

BCA- Semester-6



CC-310

Web Applications Development II

Practical

- PHP Basics
- PHP and MYSQL Programming
- AJAX and Validation
- AJAX, PHP, MSQQL all Combined & Project



A Division of Live Education System Pvt. Ltd.

Index

UNIT-1 PHP Basics and Form Handling 07 to 72

- 1.1 Different Kind of Programming Languages
- 1.2 What is PHP?
- 1.3 Apache Web Server and It's Flavours
- 1.4 PHP Syntax and Technics
- 1.5 PHP printing statement
- 1.6 PHP Comments
- 1.7 PHP Variable, Data type and Scope of Variable
- 1.8 Operators in PHP
- 1.9 Flow Control Statements
- 1.10 Array in PHP
- 1.11 PHP Function
- 1.12 PHP Form Handling and PHP Global
- 1.13 Include and require function to include a PHP file in other PHP file
- 1.14 PHP Date and Time function
- 1.15 PHP Maths functions
- 1.16 PHP String Functions

Practical

1. Write a program in PHP to display "Learning PHP" in bold format.
2. Write a program in PHP to demonstrate the use of comments, echo and print.
3. Create a program in PHP to demonstrate the use of If ... Else and switch statements.
4. Create an array named \$sub, assign five elements to it and display the elements assigned using for loop and for each statement.

5. Create an array named \$student that stores 5 elements bounded to different keys and access the same using the key element.
6. Write a program in PHP to demonstrate the use of multidimensional arrays.
7. Create two functions in PHP, parameterized and non-parameterized for implementing string concatenation operation.
8. Write a PHP program to display information of PHP in the browser.
9. Write a program in PHP to sort the array of given 5 numbers in ascending and descending order.
10. Write a program to count the total number of times a specific value appears in an array.

UNIT-2 Problem Solving by Searching

73 to 120

- 2.1 How to Create Database in MySQL
- 2.2 Different Data types in Database
- 2.3 How to create table in MySQL
- 2.4 Different PHP Functions related to MySQL Database
- 2.5 PHP CRUD Operations

Practical

1. Create a form containing two input fields (Name, Email ID) and a submit button. When the user clicks on submit button, the form data should be sent for processing to PHP file ,which should display the welcome message with the email_id on the PHP page. Form data should be sent by HTTP GET/POST method.
2. Write a PHP script for that creates a database named "DB-1" in MySQL.
3. Write a PHP script for creating a product table in the specified database with fields Pro_id, Pro_name, Pro_price, QOH. Also display an acknowledgement for the same.
4. Create a form containing four input fields(Pro_id, Pro_name, Pro_price, QOH) and Submit button. When the user clicks on the submit button an PHP script should be executed which inserts the record in the product table.

5. Create a form containing one input field(Pro_id) and a search button. When the user clicks on the Search button a PHP script should get executed and should display the details of the product for the Pro_id specified.
6. Create a form containing two input fields (Pro_id, QOH) and Update button. When the user clicks on the Update button the quantity of the Pro_id specified should get updated using a PHP script.
7. Create a form containing one input field(Pro_id) and a Delete button. When the user clicks on the Delete button a PHP script should get executed and should delete the record of the product for the Pro_id specified.

UNIT-3 AJAX and Validation

121 to 160

- 3.1 What is AJAX?
- 3.2 AJAX Request Object
- 3.3 XMLHttpRequest Methods
- 3.4 XMLHttpRequest Object Properties

Practical

1. Create a form containing one input field (Name). When the user enters his/her name and as any key is released , the form should display a welcome message for the user.Implement using AJAX.
2. Repeat the above question to demonstrate the use of keydown and keypress events.
3. Write a program for converting a string into uppercase using AJAX.
4. Create a form containing a combobox with some product names as items. Whenever a user selects a particular product from the combobox, it shuold be sent to the server asynchronously (i.e. without pressing submit button). Implement using AJAX.
5. Write a program to demostrate the example of sending items selected from radio and checkbox to server asynchronously.
6. Write a program to validate a blank field and also validate the length of the data entered(i.e. minimum lenght of 5).

7. Write a program to validate an Email ID using regular expression and by using DOM.
8. Write a program that checks a particular stuId already exists in the student(stuId,stu_name,mob,country) table or not. If stuId exists then display a message. "User Already Exist. Try another stuId". If it does not exist then add the data in the student table. Implement using AJAX.

UNIT-4 AJAX, PHP, MSQQL all combined and Project

161 to 177

- 4.1 Project Management
- 4.2 Modules of E-commerce Application
- 4.3 Logical Steps with screen shot for Admin Module
- 4.4 Logical Steps with screen shot for UserModule

Practical

1. Create a Shopping Cart Project in PHP Script.
2. Create a Shopping Cart Project in AJAX.

UNIT-1

PHP Basics and Form Handling

1. Write a program in PHP to display "Learning PHP" in bold format.
2. Write a program in PHP to demonstrate the use of comments, echo and print.
3. Create a program in PHP to demonstrate the use of If ... Else and switch statements.
4. Create an array named \$sub, assign five elements to it and display the elements assigned using for loop and for each statement.
5. Create an array named \$student that stores 5 elements bounded to different keys and access the same using the key element.
6. Write a program in PHP to demonstrate the use of multidimensional arrays.
7. Create two functions in PHP, parameterized and non-parameterized for implementing string concatenation operation.
8. Write a PHP program to display information of PHP in the browser.
9. Write a program in PHP to sort the array of given 5 numbers in ascending and descending order.
10. Write a program to count the total number of times a specific value appears in an array

Types of Operator:

- Arithmetic
- Comparison
- Logical
- Assignment
- Conditional
- Increment/Decrement
- String

1.8.1 Arithmetic Operator:

Arithmetic Operator is used to do some arithmetic Operations in program like addition, subtraction, multiplication, division, modulation. This operator is belonging to binary operator where two operands and one operator are used.

Operator	Description	Example
+	Used for Addition/Sum of two number or two variables.	10+90; \$a+\$b;
-	Used for Subtraction of two number or two variables.	100-50; \$a-\$b;
*	Used for Multiplication of two number or two variables.	50*50; \$a*\$b;
/	Used for Division of two number or two variables.	50/2; \$a/\$b;
%	Used for get Reminder of two number or two variables.	50%4; \$a%\$b;

```
<?php
$a=40;
$b=60;
$sum=5+10;
$subtraction=$a-$b;s
$multiply=5*$b;
$division=$b/3;
$modulo=$a%3;
echo("Sum=$sum <br>");
echo("Subtraction=$subtraction <br>");
```

```

echo("Multiplication=$multiply <br>");  

echo("Division=$division <br>");  

echo("Modulation=$modulo <br>");  

?>

```

1.8.2 Comparison Operator:

This Operator is used to compare two operands and return logical value based on comparison. It return Boolean value based on Comparison (True/False). We can use any numerical or string operand to compare. Following are the Comparison Operators. It is also belonging to binary operator where two operands and one operator are used.

Operator	Description	Example
Double Equals (==)	Used for compare two operand. Returns true if both are equal. This operator ignore the data type of variable and value at the time of Comparison	\$a=3;\$b=3 \$a==\$b; true \$a=='3'; true \$a=="3" ;true
Not Equals (!=) Not Equals (<>)	Used for compare two operands. Returns true if both are not equal.	\$a=3;\$b=4; \$a!=\$b; true \$a<>\$b; true
Equals(==)	Used for compare two operands. Returns true if both are equal. This operator compares data type of variable and value at the time of Comparison, if both are same then it will return true.	\$a=5;\$b=5; \$a==\$b; true \$a=="5";False
Greater than (>)	This operator is used to compare two operands. It will return true if left operand vale is greater than right operand value;	\$a=20; \$b=10; \$a>\$b; true.
Less than (<)	This operator is used to compare two operands. It will return true if left operand vale is less than right operand value;	\$a=5;\$b=20; \$a<\$b; true.
greater Equals (>=) or	This Operator is used to compare greater and equals both at same time. It will return true if left operand value is either greater than or equals to the right operand value	\$a=10; \$b=4; \$a>=\$b; true; \$a>=10; true;
Less Equals (<=) or	This Operator is used to compare less and equals both at same time. It will return true if left operand value is either less than or equals to the right operand value.	\$a=10; \$b=5; \$b<=\$a; true; \$a<=10; true;

```

<?php
    $a=20;
    $b=10;
    if($a>=$b)
    {
        echo("A is greater or Equal to B");
    }
    else if($a<=$b)
    {
        echo("B is Greater or Equal to A");
    }
    else if($a===$b)
    {
        echo("A and B are same and their Data types are also same");
    }
    else if($a===$b)
    {
        echo("A and B are same but their Data types are not same");
    }
    else
    {
        echo("A and B are not same");
    }
?>

```

1.8.3 Logical Operator:

Logical Operator is used to check logical value. They return Boolean value. It is used to compare two different Logic or Conditions.

Operator	Description	Example
Logical And (&)	This operator returns true if both expressions are fulfil it's condition.	\$a>\$b && \$a>\$c;
Logical Or ()	This operator returns true if either first or second one of the expression is fulfil it's condition.	\$a>\$b \$a>\$c;
Logical Not (!)	This operator is reversing the Boolean result of the Operand or Condition.	!(\$a>\$b) return true !(\$a<\$b) return false

```

<?php
    $a="";
    $b="";
    if($a>100 || $a<=200 && $a!=0)
    {
        echo ("A is between 100 to 200");
    }
    if($b>=50 && $b<=100)
    {
        echo ("$b is between 50 to 100");
    }
    if(!$a && !$b)
    {
        echo ("A and B is Empty");
    }
?>

```

1.8.4 Assignment Operator:

This operator is used to assign value to a variable. It will always assign new value in left side of operator variable. We can also use this operator to assign some computed Expression result value to the variable.

Operator	Description	Example
Equalsto (=)	This operator is used to assign direct value to the left side operand. It is also used to assign some Computed Expression result to the left side operand.	a=3;b=3 a=b+10;
Plus and Equals to (+=)	This operator is first of all Add value of left side and right side operand and then assign result to left side operand. It is equivalent of $a=a+b$;	a=3;b=4; a+=b; now a=7
Minus and Equals to (-=)	This operator is first of all Minus value of left side and right side operand and then assign result to left side operand. It is equivalent of $a=a-b$;	a=5;b=5; a-=b; now a=0;
Minus and Equals to (*=)	This operator is first of all Multiply value of left side and right side operand and then assign result to left side operand. It is equivalent of $a=a*b$;	a=5;b=5; a*=b; now a=25;
Minus and Equals to (/=)	This operator is first of all Divide value of left side and right side operand and then assign result to left side operand. It is equivalent of $a=a/b$;	a=5;b=5; a/=b; now a=1;

Minus and Equals to (%)	This operator is first of all Modulo value of left side and right side operand and then assign result to left side operand. It is equivalent of $a=a*b;$	a=5;b=5; a% = b; now a=0;
-------------------------	--	---------------------------------

```
<?php
    $a=20; $b=10; $c=0;
```

```
    $c+=$a;
    echo("sum=". $c. "<br>");
    $c-=$b;
    echo("Subtraction=". $c. "<br>");
    $b*=$a;
    echo("Multiplication=". $b. "<br>");
    $b/=$c;
    echo("Division=". $b. "<br>");
    $b% = 3;
    echo("Modulo=". $b. "<br>");
```

```
?>
```

1.8.5 Conditional or Ternary Operator:

Ternary operator starts with conditional expression followed by ? operator. Second part (after ? and before : operator) will be executed if condition turns out to be true. If condition becomes false then third part (after :) will be executed.

```
<?php
    $a = 10; $b = 5;
    $c = $a > $b ? $a : $b;
    echo("Value of C=$c");
?>
```

1.8.6 Increment or Decrement Operator:

These operators are also known as unary operator, means it has only single operand. This operator will increment or decrement the value of operand by 1, if you use ++ sign then it will add 1 in actual value, if you used -- then it will decrease the value by 1 in actual value. We can add this operator before or after the operand.

- If you use the ++ operator as prefix like: ++variable. The value of variable is incremented by 1 then, it returns the value.
- If you use the ++ operator as postfix like: variable ++. The original value of variable is returned first then, variable is incremented by 1.

```
<?php
```

```
$a=5;
```

```
echo $a++;
```

```
$c=$a;
```

```
echo $c;
```

```
--$c;
```

```
$d=$c;
```

```
echo($d);
```

```
?>
```

1.8.7 Concatenation operator:

In PHP for merge two string , to merge string with number or to merge string with variable we use ‘.’ operator. This operator always merge string data with other data types.

```
<?php
```

```
echo ("hello". "Student");
```

```
echo ("Your Registration no=".123);
```

```
$a=10; $b="Diya Patel";
```

```
echo($b." your age=". $a);
```

```
?>
```

1.9 Flow Control Statements:

Flow controls is one of the powerful tools in all programming language and it is also used in PHP .Flow control statements are used to build intelligent and logic in your web page .It is used to constructing a script for decisions and repeating a task until specified criteria are not met. It is always return Boolean value while it fulfils the criteria. PHP use two types of Flow Control Statements.

1. Conditional Flow Control Statements.

2. Looping Flow Control Statements.

1.9.1 Conditional Flow Control Statements:

This types of flow control use with decision. In our daily life we have to make decisions. Like which dress to wear, which task we have to give first priority and many more. Such kind of decision statements also used in programming language as well as scripting language. Decision making statement always return Boolean value either true or false depend on that program flow will execute. In PHP we use 4 different types of conditional statements:

1. if
2. if.... Else
3. nested if
4. Switch

1) if Statement:

The if statement is a fundamental conditional statement in programming language. This statement is verified a condition or set of conditions. After verification it defines certain statements or block of statements will be executed or not. If certain condition is true then block of statement inside that condition is execute otherwise not execute.

