?php
    $a = 10;
    if (0)
        print "all";
        if
        else
            print "some"
        ?>
```

Output: _____

L. References / Suggestions

1) https://www.w3schools.com/php/default.asp
2) https://www.tutorialspoint.com/php/
3) https://tutorialehtml.com/en/php-tutorial-introduction/
4) www.tizag.com/phpT/
5) https://books.goalkicker.com/PHPBook/
6) https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
7) https://codecourse.com/watch/php-basics
8) https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

M. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _____

Practical No.4: Looping Structures

a. Write PHP Script to print Fibonacci series in html tabular format.
b. Write a PHP script to print below number triangle.

```
1
2 3
4 5 6
7 8 9 10
```

c. Write a PHP script to create chess board (tabular structure).

A. Objective:

Looping structure is used in programming to repeat a specific block of code until certain condition is true. Students will be able to use while, do-while and for loop to replace the repetition of statements. Foreach loop is used for iteration within array.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.

Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.

Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"

This practical is expected to develop the following skills.

1. Understand the use of different looping structures in PHP.
2. Write a PHP script for given problems to use different form of looping structure and execute it.
3. Debug code and solve the errors.

D. Expected Course Outcomes(Cos)

CO1: Create small programs using basic PHP concepts.

E. Practical Outcome(PRo)

Students will be able to write a PHP script for different looping structures.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Loops are used to execute the same block of code again and again, as long as a certain condition is met. The basic idea behind a loop is to automate the repetitive tasks within a program to save the time and effort. PHP supports four different types of loops.

- while
- do…while
- for
- foreach

while loop:

while loop executes a block of code as long as the specified condition is true.

Syntax:

```
while(condition) {
Block of code          // statements to execute till condition is true
}
```

do...while loop:

do...while loop executes a block of code once, and then repeats the loop as long as the specified condition is true

Syntax:

```
do {
Block of code          // statements to execute till condition is true.
                       // Executed once before checking condition.
} while(condition);
```

for loop:

for loop executes a block of code a specified number of times.

Syntax:

```
for(init counter; condition; increment/decrement counter)  {
    Block of code          // statements to execute till condition is true
}
```

foreach loop:

foreach loop executes a block of code for each element in an array.

Syntax:

```
foreach ($array as $val)  {
    Block of code  // statements to execute for each element in an array
}
```

break statement:

break statement is used to terminate the execution of a loop prematurely.

Syntax:

```
while(condition1) {
    Block of code          // statements to execute till condition1 is true
    if(condition2) {
            break;         // exit while loop is condition2 is true
    }
}
```

continue statement:

continue statement is used to halt the current iteration of a loop and start next iteration of loop. It does not terminate the loop.

Syntax:

```
while(condition1) {

    if(condition2) {
            continue;      // if condition2 is true then
            //remaining part of the loop  will not be executed in this  iteration
    }
    Block of code  // statements to execute  till condition1 is true
}
```

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Write PHP Script to* print *Fibonacci series in html tabular format.*

A) Input-Output:

B) *Write a PHP script to print below number triangle.*

```
1
2 3
4 5 6
7 8 9 10
```

B) Input-Output:

C) *Write a PHP script to create chess board (tabular structure).*

C) Input-Output:

K. Practical related Quiz.

1) Which of the following loops can be used in PHP?
   a) while
   b) do ... while
   c) for
   d) foreach

   Options:
   1. A and B
   2. A, B, and C
   3. A and C
   4. A, B, C, and D

2) What is the correct output of the given code in PHP?

```
<?php
$iLoop = 1;
while ($iLoop <= 5)
{
   echo $iLoop;
   iLoop++;
}
?>
```

   Options:
   1. 12345
   2. 01234
   3. Error
   4. Infinite loop

3) What is the correct output of the given code in PHP?

```
<?php
    $iLoop = 1;
    while ($iLoop <= 5) echo ++$iLoop;
?>
```

Options:

1. 12345
2. 01234
3. 23456
4. Error

4) Is do… while loop executes at least once?

1. Yes
2. No

L. References / Suggestions

1) https://www.w3schools.com/php/default.asp
2) https://www.guru99.com/php-tutorials.html
3) https://www.tutorialspoint.com/php/
4) https://tutorialehtml.com/en/php-tutorial-introduction/
5) www.tizag.com/phpT/
6) https://books.goalkicker.com/PHPBook/
7) https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
8) https://codecourse.com/watch/php-basics
9) https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

M. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _____

Practical No.5: Arrays

a. Write PHP Script for addition and multiplication of two 2x2 matrices.
b. Write a PHP Script to count the number of elements in an array without using built-in function (use numeric and associative arrays).

A. Objective:

Array in PHP is a type of data structure that allows us to store multiple elements of similar data types under a single variable thereby saving us the effort of creating a different variable for every data. Students will be able to use indexed array, associative array and multidimensional array.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"

This practical is expected to develop the following skills.

1. Understand the use of types of array structures in PHP.
2. Write a PHP script for addition and multiplication of 2x2 matrices.
3. Write a code to count number of elements in an array without using built-in functions of PHP.

D. Expected Course Outcomes(Cos)

CO1: Create small programs using basic PHP concepts.

E. Practical Outcome(PRo)

Students will be able to write PHP script using array data structures.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

PHP In PHP, an array is a variable that can hold multiple values of different data types. Arrays are incredibly useful when you need to work with multiple values of the same type, or when you need to store related data together.
There are two types of arrays in PHP: indexed arrays and associative arrays.

Indexed Arrays

An indexed array is an array where each value is assigned a numeric index, starting from zero. Here's an example of how to create an indexed array in PHP:

```
$dept = array("IT","CE","ME","EE");
```

In this example, the **$dept** array contains five elements, each with a numeric index. You can access the values of an indexed array by referring to their index:

```
echo $dept[0]; // Output: IT
echo $dept[1]; // Output: CE
echo $dept[2]; // Output: ME
```

Associative Arrays

An associative array is an array where each value is assigned a string key. Here's an example of how to create an associative array in PHP:

```
$person = array(
   "name" => "Sachin",
   "age" => 30,
   "location" => "Ahmedabad"
);
```

In this example, the **$person** array contains three elements, each with a string key. You can access the values of an associative array by referring to their key:

```
echo $person["name"];          // Output: Sachin
echo $person["age"];           // Output: 30
echo $person["location"];      // Output: Ahmedabad
```

Multidimensional Arrays

A multidimensional array is an array where each element can also be an array. Here's an example of how to create a multidimensional array in PHP:

```
$students = array(
   array("name" => "Sachin", "age" => 30),
   array("name" => "Saurav", "age" => 31),
   array("name" => "Rahul", "age" => 29)
);
```

In this example, the **$students** array is a multidimensional array containing three elements, each of which is an associative array with two keys (**name** and **age**). You can access the values of a multidimensional array by referring to their indexes and keys:

```
echo $students[0]["name"]; // Output: Sachin
echo $students[1]["age"]; // Output: 21
echo $students[2]["name"]; // Output: Rahul
```

So, arrays are an essential and important part of PHP, and they allow developers to store and manipulate multiple values in a single variable. Indexed arrays are useful for working with numeric data, while associative arrays are useful for working with non-numeric data. Multidimensional arrays are useful for storing complex data structures.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Write PHP Script for addition and multiplication of two 2x2 matrices.*

A) Input/Output:

B) *Write a PHP Script to count the number of elements in an array without using built-in function (use numeric and associative arrays).*

B) Input/Output:

K. Practical related Quiz.

1) PHP's numerically indexed array begin with position _____
2) Which of the following are correct ways of creating an array?
   a. state[0] = "GUJARAT";
   b. $state[] = array("GUJARAT");
   c. $state[0] = "GUJARAT";
   d. $state = array("GUJARAT");
3) _____ and _____ are types of array in PHP.
4) In associative array element value is assigned to _____ .
5) What will be the output of the following PHP code?

L. References / Suggestions

1) https://www.w3schools.com/php/default.asp
2) https://www.guru99.com/ php-tutorials.html
3) https://www.tutorialspoint.com/php/
4) https://tutorialehtml.com/en/php-tutorial-introduction/
5) www.tizag.com/phpT/
6) https://books.goalkicker.com/PHPBook/
7) https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
8) https://codecourse.com/watch/php-basics
9) https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

M. Assessment-Rubrics

Marks Obtained				Faculty Signature	Date
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _____

Practical No.6: User defined Functions

a. Write a PHP script to call by reference and call by value.
b. Write a PHP Script for performing function that takes arguments with default argument and returns value.
c. Write a PHP Script to show the use of variable length argument.

A. Objective:

A function is a block of reusable code that is used to perform a specific task. Functions enhances the readability of a program, Reduce duplication of the code, reduces the complexity of a program. User-defined functions help to decompose a large program into small segments which makes program easy to understand, maintain and debug. This practical will help student to practice writing PHP scripts using user defined functions.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"

This practical is expected to develop the following skills.

1. Understand the use of calling function by value and by reference.
2. Write PHP script for functions having return value, default argument, variable length argument etc.
3. Follow coding standards and debug program to fix errors.

D. Expected Course Outcomes(Cos)

CO2: Create User defined functions in PHP programming.

E. Practical Outcome(PRo)

Students will be able to create user defined functions for different tasks.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

User defined functions

User-defined functions can be used to group code into reusable blocks and make the code more modular and easier to maintain.

Syntax:

```
function function_name(arg1, arg2, ..., argN)
{
   code to be executed
}
```

Here,

- **function**: This is the keyword that tells PHP that you are defining a function.
- **function_name**: This is the name of the function.
- **(arg1, arg2, ..., argN)**: These are the arguments that the function accepts. Arguments are optional, and you can define as many or as few as you need.
- **code to be executed**: This is the code that the function executes when it is called.

```
<?php

  function test()
    {
    echo "Welcome Students.";
  }

  test();
?>
```

O/P:
Welcome Students.

Function with parameters

```
<?php

  function sum($num1, $num2)
    {
          $sum = $num1 + $num2;
          echo "Sum of the two numbers is: $sum";
  }

  sum(10,20);
?>
```

O/P:
Sum of the two numbers is: 30

Function with returning value

```
<?php

  function sum($num1, $num2)
    {
          $sum = $num1 + $num2;
          return $sum;
  }

    $ret_sum = sum(10,20);
    echo "Returned value from function is: $sum";
?>
```

O/P:
Returned value from function is: 30

Function with default argument

```
<?php

  function sum($num1, $num2 = 5)
    {
          $sum = $num1 + $num2;
          echo "Sum of the two numbers is: $sum <br/>";
    }

    sum(10);
    sum(10,20);
?>
```

O/P:

Sum of the two numbers is: 15
Sum of the two numbers is: 30

Variable length argument function

Normally when we create function, the number of arguments are created according to requirement of function. Also we pass argument in function calling according to argument of the function. But in PHP, we have facility to call function according to arguments. We can pass arguments to a function that has no argument defined. PHP allows to access arguments provided in the function call operation without using argument variable in function declaration.

PHP supports variable-length argument with below basic rules :

- The function call can provide more arguments than the number of arguments defined in function declaration.
- We can pass arguments to a function that has no argument variable defined.

To support variable length arguments, there are predefined functions :

- func_num_args()
- func_get_args()
- func_get_arg()

func_num_args()

It returns the total number of arguments provided in the function call operation.

```
<?php
  function sum()
    {
        echo func_num_args();
  }
    sum(10,20,30,40);
?>
```

O/P: 4

func_get_args()

It creates and returns an array which contains values of arguments provided in the function call operation.

```
<?php
  function dept()
    {
        $arr = func_get_args();
        Print_r($arr);
  }
    dept("IT", "CE", "ME");
?>
```

O/P: Array ([0] => "IT" [1] => "CE" [2] => "ME")

func_get_arg(int position)

It returns the value of specified argument provided in the function call operation. Here 'position' is the position index of specified argument.

```
<?php
  function dept()
    {
        echo func_get_arg(2);
  }
    dept("IT", "CE", "ME");
?>
```

O/P: ME

Function call by value

In this method, only values of actual parameters are passing to the function. So there are two addresses stored in memory. Making changes in the passing parameter does not affect the actual parameter.

```
<?php
  function incr($val)
    {
        $val = $val + 1;
        echo $val . "<br/>";
  }
    $a = 1;
    incr($a);
    echo $a;
?>
```

O/P: 2

1

Function call by reference

In this method, the address of actual parameters is passing to the function. So any change made by function affects actual parameter value.

