Section-III

PHP

Assignment 1:Basics in PHP

Basics of PHP

For learning PHP, we have to learn the following basic points:

PHP Delimiters:

PHP is an embedded application, for writing PHP code we have to use its delimiters

<? php => starting delimiter and

?> => ending delimiter.

Syntax:

```
<? php

?>
```

Data Types:

PHP supports the following data types:

- String
- Integer
- Float (floating-point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

Operators in PHP

Types of operators that can be used in PHP programs are

There are the following types of operators:

- Arithmetic operators:- + , - , * , / , % ,**
- Assignment operators :- =
- Comparison operators :- <, >, <=, ==, ===, !=, <>, !==
- Increment/Decrement operators :- ++,--
- Logical operators: - &&, ||, !
- String operators:- .(concatenation)
- Conditional assignment operators: - ?,:

***Note: S**tudents can design HTML form to accept input from the user as per the requiremet of the program.

Practice Programs:

1. Write a PHP script to perform arithmetic operations on two numbers (Addition, Subtraction, Multiplication Division).
2. Write a PHP script to display a maximum of two numbers using a conditional operator.
3. Write a PHP script that will perform pre and post-increment of a number. (Example ++a, a++).

SET A:

1. Write a PHP Script to display Quotient and Remainder of the division of two variables.
2. Write a PHP Script to swap the values of two variables.
3. Write a PHP Script which will convert temperatures from Celsius(C)to Fahrenheit (F). (Hint: C=5.0/9(F-32)

SET B:

1. Write a PHP Script to display the surface area and volume of a cuboid.
   (Hint: surface area=2(lb+lh+bh), volume = l*b*h)
2. Write a PHP Script to calculate the area of Circle, Square, and Rectangle.
3. Write a PHP Script to display the total and percentage of Marks of Subjects (Out of 100) Data Structure, Digital Marketing, PHP, SE, and Bigdata.

SET C:

1. Write a PHP Script to calculate the total cost of AIR Ticket Reservation and display the details for Name, Address, Contact No, Source, Destination, Date of journey, Gender of passenger, No of Persons, Price per Ticket, etc.

Assignment Evaluation

0: Not Done [ ] **1: Incomplete [ ]** **2: Late Complete [ ]**
3: Need Improvement [ ] **4: Complete [ ]** **5: Well Done [ ]**

Signature of Instructor

Assignment 2:Control Structures and Loops

Conditional Statements

Conditional statements are used to check conditions and the programmer can accordingly display the results. PHP supports the following conditional statements.

1. if Statement
2. if else Statement
3. elseif Statement
4. switch Statement

Name/Use	Syntax	Example

if Statement:
It is used to check a condition. If the condition is true the corresponding body of if statement is executed.
	if(Condition)
	{
	Statements;
	}
		<?php
		$n=10;
		if ($n%2==0)
		{
		echo “Number Is Even”;
		}
		?>
		Output: Number Is Even

if else Statement:
It checks a condition and if the condition is true the corresponding body of the if statement is executed otherwise else part is executed.
	if(Condition)
	{
	Statements;
	}
	else
	{
	Statements;
	}
		<?php
		$n=10;
		if ($n%2==0)
		{ echo “Number Is Even”;}
		else
		{ echo “Number Is Odd”;}
		?>
		Output: Number Is Even

elseif Statement:
It checks more than one condition.
	if (condition)
	{ Statements;}
	elseif (condition)
	{ Statements;}
	else
	{ Statements;}
		<?php
		$a=10;
		$b=20;
		if ($a==$b)
		{ echo “a and b are same”;}
		elseif ($a<$b)
		{ echo “a is less than b”;}
		else
		{ echo “a is greater than b”;}
		
		?>
		Output:a is less than b

switch Statement It checks the result of the expression with multiple conditions (cases). The case is	switch (expression) { case value1: Statements	<?php $num=2; switch($num) {

executed for which the match is found.	break; case value2: Statements break;..... default: Code to be executed if all cases are not matched; }	Case 1: echo “One”; break; Case 2: echo “two”; break; Case 3: echo “Three” break; Case 4: echo “Four; break; Case 5: echo “Five”; break; default: echo “Invalid Number”; ?> **Output:** Two

Loops

Loops in php are used to execute a similar group of statements. PHP supports the following 4 types of loops.

1. for Loop
2. while Loop
3. do…while Loop
4. foreach Loop

Name/Use	**Syntax**	**Example**

for Loop
It executes a block of code for a specified number of times.
	for (initialization; condition; increment)
	{
	code to be executed;
	}
		<?php
		For($j=1; $j<=5; $j++)
		{
		echo $j;
		}
		?>
		Output: 12345

while Loop
It executes a block of code until the condition specified is true.
	while(expression)
	{
	Statements;
	}
		<?php
		$j=1;
		while($j<=5)
		{
		echo $j;
		$j++;
		}
		?>

		Output: 12345

do…while Loop
It executes ablock of code once and then repeats the loop as long as a special condition is true.
	do
	{
	code to be executed;
	}
	while (condition);
		<?php
		$j=1;
		do{
		echo $j;
		$j++;
		} while($j<=5);
		
		?>
		Output: 12345

foreach Loop
It is used to traverse the array and the block of code is executed for each element of the array.
	foreach (array as value) {
	code to be executed;
	}
		<?php
		$array = array(1, 2, 3, 4, 5);
		foreach($array as $value)
		{
		echo "Value is $value";
		}
		?>
		Output:
		Value is 1Value is 2Value is 3Value is 4Value is 5

***Note: S**tudents can design HTML form to accept input from the user as per the requirement of the program.

Practice Programs:

1. Write a PHP Script to display a maximum of two numbers.
2. Write a PHP Script to check whether a number is positive or negative.
3. Write a PHP Script to display a Multiplication table of a number

SET A:

1. Write a PHP Script to check whether a year is a leap or not.
2. Write a PHP Script which will perform the Addition, Subtraction, Multiplication, and Division of two numbers as per the choice. (Use Switch Case)
3. Write a PHP Script to display the grade of the student according to percentage. Use the following conditions:
   Percentage <40 => Grade=”Fail”
   Percentage >= 40 and Percentage <=50 => Grade= “Pass Class”
   Percentage >=50 and Percentage <=60 => Grade= “Higher Second Class”
   Percentage >60 and Percentage <=70 => Grade= “First Class”
   Percentage >70 => Grade= “First Class with Distinction”

SET B:

1. Write a PHP Script to display prime numbers between 1 to 50.

2. Write a PHP Script to display a perfect numbers between 1 to100.
3. Write a PHP Script to display the reverse of a number. E.g. 607 =>706
4. Write a PHP Script to display Armstrong numbers between 1 to 500.