Syntax:

```
If(condition)
{
    Body of if block...
}
```

```
<?php
    $age=65;
    if($age>60)
    {
        echo("Age is greater than 60");
    }
?>
```

2) if ...else Statement:

It is a similar like If statement. But in If statement only one block of If condition is execute. If condition return false then it is execute second block of statement where we use else block. This will execute while if block return false, means else block return true. It is always used in group. It will execute else block bases on true and false of if condition.

Syntax:

```
if(conditional statements)
{
    Body of if block
}
else
{
    Body of else block
}
```

```
<?php
    $A=5;
```

```

if($A%2==0)
{
    echo("A is Even no");
}
else
{
    echo("A is Odd no");
}
?>

```

3) Nested if Statement:

PHP allows us to add if statement inside another if or else statements. It is called nested if statement. We can use more than one if statements inside another if statements. It will check conditions one by one in order of nesting. Nested if is also used to check dependent conditions.

Syntax:

```

if (condition)
{
    if (condition)
    {
    }
}

```

else

```

{
    if (condition)
    {
    }
}

```

<?php

\$A=12;

\$B=5;

```

if ($A!=$B)
{

```

```

    echo("A and B are not same<br>");
    if($A>$B)
    {

```

```

        echo("A is greater than B");
    }
    else
    {

```

```

        echo("B is greater than A");
    }
}
else
{
    echo("A and B are same");
}
?>

```

4) Switch Statement:

In previous topic we learn to check multiple conditional statements, but it quite difficult to develop a nesting structure of multiple conditions. There is another option we have to check multiple conditional statements and it is easy to develop. Switch statement compare variable with multiple case. Once the case is match that block of statements will execute. In program we need to define one variable which use as an argument of switch() that variable value is match with all the cases under switch statement. Each case is design with different number or characters. It is used as matching of variable value. If Case is not match then default block is execute.

Syntax :

Switch(expression)

```

{
    case value1:
        statements;
        break;
    case value2:
        statements;
        break;
    default:
        statements;
}
```

```

<?php
$sw='B';
switch($sw)
{
    case 'A':
        echo("Select Block A");
        break;
    case 'B':
        echo ("Select Block B");
        break;
    case 'C':

```

```
echo ("Select Block c");
echo ("case not match in default block");
```

default:

```
echo ("case not match in default block");
```

}**?>**

1.9.2 Looping Flow Control Statements:

In PHP we can use different four types of looping statements. This four types are derived in two different categories Exit Control loop and Entry control loop. Exit control loop means such looping statement will check the condition of loop at the time of exit from loop body. If condition is true then it will come again in loop body. IN this category the body of loop will execute at least one time. Entry control loops are such looping statement will check the condition of loop at the time of entry in to the loop body. if condition is true then it will enter into the body of loop otherwise it will not go inside the body and stop the looping.

Exit control loop is Do while loop in this loop the body of loop will execute first then it will check while condition if it is true then it will come again in to the body otherwise it will come out from loop and execute next statement of program.

Exit Control Loop

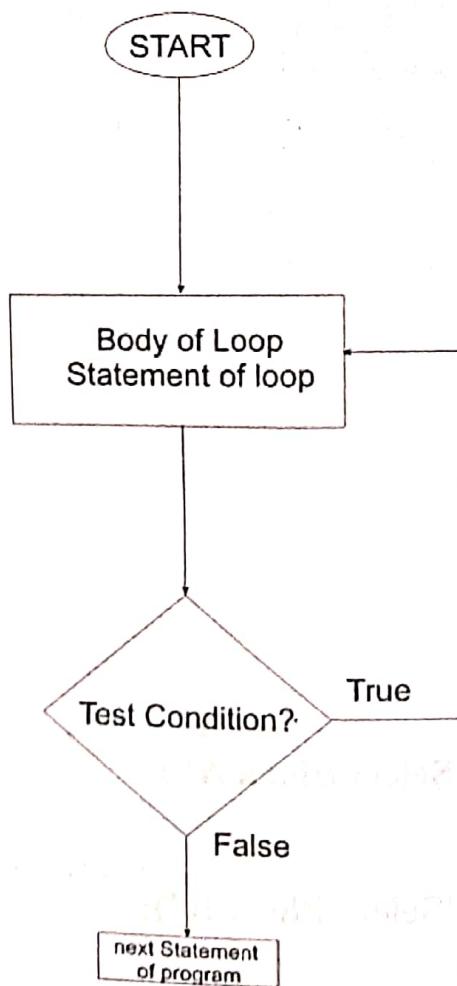
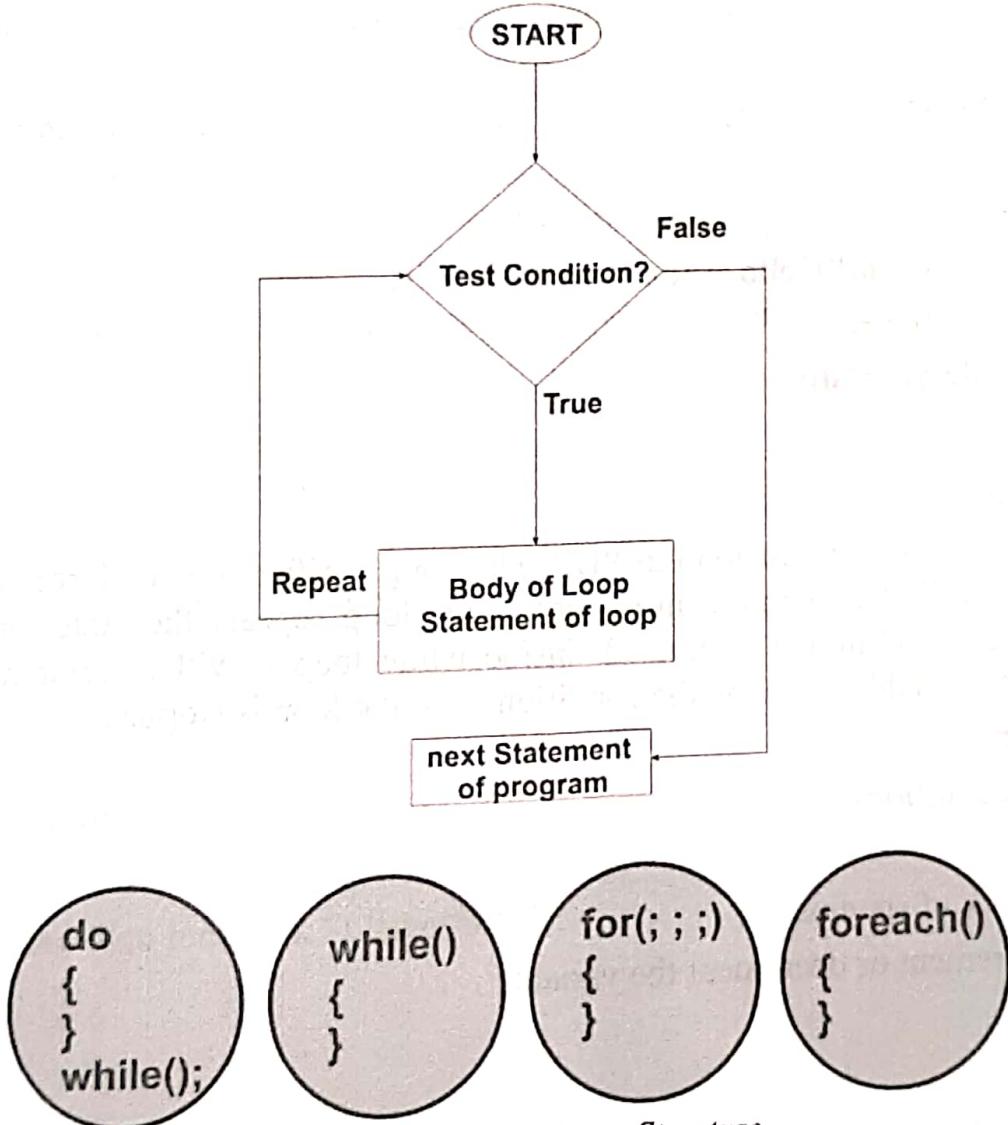


Fig 1.5 Exit Control Loop Structure.

Entry control loops are while loop, for loop and foreach loop. In these loops the body of loop will not execute first, it will check condition if it is true than it will enter in to the body otherwise it will come out from loop and execute next statement of program. Every time it will check condition first then execute body of loop.

Entry Control Loop

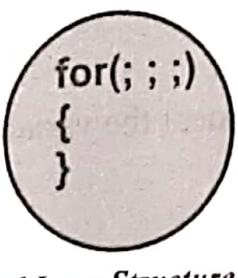
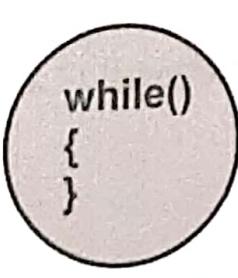


- **Do While:**

Do while loop is evaluate the condition at the end of each loop iteration. In do while loop it will execute block of code at least one time. Then the condition is evaluated, if the condition is true, the statement is repeated as long as the specified condition evaluated to is true. Before while condition inside do it will increment or decrement the value of variable until the condition become true.

Syntax:

do



Block of code execute repeat

Increment or decrement;

}

while(condition);

<?php

\$i=1;

do

{

echo("Hello student no \$i
");

\$i++;

}while(\$i<=10);

<?

- **While:**

While loop is simplest loop in PHP. This loop verifies the condition at the time of entry so it is called as entry control loop. This loop repeats the statements as long as the specified condition is evaluated. Inside while loop it will increment or decrement the value of variable. As soon as the condition fails, the loop is stopped.

Syntax:

while(condition)

{

Block of statements;

Increment or decrement the variable

}

<?php

\$i=1;

while(\$i<=5)

{

echo("PHP while loop \$i
");

\$i++;

}

?>

- **For:**

For loop is entry control loop means it will execute the block of statement after checking the conditional statement. In for loop statement divide in three different parts. First part is initialization, second is conditional part and third is increment or decrement part.

First two parts execute then if condition return true then loop transfer control inside body of loop. After execution of body third part is execute. Each three parts are separated by semicolon (;).

Initialization part: -initialization part is used to initialize the variable with value.

Condition part:- in condition part it will check the initialize variable met the condition, if it return true then control is transfer to the body of loop otherwise it will stop the execution of loop.

Increment/Decrement: -in this part it will increment or decrement the value of initialize variable.

Syntax:

```
for(initialization;condition;increment)
```

```
{
```

Body of for loop execute statements.

```
}
```

```
<?php
```

```
for($i=0;$i<5;$i++)
```

```
{
```

```
    echo("Foo loop run $i time<br>");
```

```
}
```

```
?>
```

- **Foreach:**

This loop is used with only array data type in PHP. It is easy to traverse in array using this loop. It will select each element of array instead of their index or key and access the value. In foreach loop we do not need to increment the value. There are two argument pass in foreach, first one is an array variable which we want to access and second is user define variable which will access and receive the value of all the array elements.

There are two syntaxes are used.

Syntax :

```
foreach(array variable as user define variable)
```

```
{
```

```

}

foreach(array variable as user define key=> value)
{
    // access both key & value
}

}

<?php
    $a=array(1,2,3,4,5);
    foreach($a as $ans)
    {
        echo($ans."<br>");
    }

    $product=array("name"=>"mouse","color"=>"black","price"=>150);
    foreach($product as $p=>$v) //access key and value both.
    {
        echo("[ ".$p." ]=>".$v."<br>");
    }

    foreach($product as $p=>$v)
    {
        echo($p."<br>"); //print only key part of array
    }

    foreach($product as $v) //access only value part of array
    {
        echo($v."<br>");
    }
?>

```

1.10 Array in PHP:

As we know in real life data are come in multiple formats. These all data are store under common variable. In PHP array variable is used to store such kind of data in single variable. PHP array is different than other programming language like c, c++, java etc. in PHP array is declare without data type, means like PHP variable no need to define any data type with array variable also. As we know array is store only similar data type values but in PHP we can store any data type value in array, because there is no fix data type at the time of declaration of array variable. We can declare array in two different ways. we can use short syntax like variable name then initialize values of array in square bracket.

Syntax:

```
$arr=[1,3,5,7,9];
```

```
$a=['a','b',11,22,"abcd"];
```

We can use array function to declare array in PHP. In this function we will pass the values of array. It is the most popular method to declare an array in php.

Syntax:

```
$arr=array(1,2,3,4,5);
```

```
$std=array("miten","miten@gmail.com",30,'M',"Ahmedabad");
```

There are three types of array in PHP

- Index Array
- Associative Array
- Multidimensional Array

1.10.1 Index Array:

This array is use numeric index to store a value. It store value in linearly. the first value of index array is place on 0 index then second value is on 1 index and so on up to n umber index will define automatic based on number of values in array.

We can access index array value by using index of array element.

For example:

```
<?php  
$arr=array(11,22,33,44,55,66);  
Echo $arr[3] ;  
?>
```

In above example array is define with \$arr and there are 6 elements in array first element on 0 index and so on. I have access 3rd index array element so it will print 44 on screen.

1.10.2 Associative Array:

In PHP this array is flexible to store value. This array is use key name as an index of array and with key it store value. It is a pair of key and value. It is define with => equals to and greater than sign. Left side of this sign we define key and right side of this sign we define value. For example we store carname and price in array then we define array as follow.

```
$carlist=array("ertiga"=>900000,"i20"=>600000,"scorpio"=>1000000);  
If we want to access this car then we access it with its key name, it will give value of that key name. for example we print this array.  
echo $carlist["ertiga"];
```

```
echo $carlist["i20"];
echo $carlist["scorpio"];
```

when we access array in above method then it will print the value. echo \$carlist["ertiga"]; will print 900000.