```
<?php
  function incr(&$val)
    {
        $val = $val + 1;
        echo $val . "<br/>";
  }
    $a = 1;
    incr($a);
    echo $a;
?>
```

O/P: 2

2

H. Resources/Equipment Required

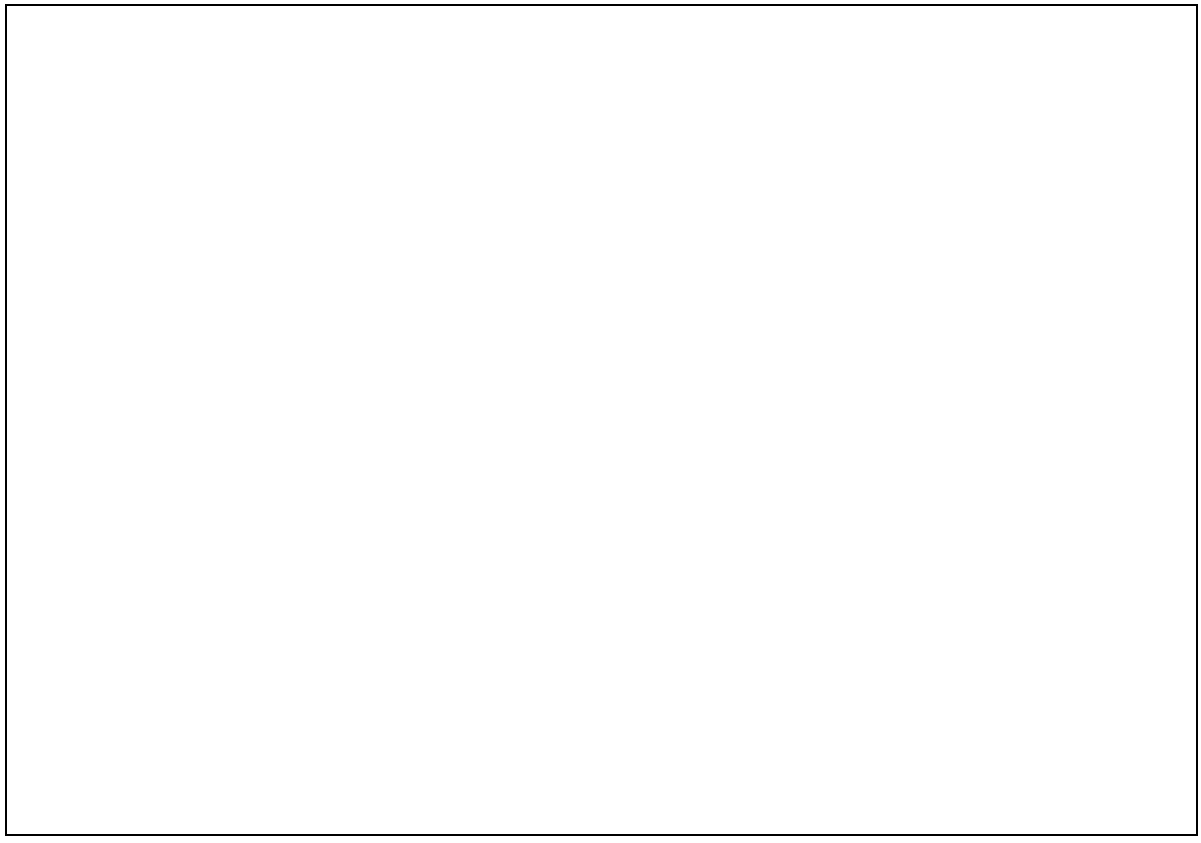
Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

I. Safety and necessary Precautions followed

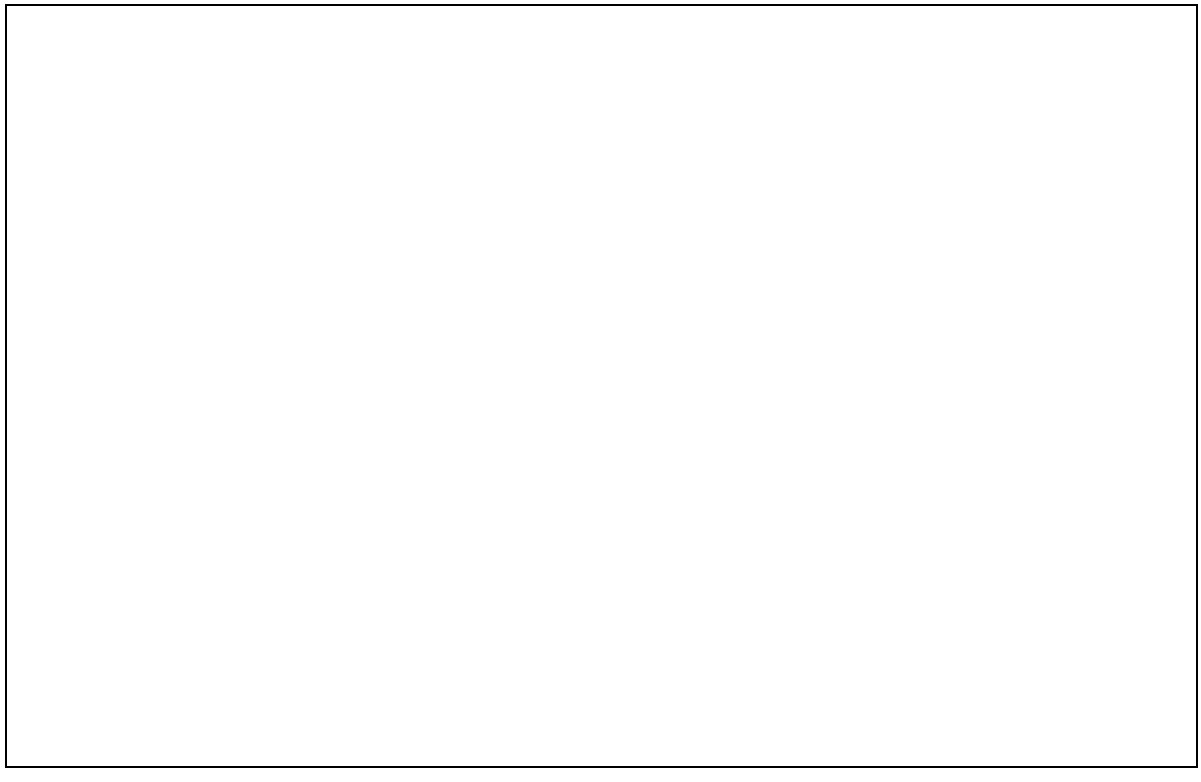
NA

J. Source code:

A) *Write a PHP script to call by reference and call by value.*



A) Input/Output:



B) *Write a PHP Script for performing function that takes arguments with default argument and returns value.*

B) Input/Output:

C) *Write a PHP Script to show the use of variable length argument.*

C) Input/Output:

K. Practical related Quiz.

1) How to define a function in PHP?

a) function {function body}

b) data type functionName(parameters) {function body}

c) functionName(parameters) {function body}

d) function functionName(parameters) {function body}

2) Which type of function call is used in line 8 in the following PHP code?

```
<?php
    function calc($price, $tax)
    {
       $total = $price + $tax;
    }
    $pricetag = 15;
    $taxtag = 3;
    calc($pricetag, $taxtag);
 ?>
```

a) Call By Value

b) Call By Reference

c) Default Argument Value

d) Type Hinting

3) What will be the output of the following PHP code?

```
<?php
   function calc($price, $tax="")
   {
      $total = $price + ($price * $tax);
      echo "$total";
   }
   calc(42);
?>
```

a) Error
b) 0
c) 42
d) 84

4) A function name cannot start with a ____

a) Alphabet
b) Underscore
c) Number
d) Both c and b

5) Function names are case-sensitive.

a) True
b) False
c) Only Built-In function
d) Only User-defined function.

K. References / Suggestions

1) https://www.w3schools.com/php/default.asp
2) https://www.guru99.com/php-tutorials.html
3) https://www.tutorialspoint.com/php/
4) https://tutorialehtml.com/en/php-tutorial-introduction/
5) www.tizag.com/phpT/
6) https://books.goalkicker.com/PHPBook/
7) https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
8) https://codecourse.com/watch/php-basics
9) https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

L. Assessment-Rubrics

Marks Obtained				Faculty Signature	Date
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _____

Practical No.7: Built-In functions

a. Write PHP script to demonstrate the use of various strings handling function.
b. Write a PHP script to demonstrate the use of Include() and require() function.
c. Write PHP script to demonstrate the use of Array functions.
d. Write PHP script to demonstrate the use of fopen(), fread(), fwrite() and fclose() file functions.

A. Objective:

Built-in functions are predefined functions in PHP that exist in the PHP library. These PHP inbuilt functions make PHP a very efficient and productive scripting language. The built in functions of PHP classified into many categories. Students will be able to learn various built-in functions like string functions, array functions, file functions etc.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"

This practical is expected to develop the following skills.

1. Write PHP script for various string handling functions.
2. Write PHP script to demonstrate Include() and Require() functions.
3. Write PHP script for various array and file functions.
4. Follow coding standards and debug program to fix errors.

D. Expected Course Outcomes(Cos)

CO2: Create User defined functions in PHP programming.

E. Practical Outcome(PRo)

Students will be able to use different built-in functions of string, file and array.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

PHP String Functions

echo	
Use:	To output/display one or more strings or variables to the screen. Note: The echo() function is not actually a function, so you are not required to use parentheses with it. However, if you want to pass more than one parameter to echo(), using parentheses will generate an error.
Syntax:	echo(*strings*)
Example:	<?php echo "Hello world"; ?>
Output:	Hello world

print	
Use:	To output/display one or more strings or variables to the screen. Note: The print() function is not actually a function, so you are not required to use parentheses with it.
Syntax:	print(*strings*)
Example:	<?php print "Hello world"; ?>
Output:	Hello world

strlen()	
Use:	To find the length of strings.
Return:	The length of a string.
Syntax:	strlen(String)

Example:	<?php echo strlen("Hello world"); ?>
Output:	11

str_word_count()	
Use:	To count the words of strings.
Return:	The number of words in the string.
Syntax:	str_word_count(String)
Example:	<?php echo str_word_count("Hello world"); ?>
Output:	2

strrev()	
Use:	To reverse the given strings.
Return:	The reversed string.
Syntax:	strrev(String)
Example:	<?php echo strrev("Hello world"); ?>
Output:	dlrow olleH

str_replace()	
Use:	To replace some of the characters in a string with some other.
Return:	The string with replacements.
Syntax:	str_replace(search, replace, string) Here; Search: Required. This will be the string to search for replacing. Replace: Required. This will replace the searched string. String: Required. This will the string which we have to search and replace.
Example:	<?php echo str_replace("Hello","Hi","Hello world"); ?>
Output:	Hi world

strstr()	
Use:	To display the part of the string from the first search occurrence. Note: This function is case-sensitive. For a case-insensitive search, use stristr() function.
Return:	The rest of the string.
Syntax:	strstr(string,search,before_search)

	Here; String: Required. This will be the string to search. Search: Required. Specifies the string to search for. If this is a number, it will search for the character matching the ASCII value of the number. Before_search: Optional. A boolean value whose default is "false". If set to "true", it returns the part of the string before the first occurrence of the search parameter.
Example:	<?php echo strstr("Hello world","world") . " "; echo strstr("Hello world","world",true); ?>
Output:	world Hello

substr()	
Use:	To extract the string.
Return:	The a part of the string.
Syntax:	substr(string,start,length) Here; String: Required. This will be the string to return a part of. Start: Required. Specifies where to start in the string - A positive number - Start at a specified position in the string - A negative number - Start at a specified position from the end of the string - 0 - Start at the first character in string Length: Optional. Specifies the length of the returned string. - A positive number - The length to be returned from the start - Negative number - The length to be returned from the end - If the length parameter is 0, NULL, or FALSE - returns empty string
Example:	<?php echo substr("Hello world",7) ." "; echo substr("Hello world",-7) ." "; echo substr("Hello world",7,2) ." "; echo substr("Hello world",1,-7) ." "; echo substr("Hello world",-7,-1); ?>
Output:	orld o world or ell o worl

strtolower()

Use:	To extract the string.
Return:	String in lower case.
Syntax:	strtolower(string)
Example:	<?php echo strtolower("Hello World"); ?>
Output:	hello world

strtoupper()	
Use:	To extract the string.
Return:	String in upper case.
Syntax:	strtoupper(string)
Example:	<?php echo strtoupper("Hello World"); ?>
Output:	HELLO WORLD

ucwords()	
Use:	To convert the first character of each word in a string to uppercase.
Return:	String with first character of each word in uppercase.
Syntax:	ucwords(string, delimiters) Here; String: Required. This will be the string to convert. Delimiters: Optional. Word separator character.
Example:	<?php echo ucwords("hello world") ." "; echo ucwords("hello-world", "-"); ?>
Output:	Hello World Hello-World

strpos()	
Use:	To find the position of the first occurrence of a string inside another string. Note: The strpos() function is case-sensitive.
Return:	The position of the first occurrence of a string inside another string, or FALSE if the string is not found. (String positions start at 0).
Syntax:	strpos(string, find, start) Here; String: Required. This will be the string to search. Find: Required. This will be the string to find. Start: Optional. Specifies where to begin the search. (If start is a negative number, it counts from the end of the string.)

Example:	<?php echo ucwords("hello world") ." "; echo ucwords("hello-world", "-"); ?>
Output:	Hello World Hello-World

ltrim()	
Use:	To remove whitespace and other predefined characters from left side of a string.
Return:	String after removing of space/characters.
Syntax:	ltrim(string, charlist) Here; String: Required. This will be the string to check. Charlist: Optional. This will be characters to remove from the string.
Example:	<?php echo ltrim("Hello world","Held"); ?>
Output:	o world

rtrim()	
Use:	To remove whitespace and other predefined characters from right side of a string.
Return:	String after removing of space/characters.
Syntax:	rtrim(string, charlist) Here; String: Required. This will be the string to check. Charlist: Optional. This will be characters to remove from the string.
Example:	<?php echo rtrim("Hello world","Held"); ?>
Output:	Hello wor

trim()	
Use:	To remove whitespace and other predefined characters from both sides of a string.
Return:	String after removing of space/characters.
Syntax:	trim(string, charlist) Here; String: Required. This will be the string to check. Charlist: Optional. This will be characters to remove from the string.
Example:	<?php

	echo trim("Hello world","Held"); ?>
Output:	o wor

implode()	
Use:	To convert the array to strings. It takes an array and converts those array to strings by using any separator.
Return:	String from elements of an array.
Syntax:	implode(separator, array) Here; Separator: Optional. This will be the separator to put between the array elements. Default is "" (an empty string). Array: Required. This will be an array.
Example:	<?php $arr = array('Hello','World'); echo implode(" ",$arr) ." "; echo implode("-",$arr); ?>
Output:	Hello World Hello-World

explode()	
Use:	To break a string into an array.
Return:	An array of strings.
Syntax:	explode(separator, string, limit) Here; Separator: Required. This will be the for where to break the string. String: Required. This will be string to split. Limit: This will be the number of array elements to return. limit can be, >0 - Returns an array with a maximum of limit element(s) <0 - Returns an array except for the last -limit element(s) 0 - Returns an array with one element

Example:	<?php
	$str = "Welcome to the PHP world.";
	print_r (explode(" ",$str));
	echo "
	";
	print_r (explode(" ",$str,0));
	echo "
	";
	print_r (explode(" ",$str,3));
	echo "
	";
	print_r (explode(" ",$str,-1));
	?>

Output:	Array ([0] => Welcome [1] => to [2] => the [3] => PHP [4] => world.) Array ([0] => Welcome to the PHP world.) Array ([0] => Welcome [1] => to [2] => the PHP world.) Array ([0] => Welcome [1] => to [2] => the [3] => PHP)

PHP include() and require() function

In PHP, the include() and require() functions are used to include and execute the contents of another PHP file within the current PHP file. This is useful when you want to reuse code across multiple files, or when you want to break up a large PHP file into smaller, more manageable pieces.

While using include() if there are any kind of errors then this include() function will pop up a warning but, it will not stop the execution of the script rather the script will continue its process.

While using require() if there are any kind of errors then this require() function will pop up a warning along with a fatal error and it will immediately stop the execution of the script.

Syntax:

```
include 'filename';
or
require 'filename';
```

Let's assume that we have a file named 'two.php' and we want to include it in our page named 'one.php'.

two.php:

```
<?php
echo "<p>Thank  you for visiting our website.</p>";
?>
```

one.php:

```
<html>
<body>

<h1>Welcome to the PHP world!</h1>
<p>This is demo of include/require() functions.</p>
<p>Let's test it.</p>
<?php include 'two.php';?>