SET C:

1. Write a PHP script to display a number in words (Use Switch case)
   e.g. 345–three four five
2. Write a PHP script to change the background color of the browser using a switch statement according to the day of the week.
3. Write a PHP script to count the total number of even and odd numbers between 1 to 1000.

Assignment Evaluation

0: Not Done [ ] **1: Incomplete [ ]** **2: Late Complete [ ]**
3: Need Improvement [ ] **4: Complete [ ]** **5: Well Done [ ]**

Signature of Instructor

Assignment 3: Arrays and Strings

Strings in PHP

A string is a sequence of characters. There are two types of strings.

1. **Single-Quoted String:**In this type, characters are enclosed with a single quotation mark(‘’);

Examples:

‘Hello World’
‘Amar’
‘Pune’

The limitation of a single-quoted string is that variables are not interpolated.

Example:

```
<?php
$name=‘Amar’;
$str='Hello $name';
echo $str;
?>
```

Output:

Hello $name

2. **Double-Quoted String:** In this type,characters are enclosed with double quotation marks (“ ”).

PHP interpreter interprets variables and special characters inside double-quotes.

Example:

```
<?php
    $name=‘Amar’;
    $str=“Hello $name”;
    echo $str;
?>
```

Output:

Hello Amar

It expands the many PHP escape sequences. The escape sequences recognized by PHP in double-quoted strings are as follows:

Escape Sequence	Meaning
\n	New Line
\r	carriage return
\t	horizontal tab
\v	vertical tab
\e	escape
\f	form feed
\\	Backslash
\$	dollar sign
\"	double-quote

String Functions

PHP provides approximately one hundred functions for string manipulations. Some of the functions that can be performed on strings are:

- Compare two strings
- Find a String In AnotherString
- Find Out How Many Instances of A String Occur In AnotherString
- Return Part of aString
- Replace Part of aString
- Trim Whitespace From The Ends of aString
- Make An Entire String Lowercase or uppercase

Name	Use	Example
strlen()	It is used get string length.	<?php $input = 'Sunayana'; echo strlen($input); ?> **Output:**8
trim()	It used to remove the whitespaces and other characters.	<?php $input = " Programming in PHP \n"; echo trim($input); ?> **Output:** Programming in PHP
ltrim()	It used to strip whitespace or other characters from the beginning of a string.	<?php $input = " Programming in PHP \n"; echo trim($input); ?> **Output:** Programming in PHP \n
rtrim()	It is used to remove the	<?php

	white spaces from end of the string.	$input = " Programming in PHP \n"; echo trim($input); ?> **Output:** Programming in PHP
strtolower()	It converts the whole string into lower case.	<?php echo strtolower("DYPATIL ACS"); ?> **Output:** dypatil acs
strtoupper()	It converts the whole string into upper case.	<?php echo strtoupper("d y patil acs "); ?> **Output:** D Y PATIL ACS
ucfirst()	It used to convert the first character of a string to upper case.	<?php echo ucfirst("dypatil"); ?> **Output:**Dypatil
ucwords()	It used to convert the first character of a string to upper case in each string	<?php echo ucwords("d y patil pimpri"); ?> **Output:** D Y Patil Pimpri
strcmp()	It is used to compare two strings. If two string are equal it returns 0 otherwise 1.	<?php echo "The result is "; echostrcmp("Hello world!","Hello world!"); ?> Output: The result is 0
substr()	Returns a part of a string	<?php echo substr("D Y Patil",2); ?> **Output:** Y Patil
substr_replace()	It used to replace the part of string with another string	<?php echosubstr_replace("HelloWorld","Good Morning",0); ?> **Output:** Good Morning
substr_compare()	It used to compare two string format with a specific start position	<?php echo substr_compare("Hello","world",0)." "; echo substr_compare("abcde","de",1,3)." "; ?> **Output:** -1 -1
substr_count()	It used to count the number of sub strings	<?php echo substr_count("HelloWorld","World"); ?> **Output:** 1
strrev()	It is used to reverse a	<?php

	string.	echo strrev("sairamkrishna"); ?> **Output:**anhsirkmarias

str_pad()	It pads a string to a new length.	<?php
		//Pad to the right side of the string, to a new length of 20 characters:
		
		$str = "Hello World";
		echo str_pad($str,15,"=");
		?>
		Output: Hello World====

explode()	It is used to split a string by string	<?php
		//Decomposing string
		//Break a string into an array:
		$str = "Hello world. It's a beautiful day.";
		$arr=explode(" ",$str);
		print_r ($arr);
		echo"
		
		";
		$str = "one|two|three|four";
		$arr=explode('\|',$str);
		print_r ($arr);
		?>
		Output:
		Array ([0] => Hello [1] => world. [2] =>It's [3] => a [4] => beautiful [5] => day.)
		
		Array ([0] => one [1] => two [2] => three [3] => four)

implode()	It creates a string from an array of smaller string.	<?php
		$arr = array('Hello','World!','Beautiful','Day!');
		echo implode(" ",$arr);
		echo "
		
		";
		$arr = array('Hello','World!','Beautiful','Day!');
		echo implode(",",$arr);
		?>
		Output:
		HelloWorld! Beautiful Day!
		
		Hello,World!,Beautiful,Day!

strpos()	It is used to find the position of first occurrence of a string inside another string.	<?php echostrpos("I love php, I love php too!","php")." "; ?> **Output:** 7

strstr()	It is used to find the first occurrence of a string and returns from that small string onwards.	<?php echostrstr("Hello world!","world")." "; echostrstr("w3resource.com","."); ?> **Output:** world! .com

Arrays in PHP

An array is a collection of different data elements. Multiple elements can be stored using an array under a single name.

Declaration of an Array

An array can be defined/declared by using **array ()** function.

Syntax:

```
$a=array (10, 20, 30, 40);
$colors = array("Red", "Blue, "Yellow");
```

There are following types of an array:

1. Indexed array.
2. Associative array.
3. Multidimensional array.

An array is organized as an ordered collection of (key,value) pairs. In PHP there are three types of arrays:

a) Indexed array: It is an array with a numeric index starting with 0. There are two ways to create an Indexed array:

Example:

```
$num=array (10, 20);
        OR
$num[0]=10;
$num[1]=20;
```

b) Associative array: Associative arrays are arrays that use named keys that you assign to them. There are two ways to create an associative array:

Example:

```
$age = array("Sagar"=>"35", "Abhijeet"=>"37", "Ishwar"=>"43");
                    OR
```

```
$age['Sagar'] = "35";
$age['Abhijeet'] = "37";
$age['Ishwar'] = "43";
```

c)Multidimensional array: A multidimensional array is an array containing one or more arrays. In this type of array, multiple arrays can be defined in a single array.