For example:

```
<?php
```

```
$student=array("name"=>"Suhit patel","age"=>32,"height"=>5.4);
echo $student["name"];
echo $student["age"];
echo $student["height"];
```

```
?>
```

1.10.3 Multi-dimensional Array:

This array is also using associative array. In this array we develop other nested array. This array allows us to store relative data in common group. Sometimes when we want to store more than one key with different values we use multi-dimensional array. In multi-dimensional array dimensions are used for indices. If you use two dimensional array then there are two indices are use. In PHP multi-dimensional array can work with index as well as with key, in bellow example Columns name, Salary and designation are key and ram, 2000, Manager are on index 0 or in row 0.

Syntax:

```
$arr=array("key1"=>array("value1","value2","value3"),
"key2"=>array("v1","v2","v3"));
```

Row/Colum	Colom 1	Colom 2	Colom 3
	Name	Salary	Designation
Row 0	Ram	20000	Manager
Row 1	Mohan	15000	Sr.Developer
Row 2	Shiva	30000	Team Leader

```
<?php
```

```
$emp=array("Name"=>array("ram","Mohan","Shiva"),
"Salary"=>array(10000,20000,30000),
"Designation"=>array("Manager","Sr.Developer","Team Leader"));
echo $emp["Name"][1];
echo $emp["salary"][1];
```

```
echo $emp["Designation"]的文化
```

?>

In above example we have make two dimensional array where name and salary are the colom name and ram,20000 is the row 1 record. When we access name and salary we define \$emp["Name"]的文化 where Name is Colom name, and 1 is an index of record.

1.10.4 Important inbuilt functions in Array.

- **print_r()** : This function is used to print hole array with its index and value if your array is associative array then it will print hole array with key and value.we will pass an array variable as an argument of print_r() function.we can use this function for code tracing, if we fetch record from data base then we want to confirm that data are fetch in order, then we print hole array using this function.

```
<?php
```

```
$arr=array(1,2,3,4,5,6);
```

```
print_r($arr);
```

?>

- **count()**: This function is used to count the total no of elements in array.it is also used to get the length of array. In count() function we pass array variable as an argument.

```
<?php
```

```
$a=array(1,2,3,4,5,6,7);
```

```
$len=count($a);
```

```
echo "<br>".$len;
```

```
for($j=0;$j<count($a);$j++)
```

```
{
```

```
echo $a[$j];
```

```
}
```

?>

- **sizeof()**: This function is used to count the total no of elements in array.it is also used to get the length of array. It is an alias of count() function. In sizeof() function we pass array variable as an argument.

```
<?php
```

```
$a=array(1,2,3,4,5,6,7);
```

```
$len(sizeof($a));
```

```
echo "<br>".$len;
```

```
for($j=0;$j<sizeof($a);$j++)
```

```
{
```

```
echo $a[$j];
}
?>
```

- **sort()**: This function is used to sort an index array in ascending order. If your value is alphabets in array then it will sort array in ascending, all elements in a to z format. In sort() function we pass array variable as an argument.

```
<?php
$a=array("Mobile","Ac","Television","Computer");
sort($a);
print_r($a);
?>
```

- **rsort()**: This function is used to sort an index array in descending order. If your value is alphabets in array then it will sort array in descending, all elements in z to a format. In rsort() function we pass array variable as an argument.

```
<?php
$a=array("Mobile","Ac","Television","Computer");
rsort($a);
print_r($a);
?>
```

- **asort()**: This function is used to sort an associative array in ascending order. It will observe only value part and make sorting, it will ignore key at the time of sorting. If your value is alphabets in array then it will sort array in ascending, all elements in a to z format. In asort() function we pass array variable as an argument. If your array is mixing of all data type value then it will first sort string or character then float or double or integer value in ascending order.

```
<?php
$emp=array("name"=>"Ram","designation"=>"Accountant","city"=>"Mumbai");
asort($emp);
print_r($emp);

$std=array("fullname"=>"Satisfi","Age"=>22,"height"=>5.10,"Gender"=>'M');
asort($std);

foreach($std as $x=>$x_value)
```

```
{  
echo "Key=" . $x . ", Value=" . $x_value;  
echo "<br>";  
}  
?>
```

- **arsort()**: This function is used to sort an associative array in descending order. It will observe only value part and make sorting, it will ignore key at the time of sorting. If your value is alphabets in array then it will sort array in descending, all elements in z to a format. In arsort() function we pass array variable as an argument. If your array is mixing of all data type value then it will first sort string or character then float or double or integer value in ascending order.

```
<?php  
$emp=array("name"=>"Ram","designation"=>"Accountant","city"=>"Mumbai")  
;  
arsort($emp);  
print_r($emp);  
  
$std=array("fullname"=>"Satish","Age"=>22,"height"=>5.10,"Gender"=>'M');  
arsort($std);  
  
foreach($std as $x=>$x_value)  
{  
echo "Key=" . $x . ", Value=" . $x_value;  
echo "<br>";  
}  
?>
```

- **ksort()**: This function is used to sort an associative array in ascending order. It will observe only key part and make sorting, it will ignore value at the time of sorting. If your value is alphabets in array then it will sort array in ascending, all elements in a to z format. In ksort() function we pass array variable as an argument. If your array is mixing of all data type value then it will first sort string or character then float or double or integer value in ascending order.

```
<?php  
$emp=array("name"=>"Ram","designation"=>"Accountant","city"=>"Mumbai")  
;
```

```

ksort($emp);
print_r($emp);

$std=array("fullname"=>"Satisf","Age"=>22,"height"=>5.10,"Gender"=>'M');
ksort($std);

foreach($std as $x=>$x_value)
{
echo "Key=" . $x . ", Value=" . $x_value;
echo "<br>";
}
?>

```

- **krsort()**: This function is used to sort an associative array in descending order. It will observe only key part and make sorting; it will ignore value at the time of sorting. If your key is alphabets in array then it will sort array in descending, all elements in z to a format. In krsort() function we pass array variable as an argument. If your array is mixing of all data type value then it will first sort string or character then float or double or integer value in ascending order.

```

<?php
$emp=array("name"=>"Ram","designation"=>"Accountant","city"=>"Mumbai");
;
krsort($emp);
print_r($emp);

$std=array("fullname"=>"Satisf","Age"=>22,"height"=>5.10,"Gender"=>'M');
krsort($std);

foreach($std as $x=>$x_value)
{
echo "Key=" . $x . ", Value=" . $x_value;
echo "<br>";
}
?>

```

- **array_unique()**: This function will removes duplicate value from array and show only unique or single value. It will return whole array so we can use this function with other

variable which will accept the return array. we will pass array variable as an argument inside the function.

```
<?php  
$emp=array("name"=>"Ram","designation"=>"Accountant","name"=>"Ram");  
$k=array_unique($emp);  
print_r($k);  
?>
```

- **array_reverse()**: This function will convert an array in to reverse order. It will return whole array so we can use this function with other variable which will accept the return array. We will pass array variable as an argument inside the function.

```
<?php  
$emp=array("name"=>"Ram","designation"=>"Accountant","city"=>"Baroda");  
$k=array_reverse($emp);  
print_r($k);  
?>
```

- **Array_search()**: This function will search the value inside the array, if it found on any index or any key then it will print that index and key. We will pass two arguments in function first one is a search value and second is an array variable in which we want to search. If you have enter same value on different indices or keys then it will print only first index or key in which it match the word.

```
<?php  
$arr=array("a"=>"ram","b"=>"shiva","c"=>"ram");  
echo array_search("red",$arr);  
?>
```

- **Array_replace()**: This function will replace the array value from one array to another array. We have to pass two arguments in this function first one is an array variable which will replace its value and second is also an array variable whose value will replace in first one.

```
<?php  
$a=array("mobile","Television");  
$b=array("Computer","Leptop");  
Print_r(array_replace($a,$b));  
?>
```

- **Current()**: This function will print current element value of array. It will set the pointer on index inside the array and identify the value of current pointer.

- **next()**: This function moves the pointer to the next element of array. It will move pointer based on current pointer place.
- **previous()**: This function moves the pointer to the previous element of array. It will move pointer based on current pointer place.
- **end()**: This function moves the pointer to the last element of array.

```
<?php
$a=array(1,2,3,4,5);
echo "<br>";print_r($a);

echo "<br>".current($a)."current<br>";
echo next($a)."next<br>";
echo current($a)."current<br>";
echo prev($a)."prev<br>";
echo current($a)."<br>";
echo end($a);
?>
```

1.11 PHP Function:

In PHP we can also use two different types of functions one is User define function and other is inbuilt function. Function is a logical block of code. Where we add some logical syntax and we use that code repeatedly in our program by calling a function. Functions allow us to group blocks of related code that perform a specific task together. Functions can either return values when called or can simply perform an operation without returning any value. This saves us time of redeveloping a code.

1.11.1 User Define Function:

User define functions are design for some routing task. Like adding a data in database, adding some specific types of search, authenticating user data etc. In user define function we have to follow some syntax to declare and call a function. In such function it is declare or design with key word function, after function key word user can give it's own name of function, after that it will make body of function. In function declaration no need to define any data type it is same like variable, function will interpret return type at run time.

Syntax of function

Function functionname()

{

//body of function.

}

There are four different types of functions:

- Without argument without return type.
- With argument without return type.
- Without argument with return type
- With argument with return type.

Before we create our first user defined function, let's look at the rules that we must follow when creating our own functions.

- Function names must start with a letter or an underscore but not a number.
- The function name must be unique.
- The function name must not contain spaces.
- It is considered a good practice to use descriptive function names.
- Functions can optionally accept parameters and return values too.

- **Without argument without return type function:**

In such function we declare a function without any variable as an argument inside the round braces. Also will not enter any value at the time of calling a function.it will not return any answer or value.it will use that value inside the function only.

```
<?php
//define a function that displays hello function
```

```
function hello_msg()
{
    echo "Hello student this is first function.";
}
hello_msg();
```

- **With argument without return type function:**

In this function while we declare a function we will pass some variable inside the round braces as an argument. While function is call we enter some value that value will pass to the function as an argument. It will not return any answer or value. it will use that value inside the function only.

```
<?php
//define a function that calculate sum
```

```

function sum($no1,$no2)
{
    $ans=$no1+$no2;
    Echo $ans;
}
sum (10,20);
?>

```

- Without argument with return type:

In such function we declare a function without any variable as an argument inside the round braces. Also will not enter any value at the time of calling a function. It will return any answer or value. The value which is returned by function will receive or catch at time of calling a function.

```

<?php
// function along with three parameters
function make_product()
{
    $product = 5*5*5;
    return $product; //returning the product
}

```

```

// storing the returned value
$ans = make_product();
echo "The product is $ans ";
?>

```

- With argument with return type:

In this function while we declare a function we will pass some variable inside the round braces as an argument. While function is called we enter some value that value will pass to the function as an argument. It will return any answer or value. The value which is returned by function will receive or catch at time of calling a function.

```

Program_1.php
<!doctype html>
<html>
<head>

```

CC-310 Web Applications Development – II Practical

```
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<form action="product_calc.php" method="post">
    <label for="no1">No1</label><input type="number" name="no1" min=1 required><br>
    <label for="no2">No2</label><input type="number" name="no2" min=1 required><br>
    <label for="no3">No3</label><input type="number" name="no3" min=1 required><br>
    <input type="submit" name="submit" value="Submit">
    <input type="reset" name="reset" value="reset">
</form>
</body>
</html>
product_calc.php
<?php
    $n1=$_POST["no1"];
    $n2=$_POST["no2"];
    $n3=$_POST["no3"];
    // function along with three parameters
    function make_product($no1,$no2,$no3)
    {
        $product = $no1 * $no2 * $no3;
        return $product; //returning the product
    }
    // storing the returned value
    $ans = make_product($n1,$n2,$n3);
    echo "The product is $ans";
?>
```

in this file.”, means require function will stop execution of rest of PHP program, it will not continue if it is not finding file.

1.14 PHP Date and Time function:

In PHP Date and Time function is used to get current system date and time, it will get system date from server where this script will work. The *format* parameter in the *date()* function is required which specifies the format of returned date and time. This functions work with different arguments based on the argument we get different formats of date and time.

- **Date()**: this function will return current server date. There are two arguments are passed one is format parameters and second is timestamp, timestamp is optional.

Following are the format parameters used in date function.

d- Represent day of the month in double digit number (01 to 31).

D - Represent day of the week in text as an abbreviation (Mon to Sun).

m- Represent month in double digit number (01 to 12).

M- Represent month in text format (Jan to Dec).

y- Represent year in double digit number (01 or 21).

Y- Represent Year in four digit format (2001 or 2021).

We can arrange format parameters in single or double quotation marks. we can also use this parameter for getting only day, month or year,as as single argument of function.

Syntax *date("format parameters with special char - / : or space")*;

<?php

```
echo "Today Date: ";
echo date("d-m-y");
echo "<br>Today Date: ";
echo date('D-M-Y');
```

?>

Similar you can use current time format parameters in *date()* function.

h- Represent hour in 12-hours format(01 to 12).

H- Represent hour in 24-hours format(00 to 23).

i - Represent minutes with leading zeros (00 to 59).

s - Represent seconds with leading zeros (00 to 59).

a - Represent lowercase ante meridiem and post meridiem (am or pm).

A - Represent uppercase Ante meridiem and Post meridiem (AM or PM).

<?php

```

echo "Today Date: ";
echo date('h-i-s a');
echo "<br>Today Date: ";
echo date('H-i-s A');
?>

```

- **time()**: This function is used to get current timestamp. Using this timestamp we can get current date and time in date function.

```

<?php
    $time=time();
    echo $time."<br>";
    echo date('D-M-Y h:i:s',$time);
?>

```

- **mktim()**: This function is used to make timestamp based on date and time parameters. We will pass hour, minute, second, day, month and year by comma separated in this function as an argument. If you are not passing any parameter inside this then it will make timestamp based on current system date and time.