</body>
</html>
```

Output:

Welcome to the PHP world!

This is demo of include/require() functions..
Let's test it.

Thank you for visiting our website.

Same way you can use require() function and test it.

PHP Array Functions

in_array()	
Use:	To search an array for a specific value.
Return:	TRUE if the value is found in the array, or FALSE otherwise
Syntax:	in_array(search, array, type) Here; Search: Required. This will be the what to search for. Array: Required. This will be array to search. Type: Optional. If this is set to TRUE, it searches for the search-string and specific type in the array.

Example:	<?php
	$dept = array("IT", "CE", "ME", 16);
	if (in_array("16", $dept, TRUE)) {
	echo "Match found
	";
	}
	else {
	echo "Match not found
	";
	}
	if (in_array("IT",$dept)) {
	echo "Match found
	";
	}
	else {
	echo "Match not found
	";
	}
	?>

Output:	Match not found Match found

in_array()	
Use:	To merge one or more arrays into one array.
Return:	Merged array.
Syntax:	array_merge(array1, array2, …., arrayN)
Example:	<?php $a1=array("IT","CE"); $a2=array("ME","EE"); print_r(array_merge($a1,$a2)); ?>
Output:	Array ([0] => IT [1] => CE [2] => ME [3] => EE)

array_push()	
Use:	To insert one or more elements to the end of an array.

Return:	The new number of elements in the array.
Syntax:	array_push(array, value1, value2, ….)

Example:	<?php
	$a=array("IT","CE");
	array_push($a,"CE");
	print_r($a);
	?>

Output:	Array ([0] => IT [1] => CE [2] => ME)

array_pop()	
Use:	To delete the last element of an array.
Return:	The new array after deletion.
Syntax:	array_pop(array)

Example:	<?php
	$a=array("IT","CE","ME");
	array_pop($a);
	print_r($a);
	?>

Output:	Array ([0] => IT [1] => CE)

array_replace()	
Use:	To replace the values of the first array with the values from following arrays.
Return:	The replaced array, or NULL if an error occurs.
Syntax:	array_replace(array1, array2, …)
Example:	<?php $a1=array("IT","CE"); $a1=array("EE","ME"); print_r(array_replace($a1,$a2)); ?>
Output:	Array ([0] => EE [1] => ME)

array_reverse()	
Use:	To reverse the specified arrays.
Return:	The reversed array.
Syntax:	array_reverse(array)
Example:	<?php $a=array("IT","CE"); print_r(array_reverse($a)); ?>
Output:	Array ([0] => CE [1] => IT)

array_search()	
Use:	To search an array for a value.
Return:	The key of a value if it is found in the array, and FALSE otherwise.

Syntax:	array_search(value, array, strict) Here; Value: Required. This will be the value to search for. Array: Required. This will be array to search. Strict: Optional. If this parameter is set to TRUE, then this function will search for identical elements in the array. Default is FALSE.
Example:	<?php $a=array("a"=>"IT","b"=>"CE"); echo array_search("IT",$a); ?>
Output:	a

count()	
Use:	To count of elements in an array.
Return:	The number of elements in an array.
Syntax:	count(array, mode) Here; Array: Required. This will be an array. Mode: Optional. It will be 0 or 1. Default is 0 and it does not count all elements of multidimensional arrays. If it is 1 then it counts the array recursively (counts all the elements of multidimensional arrays).
Example:	<?php $a=array("IT","CE"); echo (count($a)); ?>
Output:	2

current(), next(), prev(), reset(), end()	
Use:	To get appropriate element from array (current, next, previous etc).
Return:	current() - The current element of an array. next() - The next element of an array. (internal pointer move to here) prev() - The previous element of an array. (pointer move to here) reset() - The first element of an array. (internal pointer move to first) end() - The last element of an array. (internal pointer move to last)
Syntax:	current(array), next(array), prev(array), reset(array), end(array)
Example:	<?php $dept = array("IT", "CE", "ME", "EE"); echo current($dept) . " "; echo next($dept) . " "; echo current($dept) . " "; echo prev($dept) . " "; echo end($dept) . " "; echo prev($dept) . " ";

	echo current($dept) . " "; echo reset($dept) . " "; echo next($dept); ?>
Output:	IT CE CE IT EE ME ME IT CE

list()	
Use:	To assign values to a list of variables in one operation.
Return:	The assigned array.
Syntax:	list(var1, var2, var3, ...)
Example:	<?php $arr = array("IT","CE"); list($a, $b) = $arr; echo "Our departments are $a and $b."; ?>
Output:	Our departments are IT and CE.

sort()	
Use:	To sort an indexed array in ascending order.
Return:	TRUE on success. FALSE on failure
Syntax:	sort(array)

Example:	<?php
	$arr=array("IT","CE","ME","EE");
	sort($arr);
	$alen=count($arr);
	for($x=0;$x<$alen;$x++)
	{
	echo $arr[$x] . "
	";
	}
	?>

Output:	CE EE IT ME

PHP File Functions

fopen()	
Use:	To open a file or URL.
Return:	A file pointer resource on success, FALSE and an error on failure.
Syntax:	fopen(filename, mode) Here; Array: Required. This will be file or URL to open. Mode: Required. This will be the type of access you require to the file. Modes can be r, r+, w, w+, a, a+, x, x+, c, c+.

fclose()	
Use:	To close a file.
Return:	TRUE on success, FALSE on failure.
Syntax:	fclose(filepointer)

fread()	
Use:	To read from an open file.
Return:	A file pointer resource on success, FALSE and an error on failure.
Syntax:	fread(file, length) Here; File: Required. This will be file to read from. Length: Required. This will be the maximum number of bytes to read.

Example:	<?php
	$file = fopen("test.txt","r");
	fread($file,"5");
	fclose($file);
	?>

fwrite()	
Use:	To write to an open file.
Return:	The number of bytes written, FALSE on failure.
Syntax:	fwrite(file, string, length) Here; File: Required. This will be file to write to. string: Required. This will be string to write in file. Length: Optional. This will be maximum number of bytes to write.
Example:	<?php $file = fopen("test.txt","w"); fwrite($file,"Hello World") fclose($file); ?>
Output:	11

PHP Variable Functions

gettype()	gettype(variable name);	It accepts variable as an argument and returns the data type of that variable.
settype()	settype(Variable name, Data type);	It accepts a variable as an argument and set it to specific data type passed as an argument.
isset()	isset(Variable Name);	It accepts a variable as an argument and determines whether the variable exists and it is assigned value or not.
unset()	unset(Variable Name);	It accepts a variable as an argument and destroys that variable from memory.
strval()	strval(variable name);	It accepts variable as an argument and returns the string value of that variable.
floatval()	floatval(variable name);	It accepts variable as an argument and returns the Float value of that variable.
intval()	intval(variable name);	It accepts variable as an argument and returns the Integer value of that variable.
print_r()	print_r(variable name);	It accepts variable as an argument and display it in a human readable format.

PHP Math Functions

abs()	abs(Number);	Accepts numbers an argument and Returns the absolute value of a number.
ceil()	ceil(Number);	Accepts number as an argument and Returns the number which is rounded upwards to the nearest integer value.
floor()	floor(Number);	Accepts number as an argument and Returns the number which is rounded downwards to the nearest integer value.
round()	round(Number [,Precision]);	Accepts number as an argument and Returns the number rounded to the nearest integer.
fmod()	fmod(Number 1,Number2);	Accepts two numbers as an argument and divides num1 by num2 and returns reminder of division.
min()	min(Number 1,Number2);	Accepts two numbers as an argument and returns lowest value among them.
max()	max(Number 1,Number2);	Accepts two numbers as an argument and returns highest value among them.
pow()	pow(Number 1,Number2);	Accepts two numbers as an argument and raises num1 to the power of num2 and returns result.
sqrt()	sqrt(Number);	Accepts a number as an argument and Returns square root of a number.
rand()	rand([min],[max]);	Generate a random integer between the ranges of 0 to ROUND_MAX.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Write PHP script to demonstrate the use of various strings handling function.*

B) Input/Output:

C) *Write a PHP script to demonstrate the use of Include() and require() function.*

D) Input/Output:

E) *Write PHP script to demonstrate the use of Array functions.*

F) Input/Output:

G) *Write PHP script to demonstrate the use of fopen(), fread(), fwrite() and fclose() file functions.*

H) Input/Output:

K. Practical related Quiz.

1. Write difference between echo and print.

_____

_____

_____

_____

_____

2. _____ function is used for replacing the whole string with an alternate string.

3. A function that capitalizes the first letter of each word in a string, is a _____.

4. _____ in-built function will add a element to the end of an array.

5. _____ function is used to get the value of the previous element in an array.

6. What will be the output of following code? _____

```
<?php
        $a=array("a"=>"IT","b"=>array("CE"));
        echo (count($a,1));
?>
```

L. References / Suggestions

1) https://www.w3schools.com/php/default.asp
2) https://www.guru99.com/php-tutorials.html
3) https://www.tutorialspoint.com/php/
4) https://tutorialehtml.com/en/php-tutorial-introduction/
5) www.tizag.com/phpT/
6) https://books.goalkicker.com/PHPBook/
7) https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
8) https://codecourse.com/watch/php-basics
9) https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

M. Assessment-Rubrics

Marks Obtained				Faculty Signature	Date
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _____

Practical No.8: Form Handling

a. Create student registration form using text box, check box, radio button, select, submit button. And display user inserted value in new PHP page using GET or POST Method.
b. Write a PHP script to explain the concept of $_REQUEST.

A. Objective:

Forms are important component of the web application that allows to collect information from the users. Forms are used for various tasks such as login, registration, contact us, and application specific information collection. This practical will help students to design a form and to collect data entered in form input by users using PHP.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"

This practical is expected to develop the following skills.

Create student registration form using tags.

1) Understand the use of Get and Post method of form.
2) Apply $_GET[], $_POST[] and $_REQUEST for collecting user inputs.
3) Follow coding standards and debug program to fix errors.

D. Expected Course Outcomes(Cos)

CO3: Design and develop a Web site using form controls for presenting web-based content.

E. Practical Outcome(PRo)

Students will be able to use a form and to collect data entered in form input by users using PHP.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

From Tag

It is used to design form which contains various input controls that allows user to input information.

Input Element

It allows you to place various controls in the form to accept input from user.

Sr. No.	Type	Control
1	Text	Textbox
2	Hidden	Hidden Textbox
3	Password	Textbox with Password
4	Submit	Submit Button
5	Checkbox	Check Box
6	Radio	Radio Button
7	Reset	Reset Button
8	File	File Upload
9	Button	Button
10	Image	Image button

→ Textbox Syntax:-

```
<input type="text" name="Text Name" value="Default value">
```

→ Hidden field Syntax:-

```
<input type="hidden" name="field Name" value="Default value">
```

→ Password Syntax:-

```
<input type="password" name="psw Name" value="Default value">
```

→ TextArea Syntax:-

```
<textarea name ="Text name" rows="rowsize" cols="columnsize">
```

→ checkbox Syntax:-

```
<input type="checkbox" name="checkBoxName" value="Default value"
checked> TextToDisplay </input>
```

→ radio Button Syntax:-

```
<input type="radio" name="radioname" value="Default value"
checked> TextToDisplay </input>
```

→ List box Syntax:-

```
<select name="List name" size="value">
<option>Value1</option>
<option>Value1</option>
</select>
```

→ Submit Button Syntax:-

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

→ Image Button Syntax:-

```
<input type="image" SRC="URL">
```

Submit form using GET method:

- It will pass variable through URL.
- In this Method Information will be sent to destination file through URL using concept of Query String.
- The GET method is restricted to send up to 1024 characters only.
- Never use GET method if you have password or other sensitive information to be sent to the server.
- GET can't be used to send binary data, like images or word documents, to the server.
- The PHP provides **$_GET** associative array to access all the sent information using GET method.

Syntax:

```
$VariableName = $_GET["fieldname"];
```

Disadvantages:-

- Information trying to pass to destination file is visible in URL so it is Insecure.
- Transfer limited amount of Information.
- GET can't be used to send binary data, like images or word documents, to the server.
- The GET method is restricted to send up to 1024 characters only.

Submit form using POST method:

- The POST method transfers information via HTTP headers.
- It will transfer Information between pages through FORM body.
- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
- The PHP provides $_POST associative array to access all the sent information using POST method
- Information is transferred through FORM body, not visible to everyone. Hence secure method.
- It allows you to transfer larger amount of data.

Syntax:

$VariableName = $_POST[“fieldname”];

Using $_Request method:

- $_REQUEST is a super global variable which is widely used to collect data after submitting html forms.
- $_REQUEST method is used to retrieve value from URL as well as FORM collection.
- It is used to get the result from form data sent with both the GET and the POST methods.
- PHP provides the super global variable $_REQUEST that contains the contents of both the $_GET and $_POST variables as well as the values of the $_COOKIE variable.

Syntax:

$VariableName = $_REQUEST[“fieldname”];

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

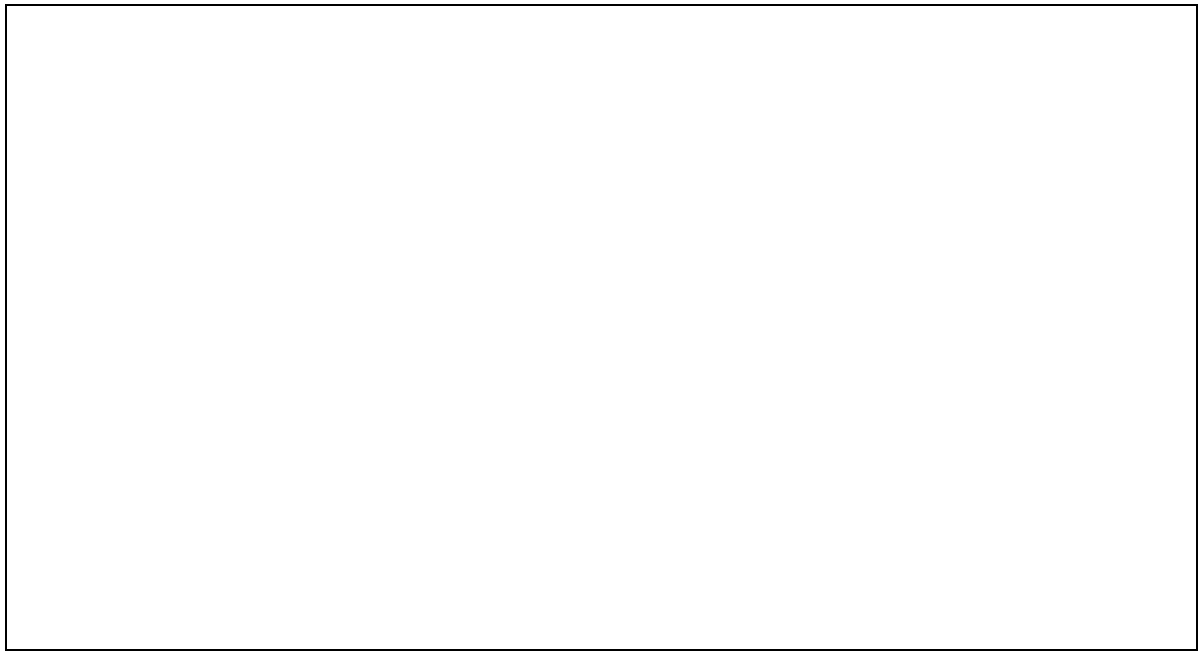
I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Create student* registration *form using text box, check box, radio button, select, submit button. And display user inserted value in new PHP page using GET or POST Method.*

A) Input/Output:



B) *Write a PHP script to explain the concept of $_REQUEST.*

B) Input/Output:

K. Practical related Quiz.

1. _____ variable is used to collect form data sent with both the GET and POST methods.
2. _____ should not be used while sending passwords or other sensitive information.
3. Write any one difference between get and post method.

   _____

   _____

   _____

   _____

   _____

   _____

4. Write attributes of <form> tag.

   _____

   _____

   _____

5. For password input in form, you should use _____ attribute and _____ value of <input> tag.

L. References / Suggestions

1) https://www.w3schools.com/php/default.asp
2) https://www.guru99.com/php-tutorials.html
3) https://www.tutorialspoint.com/php/
4) https://tutorialehtml.com/en/php-tutorial-introduction/
5) www.tizag.com/phpT/
6) https://books.goalkicker.com/PHPBook/
7) https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
8) https://codecourse.com/watch/php-basics
9) https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

M. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _____

Practical No.9: Email and Validation

a. Write PHP script to validate form including name, email using appropriate functions.
b. Write PHP script for sending plain text email, HTML email and attachments with email.

A. Objective:

Forms validation and sending email are important if we talk about input of form. By practical students will be able to learn from validation and sending email details from form.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"

This practical is expected to develop the following skills.

1. Apply validation to name, email using appropriate function.
2. Create script for sending email, HTML email and attachments with email.
3. Follow coding standards and debug program to fix errors.

D. Expected Course Outcomes(Cos)

CO2: Create User defined functions in PHP programming.

E. Practical Outcome(PRo)

Students will be able to perform validation and to email data entered in form by users using PHP.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Form Validation

Form validation in PHP is the process of verifying that the data submitted by a user through a web form is valid, complete, and meets the required format. An HTML form contains various input fields such as text box, checkbox, radio buttons, submit button, and checklist, etc. These input fields need to be validated, which ensures that the user has entered information in all the required fields and also validates that the information provided by the user is valid and correct.

PHP validates the data at the server-side, which is submitted by HTML form. You need to validate a few things:

Empty String

The code below checks that the field is not empty. If the user leaves the required field empty, it will show an error message. Put these lines of code to validate the required field.