Example:

```
$cars = array(
  array(“Swift”,20,30),
  array(“Dezire,40,50),
  array(“Mercedez”,6,7),
  array("Scoda",12,15)
  );
```

Array Functions

Name	Use	Example
array_chunk()	It is used to split an array into chunks of a given size	<?php $a=array("10","20","30","40"); print_r(array_chunk($a,2); ?> **Output:** Array([0] => Array([0] =>10 [1] =>20) [1] => Array ([0] =>30 [1] =>40))
array_combine ()	It is used to combine two arrays into one, values of the first array are the keys and values of the second array are the values in the combined array.	<? php $X=array("a","b","c"); $Y=array("100","200","300"); $Z=array_combine($X,$Y); print_r($Z); ?> **Output :**Array([a]=>100, [b]=>200, [c]=>300)
array_diff ():	It is used to compare the values of two arrays and return the difference	<?php echo "<font size=14>"; $a=array(1,2,3,4,5); $b=array(4,2,6); $c=array_diff($a,$b); print_r($c); ?> **Output:** Array ([0] => 1 [2] => 3 [4] => 5)
array_intersect()	It returns the common elements of two arrays.	<?php $a=array(1,2,3,4); $b=array(4,5,6,2); $c=array_intersect($a,$b);

		print_r($c); ?> **Output:** Array([1]=>2, [2]=>4);
array_flip()	Exchanges all keys with their associated values in an array.	<?php $a = array("a"=>1, "b"=>2, "c"=>3, "d"=>4, "e"=>5); print_r(array_flip($a)); ?> **Output:** Array ([1] => a [2] => b [3] => c [4] => d [5] => e)

array_splice()	It removes and replaces specified elements of an array	<?php
		$a=array(10,20,30,40,50,60);
		$b=array_splice($a,2,3);
		print_r($a);
		print_r($b);
		?>
		Output: Array ([0] => 10 [1] => 20 [2] => 60)
		Array ([0] => 30 [1] => 40 [2] => 50)

array_slice()	It returns selected parts of an array. It returns the sequence of elements from the array array as specified by
	the offset and length para meters.
		<?php
		$a=array(1,2,3,4,5,6);
		$b=array_slice($a,2,3);
		print_r($a);
		echo "
		";
		print_r($b);
		?>
		Output:
		Array ([0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5 [5] => 6)
		Array ([0] => 3 [1] => 4 [2] => 5)

array_reverse()	Returns an array in the reverse order.	<?php $a=array(1,2,3); $d=array_reverse($a); print_r($d) ?> **Output:** Array ([0] => 3 [1] => 2 [2] => 1)

array_key_exists( )	This function is used to check if an element exists in thearray	<?php
		$a=array("a"=>"ABC","p"=>"PQR","x"=>"XYZ");
		if(array_key_exists("p",$a))
		echo "Key Exists";
		else
		echo "Key Does not Exists";
		?>
		Output: Key Exists

array_push()	This function add the	<?php

	new element at the end of an array.	$a = array("a"=>"banana","b"=>"apple","c"=>"orange");
		print_r(array_push($a, "Straberry"));
		print_r($input);
		?>
		Output: 4 Array ([a] => banana [b] => apple [c] => orange [0] =>Straberry)

array_pop()	This functions remove last element of an array.	<?php $a=array("a"=>"banana","b"=>"apple","c"=>"orange"); print_r(array_pop($a)); ?> **Output:** orange
array_shift()	It removes the first element from an array, and returns the value of the removed element	<?php $a = array("a"=>"banana","b"=>"apple","c"=>"Mango"); print_r(array_shift($a)); ?> **Output:** banana

array_unshift()	It adds one or more elements to the beginning of an array	<?php
		$a = array("orange", "banana");
		array_unshift($a, "apple");
		print_r($a);
		?>
		Output: Array ([0] => apple [1] => orange [2] => banana)

array_sum()	It returns the addition of array elements.	<?php $a=array(1,2,3); $sum=array_sum($a); echo "Sum=$sum"; ?> **Output:** Sum=6
array_product()	It returns the product of array elements.	<?php $a = array(5,6); print_r(array_product($a)); ?> **Output:**30
array_unique()	It removes duplicate values from an array	<?php $a = array("a" => "green", "red", "b" => "green", "blue", "red"); $result = array_unique($a); print_r($result); ?>

		Output: Array ([a] => green [0] => red [1] => blue)
extract()	It creates local variables from an array.	**?**php $a = "Original"; $my_array = array("a" => "Cat","b" => "Dog", "c" => "Horse"); extract($my_array); echo "\$a = $a; \$b = $b; \$c = $c"; ?> **Output:** $a = Cat; $b = Dog; $c = Horse
compact()	Create array containing variables and their values	<?php $city = "Pune"; $state = "Mumbai"; $result = compact("city", "state"); print_r($result); ?> **Output:**Array ([city] => Pune [state] => Mumbai)

in_array()	Checks if a specified value exists in an array	<?php
		$a=array(10,20,30,40,50,60);
		if(in_array(40,$a))
		echo "Element Available in array";
		else
		echo "Element not available in array";
		?>
		Output: Element Available in array

count()	It gives number of elements in an array.	<?php $a=array(10,20,30,40,50,60); echocount($a); ?> **Output:** 6

Array Sorting Functions

Name	Use	Syntax

sort()	It sorts array in ascending order.	<?php
		$a=array("mh","ap","LM","za");
		print_r($a);
		echo "
		";
		sort($a);
		echo "
		 After Sorting
		";
		print_r($a);
		?>