Syntax: **mktim(hour, minute, second, month, day, year);**

```

<?php
    $time=mktim(20,51,40,28,03,2021);
    echo $time;
    echo date('D-M-Y h:i:s',$time);
    $time2=mktim()
    echo $time2;
    echo date('D-M-Y h:i:s',$time2);
?>

```

- **date_create()**: This function will return a new Date time object. It will use to convert date format.
- **date_format()**: This function returns a date formatted according to the specified format. In date format we require two arguments one argument is from **date_create()** function and second is format parameters in which format you want to convert date. date format parameters are listed as above in **date()** function.

```

<?php
    $date_con=date_create("2021-01-02");
    echo date_format($date_con,"d/m/Y");
?>

```

?>

Above program will print date in format of dd-mm-yyyy it is converted from yyyy-mm-dd.

- **date_Diff()**: This is an inbuilt function in PHP which is used to calculate the difference between two dates. This function returns a Date Interval object on the success and returns FALSE on failure. There are two different dates are pass as an argument of date_diff(). There should be compulsory both date format are same either dd-mm-yyyy or yyyy-mm-dd.

<?php

```
$dc1=date_create("2021-01-02");
$dc2=date_create("2024-01-02");
$interval=date_diff($dc1,$dc2);
echo $interval->format('%R%a days');
```

?>

1.15 PHP Maths functions.

- **round()**: This function rounds a floating-point number. In round function if the value is 5 or above the 5 then will round the value next highest value. If you enter value round(0.78) then it will round this value to 1 if you pass round(0.7823,2) then it will round the value with 0.78 two digit round.
- **ceil()**: This function will return the round a fraction value near to the next highest value. If you have value 3.4 then it will round value to 4 because next highest value is 4.
- **floor()**: This function will return the round a fraction value near to the next lowest value. If you have value 3.4 then it will round value to 3 because next lowest value is 3.

<?php

```
echo round(0.98);//will prnt 1
echo round(4.96754,2);//will print4.97
```

echo ceil(3.4); // will print 4

echo ceil(9.99); //will print 10

echo ceil(-4.2); // will print -3

echo floor(4.3); // will print 4

echo floor(5.89); //will print 5

echo floor(-5.60); //will print -6

?>

- **sqrt()** : You can use the sqrt() function to find the square root of a positive number. This function returns a special value NAN for negative numbers. Here's an example:

```
<?php  
echo sqrt(9); // Outputs: 3  
echo sqrt(25); // Outputs: 5  
echo round(sqrt(10),2); // Outputs: 3.16  
echo sqrt(-16); // Outputs: NAN  
?>
```

- **rand()** : This function will generate random value between the range of min and max. We pass two arguments inside the rand() function. Both are optional, user can pass min value or max value as an argument if user can not pass any value then default min value is 0 and max value is getrandmax() function value.

```
<?php  
echo rand();  
echo rand(10,1000);  
//it will generate random value between 10 to 1000 numbers.  
?>
```

- **decbin()** and **bindec()**: The decbin() function is used to convert a decimal number into binary number, also its counterpart the bindec() function converts a number from binary to decimal. we have to pass decimal number or binary number as an argument.

```
<?php  
echo decbin(123)."  
"; // Outputs: 1111011  
echo decbin(12)."  
"; // Outputs: 1100  
echo decbin(100)."  
"; // Outputs: 1100100  
echo bindec(100100)."  
"; // Outputs: 36  
echo bindec(101010)."  
"; // Outputs: 42  
echo bindec(1000111)."  
"; // Outputs: 71  
?>
```

- **dechex()** and **hexdec()**: The dechex() function is used to convert a decimal number into Hexadecimal number, also its counterpart the hexdec() function converts a number from Hexadecimal to decimal. we have to pass decimal number or binary number as an argument. Same for decimal to octal or octal to decimal conversion we use these functions decoct() and octdec().

```
<?php  
echo dechex(35)."  
"; //output 23
```

```
echo dechex(289)."  
";//output 121  
echo hexdec(186)."  
";//output 390  
echo hexdec(654)."  
";//output 1620
```

```
?>
```

- **max()**: This function will get the maximum number from array as well as list of numbers.
- **min()**: This function will get minimum number from array as well as list of numbers.

```
<?php
```

```
echo max(12,22,4,1,57,21);  
echo max(array(3,7,1,2,9,4,6));  
echo min(12,22,4,1,57,21);  
echo min(array(3,7,1,2,9,4,6));
```

```
?>
```

1.16 PHP String Functions.

- **join()**: This function is used to join array elements in single string. We pass two arguments, first one is a separator which is used to join string and second one is an array variable. It is an alias of implode() function.

Syntax: **join(separator, array variable);**

```
<?php
```

```
$arr=array("hello","have","a","nice","day");  
echo join(" ",$arr);
```

```
?>
```

- **implode()**: This function is used to join array elements in single string. We pass two arguments, first one is a separator which is used to join string and second one is an array variable.

Syntax: **implode(separator, array variable);**

```
<?php
```

```
$arr=array("one","two","three","four");  
echo implode(":",$arr);
```

```
?>
```

- **explode()**: This function will create an array of string. It will separate the string and create array of string. It will use two arguments one is separator and second is string or variable. It will separate the string with special character which is mention in variable or string.

string. For example in your string there is a space between all the words then you can use space as a separator in argument.

```
<?php
    $str="Hello student good morning and have a nice day.";
    $arr=explode(" ",$str);
    print_r($arr); // this will print array of string with index value.
    Echo $arr[0];
?>
```

- **ltrim()**: This function will strips whitespace from the beginning or left side of string.
- **rtrim()**: This function will strips whitespace from the end or right side of string.

```
<?php
    $str=" this will trim from beginning ";
    $ltrim=ltrim($str);
    $rtrim=rtrim($str);
    echo "<pre><mark>$ltrim</mark></pre>";
    echo "<pre><mark>$rtrim</mark></pre>";
?>
```

- **md5()**: This function will convert string in to Message Digest Algorithm 5 of string using the RSA Data Security. There are 2 different arguments are pass first one is string and second one is raw format, in raw format default value is false means it will convert in to 32 character hexadecimal number and if it is true then it will convert in to 16 character. Raw format is optional.

```
<?php
    $str="admin@123";
    echo md5($str); // e6e061838856bf47e1de730719fb2609
?>
```

- **parse_str()**: This function will parse a string in to variable. This function is useful when wan to store string value in to different variables. We need to pass two arguments first one is string and second is variable name in which it is convert. It will convert it into associative array variable.

```
<?php
    $str="name=Hima&age=3";
    Parse_str($str,$result);
```

Echo \$result["name"]; // it will print name Hima

Echo \$result["age"]; // it will print age 3

?>

- **str_replace()**: This function will replace all the occurrence of word in string which matching a replace word. There are three arguments are pass first one is find word second one is replace word and third is string in which we match a word. str_ireplace() function also perform same task but it is used with case-insensitive version of str_replace().

<?php

```
$str="Hello Student,Hello Parents and Hello Faculty member";
echo str_replace("Hello","Goo Morning",$str);
```

?>

- **str_split()**: This function will split sting in to array of chunks. There are two arguments are pass first one is string and second is element of array, default element is 1 it is optional.

<?php

```
$str="Have a nice Day! ";
$split=str_split($str);
print_r($split); // will print string in to individual array element.
echo $split[0]; // will print H
echo $split[1]; // will print a
```

?>

- **strcmp()**: This function will compare two string with case-sensitive. There are two different string are pass as an argument to compare sting. If both strings are same then it will return 0 otherwise it will return length of character which are not match. For case-insensitive searches, use thestrcasecmp() function.

<?php

```
$str1="Hypertext";
$str2="Hypertext";
echo strcmp($str1,$str2);
if(strcmp($str1,$str2)==0)
{
    echo "Both Strings are Equal";
}
?>
```

- **strpos()**: The strpos() function finds the position of the first occurrence of a string inside another string. This function is case-sensitive. For case-insensitive searches, use the stripos() function. There are two arguments are pass first one is actual string and second is a word or portion of string which we want to compare.

```
<?php
    $str1="Hypertext";
    $str2="Hypertext";
    echo strcmp($str1,$str2);
    if(strcmp($str1,$str2)==0)
    {
        echo "Both Strings are Equal";
    }
?>
```

- **strlen()**: This function will return the length of string. We have to pass string as an argument of this function.

```
<?php
    $str="Hypertext Pre-processor (PHP)";
    $len=strlen($str);
    Echo "Total Length of String=$len";
?>
```

- **substr()**: This function extracts a part of a string. There are three arguments are pass. First argument is string from which we create sub string. Second is start, specifies the position in the string from where the extraction begins. Third is length, specifies how many characters to extract.

```
<?php
    $substr1=substr($str,9); // it will start 9th letter to end
    echo nl2br("\n $substr1");
    $substr2=substr($str,5,8); // it will start 5th letter to size of 8
    echo nl2br("\n $substr2");
    $substr3=substr($str,-4); // it will start from back side of string.
    echo nl2br("\n $substr3");
    $substr4=substr($str,-5,5); // it will star from back side.
    echo nl2br("\n $substr4");
?>
```

UNIT- 1 Practical

1. Write a program in PHP to display "Learning PHP" in bold format.

```
<?php  
    echo "hello World";  
    echo " Learning PHP ";  
    echo 123;  
?>
```

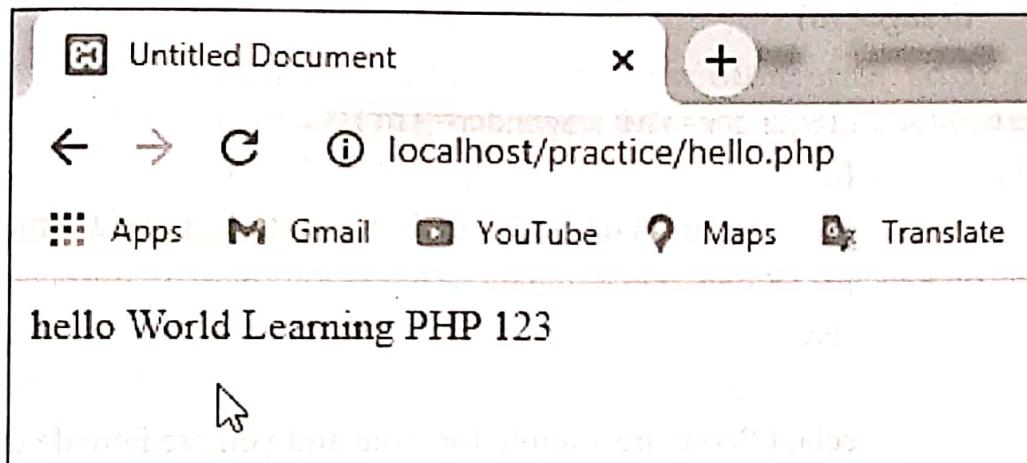


Fig 1.11 echo function to print message on browser window.

2. Write a program in PHP to demonstrate the use of comments, echo and print.

```
<?php  
    echo("below line is in comment");  
    // echo("this line is in single line comment");  
    print "hello Asia";  
    print 100;  
    /* $age=35;  
    print "Payal Patel age is ". $age ;  
    print("this is a multiline comment in program");*/  
?>
```

3. Create a program in PHP to demonstrate the use of If ... Else and switch statements.

```
<!doctype html>
```

```
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<?php
    $age="15";
    $gender="f";
    if($age>18)
    {
        if($gender=='M' || $gender=='m')
        {
            echo "You are eligible for Vote and you are male candidate";
        }
        else
        {
            echo ("You are eligible for Vote and you are Female candidate");
        }
    }
    else
    {
        print("You are not eligible for Vote....");
    }
?
</body>
</html>
```

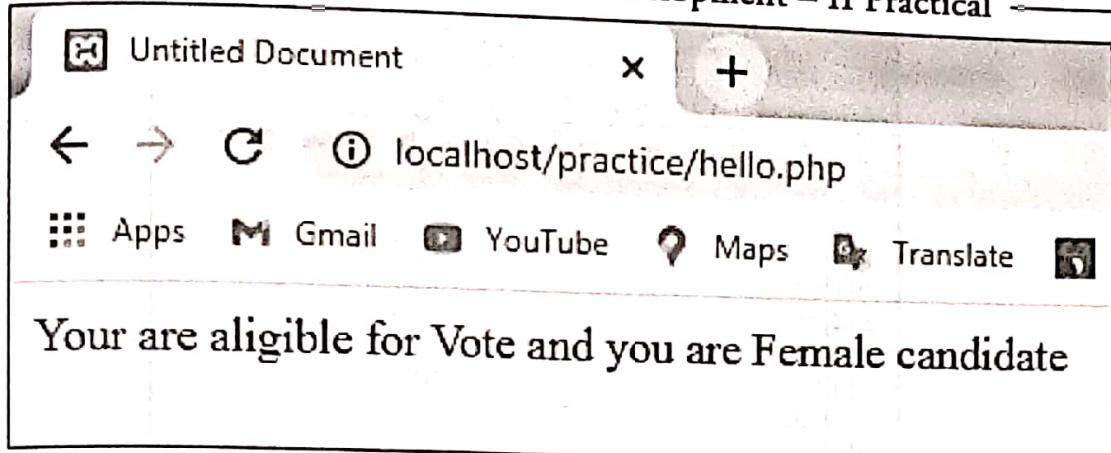


Fig 1.12 If else condition example.

4. Create an array named \$sub, assign five elements to it and display the elements assigned using for loop and for each statement.

```
<?php
    $sub=array(1,2,3,4,5,55,66,77,88);
    $cnt=count($sub); //count function will give the total number of elements in array
    echo $cnt;
    for ($i=0;$i<$cnt;$i++)
    {
        echo "<br>arr[\".$i.\"]=".$sub[$i];
    }
    foreach($sub as $val)
    {
        echo "<br>".$val;
    }
    $m=array("banana",11,333.333,'A');
    print_r($m); // this function will print an array with key and value also with index and value.
    for($j=0;$j<count($m);$j++)
    {
        echo "<br>arr[\".$j.\"]=".$m[$j];
    }
?>
```

```

9
arr[0]=1
arr[1]=2
arr[2]=3
arr[3]=4
arr[4]=5
arr[5]=55
arr[6]=66
arr[7]=77
arr[8]=88
1
2
3
4
5
55
66
77
88
Array ( [0] => banan [1] => 11 [2] => 333.333 [3] => A )
arr[0]=banan
arr[1]=11
arr[2]=333.333
arr[3]=A

```

Fig 1.13 Array function and different types of Array.