```
if (empty ($_POST["name"])) {
   $errMsg = "Error! Kindly Enter the Name.";
         echo $errMsg;
} else {
   $name = $_POST["name"];
}
```

Validate String

The code below checks that the field will contain only alphabets and whitespace, for example - name. If the name field does not receive valid input from the user, then it will show an error message:

```
$name = $_POST ["Name"];
if (!preg_match ("/^[a-zA-z]*$/", $name) ) {
   $ErrMsg = "Only alphabets and whitespace are allowed.";
         echo $ErrMsg;
} else {
   echo $name;
}
```

Validate Numbers

The below code validates that the field will only contain a numeric value. For example - Mobile no. If the Mobile no field does not receive numeric data from the user, the code will display an error message:

```
$mobileno = $_POST ["Mobile_no"];
if (!preg_match ("/^[0-9]*$/", $mobileno) ){
   $ErrMsg = "Only numeric value is allowed.";
   echo $ErrMsg;
} else {
   echo $mobileno;
}
```

Validate Email

A valid email must contain @ and . symbols. PHP provides various methods to validate the email address. Here, we will use regular expressions to validate the email address. The below code validates the email address provided by the user through HTML form. If the field does not contain a valid email address, then the code will display an error message:

```
$email = $_POST ["Email"];
$pattern = "^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-
z]{2,3})$^";
if (!preg_match ($pattern, $email) ){
   $ErrMsg = "Email is not valid.";
         echo $ErrMsg;
} else {
   echo "Your valid email address is: " .$email;
}
```

Validate URL

The below code validates the URL of website provided by the user via HTML form. If the field does not contain a valid URL, the code will display an error message, i.e., "URL is not valid".

```
$websiteURL = $_POST["website"];
if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~_|!:,.;]*[-
a-z0-9+&@#\/%=~_|]/i",$website)) {
  $websiteErr = "URL is not valid";
  echo $websiteErr;
} else {
   echo "Website URL is: " .$websiteURL;
}
```

Input length

The input length validation restricts the user to provide the value between the specified range, for Example - Mobile Number. A valid mobile number must have 10 digits. The given code will help you to apply the length validation on user input:

```
$mobileno = strlen ($_POST ["Mobile"]);
$length = strlen ($mobileno);
if ( $length < 10 && $length > 10) {
   $ErrMsg = "Mobile must have 10 digits.";
        echo $ErrMsg;
} else {
   echo "Your Mobile number is: " .$mobileno;
}
```

Sending Email in PHP

Use the mail() function in PHP to send the email. The mail() function requires at least three parameters: the recipient email address, the subject, and the message body.

Syntax:

```
mail(to, subject, message, headers, parameters);
```

Here,
To: Required. Specifies receiver or receivers of the mail.
Subject: Required. Specifies the subject of the email.
Message: Required. Defines the message to be sent.
Headers: Optional. Specifies additional headers, like From, Cc, and Bcc.
Parameters: Optional. Specifies an additional parameter to the sendmail program

Sending plain text email

```
<?php
$to = "receiver@example.com";
$subject = "My subject";
$txt = "Hello world!";
$headers = "From: sender@example.com" . "\r\n" .
"CC: somebodyelse@example.com";

mail($to,$subject,$txt,$headers);
?>
```

Sending HTML email

```
<?php
  $to = "receiver@example.com";
  $subject = "This is subject";
  $message = "<h1>This is HTML Email</h1> \r\n";
  $message .= "<h1>This is HTML Heading.</h1>";

  $header = "From:sender@example.com \r\n";
  $header .= "MIME-Version: 1.0 \r\n";
  $header .= "Content-type: text/html;charset=UTF-8 \r\n";

  $result = mail ($to,$subject,$message,$header);
  if( $result == true ){
    echo "Message sent successfully...";
  }else{
    echo "Sorry, unable to send mail...";
  }
?>
```

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

I. Safety and necessary Precautions followed

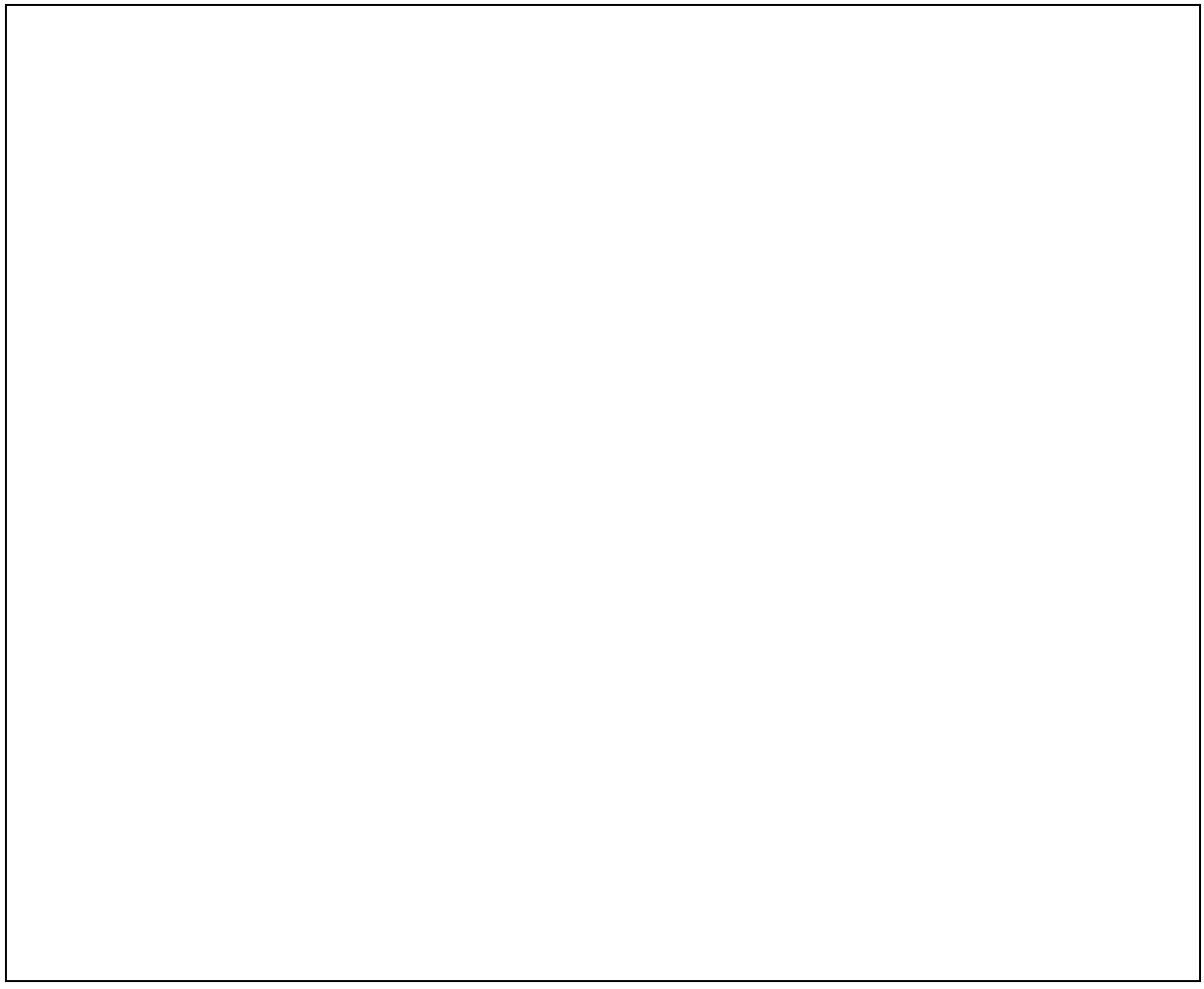
NA

J. Source code:

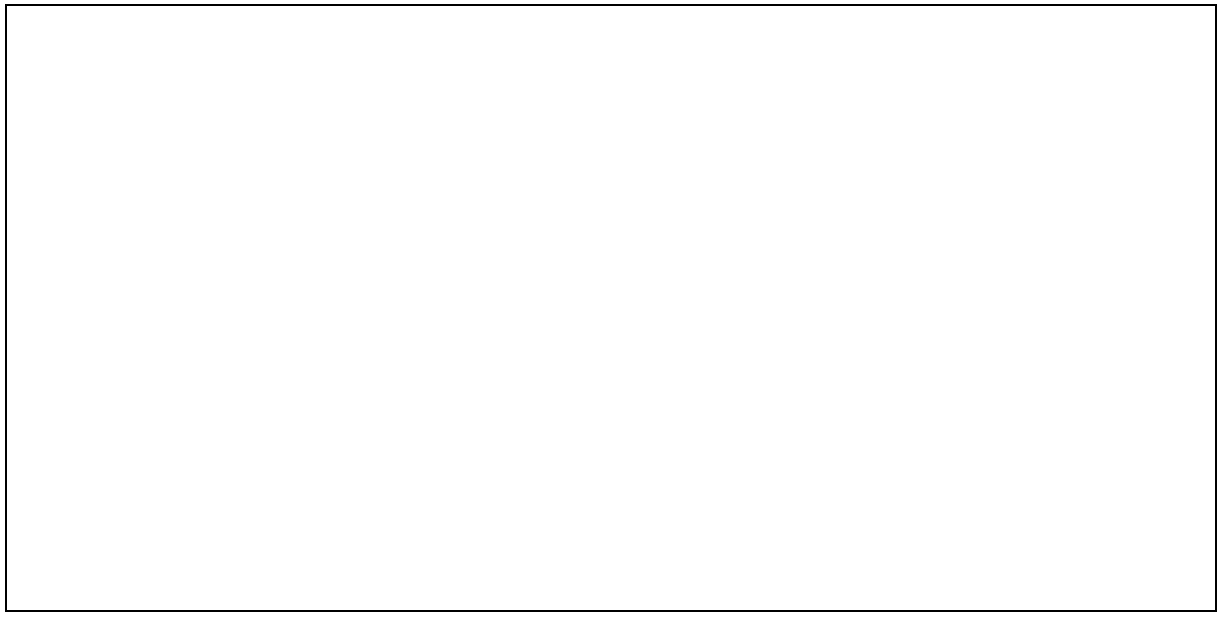
A) Write *PHP script to validate form including name, email using appropriate functions.*

B) Input/Output:

C) *Write PHP script for sending plain text email, HTML email and attachments with email.*



D) Input/Output:



K. Practical related Quiz.

1. preg_match() function is used to _____ .
2. For sending email _____ function is used in PHP.
3. Empty() function checks _____.

L. References / Suggestions

1) https://www.w3schools.com/php/default.asp
2) https://www.guru99.com/php-tutorials.html
3) https://www.tutorialspoint.com/php/
4) https://tutorialehtml.com/en/php-tutorial-introduction/
5) www.tizag.com/phpT/
6) https://books.goalkicker.com/PHPBook/
7) https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
8) https://codecourse.com/watch/php-basics
9) https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

M. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _____

Practical No.10: Session and Cookies

a. Write a PHP script to demonstrate creating, deleting, updating, retrieving and passing data with Cookie.
b. Write PHP script to demonstrate passing information using Session.

A. Objective:

A cookie is often used to identify a user. It is a small file that the server stores on the user's computer. Each time the same computer requests a page with a same browser, it will send the cookie too A session is a way to store information which is to be used across multiple pages. In this practical student will learn how to create cookie, modify it and delete it. Also, they will learn how to start a session, fetch session variables and destroy a session.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"

This practical is expected to develop the following skills.

1. Understand the difference between session and cookies.
2. Develop an application to start session, get session and destroy session.
3. Demonstrate the use of cookies by creating, deleting, updating, retrieving and passing data with it.

D. Expected Course Outcomes(Cos)

CO4: Debug the Programs by applying state management concepts and error handling techniques of PHP.

E. Practical Outcome(PRo)

Students will be able to create, store, fetch, delete cookies and session.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Cookies:

Cookies are small files that are stored on the client's computer by the web server. Cookies are used to store small amounts of data that can be accessed by the web server when the client requests a page from the website. Cookies are commonly used to store user preferences, shopping cart items, login credentials, and other user-specific information.

To set a cookie in PHP, you can use the setcookie() function.

Syntax:
setcookie(Name, value, Expire Time)

To retrieve a cookie value in PHP, you can use the $_COOKIE superglobal array. The $_COOKIE array contains all of the cookies that have been set by the web server.

Syntax:
$variable_name = $_COOKIE['cookie_name'];

To delete a cookie create a cookie using setcookie() function without specifying any Expire Time or give past time as an expiry time.

Syntax:
setcookie (‘Cookie_Name’, "") /OR/ setcookie ("name","",time()-60)

Example:

```
<?php
        setcookie("user", "IT");
?>
<html>
        <body>
        <?php
                if(!isset($_COOKIE["user"])) {
                        echo "Sorry, cookie is not found!";
                } else {
                        echo "<br/>Cookie Value: " . $_COOKIE["user"];
                        }
                ?>
        </body>
</html>
```

Session

Session is a way to store data on the server side that is associated with a specific user or client. Sessions are used to maintain stateful information across multiple requests from the same client. To start a session in PHP, you need to call the session_start() function at the beginning of your script. This function creates a new session or resumes an existing session if one exists.

Syntax:

```
session_start();
```

Once the session has been started, you can store data in the session by setting values in the $_SESSION superglobal array.

Syntax:

```
$_SESSION['session_name'] = 'value';
```

To retrieve data from the session, you can simply access the $_SESSION superglobal array.

Syntax:

```
$variable_name = $_SESSION['session_name'];
```

You can destroy a session and all of its associated data by calling the session_destroy() function. This function will remove all session.

To unset specific session variables without destroying the entire session, you can use the unset() function.

Syntax:

```
session_destroy();
unset($_SESSION['session_name']);
```

Example: session1.php

```
<?php
    session_start();
?>
<html>
    <body>
    <?php
        $_SESSION["user"] = "IT";
        echo "Session information are set successfully.<br/>";
    ?>
    <a href="session2.php">Visit session2.php</a>
    </body>
</html>
```

session2.php

```
<?php
    session_start();
?>
<html>
    <body>
        <?php
            echo "User is: ".$_SESSION["user"];
        ?>
    </body>
</html>
```

H. Resources/Equipment Required

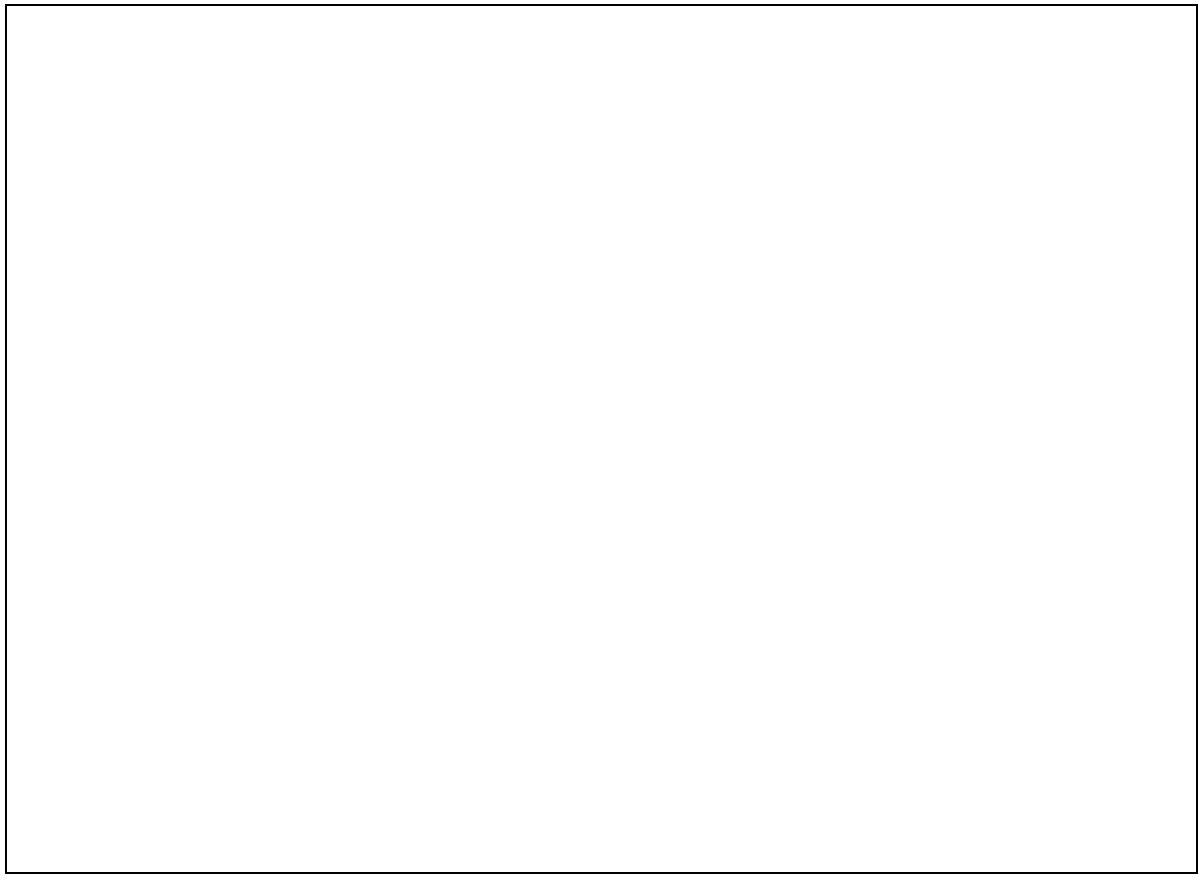
Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

I. Safety and necessary Precautions followed

NA

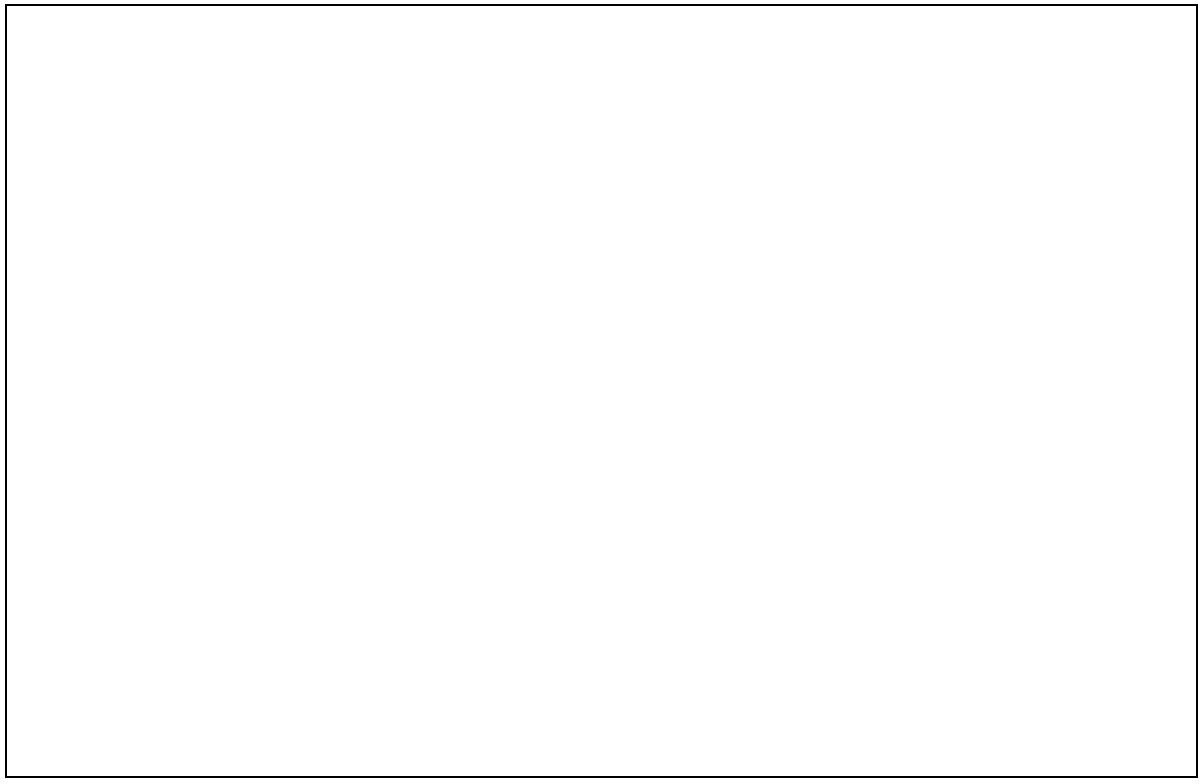
J. Source code:

A) Write *a PHP script to demonstrate creating, deleting, updating, retrieving and passing data with Cookie.*

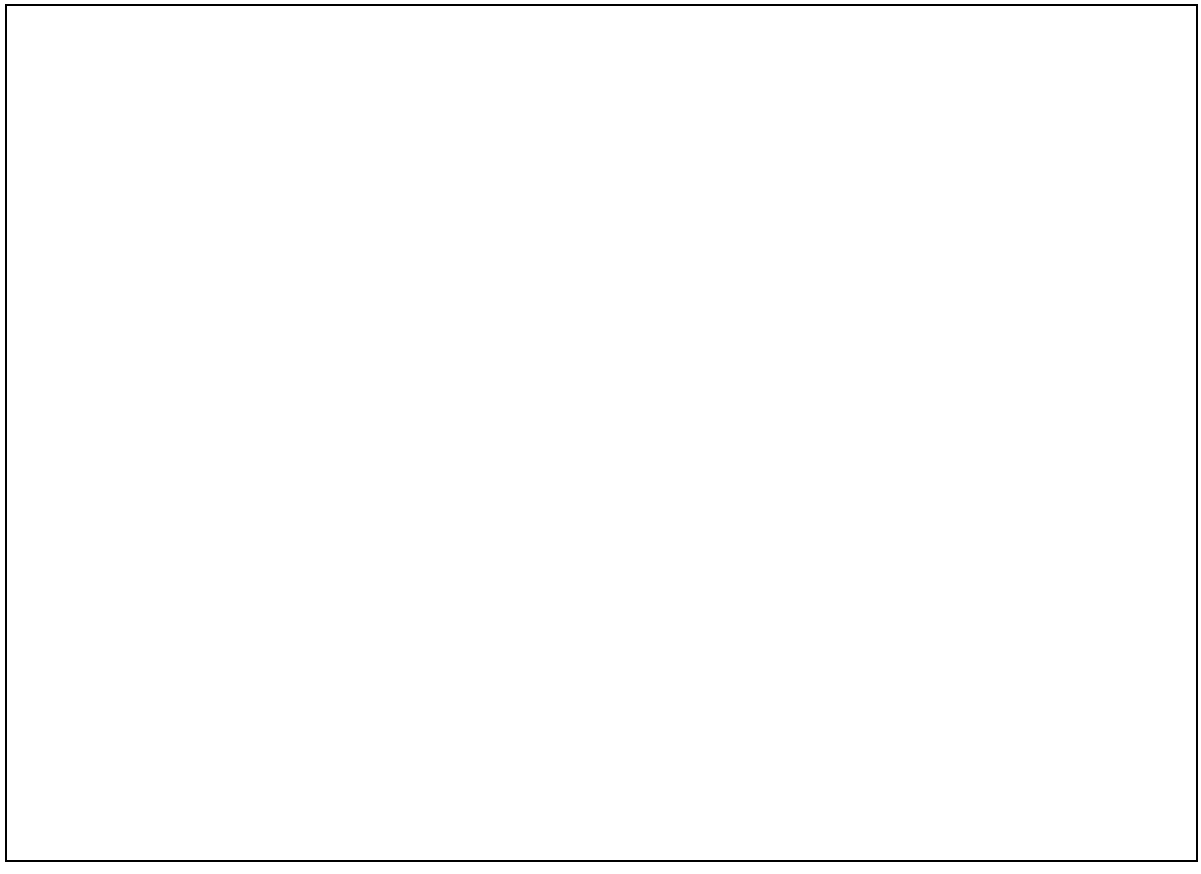


A) Input/Output:

B) *Write PHP script to demonstrate passing information using Session.*



C) Input/Output:



K. Practical related Quiz.

1. In PHP, cookies are set with _____ function.
2. The session_start() function must appear.
   a. after HTML tag
   b. before HTML. Tag
   c. after BODY tag
   d. before BODY tag
3. _____ function is used to erase all session variables stored in the current session.
4. Cookie is stored at _____ side and Session is stored at _____ side.
5. _____ superglobal variable is used to fetch information stored cookies.
6. _____ supergloabl variable is used to fetch information stored in session.
7. _____ function deletes only specified session.

L. References / Suggestions

1) https://www.w3schools.com/php/default.asp
2) https://www.guru99.com/php-tutorials.html
3) https://www.tutorialspoint.com/php/
4) https://tutorialehtml.com/en/php-tutorial-introduction/
5) www.tizag.com/phpT/
6) https://books.goalkicker.com/PHPBook/
7) https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
8) https://codecourse.com/watch/php-basics
9) https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

M. Assessment-Rubrics

Marks Obtained				Faculty Signature	Date
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _____

Practical No.11: Error Handling

a. Write a PHP script to demonstrate Error Handling.
b. Write a PHP script to demonstrate Use of Try and Catch Error Handling.

A. Objective:

When creating scripts and web applications, error handling is an important part. If your code lacks error checking code, your program may look very unprofessional and you may be open to security risks. Students will be able to learn handling of error in PHP.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"

This practical is expected to develop the following skills.

1. Understand error handling techniques and solve problem using it.
2. Follow coding standards and debug program to fix errors.

D. Expected Course Outcomes(Cos)

CO4: Debug the Programs by applying state management concepts and error handling techniques of PHP.

E. Practical Outcome(PRo)

Students will be able to perform various error handling techniques.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Error Handling in PHP

Error handling is the process of catching errors raised by your program and then taking appropriate action. When creating scripts and web applications, error handling is an important part. If your code lacks error checking code, your program may look very unprofessional and you may be open to security risks.

3 ways of error handling mechanism

- Using “die()” function
- Custom Errors
- Exception Handling

Using die() function

It will display error message that user can easily understand and it will stops the execution of script. The die() function prints a message and exits the current script.