		Output: Array ([0] =>mh [1] =>ap [2] => LM [3] =>za) After Sorting Array ([0] => LM [1] =>ap [2] =>mh [3] =>za)

rsort()	It sorts array in descending order.	<?php
		$a=array("mh","ap","LM","za");
		print_r($a);
		echo "
		";
		rsort($a);
		echo "
		 After Sorting
		";
		print_r($a);
		?>
		Output: Array ([0] =>mh [1] =>ap [2] => LM [3] =>za)
		After Sorting
		Array ([0] =>za [1] =>mh [2] =>ap [3] => LM)

asort()	It sorts associative array in ascending order as per the values	<?php
		$b=
		array("X"=>"XYZ","A"=>"ABC","L"=>"LMN");
		print_r($b);
		asort($b);
		echo "
		 After Sorting
		";
		print_r($b);
		?>
		Output: Array ([X] => XYZ [A] => ABC [L] => LMN)
		After Sorting
		Array ([A] => ABC [L] => LMN [X] => XYZ)

arsort()	It sorts associative array in descending order as per the values.	<?php
		$b=
		array("X"=>"XYZ","A"=>"ABC","L"=>"LMN");
		print_r($b);
		arsort($b);
		echo "
		 After Sorting
		";
		print_r($b);
		?>
		Output:
		Array ([X] => XYZ [A] => ABC [L] => LMN)
		After Sorting
		Array ([A] => ABC [L] => LMN [X] => XYZ)

ksort()	It sorts associative array in ascending order as per the keys.	<?php
		$b=
		array("X"=>"XYZ","A"=>"ABC","L"=>"LMN");
		print_r($b);
		ksort($b);

		echo " After Sorting "; print_r($b); ?> **Output:** Array ([X] => XYZ [A] => ABC [L] => LMN) After Sorting Array ([A] => ABC [L] => LMN [X] => XYZ)

krsort()	It sorts associative array in descending order as per the keys.	<?php
		$b=
		array("X"=>"XYZ","A"=>"ABC","L"=>"LMN");
		print_r($b);
		krsort($b);
		echo "
		 After Sorting
		";
		print_r($b);
		?>
		Output:
		Array ([X] => XYZ [A] => ABC [L] => LMN)
		After Sorting
		Array ([X] => XYZ [L] => LMN [A] => ABC)

***Note: S**tudents can design HTML form to accept input from the user as per the requirement of the program.

Practice Programs:

1. Write a PHP Script to define an array. Find the element from the array that matches the given value using the appropriate search function.
2. Write a PHP script to count the total number of vowels (a,e, i,o,u) from the string. Show the occurrences of each vowel from the string.
3. Write PHP program to perform the following operations on Indexed Array:
   a) Check the array element is positive or negative
   b) Calculate the average of array elements
   c) Calculate the sum of array elements

SET A:

1. Write PHP program to perform the following operations on Indexed Array:
   a) Union of two arrays
   b) Traverse the array elements in random order
2. Write a PHP program to perform the following operations on an associative array:
   a) Display the elements of an array along with the keys.
   b) Display the size of an array
   c) Delete an element from an array from the given index.
   d) Reverse the order of each element's key-value pair

e) Traverse the elements in an array in random order.

3. Write a PHP Script for the following:
   a) Declare and Display a multidimensional Array.
   b) Search and display a specific element from a Multidimensional array.

SET B:

1. Write a PHP script to perform the following operations on string :
   i) Compare string 2 with string3.
   ii) Convert all the strings to Upper case
   iii) Convert all the strings to Lowercase
2. Write a PHP script to perform the following operations on string :
   i) Convert each word of a string to Lowercase and Uppercase.
   ii) Find the first and last occurrence of string2 in string1.
3. Write a menu-driven program in PHP to perform the following operations on associative arrays:
   i) Sort the array by values (changing the keys) in ascending, descending order.
   ii) Also, sort the array by values without changing the keys.
   iii) Find the intersection of two arrays.
   iv) Find the union of two arrays.

SET C:

1. Write a PHP script to perform the following operations on string :
   i) Replace the string2 by string3 in string1.
   ii) Reverse and display the string.

Assignment Evaluation

0: Not Done [ ] **1: Incomplete [ ]** **2: Late Complete [ ]**
3: Need Improvement [ ] **4: Complete [ ]** **5: Well Done [ ]**

Signature of Instructor

Assignment 4:Functions, Class, and Object

Functions

A function is a block of code that performs a specific task. It can be called from anywhere from the program. It takes zero or any number of parameters and does some processing and returns a value.

PHP Built-in Functions

Name	Use	Example
echo	This construct is used to display many values at once on the screen.	echo “Hello”; echo (“Hello”);
print()	It prints data to the screen	print (“Hello”);
print_r()	It prints the contents of arrays and objects.	<?php $array = array(1, 2, 3); print_r($array); ?> **Output:** Array ([0] => 1 [1] => 2 [2] => 3)
var_dump()	It returns the value and data type of a given variable.	<?php $a=50; echo var_dump($a); ?> **Output:**Int(50);

isset()	It returns true value, if a given parameter is initialized with a value otherwise it returns false value.	<?php
		$n = 0;
		if (isset($n))
		{
		echo "Variable 'a' is set.”;
		}
		Output: Variable 'a' is set.

unset()	It unsets a variable.	<?php $a = 10; echo "The value of variable 'a' before unset: " . $a . " "; unset($a); echo "The value of variable 'a' after unset: " . $a; ?> **Output:** The value of variable 'a' before unset:10

		The value of variable 'a' after unset:
define()	It is used to define constant.	<?php define (“PI”,3.14); echo PI ?> **Output**:3.14.
date()	The date() function formats a local date and time, and returns the formatted date string. Syntax: date(format,timestamp) List of characters commonly used for date: d - Represents the day of the month m - Represents a month Y - Represents a year l - Represents the day of the week	<?php echo "Today is " . date("Y/m/d") . " "; echo "Today is " . date("Y.m.d") . " "; echo "Today is " . date("l"); ?> **Output:** Today is 2021/01/19 Today is 2021.01.19 Today is Tuesday

Defining a user-defined function

While creating a user-defined function its name should be preceded by with keyword **function** and the function code should be put inside { and } braces.

Syntax:

```
functionfunction_name([parameters])
{
Statements;
}
```

Example:

```
<?php
        /* Defining a PHP Function */
    functionHelloWorld()
    {
    echo "HelloWorld Good Morning!!";
    }
        /* Calling a PHP Function */
    HelloWorld();
?>
```

Output: HelloWorld Good Morning!!

Passing Parameters to Functions

1. Call By Value

When a PHP function is called by value then actual values of variables are not modified if it is modified into the function.

Example:

```
<?php
        functionaddFun($num1, $num2)
        {
                $sum = $num1 + $num2;
        echo "Sum of the two numbers is : $sum";
        }
        addFun(15, 20);
 ?>
```

2. Call By Reference

When a PHP function is called by reference then the actual values of the parameters are modified by the function.