5. Create an array named \$student, that stores 5 element bounded to a different keys and access the same using the key element.

```

<?php
$student=array("Name"=>"Hima
Patel","Age"=>5,"Height"=>3.4,"City"=>"Ahmedabad");
//print_r($student);
echo $student["Name"]."<br>";
echo $student["Age"]."<br>";
echo $student["Height"]."<br>";
echo $student["City"]."<br>";
foreach($student as $key=>$val)
{
    echo "<br>[$key]=>$val;
}
?>

```

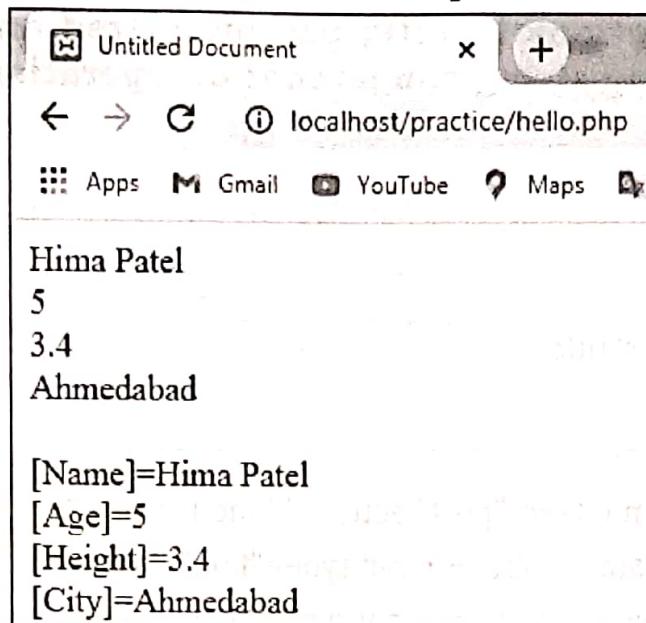


Fig 1.14 foreach Loop Structure.

6. Write a program in PHP to demonstrate the use of multidimensional arrays.

```
<?php
$shop=array("mobile"=>array("samsung"=>"m12","mi"=>"note9 pro","apple"=>"i12"),
            "leptop"=>array("dell"=>"insp12","hp"=>"hp2200","lenovo"=>"Z350")
);
echo $shop["mobile"]["samsung"]."<br>";
echo $shop["mobile"]["mi"]."<br>";
echo $shop["mobile"]["apple"]."<br>";
echo $shop["leptop"]["lenovo"]."<br>"
?>
```

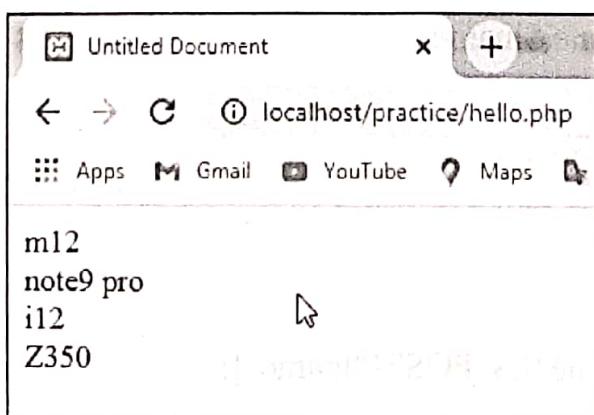


Fig 1.15 multi-dimensional array.

7. Create two functions in PHP, parameterized and non-parameterized for implementing string concatenation operation.

post_data.php

```
<!doctype HTML>
<html>
<head>
    <title>Post data </title>
</head>
<body>
<form name="add" method="post" action="function.php">
    <label for="n1">Fname</label><input type="text" name="fname" value="" /><br>
    <label for="n2">Lname</label><input type="text" name="lname" value="" /><br>
    <input type="submit" name="submit" value="submit" />
</form>
</body>
```

function.php

```
<?php
//function call without argument.
function string_cat()
{
    $fname=$_POST["fname"];
    $lname=$_POST["lname"];
    $fullname=$fname.$lname."<br>";
    echo $fullname;
}

//function call with argument.
function str_cat($fname,$lname)
{
    $flname=$fname.$lname;
    echo $flname;
}

string_cat();
str_cat($_POST["fname"],$_POST["lname"]);
?>
```

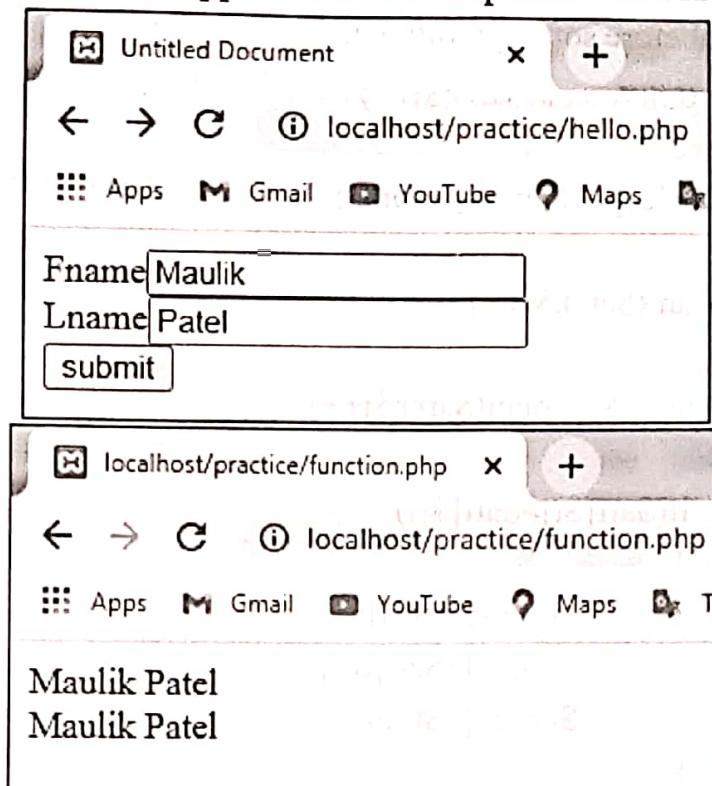


Fig 1.16 Function Example.

8. Write a PHP program to display information of PHP in the browser.

```
<?php  
echo "php version=".phpversion();  
Phinfo();  
phinfo(INFO_MODULES);  
?>
```

9. Write a program in PHP to sort the array of given 5 numbers in ascending and descending order.

```
<!doctype html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>Untitled Document</title>  
</head>  
  
<body>  
<?php  
$arr=array(11,4,23,3,12,59,2,34);
```

```

echo nl2br("\nbefore sorting array\n");
for($i=0;$i<count($arr);$i++)
{
    echo "arr[$i] = ". $arr[$i]. "<br>";
}
for($i=0;$i<count($arr);$i++)
{
    for($j=$i+1;$j<count($arr);$j++)
    {
        if($arr[$i]>$arr[$j])
        {
            $tmp=$arr[$i];
            $arr[$i]=$arr[$j];
            $arr[$j]=$tmp;
        }
    }
}
echo nl2br("\nafter sorting array in Assending\n");
for($i=0;$i<count($arr);$i++)
{
    echo $arr[$i]. "<br>";
}
for($i=0;$i<count($arr);$i++)
{
    for($j=$i+1;$j<count($arr);$j++)
    {
        if($arr[$i]<$arr[$j])
        {
            $tmp=$arr[$i];
            $arr[$i]=$arr[$j];
            $arr[$j]=$tmp;
        }
    }
}

```

```
} after sorting array in Ascending order  
echo "<br>after sorting array in Decending<br>";  
  
for($i=0;$i<count($arr);$i++)  
{  
    echo $arr[$i]."<br>";  
}  
?  
</body>  
</html>
```

```
before sorting array
arr[0]=11
arr[1]=4
arr[2]=23
arr[3]=3
arr[4]=12
arr[5]=59
arr[6]=2
arr[7]=34

after sorting array in Assending
2
3
4
11
12
23
34
59

after sorting array in Decending
59
34
23
12
11
4
3
2
```

Fig 1.17 Sorting of Array.

10. Write a program to count the total number of times a specific value appears in an array.

```
<?php
$cnt=0;
$match=1;
$arr=array(1,3,22,1,4,35,21,1,5,33);
for($i=0;$i<count($arr);$i++)
{
    if($arr[$i]==$match)
    {
        $cnt++;
    }
}
echo $match." is found <b>".$cnt."</b> time <br>";
$cnt=0;
$word="hello";
$string="hello student hello have a nice day";
$v=explode(" ",$string);
//explode function is used to split a string with matching delimiter with string variable.
//explode(delimiter, string variable);
for($i=0;$i<count($v);$i++)
{
    if($v[$i]==$word)
    {
        $cnt++;
    }
}
echo $word." is found <b>".$cnt."</b> time";
?>
```

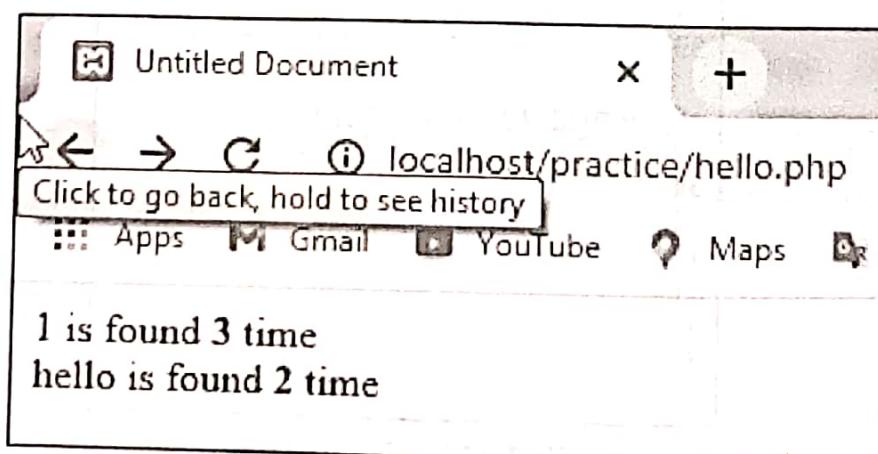


Fig 1.18 Total Number of character found.

UNIT-2

PHP and MYSQL Programming

1. Create a form containing two input fields (Name, Email ID) and a submit button. When the user clicks on submit button, the form data should be sent for processing to PHP file ,which should display the welcome message with the email_id on the PHP page. Form data should be sent by HTTP GET/POST method.
2. Write a PHP script for that creates a database named "DB-1" in MySQL.
3. Write a PHP script for creating a product table in the specified database with fields Pro_id, Pro_name, Pro_price, QOH. Also display an acknowledgement for the same.
4. Create a form containing four input fields(Pro_id, Pro_name, Pro_price, QOH) and Submit button. When the user clicks on the submit button an PHP script should be executed which inserts the record in the product table.
5. Create a form containing one input field(Pro_id) and a search button. When the user clicks on the Search button a PHP script should get executed and should display the details of the product for the Pro_id specified.
6. Create a form containing two input fields (Pro_id, QOH) and Update button. When the user clicks on the Update button the quantity of the Pro_id specified should get updated using a PHP script.
7. Create a form containing one input field(Pro_id) and a Delete button. When the user clicks on the Delete button a PHP script should get executed and should delete the record of the product for the Pro_id specified.