```
<?php
        if(file_exists("myfile.txt"))
        {
                $file = fopen("myfile.txt", "r");
        }
        else
        {
                die("Error: The file does not exist.");
        }?>
```

Using Custom Error Handling

You can write your own function to handling any error. PHP provides you a framework to define error handling function. This function must be able to handle a minimum of

two parameters (error level and error message) but can accept up to five parameters (optionally: file, line-number, and the error context) –

Syntax:

Custom_Function_Name (Type, Message, Filename, Line Number, Context)

Using Exception Handling

PHP provides exception handling mechanism without terminating the execution with following three keywords.

1) Try 2) catch 3) Throw.

Try – A function using an exception should be in a "try" block. If the exception does not trigger, the code will continue as normal. However if the exception triggers, an exception is "thrown".

Throw – This is how you trigger an exception. Each "throw" must have at least one "catch".

Catch – A "catch" block retrieves an exception and creates an object containing the exception information.

Syntax:

```
try
{ //code to be executed
    throw new Exception("Error Message");
}
Catch (ExceptionType1 $ob1)
{// Exception Handling code for Exception Type 1 }
```

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Write a PHP script to demonstrate Error Handling.*

B) Input/Output:

C) *Write* a PHP script to demonstrate Use of Try and Catch Error Handling

D) Input/Output:

K. Practical related Quiz.

1. Which of the following is not specialized keywords in exception handling.
   a. try
   b. catch
   c. this
   d. throw
2. Die() function stops the execution of script. True/False?

L. References / Suggestions

1) https://www.w3schools.com/php/default.asp
2) https://www.guru99.com/php-tutorials.html
3) https://www.tutorialspoint.com/php/
4) https://tutorialehtml.com/en/php-tutorial-introduction/
5) www.tizag.com/phpT/
6) https://books.goalkicker.com/PHPBook/
7) https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
8) https://codecourse.com/watch/php-basics
9) https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

M. Assessment-Rubrics

Marks Obtained				Faculty Signature	Date
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _____

Practical No.12: Error Handling

a. Write a Write a PHP script to connect with database server from your webpage.
b. Create a database with student table and write a PHP script to insert a record in student table.
c. Write a program to read student records from student table and display all these information in table format on output screen.
d. Write a PHP script to delete and update a specific record from table.
e. Write a PHP script simple login system that allows user to add a new username if user doesn't exist in the database, also create a forgot password link, to redirect user to set up his new password on authentication.

A. Objective:

When we are talking about creating dynamic website, database plays an very important role. Students will be able to understand database in phpMyAdmin and can use various database operations.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Develop a webpage using PHP"

This practical is expected to develop the following skills.

1. Create database and table according to the requirements.

2. Develop an application which uses different database operations like insert, update, delete and select.
3. Follow coding standards and debug program to fix errors.

D. Expected Course Outcomes(Cos)

CO5: Create dynamic web pages using PHP and MySQL database.

E. Practical Outcome(PRo)

Students will be able to use database and perform various database operations using SQL statements.

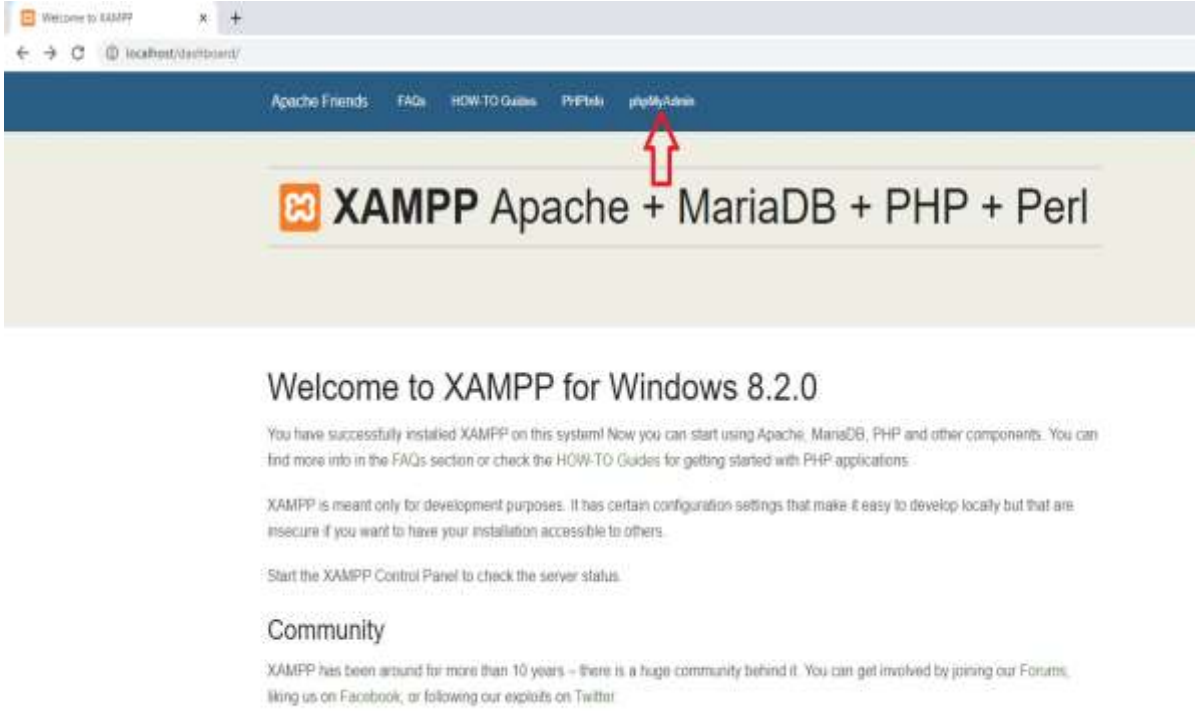
F. Expected Affective domain Outcome(ADos)

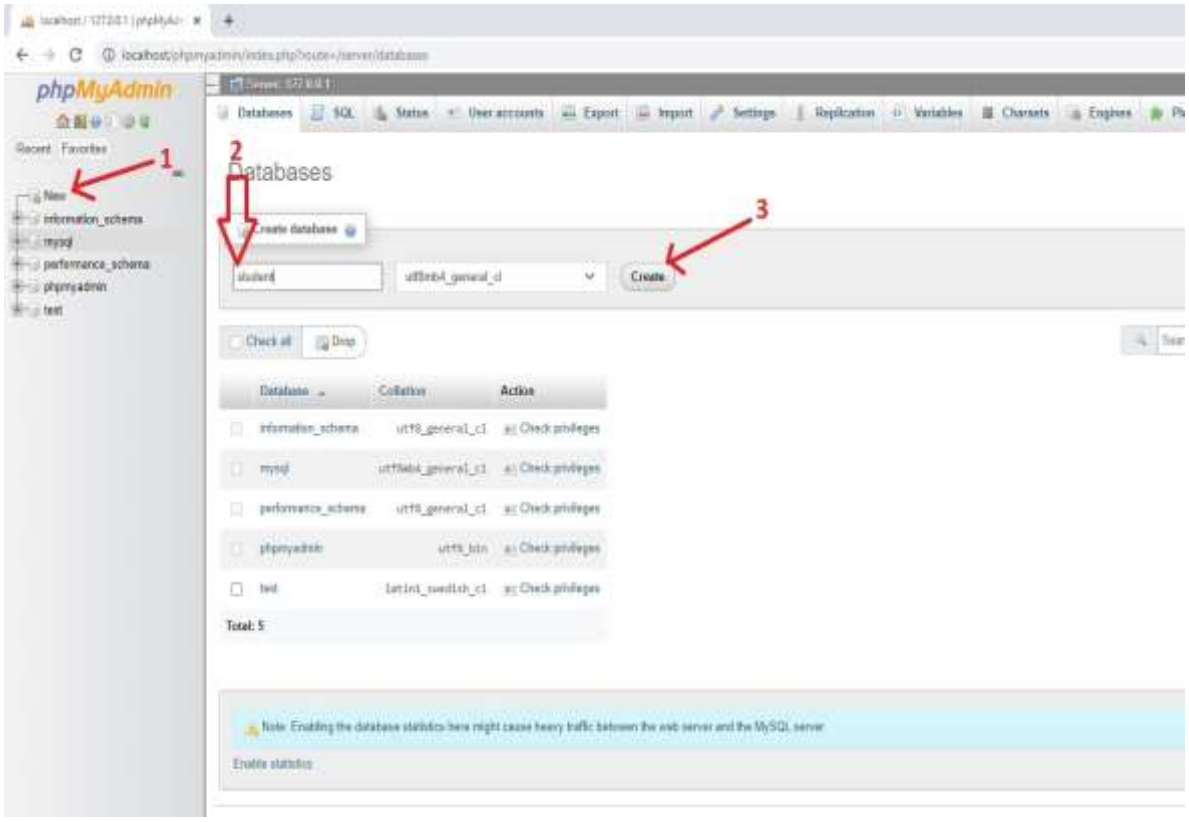
1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

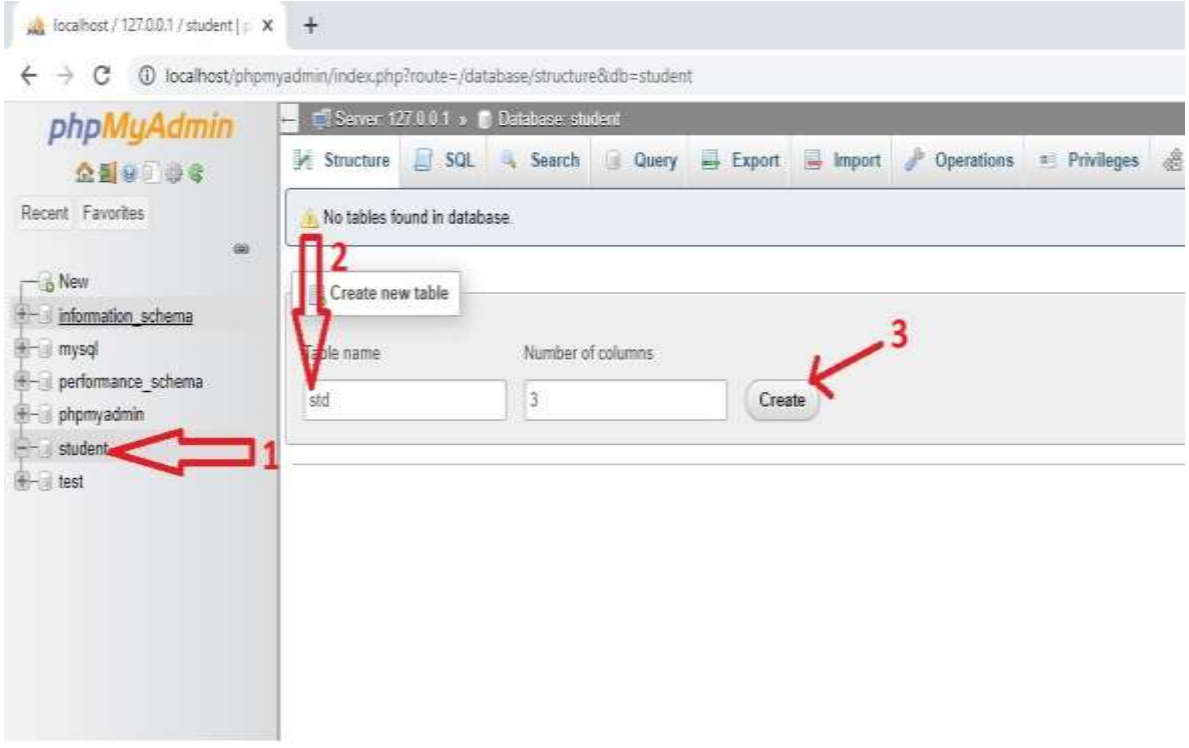
Database

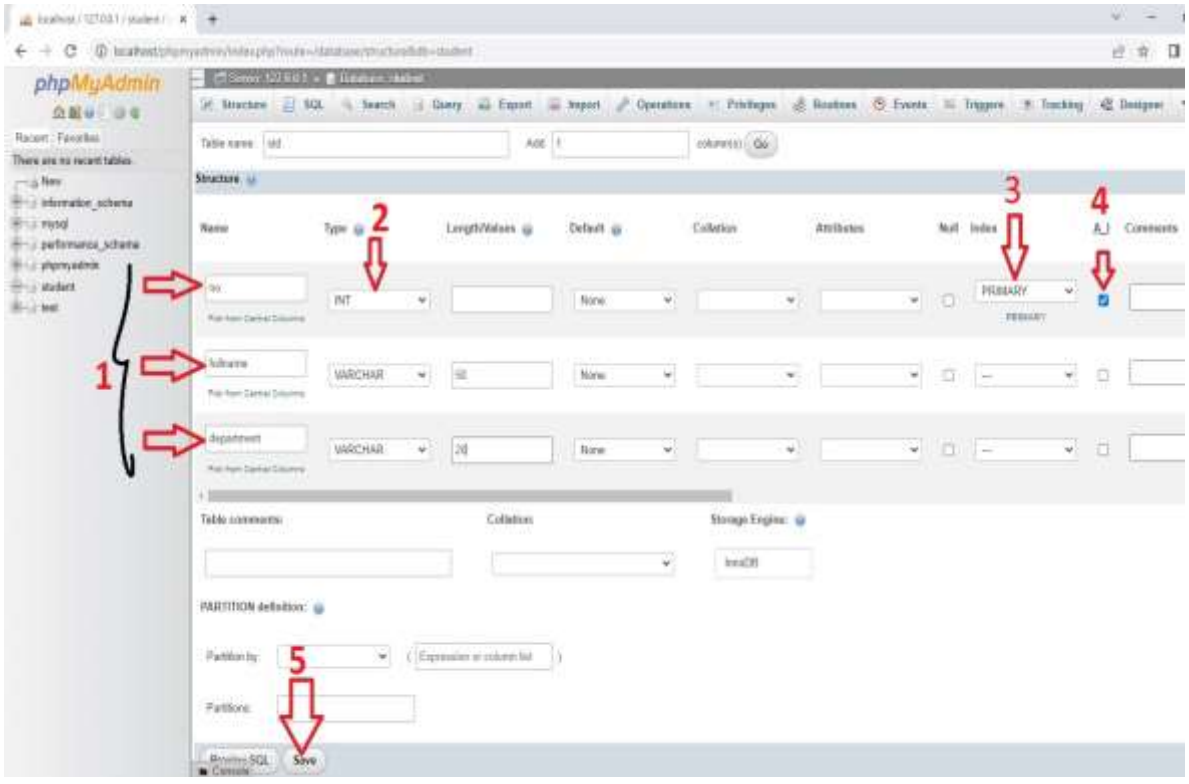
Create Database from PhpMyAdmin

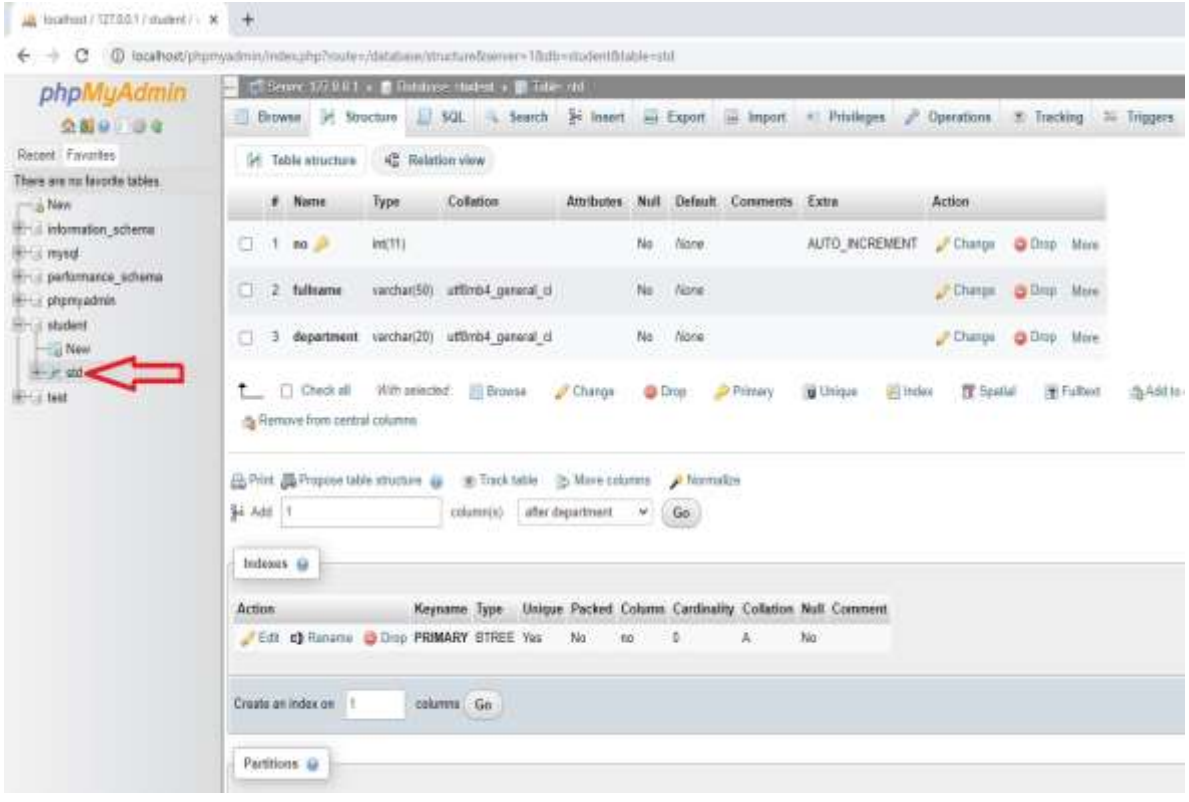




Create Table from PhpMyAdmin







mysqli_connect()

mysqli_connect() is a function used to establish a connection to a MySQL database.

Syntax:

mysqli_connect(servername, username, password, dbname);

Here,

servername: specifies the name of the MySQL server host.

username: specifies the username to connect to the MySQL server.

password: specifies the password to connect to the MySQL server.

dbname: specifies the name of the database to connect to.

mysqli_query()

mysqli_query() is a function used in PHP to execute a MySQL query on a database.

Syntax:

mysqli_query(connection, query);

Here,

connection: specifies the MySQL connection to use for the query.

query: specifies the SQL query to be executed.

mysqli_fetch_row()

mysqli_fetch_row() function fetches one row from a result-set and returns it as an enumerated array. It returns an array of strings that corresponds to the fetched row.

Syntax:

mysqli_fetch_row(result);

Here,

Result: Required. Specifies a result set identifier returned by mysqli_query().

mysqli_fetch_array()

mysqli_fetch_array() function fetches a result row as an associative array, a numeric array, or both. It returns an array of strings that corresponds to the fetched row.

Syntax:

mysqli_fetch_array(result,resulttype);

Here,

Result: Required. Specifies a result set identifier returned by mysqli_query().

Resulttype: Optional. Specifies what type of array should be produced. Can be one of the following values: MYSQLI_ASSOC, MYSQLI_NUM, MYSQLI_BOTH (default)

mysqli_error()

mysqli_error() function returns the last error description for the most recent function call, if any.

Syntax:

mysqli_error(connection)

Here,

Connection: Required. Specifies the MySQL connection to use

mysqli_close()

mysqli_close() function closes a previously opened database connection.

Syntax:

mysqli_close(connection)

Here,

Connection: Required. Specifies the MySQL connection to use

Example: (Insert)

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

        // Create connection
        $conn = mysqli_connect($servername, $username, $password,
        $dbname);
        // Check connection
        if (!$conn) {
          die("Connection failed: " . mysqli_connect_error());
        }

        $sql = "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('John', 'Doe', 'john@example.com')";

        if (mysqli_query($conn, $sql)) {
          echo "New record created successfully";
        } else {
          echo "Error: " . $sql . "<br>" . mysqli_error($conn);
        }

        mysqli_close($conn);
?>
```