Example:

```
<?php
        functionaddFun(&$num1, &$num2)
        {
                $sum = $num1 + $num2;
        echo "Sum of the two numbers is : $sum";
        }
        addFun(15, 20);
 ?>
```

Default Parameter

If we do not pass any value to the function then the function uses a default value calleddefault parameter.

Example:

```
<?php
        functionsetHeight($maxheight=100)
        {
        echo "The height is : $maxheight<br>";
        }
        setHeight(350);
        setHeight(); // will use the default value of 100
?>
```

Output:

```
The height is: 350
The height is:100
```

Classes and Objects:

PHP supports to the object oriented programming concepts.

Class:

It is user defined data type. It is a collection of data members and functionsas a single unit.

A class can be defined as:

Example:

```
<?php
   class Car
   {
        /* Member variables */
   var $price;
        /* Member functions */
   functionsetPrice($par)
   {
         $this->price = $par;
   }
   functiongetPrice()
   {
   echo $this->price ."<br/>";
   }
  }
?>
```

Object:

Any real or runtime entity is called an object. Objects are also known as instance.

Creating Objects in PHP

After defining a class, an object of that class can be created. It can be done by using a new operator as follow:

Example:

```
Object_name=new Class_Name;
$Car1=new Car;
```

For accessing data member and member functions of a class, an object is used.

Example:

```
$Car1->setPrice(5);
```

Practice Programs:

1. Write a PHP script to calculate the area and volume of a cylinder using a function.
2. Write a PHP Script to display the sum and average of array elements(Using predefined functions)
3. Write a PHP script to calculate the factorial of a number using a function.

SET A:

1. Write a PHP script to calculate x^y using a function.
2. Write a PHP script to define a function EvenOdd, which will display even and odd numbers between 1 to 50.
3. Write a PHP script to define a function Maximum, which will accept 3 numbers as parameters and returns a maximum of 3 numbers.
4. Write a PHP script to swap two numbers using a function (Use Call by value and Call by reference)

SET B:

1. Write a PHP Script to create a class Fruit that contains data members as Name, Color and Price. Write a member function to accept and display details of Fruit.
2. Write a PHP Script to create a class Student that contains data members as Roll_Number, Stud_Name, and Percentage. Write member functions to accept Student information.
3. Write a PHP Script to create a class Book (Book_id, Book_name, Publication, Author, Book_price). Write a member function to accept and display Book details.

SET C:

1. Write a PHP script to define a function "DisplayDay", which will display the day of the current date.
2. Write a PHP script to perform arithmetic operations on two numbers. Write a PHP function to display the result. (Use the concept of function and default parameters)

Assignment Evaluation

0: Not Done [ ] **1: Incomplete [ ]** **2: Late Complete [ ]**
3: Need Improvement [ ] **4: Complete [ ]** **5: Well Done [ ]**

Signature of Instructor

Assignment 5: Working with forms

Processing a Form's Data:

HTML forms are used to send the user information to the server and returns the result to the browser. For example, if you want to get the details of visitors to your website, and send them good thoughts, you can collect the user information employing form processing. Then, the information can be validated either at the client-side or on the server-side. The final result is sent to the client through the respective web browser. To create a HTML form, the following **form** tag should be used.

- Action
- Method

Form processing contains a set of controls through which the client and server can communicate and share information. The controls used in forms are:

- Text
- Textarea
- Dropdown
- Radio
- Checkbox
- Buttons

PHP methods used in form processing are:

- **$_GET[]:** It is used the retrieve the information from the form control through the parameters sent in the URL. It takes the attribute given in the URL as the parameter.
- **$_POST[]:** It is used the retrieve the information from the form control through the HTTP POST method. It takes the name attribute of the corresponding form control as the parameter.

Example Using POST Method

stud.html

```
<html>
<body>
<form method=POST action="stud.php">
        RollNo :          <input type=text name=rno><br>
        Student Name : <input type=text name=sname><br>
        Percentage :     <input type=text name=per><br>
                        <input type=submit value=submit name=submit>
</form>
</body>
</html>
```

stud.php

```
<?php
```

```
        echo $_POST[‘rno’];
        echo $_POST[‘sname’];
        echo $_POST[‘per’];
?>
```

Example Using GET Method

stud.html

```
<html>
<body>
<form method=GET action=”stud.php”>
        RollNo :          <input type=text name=rno><br>
        Student Name : <input type=text name=sname><br>
        Percentage :     <input type=text name=per><br>
                         <input type=submit value=submit>
</form>
</body>
</html>
```

stud.php

```
<?php
        echo $_GET[‘rno’];
        echo $_GET[‘sname’];
        echo $_GET[‘per’];
?>
```

PHP function used for form processing:

- **isset():** This function is used to determine whether the variable or a form control is having a value or not.

Example using isset()

stud.html

```
<html>
<body>
<form method=POST action=”stud.php”>
        RollNo :          <input type=text name=rno><br>
        Student Name : <input type=text name=sname><br>
        Percentage :     <input type=text name=per><br>
                         <input type=submit value=submit>
</form>
</body>
</html>
```

stud.php

```
<?php

if (isset($_GET[‘submit’]))
{
        if((!isset($_GET[‘rno’])) ||(!isset($_GET[‘sname’]))|| (!isset($_GET[‘per’])))
        {
                Echo “Please fill all the required fields”;
        }
}
else
{
        echo $_GET[‘rno’];
        echo $_GET[‘sname’];
        echo $_GET[‘per’];
}
?>
```

Self ProcessingPage :

Selfprocessing page means one PHP can be used to both generate a form and process it. PHP_SELF variable is used for self processing page.

PHP_SELF variable returns the name and path of the currently executing script. This variable can used in action attribute of the form.

Example:

```
<form method="Get" action="<?php $_SERVER['PHP_SELF'];?>">
```

Example Self processing page

stud.php

```
<html>
<body>
<form method=GET action="<?php $_SERVER['PHP_SELF'];?>">
        RollNo :          <input type=text name=rno><br>
        Student Name : <input type=text name=sname><br>
        Percentage :     <input type=text name=per><br>
                        <input type=submit value=submit>
</form>

<?php

if (isset($_GET[‘submit’]))
{
        if((!isset($_GET[‘rno’])) ||(!isset($_GET[‘sname’]))|| (!isset($_GET[‘per’])))
        {
                Echo “Please fill all the required fields”;
        }
}
else
```

```
{
        echo $_GET[‘rno’];
        echo $_GET[‘sname’];
        echo $_GET[‘per’];
}
?>
</body>
</html>
```

Sticky Forms:

Form remembers the values that are entered in the input fields. For example Google search box. In the sticky form, the results of a query are accompanied by a search form whose default values are those of the previous query.