Unit -2 Practical

- 1. Create a form containing two input fields (Name, Email_ID) and a submit button. When the user clicks on submit button, the form data should be sent for processing to PHP file ,which should display the welcome message with the email_id on the PHP page. Form data should be sent by HTTP GET/POST method.**

Prg1.php

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>

<body>
<form action="prg1_submit.php" method="post">
<!--we can set method get also-->
<table>
<tr>
    <td><label for="name">Name</label></td>
    <td><input type="text" name="fullname" ></td>
</tr>
<tr>
    <td><label for="email">Email</label></td>
    <td><input type="email" name="email" ></td>
</tr>
<tr>
    <td colspan="2"><input type="submit" name="Submit" value="Submit" >
    <input type="reset" name="reset" value="reset"></td>
</tr>
</table>
</form>
```

```
</body>
</html>
```

Fig. 2.21 Submit form data on another page using post method.

prg1_submit.php

```
<?php
```

```
$fullname=$_POST["fullname"];
$email=$_POST["email"];
//if form method is get then we use$_GET["fullname"]; to receive data from form.
echo "Form Submit following Data<br>";
echo "Fullname=". $fullname . "<br>";
echo "Email=$email<br>";
?>
```

Fig. 2.22 Data posted from form and received by action page.

2. Write a PHP script for that creates a database named "DB-1" in MySQL.

Note: for working with database file ,first we have to establish a connection between PHP and mysql. For that we use connection function.i have here use one external php file connection.php which I will include in my program.

connection.php

```
<?php
    error_reporting("E_ALL");// this will not show any warning msg.
    $host="localhost";
    $user="root";
    $pass="";
    $con=mysqli_connect($host,$user,$pass);
    if($con)
    {
        echo "Connection Established";
    }
    else
    {
        echo "Connection not Established";
    }
?>
```

Prg2.php

```
<?php
    include ("connection.php");
    $db_create="create database db1";
    $qry=mysqli_query($con,$db_create);
    if($qry)
    {
        echo "Database Created Successfully";
    }
    else
    {
        echo "Database not Created.";
    }
?>
```

?>

Database	Collation	Action
astrology	latin1_swedish_ci	Check privileges
augmetic	latin1_swedish_ci	Check privileges
buy_bag	latin1_swedish_ci	Check privileges
com_shop	latin1_swedish_ci	Check privileges
db1	latin1_swedish_ci	Check privileges

Fig. 2.23 Create Database using create database query.

3. Write a PHP script for creating a product table in the specified database with fields Pro_id, Pro_name, Pro_price, QOH. Also display an acknowledgement for the same.

Note: for creating a table within database, first we have to establish a connection between PHP and mysql database. For that we use connection function.i have here use one external php file connection.php which I will include in my program.

connection.php

```
<?php
error_reporting("E_ALL");// this will not show any warning msg.
$host="localhost";
$user="root";
$pass="";
$dbname="db1";
$con=mysqli_connect($host,$user,$pass,$dbname);
if($con)
{
    echo "Connection Established<br>";
}
```

```

else
{
    echo "Connection not Established<br>";
}
?>

```

Prg2.php

```

<?php
include ("uconnection.php");
$db_create="CREATE TABLE product
(Pro_Id INT(3) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
 Pro_name VARCHAR(100) NOT NULL,
 Pro_price FLOAT(7,2) NOT NULL,
 QOH INT(5)NOT NULL)";
$qry=mysqli_query($con,$db_create);
if($qry)
{
    echo "PRODUCT TABLE CREATED SUCCESSFULLY";
}
else
{
    echo "PRODUCT TABLE NOT CREATED.";
}
?>

```

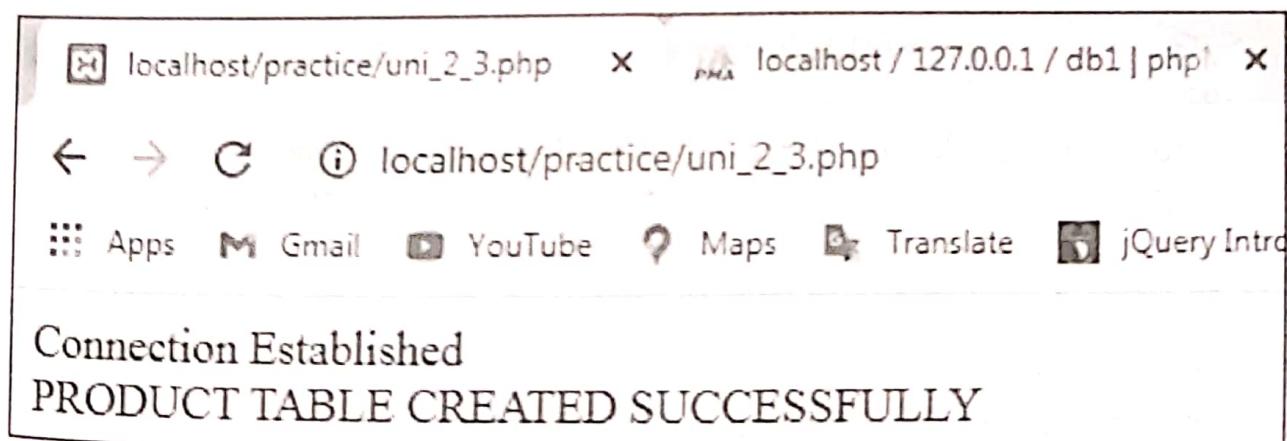


Fig. 2.24 Create Table using create table query.

4. Create a form containing four input fields(Pro_id, Pro_name, Pro_price, QOH) and Submit button. When the user clicks on the submit button an PHP script should be executed which inserts the record in the product table.

Prg3.php

```
<?php
if(isset($_GET["msg"]))
{
    echo $_GET["msg"];
}
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<form action="product_query.php" method="post">
    <table>
        <caption><b>Add Product</b></caption>
        <tr>
            <td><label for="name">Productname</label></td>
            <td><input type="text" name="Pro_name" ></td>
        </tr>
        <tr>
            <td><label for="email">Price</label></td>
            <td><input type="number" name="Pro_price" step="0.01"></td>
        </tr>
        <tr>
            <td><label for="email">Quality</label></td>
            <td><input type="number" name="QOH" ></td>
        </tr>
        <tr>
```

```

<td colspan="2"><input type="submit" name="Submit" value="Submit" >
<input type="reset" name="reset" value="reset"></td>
</tr>
</table>
</form>
</body>
</html>

```

Add Product

Productname	USB Head Phone
Price	1260.50
Quality	70

Submit **reset**

Fig. 2.25 Insert Product in product table using add product form .

➤ product_query.php

```

<?php
    include("uni_2_connection.php");
    $productname=$_POST["Pro_name"];
    $price=$_POST["Pro_price"];
    $qoh=$_POST["QOH"];
    if(!empty($productname) && !empty($price) && !empty($qoh))
    {
        $ins_qry="insert into product (Pro_name,Pro_price,QOH) values
        ('$productname','$price','$qoh')";
        $res=mysqli_query($con,$ins_qry);
        if($res)
        {
            header("location:product_record.php");
        }
    }
    else
    {

```

```
header("location:uni_2_4.php?msg=Please Try Again");
```

```
}
```

```
?>
```

➤ Product_record.php

```
<?php
include("uni_2_connection.php");
$selqry="select * from product";
$res=mysqli_query($con,$selqry);
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<h3><a href="uni_2_4.php">+Add New Record</a></h3>
<table border="1">
<tr bgcolor="#6699FF">
<th>Product Id</th>
<th>Product Name</th>
<th>Product Price</th>
<th>Product Quantity</th>
<th>Edit</th>
</tr>
<?php
while($row=mysqli_fetch_array($res))
{
?>
<tr bgcolor="#FFCC33">
<td><?php echo $row["Pro_Id"];?></td>
<td><?php echo $row["Pro_name"];?></td>
<td><?php echo $row["Pro_price"];?></td>
<td><?php echo $row["QOH"];?></td>
```

```

<td><?php echo "<a href=Prg6.php?pid=$row[Pro_Id] style=text-decoration:none>Edit</a>"?></td>
</tr>
<?php
}
?>
</table>
</body>
</html>

```

Product Id	Product Name	Product Price	Product Quantity
1	Key Board	325.85	100
2	Mouse	245.75	200
3	Pen drive SGR	420.75	150
4	USB Head Phone	1260.50	70

Fig. 2.26 Product Added in Product table and fetch record from product table .

5. Create a form containing one input field(Pro_id) and a search button. When the user clicks on the Search button a PHP script should get executed and should display the details of the product for the Pro_id specified.

Prg5.php

```

<?php
include("uni_2_connection.php");
if(!empty($_POST["search_product"]))
{
    $search=$_POST["search_product"];
    $selqry="select * from product where Pro_Id=$search";
    $res=mysqli_query($con,$selqry);
}
else

```

```
{  
$sel_qry="select * from product";  
$res=mysqli_query($con,$sel_qry);  
}  
?>  
<!doctype html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>Untitled Document</title>  
</head>  
  
<body>  
<div style="float:left; width:150px;"><h3><a href="uni_2_4.php">+Add New  
Record</a></h3></div>  
<div style="float:right; width:150px; margin-right:100px;">  
    <form action=<?php $_SERVER["PHP_SELF"];?> method="post">  
        <table>  
            <tr>  
                <td>  
                    <select name="search_product">  
                        <option value="">Please Select</option>  
                        <?php  
                            $sel_qry_search="select * from product";  
                            $res_search=mysqli_query($con,$sel_qry_search);  
                            while($row_search=mysqli_fetch_array($res_search))  
                            {  
                            ?>  
                                <option value=<?php echo $row_search["Pro_Id"];?>>  
                                    <?php echo $row_search["Pro_name"];?>  
                                </option>  
                                <?php  
                            }  
                            ?>
```

```

    </select>
    </td>
    <td>
        <input type="submit" name="submit" value="Search"></td>
    </tr>
</table>
</form>
</div>
<div style="margin-top:50px; margin-left:200px; float:left;">
<table border="1">
<caption><b>Product Detail</b></caption>
    <tr bgcolor="#6699FF">
        <th>Product Id</th>
        <th>Product Name</th>
        <th>Product Price</th>
        <th>Product Quantity</th>
    </tr>
    <?php
        while($row=mysqli_fetch_array($res))
        {
    ?>
    <tr bgcolor="#FFCC33">
        <td><?php echo $row["Pro_Id"];?></td>
        <td><?php echo $row["Pro_name"];?></td>
        <td><?php echo $row["Pro_price"];?></td>
        <td><?php echo $row["QOH"];?></td>
    </tr>
    <?php
        }
    ?>
</table>
</div>
</body>
</html>

```

Product Id	Product Name	Product Price	Product Quantity
1	Key Board	325.85	100
2	Mouse	245.75	200
3	Pen drive 8GB	420.75	150
4	USB Head Phone	1260.50	70

Fig. 2.27 Search Record from product table giving product id.

- 6. Create a form containing two input fields (Pro_id, QOH) and Update button. When the user clicks on the Update button the quantity of the Pro_id specified should get updated using a PHP script.**

➤ **Prg6.php**

```
<?php
include("uni_2_connection.php");
$pro_id=$_GET["pid"];//this id is come from product_record.php page.
```

```
$sel_product="select * from product where Pro_Id=$pro_id";
```

```
$res_procut=mysqli_query($con,$sel_product);
```

```
$row_product=mysqli_fetch_array($res_procut);
```

```
$pid=$row_product["Pro_Id"];
```

```
$qty=$row_product["QOH"];
```

```
if(isset($_GET["msg"]))
```

```
{
```

```
echo $_GET["msg"];
```

```
}
```

```
?>
```

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```

<meta charset="utf-8">
<title>Untitled Document</title>
</head>

<body>
<form action="product_update.php" method="post">
    <table>
        <caption>Product Update</caption>
        <tr>
            <td><label for="product">Product Name</label></td>
            <td>
                <select name="Pro_Id">
                    <option value="">Please Select</option>
                    <?php
                        $sel_product="select * from product order by Pro_name ASC";
                        $qry=mysqli_query($con,$sel_product);
                        while($row_search=mysqli_fetch_array($qry))
                        {
                            <option value=<?php echo $row_search["Pro_Id"];?><?php if($row_search["Pro_Id"]==$pid){echo "selected"; }else{ echo "";}?>>
                                <?php echo $row_search["Pro_name"];?>
                            </option>
                            <?php
                        }
                    <?>
                </select>
            </td>
        </tr>
        <tr>
            <td><label for="quantity">Quantity</label></td>
            <td><input type="number" name="QOH" value=<?php echo $qty;?>></td>
        </tr>
    </table>
</form>

```

```

<tr>
    <td colspan="2"><input type="submit" name="Submit" value="Submit">
    <input type="reset" name="reset" value="reset"></td>
</tr>
</table>
</form>
</body>
</html>

```

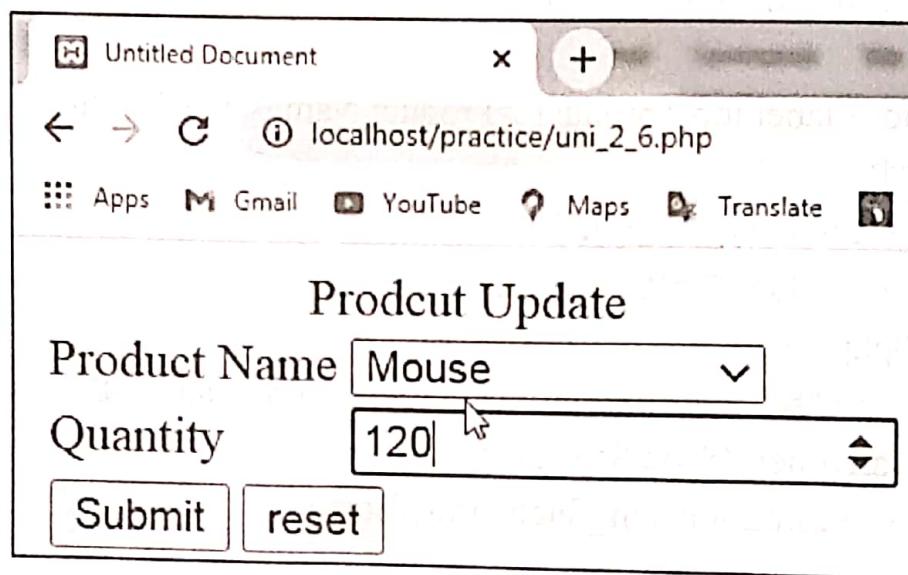


Fig. 2.28 Edit Product using product update form.

➤ Product_update.php

```

<?php
    include("uni_2_connection.php");
    echo $Pro_Id=$_POST["Pro_Id"];
    echo $qoh=$_POST["QOH"];

    if(!empty($qoh))
    {
        echo $update_qry="update product set QOH=$qoh where Pro_Id=$Pro_Id";
        $update_res=mysqli_query($con,$update_qry);
        if($update_res)
        {
            header("location:product_record.php");
        }
    }
}

```

```

else
{
    header("location:prg6.php?msg=Please Try Again");
}
?>

```

Product Id	Product Name	Product Price	Product Quantity	Edit
1	Key Board	325.85	100	Edit
2	Mouse	245.75	120	Edit
3	Pen drive 8GB	120.75	150	Edit
4	USB Head Phone	1260.50	500	Edit

Fig. 2.29 Fetch Edited Record from database

7. Create a form containing one input field(Pro_id) and a Delete button. When the user clicks on the Delete button a PHP script should get executed and should delete the record of the product for the Pro_id specified.