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	XAMPP server (PHP, Web server, Database)	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	
5	Web Browser	Edge, Firefox, Chrome or similar	

I. Safety and necessary Precautions followed

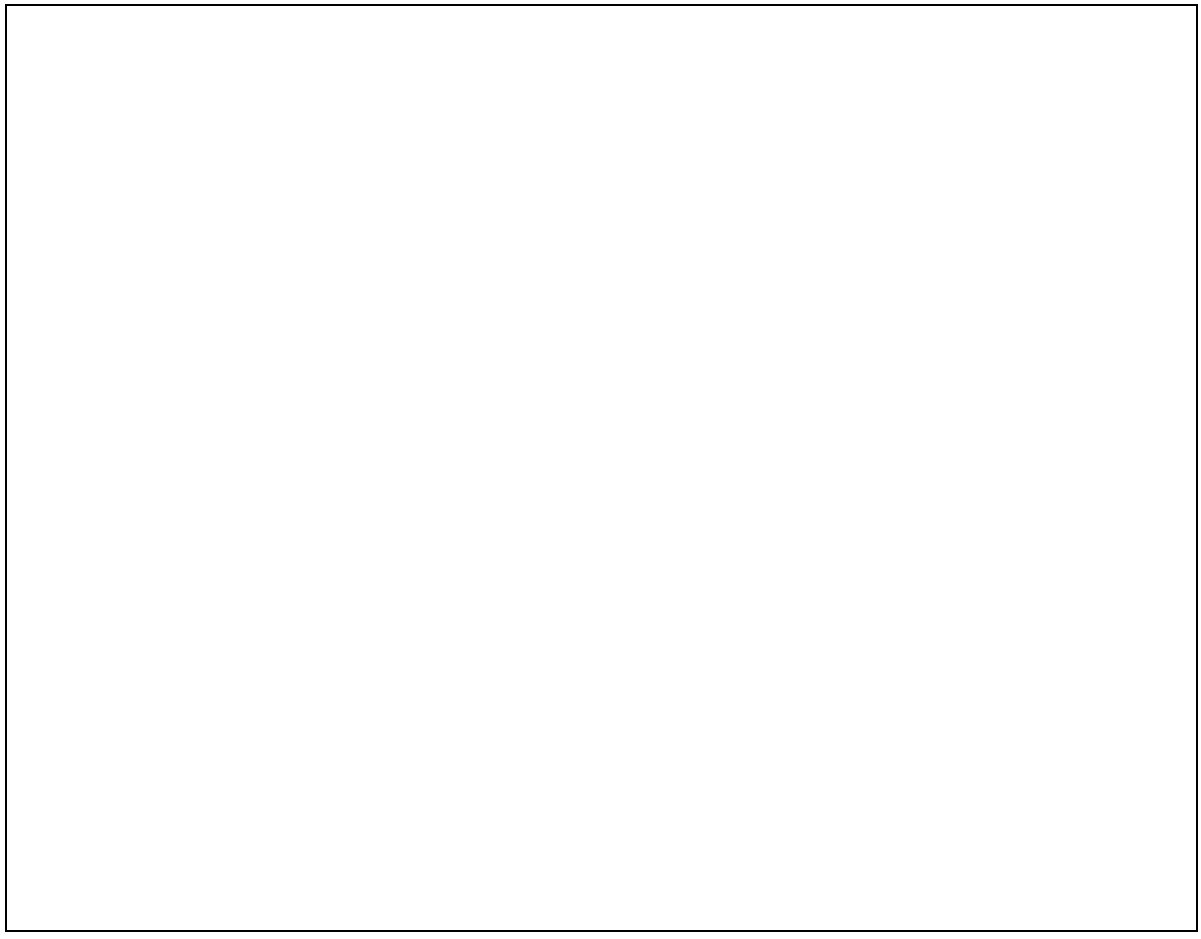
NA

J. Source code:

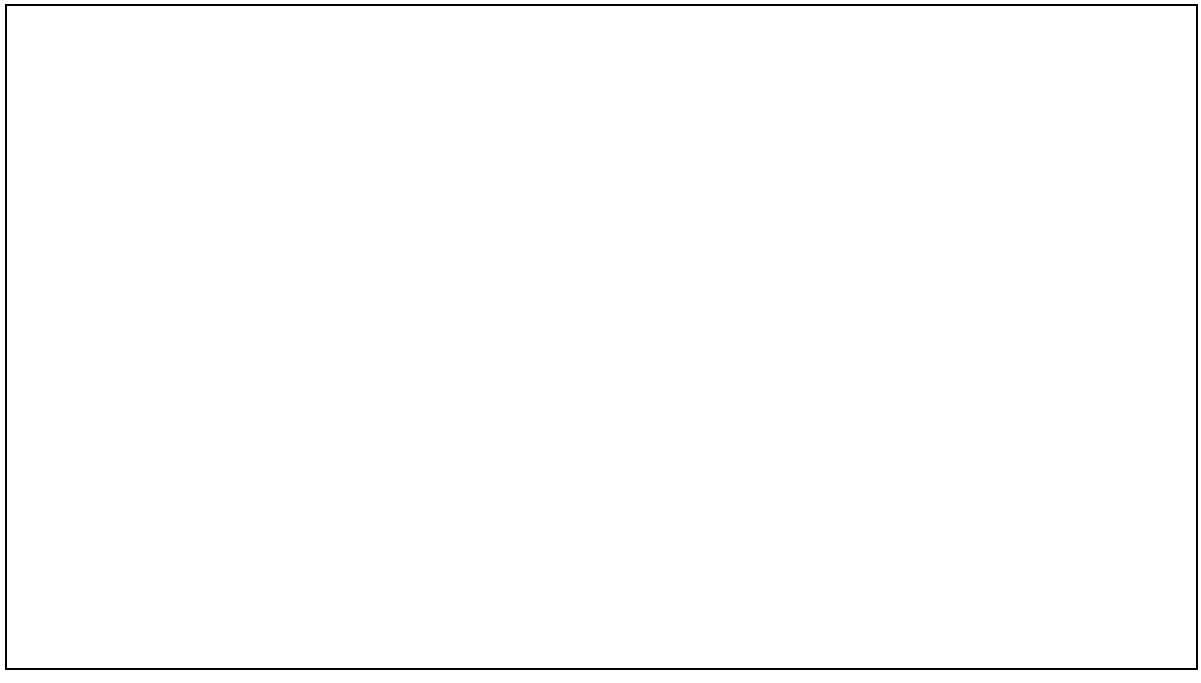
A) *Write a PHP script to connect with database server from your webpage.*

A) Input/Output:

B) *Create a database with student table and write a PHP script to insert a record in student table.*



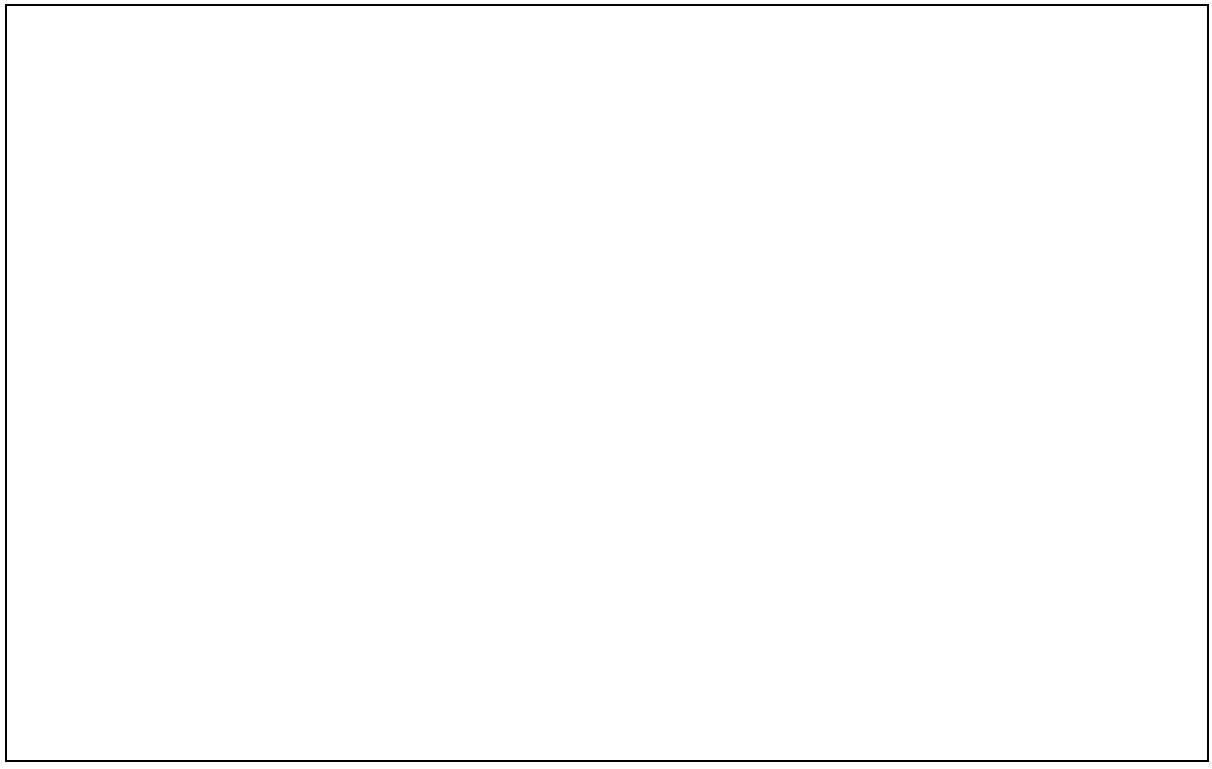
C) Input/Output:



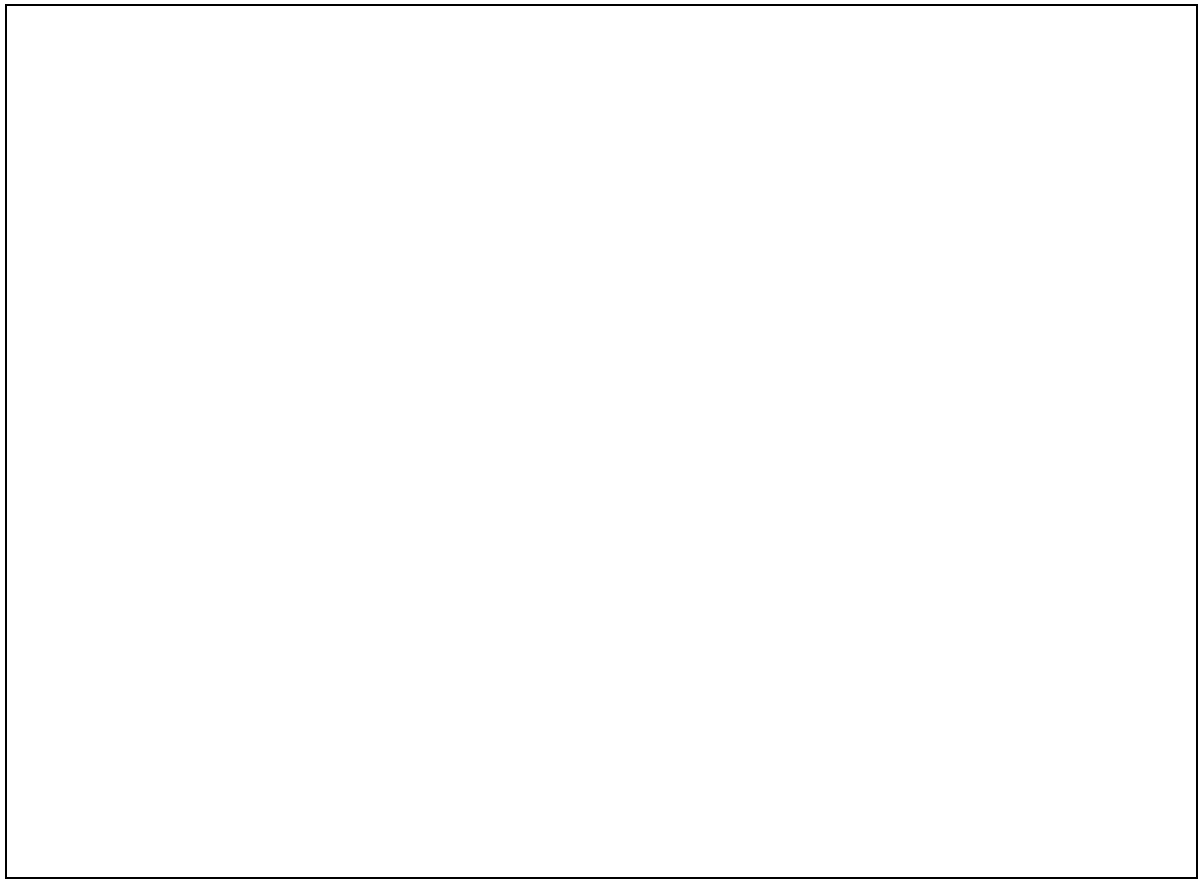
D) *Write a program to read student records from student table and display all these information in table format on output screen.*

E) Input/Output:

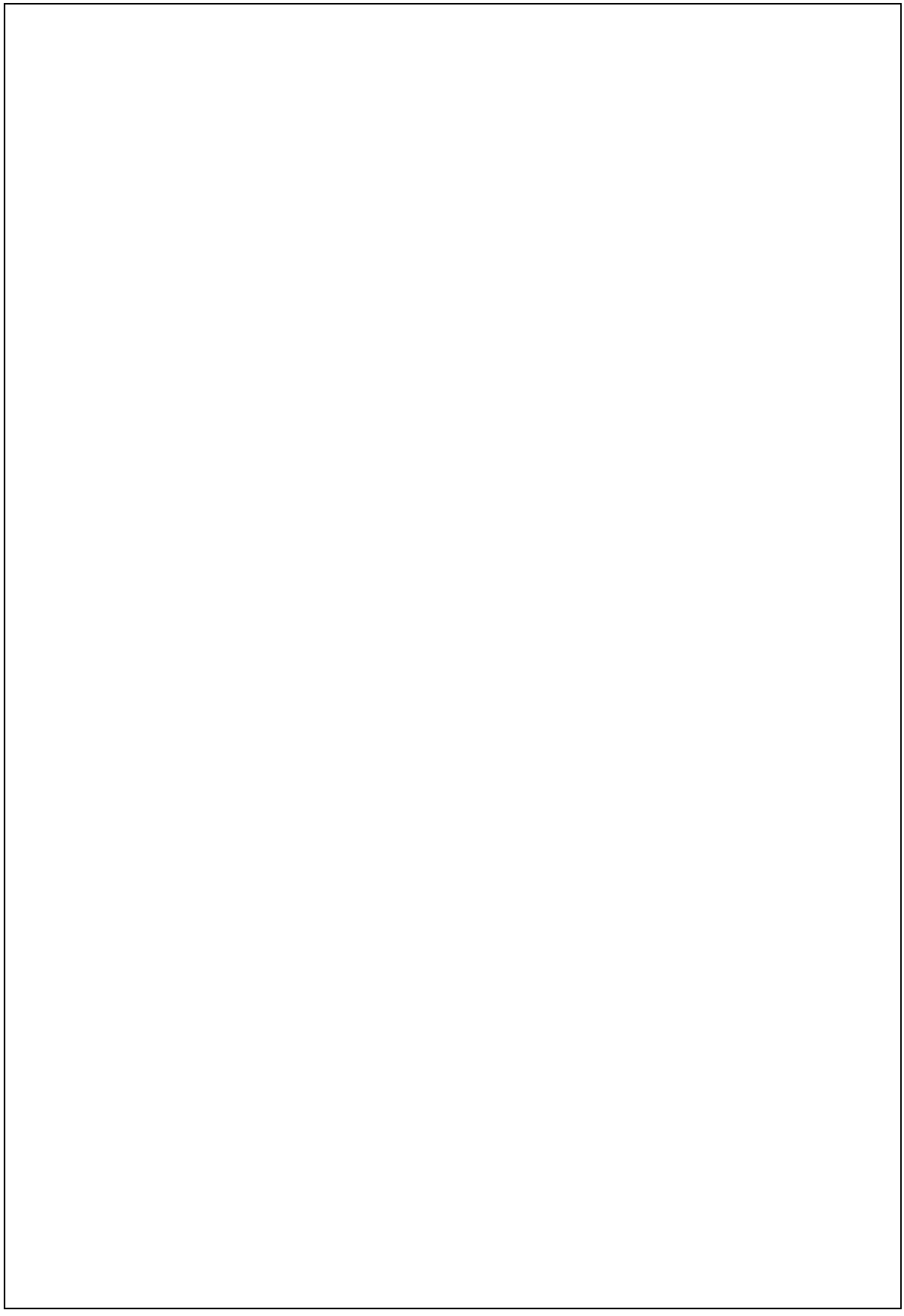
F) *Write a PHP script to delete and update a specific record from table.*

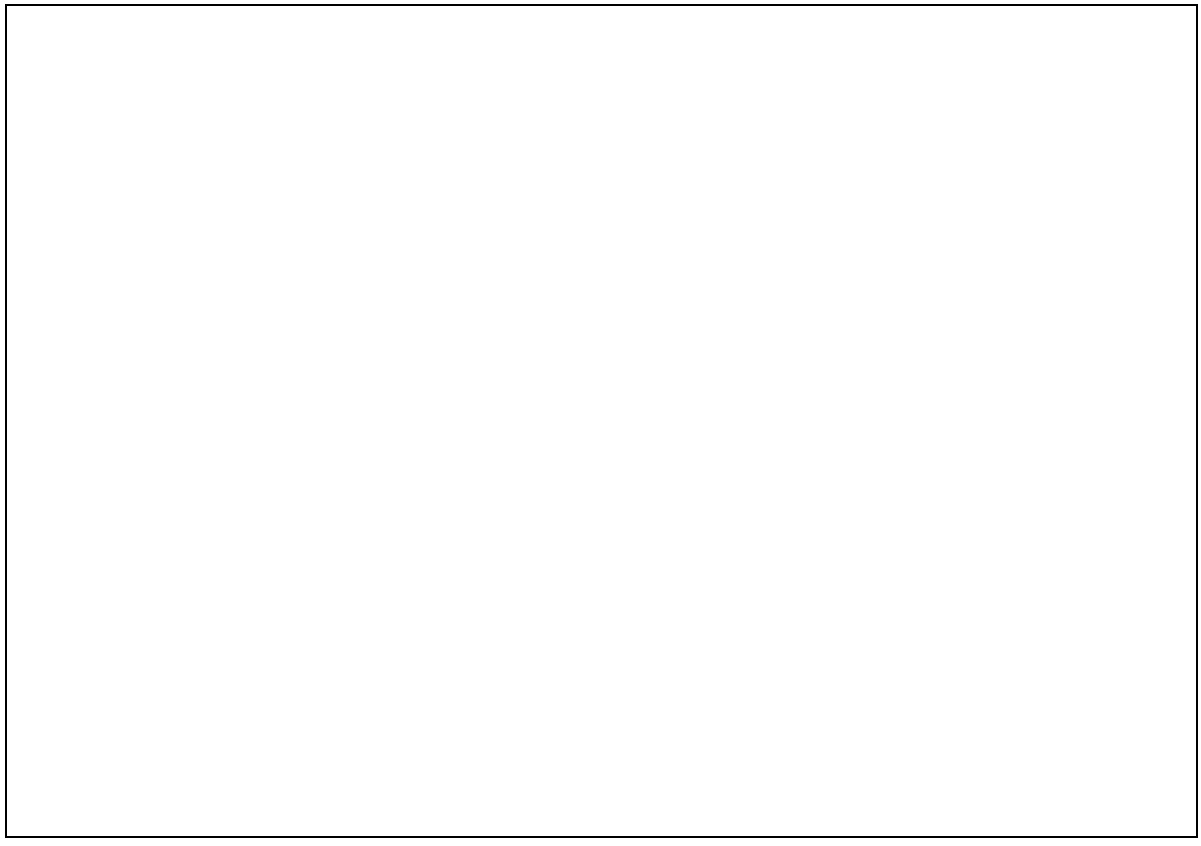


G) Input/Output:

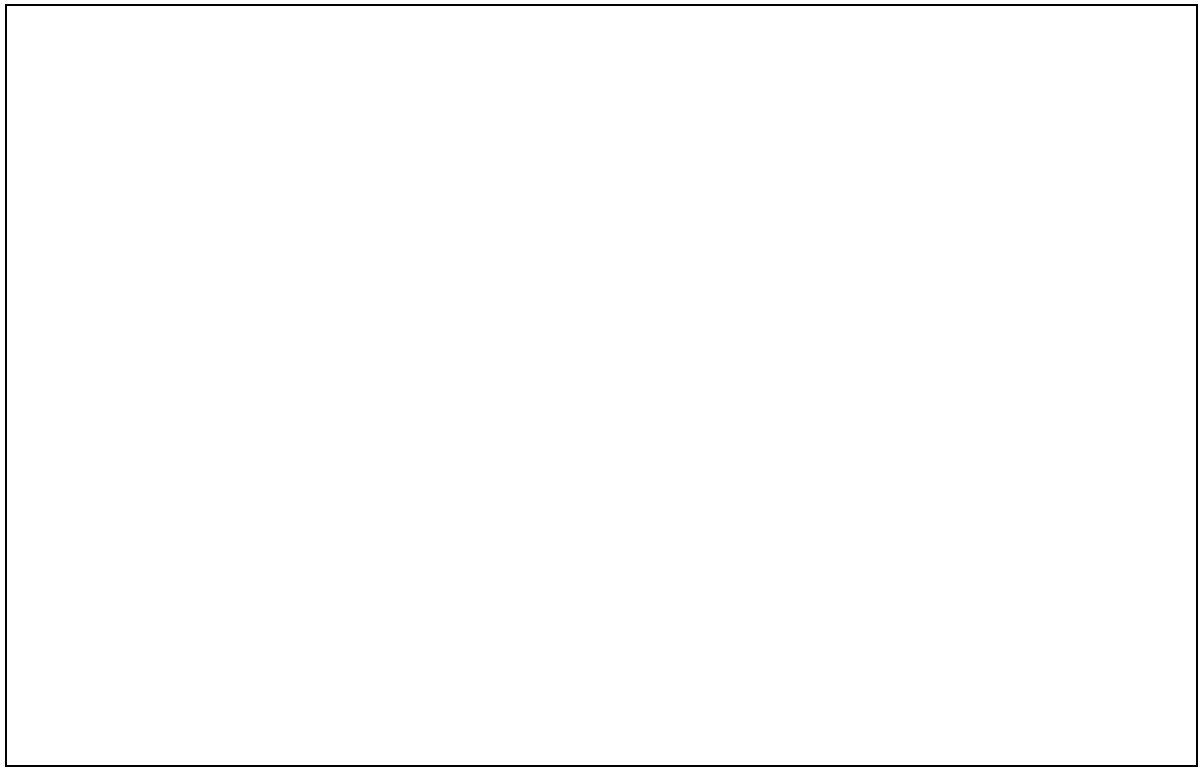


H) *Write a PHP script simple login system that allows user to add a new username if user doesn't exist in the database, also create a forgot password link, to redirect user to set up his new password on authentication.*





I) Input/Output:



K. Practical related Quiz.

1. To connect with database _____ function is used.
2. _____ function is used to write an SQL statement.
3. To fetch data from database _____ and _____ functions are used.
4. To get associative type of array in mysqli_fetch_array(), you need to specify _____ value as an argument.

L. References / Suggestions

1) https://www.w3schools.com/php/default.asp
2) https://www.guru99.com/php-tutorials.html
3) https://www.tutorialspoint.com/php/
4) https://tutorialehtml.com/en/php-tutorial-introduction/
5) www.tizag.com/phpT/
6) https://books.goalkicker.com/PHPBook/
7) https://spoken-tutorial.org/tutorial-search/?search_foss=PHP+and+MySQL&search_language=English
8) https://codecourse.com/watch/php-basics
9) https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

M. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		