To create the sticky form, we have to follow 2 steps:

- Step 1: Taking the data sent by the form by using the “GET” or “POST” method: $data= $_GET[“data”];
- Step 2: Settings that data as a value for text fields and selected or checked for other form elements.

Example Sticky Forms

stud.php

```
<html>
<body>
<form method=GET action="<?php $_SERVER['PHP_SELF'];?>">
        Your Name : <input type=text name=sname value=”<?phpecho$_POST[‘sname’] ?>”>
<br>
                <input type=submit value=submit>
</form>

<?php
        if(isset($_GET[‘sname’]))
        {
                echo $_GET[‘sname’];
        }
?>
</body>
</html>
```

Dealing with Checkbox

```
<html>
<body>
<form method=GET action="<?php echo $_SERVER['PHP_SELF']; ?>">

Select ur Choice<br>
```

```
<input type=checkbox name=ch[] value="1" <?php if($_GET['ch']=="1") echo 'checked="checked"'; ?>>
Reading

<input type=checkbox name=ch[] value="2" <?php if($_GET['ch']=="2") echo 'checked="checked"'; ?>>
Dancing<br>

<input type=Submit name="S" value=Click>
</form>

<?php
	if ((isset($_GET['S'])))
	{
		$ch=$_GET['ch'];
		if($ch=="1")
			echo "Reading";
		else
			echo "Dancing";
	}
?>

</body>
</html>
```

Dealing with Radio button

```
<html>
<body>
<form method=GET action="<?php echo $_SERVER['PHP_SELF']; ?>">
Select ur Choice<br>
<input type=radio name="r" value="1" <?php if($_GET['r']=="1") echo 'checked="checked"'; ?>> Add

<input type=radio name="r" value="2" <?php if($_GET['r']=="2") echo 'checked="checked"'; ?>> Sub<br>

<input type=Submit name="S" value=Click>
</form>

<?php
	if ((isset($_GET['S'])))
	{
		$ch=$_GET['r'];
		switch($ch)
		{
		case 1:
```

```
                        $a=$t1+$t2;
                        echo "Addition=$a";
                        break;
                case 2:
                        $a=$t1-$t2;
                        echo "Sub=$a";
                        break;
                }
        }
?>

</body>
</html>
```

Retrieving values from List:

```
<html>
<body>
<form method=GET action="<?php echo $_SERVER['PHP_SELF']; ?>">
Select ur Choice<br>
<select name=m>
        <option value="R" <?php if($_GET['m']=="R") echo 'selected="selected"'; ?>>Reading
        <option value="D" <?php if($_GET['m']=="D") echo 'selected="selected"'; ?>>Dancing
</select>
<input type=submit name=S value=Click>
</form>

<?php
        if ((isset($_GET['S'])))
        {
                $ch=$_GET['m'];
                echo $ch;

        }
?>

</body>
</html>
```

Validating and Restricting data:

Different strategies for validating form data are,

- Fields should not be empty
- Check the length of the data entered by the user.
- Check the type of data entered by the user.
- Check specific conditions for form fields

Functions for Validating and Restricting data are,

- **Empty(varName):** it is used to check whether a variable is empty or not.

- **isset():** This function is used to determine whether the variable or a form control is having a value or not.
- **filter_var()** function filters a variable with the specified filter.
  Syntax:

  filter_var(var, filtername, options)

  Parameters: This function accepts three parameters and is described below:
  1. **var** : It is the required field. It denotes the variable to filter.
  2. **filtername** : It is used to specify the ID or name of the filter to use. Default is FILTER_DEFAULT, which results in no filtering. It is an optional field.
  3. **options** : It is used to specify one or more flags/options to use. Check each filter for possible options and flags. It is also an optional field.

  Return Value: It returns the filtered data on success, or FALSE on failure.
  Filternames are,
  - FILTER_VALIDATE_INT: to check if the variable is an integer or not
  - FILTER_VALIDATE_IP: to check if the variable is a valid IP address or not.
  - FILTER_VALIDATE_EMAIL: to check if the variable is a valid email address or not.
  - FILTER_VALIDATE_URL: to check if the variable is a valid URL or not.

Example using empty()

```
<html>
<body>
<form method=GET action="<?php echo $_SERVER['PHP_SELF']; ?>">
Search<input type=text name="t1" value="<?php if(isset($_GET['t1'])) echo $_GET['t1']; ?>">
        <input type=Submit name="s" value=Click>
</form>

<?php
        if ((isset($_GET['t1'])))
        {
                $t1=$_GET['t1'];
                if ((!empty($t1)))
                {
                        echo $t1;
                }
                else
                {
                        echo "enter the value in textbox";
                }
        }
?>
</body>
</html>
```

Example using filter_var :

FILTER_VALIDATE_INT

```
<?php
$n = 200;
if (filter_var($n, FILTER_VALIDATE_INT) === 0 ||
   !filter_var($int, FILTER_VALIDATE_INT) === false)
{
   echo("Integer is valid");
}
else
{
   echo("Integer is not valid");
}
?>
```

FILTER_VALIDATE_IP:

```
<?php

$ip = "129.0.0.1";
  if (!filter_var($ip, FILTER_VALIDATE_IP) === false)
{
   echo("$ip is a valid IP address");
}
else
{
   echo("$ip is not a valid IP address");
}
 ?>
```

FILTER_VALIDATE_EMAIL:

```
<?php
$email = "abc@gmail.com";
if (!filter_var($email, FILTER_VALIDATE_EMAIL) === false)
{
   echo("$email is a valid email address");
}
else
{
   echo("$email is not a valid email address");
}
?>
```

FILTER_VALIDATE_URL:

```
<?php
  $url= "https://www.google.com";
if(!filter_var($url, FILTER_VALIDATE_URL) === false)
```

```
{
   echo("$url is a valid URL");
}
else
{
   echo("$url is not a valid URL");
}
?>
```

Practice Programs:

1. To design an application that works as a simple calculator using PHP. (use isset()).
2. Write a PHP script to check PAN number entered by the customer is valid or not and display an appropriate message.
3. Write a PHP script to check mobile number entered by the user is valid or not and display an appropriate message.

SET A:

1. Write a PHP script to accept font name, background color, and welcome message on 1st page. Display the welcome message with the given font and background color on the next page.
2. Write a PHP program to accept name, address, pincode, gender information. If any field is blank display error messages “all fields are required”.
3. Write a PHP script to accept employee details (name, address) and earning details (basic, DA, HRA). Display employee details and earning details in the proper format.