Product_record.php

```

<?php
include("uni_2_connection.php");
if(!empty($_POST["search_product"]))
{
    $search=$_POST["search_product"];
    $selqry="select * from product where Pro_Id=$search";
    $res=mysqli_query($con,$selqry);
}
else
{
    $selqry="select * from product";
    $res=mysqli_query($con,$selqry);
}
?>
<!doctype html>
<html>
<head>

```

```

<meta charset="utf-8">
<title>Untitled Document</title>
</head>

<body>
<div style="float:left; width:150px;"><h3><a href="uni_2_4.php">+Add New Record</a></h3></div>
<div style="float:right; width:150px; margin-right:100px;">
<form action="php \$_SERVER['PHP_SELF'];???" method="post">
<table>
<tr>
<td>
<select name="search_product">
<option value="">Please Select</option>
<?php
    $sel_qry_search="select * from product";
    $res_search=mysqli_query($con,$sel_qry_search);
    while($row_search=mysqli_fetch_array($res_search))
    {
        <?php echo $row_search["Pro_Id"];?>">
        <?php echo $row_search["Pro_name"];?>
        </option>
        <?php
    }
    ?>
</select>
</td>
<td>
<input type="submit" name="submit" value="Search"></td>
</tr>

```

```
</table>
</form>
</div>
<div style="margin-top:50px; margin-left:200px; float:left;">

<table border="1">
<caption><b>Product Detail</b></caption>
<tr bgcolor="#6699FF">
<th>Product Id</th>
<th>Product Name</th>
<th>Product Price</th>
<th>Product Quantity</th>
<th>Edit</th>
<th>Delete</th>
</tr>
<?php
    while($row=mysqli_fetch_array($res))
    {
        ?>
<tr bgcolor="#FFCC33">
<td><?php echo $row["Pro_Id"];?></td>
<td><?php echo $row["Pro_name"];?></td>
<td><?php echo $row["Pro_price"];?></td>
<td><?php echo $row["QOH"];?></td>
<td><?php echo "<a href=uni_2_6.php?pid=$row[Pro_Id] style=text-decoration:none>Edit</a>"?></td>
<td><?php echo "<a href=uni_2_7.php?pid=$row[Pro_Id] style=text-decoration:none>Delete</a>"?></td>
</tr>
<?php
}
?>
</table>
</div>
```

```
</body>
</html>
```

Product Detail					
Product Id	Product Name	Product Price	Product Quantity	Edit	Delete
1	Key Board	325.85	100	Edit	Delete
2	Mouse	245.75	120	Edit	Delete
3	Pen drive 8GB	420.75	150	Edit	Delete

Fig. 2.30 Delete Product

➤ Prg7.php

```
<?php
    include("uni_2_connection.php");
    echo $Pro_Id=$_GET["pid"];// this id is come from product_record.php page.

    if(!empty($Pro_Id))
    {
        echo $delete_qry="delete from product where Pro_Id=$Pro_Id";
        $delete_res=mysqli_query($con,$delete_qry);
        if($delete_res)
        {
            header("location:product_record.php");
        }
    }
    else
    {
        header("location:uni_2_7.php?msg=Please Try Again");
    }
?>
```

Product Detail					
Product Id	Product Name	Product Price	Product Quantity	Edit	Delete
1	Key Board	325.85	100	Edit	Delete
2	Mouse	245.75	120	Edit	Delete

Fig. 2.31 Deleted Record from database.

UNIT-3

AJAX and Validation

1. Create a form containing one input field (Name). When the user enters his/her name and as any key is released , the form should display a welcome message for the user.Implement using AJAX.
2. Repeat the above question to demonstrate the use of keydown and keypress events.
3. Write a program for converting a string into uppercase using AJAX.
4. Create a form containing a combobox with some product names as items. Whenever a user selects a particular product from the combox, it shuold be sent to the server asynchronously (i.e. without pressing submit button). Implement using AJAX.
5. Write a program to demostrate the example of sending items selected from radio and checkbox to server asynchronously.
6. Write a program to validate a blank field and also validate the length of the data entered(i.e. minimum lenght of 5).
7. Write a programto validate and Email ID using regular expression and by using DOM.
8. Write a program that checks a particular stuId already exists in the student(stuId,stu_name,mob,country) table or not. If stuId exists then display a message.
"User Already Exit. Try another stuId". If it does not exits then add the data in the student table.Implement using AJAX.

Unit -3 AJAX and Validation

3.1 What is AJAX?

AJAX stands for Asynchronous JavaScript and XML. It is a set of web development techniques which allow web applications to work asynchronously – processing any requests to the server in the background from client machine. Ajax is a client-side script that communicates to and from a server/database without the need for a postback or a complete page refresh. This technology is reduce the interaction between server and client machine. It is the method of exchanging data with server and updating parts of webpage without reloading an entire webpage.

JavaScript is only the client side scripting language and it is working on client web browser. Because of JavaScript support It is working Asynchronously.

XML is the acronym for Extensible Mark-up Language. It is used to encode messages in both human and machine readable formats. It's like HTML but allows you to create your custom tags.

The Document Object Model (**DOM**) for dynamic display data and its interaction.

The XMLHttpRequest object for the asynchronous communication.

➤ Why AJAX?

- IT is used to develop rich interactive web application.
- IT will validate form control without submitting a form and also provide quick request response from server side.
- Data can be retrieved from the server and only a certain part of a page updated without loading the whole page.

➤ How AJAX work?

- HTML and CSS for marking up and styling information us.
- The DOM accessed with JavaScript to dynamically display and interact with the information presented.
- A method for exchanging data asynchronously between browser and server, thereby avoiding page reloads. The XMLHttpRequest (XHR) object is usually used, but sometimes an IFrame object or a dynamically added tag is used instead.
- A format for the data sent to the browser. Common formats include XML, pre-formatted HTML, plain text, and JavaScript Object Notation (JSON). This data could be created dynamically by some form of server-side scripting.

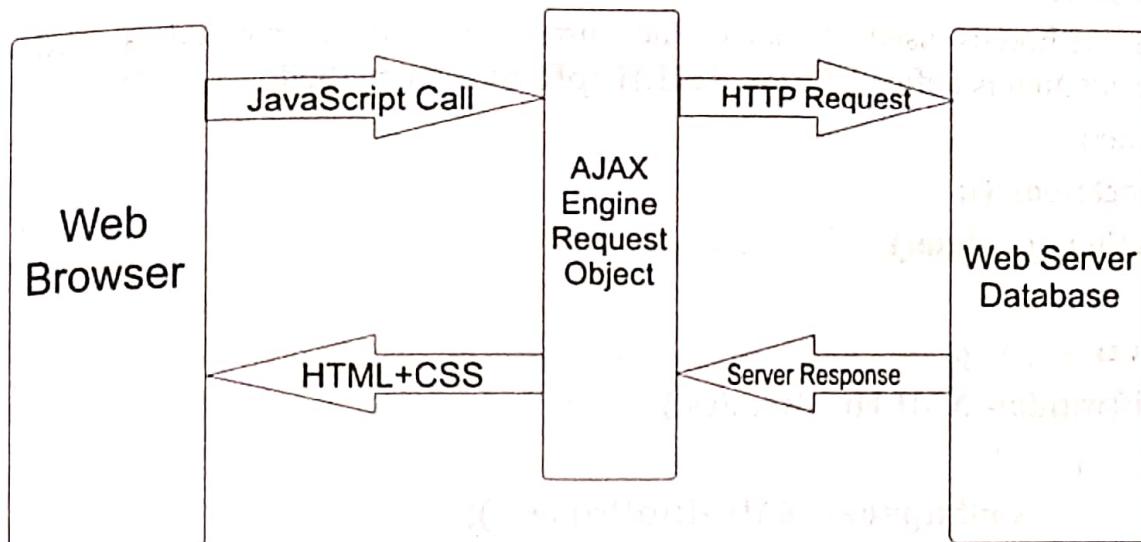


Fig. 3.1 How AJAX work?

3.2 AJAX Request Object:

In Web application AJAX Engine send request using AJAX Request Object, for that it is use XMLHttpRequest method to create XML HTTP request object. IT is used to exchange information or data with a webserver behind the scenes. It will update web page without reloading. All modern browsers (Chrome, Firefox, IE7+, Edge, Safari, Opera) have a built-in XMLHttpRequest object. It is inbuilt JavaScript library method.

Syntax for creating an XMLHttpRequest object:

Syntax:

Variable=new XMLHttpRequest();

var xmlhttp = new XMLHttpRequest();

function ajaxdata()

{

var xmlhttp;

if(window.XMLHttpRequest)

{

xmlhttp=new XMLHttpRequest();

}

}

3.3 XMLHttpRequest Methods:

Following methods are used with XMLHttpRequest object.using this object we can call any method based on our requirement. We will discuss all one by one

➤ Abort()

This method is used to cancel the current request. This method is used with request object, which is defined by new XMLHttpRequest() method.

Syntax:

```
Object.abort();  
function ajaxdata()  
{  
    var xmlhttp;  
    if(window.XMLHttpRequest)  
    {  
        xmlhttp=new XMLHttpRequest();  
    }  
    xmlhttp.abort();  
}
```

open(*method,url,async*)

This method is used to specify the request. It has three arguments one is method second is URL and third is asynchronous (true) or (False).

In first argument we pass the method which is help to send request on url page. It is used two methods GET and POST, we can use one of the methods to post our data on URL page.

In second argument URL we pass the page name where we want to send a request data. It may be any server request page. We can also append any variable with value in url to send data on request page.

In third argument we pass True or False if it is True then it will send our request asynchronously or if it is False then it will send our request synchronously.

Syntax:

```
Object.open(method,url,asyc);  
function ajaxdata()  
{  
    var xmlhttp;  
    if(window.XMLHttpRequest)  
    {  
        xmlhttp=new XMLHttpRequest();  
    }
```

```

Xmlhttp.open("GET","login_chk.php?username="+user+"&password="+password,true);
xmlhttp.abort();
}

```

➤ send()

This method is used to Sends the request. This method used with GET method.

send(string/content)

This method is used to Sends the request. This method used with POST method. We will pass some argument as a variable and value.

Syntax:

```
Object.send();
```

```
function ajaxdata()
```

```
{
```

```
    var xmlhttp;
```

```
    if(window.XMLHttpRequest)
```

```
{
```

```
        xmlhttp=new XMLHttpRequest();
```

```
}
```

```
Xmlhttp.open("GET","login_chk.php?username="+user+"&password="+password, True);
```

```
xmlhttp.send();
```

```
}
```

➤ getAllResponseHeaders()

This method is used to get the response from server it will get header information. It will bring all kind of data in string otherwise null response.

➤ getResponseHeader()

This method is used to get the response from server, it will get header information. It will bring only text data of particular header value in response.

3.4 XMLHttpRequest Object Properties:

➤ onreadystatechange

This is an event handler it will call when readyState attributes are change. It must not be used with synchronous requests.

➤ readyState

It represent the current state of XMLHttpRequestObject. Following are the different 0 to 4 states.

0 The request is not initialized.

This state is after the XMLHttpRequestObject is created but before the open() method is used.

1. The request has been set up.

This state is after the open() method is called but before the send() method.

2. The request has been sent.

This state is after the send() method is called.

3. The request is in process.

This state after the browser has established a communication with the server, but before the server has completed the response.

4. The request is completed.

This state is after the request has been completed, and the response data has been completely received from the server.

➤ **responseText**

This is returning the response text. When we get response in string then we will use this.

➤ **responseXML**

This is returning the response in XML. When we get XML Document structure in response then we use this.

➤ **status**

This will return the status number as a response. This status are define the response status of web server and web browser.

For example

200 : OK.

400 : Bed Request.

403: Forbidden

404: not Found.

500: Internal server Error.

➤ **statusText**

This will return the status Text as a response. This status are define the response status of web server and web browser. (e.g., "Not Found" or "OK" or "Bed Request" etc.).

Syntax:

xmlhttp.onreadystatechange=function()

```
{  
  
if (xmlhttp.readyState==4 && xmlhttp.status==200)  
{  
  
    document.getElementById("info").innerHTML=xmlhttp.responseText;  
}  
  
}  
  
} // End of the function
```



UNIT- 3 Practical

1. Create a form containing one input field (Name). When the user enters his/her name and as any key is released , the form should display a welcome message for the user. Implement using AJAX.