SET B:

1. Write a PHP script to accept customer name and the list of product and quantity on the first page. On the next page display the name of the customer, name of the products, rate of the product, quantity, and total price in table format.
2. Write HTML code to design multiple choice question paper for PHP subject. Display question wise marks and total marks received by the student in table format.
3. Write a PHP script to accept student name and list of programming languages (using drop down box) and display it on the next page in the proper format.
4. Write a PHP script to accept user name, email address and age. If data entered by the user is valid then display it on the next page otherwise display the appropriate message(use filter_var()).

SET C:

1. A web application that takes name and age from an HTML page. If the age is less than 18, it should send a page with "Hello <name>, you are not authorized to visit the site" message, where <name> should be replaced with the entered name. Otherwise, it should send a "Welcome <name> to this site" message.

Assignment Evaluation

0: Not Done [ ]	**1: Incomplete [ ]**	**2: Late Complete [ ]**
3: Need Improvement [ ]	**4: Complete [ ]**	**5: Well Done [ ]**

Signature of Instructor

Assignment 6: Session and Cookies

Cookies:

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

A cookie is created with the setcookie() function.

setcookie(name[, value, expire, path, domain, secure, httponly]);

where,

- **name:** A unique name for a particular cookie. You can have multiple cookies with different
- names and attributes.
- **value :** It is used to set the value of the cookie
- **expire** : The expiration date for this cookie. If no expiration date is specified, the browser
- saves the cookie in memory and not on disk. When the browser exits, the cookie disappears.
- The expiration date is specified as the number of seconds.
- **path :**It is used to specify the path on the server for which the cookie will be available.
- **domain** :It is used to specify the domain for which the cookie is available.
- **secure** : It is used to indicate that the cookie should be sent only if a secure HTTPS connection exists.

```
<?php
        //Creating a cookie
        $cookie_name = "user";
        $cookie_value = "abc";
        setcookie($cookie_name, $cookie_value, time() + (1* 24 * 60 * 60));

        //Checking a Cookie is set or not
        if(!isset($_COOKIE[$cookie_name]))
        {
        echo "Cookie named '" . $cookie_name . "' is not set!";
        }
        else
        {
        echo "Cookie '" . $cookie_name . "' is set!<br>";
        }

        //Accessing Cookie value
        echo "Value is: " . $_COOKIE[$cookie_name];

        // set the expiration date to one hour ago
        setcookie("user", "", time() - 3600);
        echo "Cookie 'user' is deleted.";

        //cookie expire after 1 day
        setcookie($cookie_name, $cookie_value, time() + (1* 24 * 60 * 60));
?>
```

Session:

A session is a way to store information (in variables) to be used across multiple pages. The information is not stored on the user's computer.By default, session variables last until the user closes the browser.Session variables hold information about one single user and are available to all pages in one application

- The first step is to start up a session. After a session is started, session variables can be created to store information. The PHP **session_start()** function is used to begin a new session. It also creates a new session ID for the user.
- **The second step is to set Session variables using** PHP global variable: **$_SESSION.**

```
<?php
        // Start the session
        session_start();

        // Set session variables
        $_SESSION["favcolor"] = "green";
        $_SESSION["favanimal"] = "cat";
        echo "Session variables are set.<br>";

        // access session data
        echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
        echo "Favorite animal is " . $_SESSION["favanimal"] . "<br>";
        print_r($_SESSION);

        // to change a session variable, just overwrite it
        $_SESSION["favcolor"] = "yellow";
        print_r($_SESSION);

        // remove all session variables
        session_unset();
        if(($_SESSION["favcolor"]!=0) && ($_SESSION["favanimal"]!=0))
                print_r($_SESSION);
        else
                echo "Session variables are unset.<br>";

        // destroy the session
        session_destroy();
        print_r($_SESSION);
        echo "Session variables are destroyed.<br>";

?>
```

Practice Programs:

1. A web application that lists all cookies stored in the browser on clicking "list cookies" button,add cookies if necessary.

2. Write a PHP program to store the current date-time in a COOKIE and display the 'Last visited on'date-time on the web page upon reopening of the same page.
3. Write a script to keep track of a number of times the web page has been accessed using the session.

SET A:

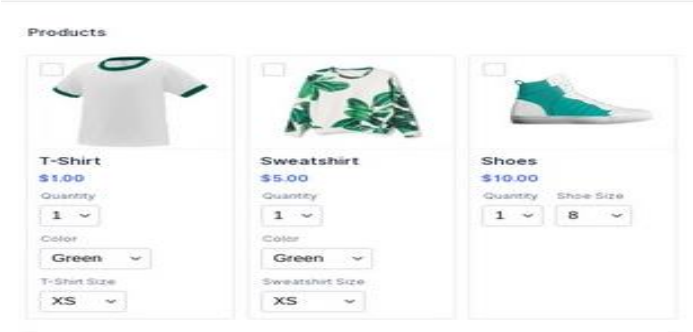
1. Write PHP program to store student information like Seat number, name, and class. On the second page, accept marks of the subject PHP, DS, CPP, and RDBMS. Display Result in table format on the third page (use cookies).
2. Write a PHP script to accept username and password. If in the first three chances, username and password entered is correct, then display the welcome message on the second form, otherwise display an error message.
3. Write a PHP script to accept font style, font size, font color, background color using a cookie. Display selected values on the next second page and actual implementation on the third web page.

SET B:

1. Create an online flight registration form. On the first page accept name, address, birthdate, and mobile number. On the second page accept flight details (flight name, source, destination, departure date-time and charges). If the user doesn't enter information within a specified time limit, expire his session and give a warning otherwise display details using sessions on the third page**.**
2. Create a form to accept patient details like name, address birthdate, and mobile number. Once thePatient information is accepted, and then accepts health details like medicare number, health fund and critical information. Display patient details and health details on the next form.
3. Write a PHP script to create an inventory management system. On the first page accept the highest sold product details like product name, total quantity and total sold. On the second page accept the latest sales details like product name, date and total sale. Display highest sold product details in one table and latest sales details in another table on the third page.

SET C:

1. Write a PHP script to create an online shopping form. On the first page accept customer name, email address, shipping address, mode of payment. Design the Second page as given below. And the third page should display a bill, which consists of customer details and purchase details in the proper format.



Assignment Evaluation

0: Not Done [ ] **1: Incomplete [ ]** **2: Late Complete [ ]**
3: Need Improvement [ ] **4: Complete [ ]** **5: Well Done [ ]**

Signature of Instructor

Assignment 7: Database

Database :

It is a collection of inter-related data that helps in efficient retrieval, insertion, and deletion of data from the database and organizes the data in the form of tables, views, schemas, reports, etc. The basic functions used in PHP for database connection are,

Function	Description	Example
mysql_connect(server, user, pwd)	It opens a database connection and returns the connection on success, or FALSE and an error on failure.	$con=mysql_connect("localhost","root","");
mysql_select_db(db name)	It is used to change the default database for the connection.	mysql_select_db("sybba");
mysql_query(query);	It executes a query on a MySQL database.This function returns the query handle for SELECT queries, TRUE/FALSE for other queries, or FALSE on failure.	$sql=”Select * from stud” $r=mysql_query($sql);
mysql_fetch_array(result,resulttype);	function fetches a result row as an associative array, a numeric array, or both. result=Required. resulttype=Optional. Specifies what type of array that should be produced. Values are, MYSQL_ASSOC, MYSQL_NUM, MYSQL_BOTH	mysql_fetch_array($r, MYSQL_ASSOC);
mysql_close(connection);	The mysql_close() function closes MySQL connection.This function returns TRUE on success, or FALSE on failure.	mysql_close($con);

1. The basic steps to create a MySQL database using PHP are:
   - Establish a connection to the MySQL server from your PHP script.
   - If the connection is successful, write a SQL query to create a database and store it in a string variable.
   - Execute the query.
   - Close the connection

```
<?php
        $con=mysql_connect("localhost","root","");
```

```
        if(!$con)
        {
        die("unable to connect");
    }
    $sql="create database sybba";
$r=mysql_query($sql);
if(! $r)
    {
        die("could not create database");
    }
echo "Database created successfully";
mysql_close($con);
?>
```

2. The basic steps to create a MySQL table using PHP are:
    - Establish a connection to the MySQL server from your PHP script.
    - If the connection is successful, then select the database.
    - Write a SQL query to create a table and store it in a string variable.
    - Execute the query.
    - Close the connection

```
<?php
        $con=mysql_connect("localhost","root","");
        if(!$con)
        {
        die("unable to connect");
    }
mysql_select_db("sybba");

    $sql="create table stud(rnoint, snamevarchar(20), per int)";
$r=mysql_query($sql);
if(! $r)
    {
        die("could not create table");
    }
    echo "Table created successfully";
mysql_close($con);
```

3. The basic steps to manipulate MySQL table using PHP are:
    - Establish a connection to the MySQL server from your PHP script.
    - If the connection is successful, then select the database.
    - Write aninsert/update/delete query to manipulate and store it in a string variable
    - Execute the query.

- Close the connection

```
<?php
        $con=mysql_connect("localhost","root","");
        if(!$con)
        {
        die("unable to connect");
      }
mysql_select_db("sybba");

      $sql="insert into stud values(1,'Neeta',84)";
$r=mysql_query($sql);
if(! $r)
      {
          die("not inserted");
      }
      echo "record added successfully";
mysql_close($con);
?>
```

4. The basic steps to fetchdata from MySQL table using PHP are:
    - Establish a connection to the MySQL server from your PHP script.
    - If the connection is successful, then select the database.
    - Write a select query and store it in a string variable
    - Execute the query
    - Display data using a while loop.
    - Close the connection

```
<?php
        $con=mysql_connect("localhost","root","");
        if(!$con)
        {
        die("unable to connect");
        }
        mysql_select_db("sybba");
        $result=mysql_query("select * from stud");
while($col=mysql_fetch_array($result,MYSQL_NUM))
        {
                echo "Rollno=".$col[0]."<br>";
                echo "Name=".$col[1]."<br>";
                echo "Per =".$col[2]."<br>";
        }
        mysql_close($con);
?>
```

Practice Programs:

1. Consider the following entities and their relationships
   Company (c_no, c_name, c_city, c_share_value)
   Person (p_no, p_name, p_city, p_ph_no)
   Relationship between Company and Person is many-to-many with descriptive attribute no_of_shares.
   Using the above database, write a PHP script to display person wise share detailsin tabular format.

2. Consider the following entities and their relationships
   Customer (c_no, c_name, c_city, c_ph_no)
   Ticket (t_no, booking_date, fare, traveling_date)

The relationship between Customer and Ticket is one-to-many. Create a RDB in 3 NF for the above.
Using the above database, write a PHP script to accept date and display,
1) The total fare collected from customers on a given date.
2) Ticket details booked by the customer.

SET A:

1. Write a PHP script to create an employee table using attributes employee number, employee name, address joining date and salary. If a table is created then display the appropriate message otherwise end the PHP script.
2. Write a PHP script to accept account details (account number, account type and balance). Store these details in the account table and display an appropriate message.
3. Write a PHP script to accept product number from the user. Update the price of the product and display an appropriate message.

SETB:

1. Consider the following entities and their relationships.
   Employee (eno, ename, sal)
   Project (pno, pname,duration)
   Employee and Project are related with a many-many relationship. Create a RDB in 3 NF for the above.
   Using the above database write a PHP script to accept the project name. Display the name of the employees and the duration of the project.
2. Consider the following entities and their relationships.
   Train(t_no, t_name)
   Passenger (p_no, p_name, addr, age)
   The relationship between Train and Passenger is many-to-many with descriptive attribute date,seat_no and amt. Create a RDB in 3 NF for the above.
   Using the above database write a PHP script to accept a date. Display train details having maximum passenger for a given date.
3. Consider the following entities and their relationships.
   Crop (c_no, c_name, c_season, pesticides)

Farmer (f_no, f_name, f_location)

The relationship between Crop and Farmer is many-to-many with descriptive attribute year.

Create a RDB in 3 NF for the above.

Using the above database write a PHP script to accept crop name and year value. Display total number of farmers harvesting given crop in a given year.

SETC:

1. Consider the following entities and their relationships.
   Client (c_no, c_name, c_addr, birth_date)
   Policy_info (p_no, p_name, maturity_amt, prem_amt, policy_term)
   The relationship between Client and Policy_info is many-to-many with descriptive attribute date_of_purchase.Create a RDB in 3NF for the above.
   Using the above database write a PHP script to display policy details of a given client for a given year in the following format.

Client Name : **Year:**

Policy Name	Maturity Amount	Premium Amount	Policy Term	Date of Purchase

Assignment Evaluation

0: Not Done [ ] **1: Incomplete [ ]** **2: Late Complete [ ]**
3: Need Improvement [ ] **4: Complete [ ]** **5: Well Done [ ]**

Signature of Instructor