➤ form_data.php

```
<!DOCTYPE>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<title>Untitled Document</title>  
<script type="text/javascript" language="javascript">  
function showdata()  
{  
    var xmlhttp;  
    if(window.XMLHttpRequest)  
    {  
  
        xmlhttp=new XMLHttpRequest();
```

```

    }
else
{ //for only ie5 and ie6
    xmlhttp=new ActiveXObject('Microsoft.XMLHTTP');
}
var unm=document.getElementById('t').value;
var pass=document.getElementById('t2').value;

xmlhttp.open('POST','ajaxfrm2.php?uname='+unm+'&Password='+pass,true);
xmlhttp.onreadystatechange=function()
{
    if(xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        document.getElementById("info").innerHTML=xmlhttp.responseText;
    }
}
xmlhttp.send();
}
</script>
</head>
<body>
<form name="frm" id="frm" action="post" onsubmit="showdata(); return false">
    Enter Uname:<input type="text" id="t" name="t"/><br />
    Enter Password:<input type="password" id="t2" name="t2"/><br />
    <input type="submit" name="submit" value="submit"/>
</form>
<h1><div id="info"></div></h1>
</body>
</html>

```

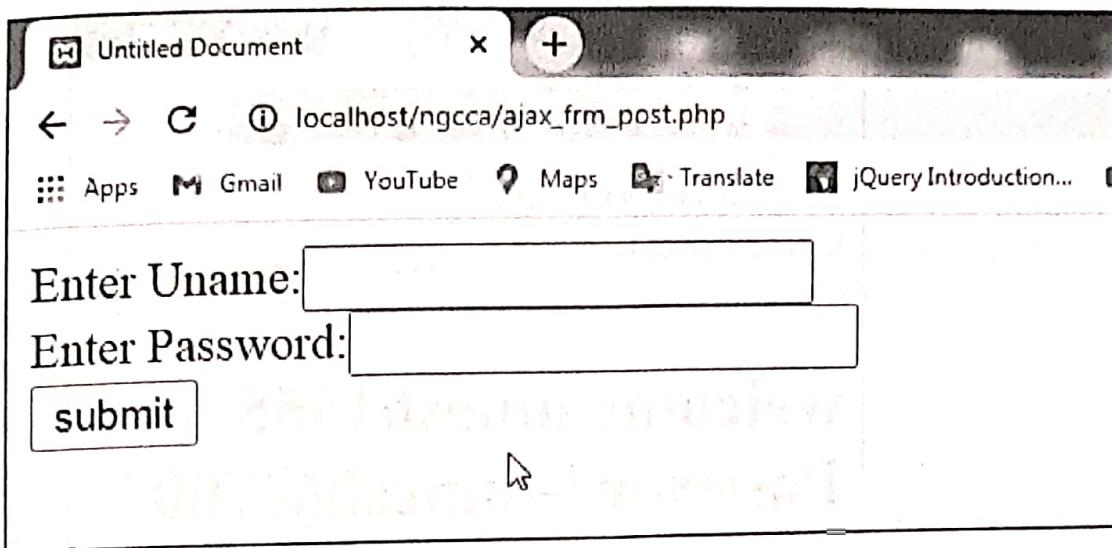


Fig 3.2 Form submission using AJAX

➤ ajaxfrm2.php

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
<?php
$name=$_GET['uname'];
$pass=$_GET['Password'];
if($name=="" || $pass=="")
{
    echo "Please Enter user name and password";
}
else
{
    echo "welcome ".$name;
    echo "<br> Password=".$pass;
}
?>
</body>
</html>
```

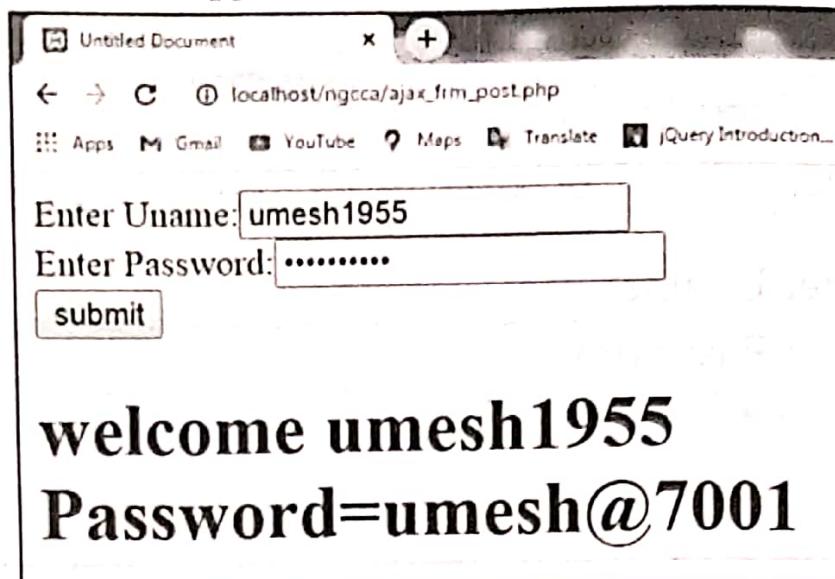


Fig 3.3 Form Submitted with Username and Password in AJAX

- 2. Create a form containing one input field (Name). When the user enters his/her name on keyup, the form should display a welcome message for the user. Implement using AJAX.**

➤ **form_data_2.php**

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script type="text/javascript" language="javascript">
function showdata()
{
    var xmlhttp;
    if(window.XMLHttpRequest)
    {
        xmlhttp=new XMLHttpRequest();
    }
    var unm=document.getElementById('t').value;
    xmlhttp.open('POST','ajaxfrm_keyup.php?uname='+unm,true);
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            document.write(xmlhttp.responseText);
        }
    }
}</script>
<body>
<input type="text" id="t" onkeyup="showdata()"/>
</body>
```

```

document.getElementById("info").innerHTML=xmlhttp.responseText;
}

else
{
document.getElementById("info").innerHTML="Response not Set Something gone
wrong...!";
}

xmlhttp.send();
}
</script>
</head>
<body>
<form name="frm" id="frm" action="post" >
    Enter Uname:<input type="text" id="t" name="t" onkeyup="showdata(); return
false"/><br />
</form>
<h1><div id="info"></div></h1>
</body>
</html>

```

➤ ajaxfrm_keyup.php

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
<?php
$name=$_GET['uname'];
echo "welcome ".$name;
?>
</body>
</html>

```



Fig 3.4 AJAX call on Keyup event

3. Write a program for converting a string into uppercase using AJAX.

Link_upper.php

```
<!DOCTYPE >
<html>
<head>
<title>Untitled Document</title>
<style type="text/css">
#info
{
    color:red;
    background-color:#FFCC33;
}
</style>
<script type="text/javascript" language="javascript" src="ajax_upper.js"></script>
</head>
<body>
Enter your name<input type="text" name="n1" id="n1" value="" />
<!--Enter your name<input type="text" name="n1" id="n1" value="" onkeyup="uppercase()"/-->
<a href="javascript:uppercase()">Uppercase</a>
<h1 id="info"></h1>
</body>
```

```

</html>
Ajax_upper.js
// JavaScript Document
function uppercase()
{
    var xmlhttp;
    if(window.XMLHttpRequest)
    {
        xmlhttp=new XMLHttpRequest();
    }
    else
    { //for only ie5 and ie6
        xmlhttp=new ActiveXObject('Microsoft.XMLHTTP');
    }
    var n1=document.getElementById('n1').value;
    xmlhttp.open('GET','ajaxfrm_upper.php?n='+n1,true);
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            document.getElementById("info").innerHTML=xmlhttp.responseText;
        }
    }
    xmlhttp.send();
}

```

➤ ajaxfrm_upper.php

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>

```

```

</head>
<body>
<?php
$name=$_GET['n'];
if($name!="")
{
    echo strtoupper($name);
}
else
{
    echo "Please Enter Name";
}
?>
</body>
</html>

```

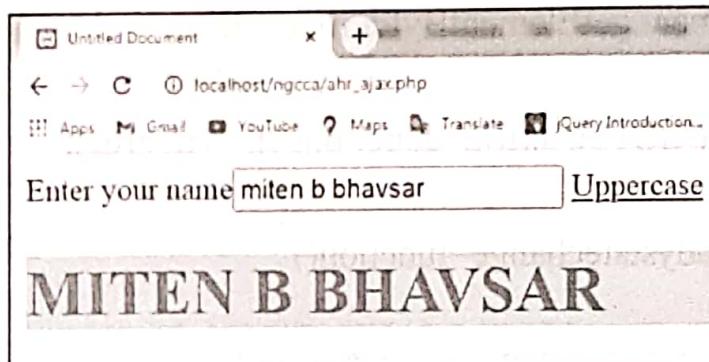


Fig 3.5 AJAX call on anchor tag to convert lowercase to uppercase.

4. Create a form containing a combobox with some product names as items. Whenever a user selects a particular product from the combobox, it shuold be sent to the server asynchronously (i.e. without pressing submit button). Implement using AJAX.

➤ **Combobox_ajax.php**

```

<!DOCTYPE html >
<html>
<head>
<title>Untitled Document</title>
<script type="text/javascript" language="javascript">
function catdisplay()

```

```
{  
var xmlhttp;  
if(window.XMLHttpRequest)  
{  
    xmlhttp=new XMLHttpRequest();  
}  
var r=document.getElementById('cat').value;  
xmlhttp.open('GET','cat_data.php?sr='+r,true);  
xmlhttp.onreadystatechange=function()  
{  
    if (xmlhttp.readyState==4 && xmlhttp.status==200)  
    {  
        document.getElementById("info").innerHTML=xmlhttp.responseText;  
    }  
}  
xmlhttp.send();  
}  
</script>  
</head>  
<body>  
<select name="cat" id="cat" onchange="catdisplay()">  
    <option value="">Please Select</option>  
    <option value="Camera">Camera</option>  
    <option value="Mobile">Mobile</option>  
    <option value="Computer">Computer</option>  
    <option value="Keyboard">Keyboard</option>  
</select>  
<h1 id="info"></h1>  
</body>  
</html>
```

➤ **Connection.php**
<?php
//connection code

```

error_reporting("E_ALL");
$server="localhost";
$username="root";
$password="";
$database="onlineshoping";
$cnn=mysqli_connect($server,$username,$password,$database);
if(!$cnn)
{
    echo "Connection is not proper";
}
?>

```

➤ Product_search.php

```

<!DOCTYPE>
<html>
<head>
<title>Untitled Document</title>
</head>
<body>
<?php
include("connection.php");
$sr=$_GET['sr'];
if($sr!="")
{
    $sel="select * from product where Product_Name like '".$sr."'";
    $query=mysqli_query($cnn,$sel);
    $num=mysqli_num_rows($query);
    if($num>0)
    {
        echo "<table width=100% border=1>
<tr>
<th>Product id</th>
<th>Product Name</th>
<th>Product Price</th>
<th>Product Quantity</th>

```

```

</tr> x si mera add operation b of array me a jaisa a
";  

while($row=mysqli_fetch_array($query))  

{  

echo "<tr>  

<td>$row[Product_ID]</td>  

<td>$row[Product_Name]</td>  

<td>$row[Product_Price]</td>  

<td>$row[Product_QTY]</td>  

</tr>  

";  

}  

echo "</table>";  

}  

else  

{  

echo "Product Not Found.";  

}  

}  

else  

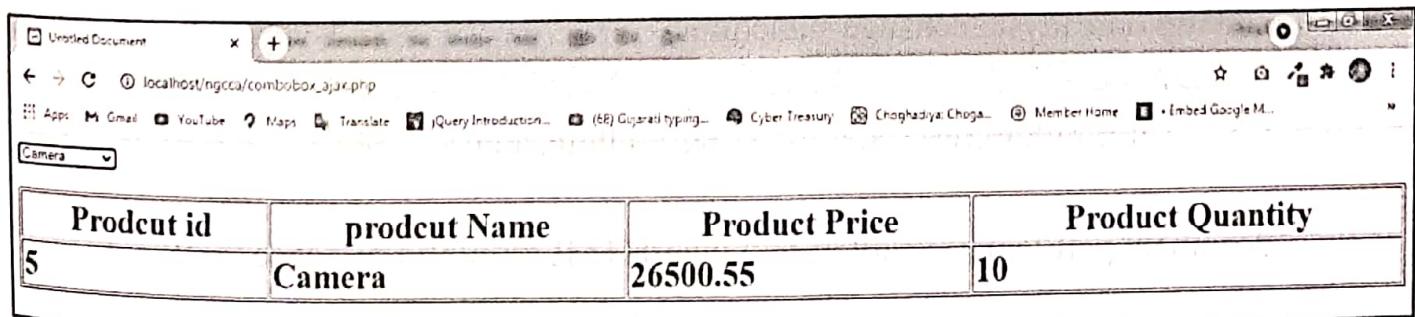
{  

echo "please Try again";  

}  

?>
</body>
</html>

```



| Prodcut id | product Name | Product Price | Product Quantity |
|------------|--------------|---------------|------------------|
| 5 | Camera | 26500.55 | 10 |

Fig. 3.6 AJAX call on select option onchange of product.

- 5. Write a program to demonstrate the example of sending items selected from radio and checkbox to server asynchronously.**

➤ **Option_frm.php**

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script type="text/javascript" language="javascript">
function selection()
{
    var xmlhttp;
    if(window.XMLHttpRequest)
    {
        xmlhttp=new XMLHttpRequest();
    }
    var r=document.getElementsByName("game");
    var game;
    for(var i=0;i<r.length;i++)
    {
        if(r[i].checked==true)
        {
            game=r[i].value;
        }
    }
    xmlhttp.open('GET','radio.php?radioselect='+game,true);
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            document.getElementById("info").innerHTML=xmlhttp.responseText;
        }
    }
}
```

```
xmlhttp.send();  
}  
</script>  
</head>  
<body style="margin-left:500px;">  
<input type="radio" name="game" value="Cricket" onclick="selection()" />Cricket<br/>  
<input type="radio" name="game" value="Football" onclick="selection()" />Foot  
Ball<br />  
<input type="radio" name="game" value="Tenis" onclick="selection()" />Tenis<br />  
<!--<input type="button" name="game" value="Tenis" onclick="selection()" /><br />-->  
  
<h1 id="info"></h1>  
</body>  
</html>
```

➤ **radio.php**

```
<?php  
$r=$_GET['radioselect'];  
echo "you have selected ". $r;  
?>
```

➤ **Checkbox_frm.php**

```
<!DOCTYPE html >  
<html>  
<head>  
<title>Untitled Document</title>  
<script type="text/javascript" language="javascript">  
function selection()  
{  
    var xmlhttp;  
    if(window.XMLHttpRequest)  
    {  
        xmlhttp=new XMLHttpRequest();  
    }  
    var r=document.getElementsByName("game");
```

```

var game="";
var cat;
for(var i=0;i<r.length;i++)
{
    if(r[i].checked==true)
        document.getElementById("checkbox").value+=r[i].value+" ";
    game+=r[i].value+" ";
}
xmlhttp.open('GET','checkbox.php?checkselect='+game,true);
xmlhttp.onreadystatechange=function()
{
    if(xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        document.getElementById("info").innerHTML=xmlhttp.responseText;
    }
}
xmlhttp.send();
}

</script>
</head>

<body style="margin-left:500px;">
<input type="checkbox" name="game" value="Cricket"/>Cricket<br />
<input type="checkbox" name="game" value="Football"/>Foot Ball<br />
<input type="checkbox" name="game" value="Tenis"/>Tenis<br />
<input type="button" name="g" value="Click" onclick="selection()"/><br />

<h1 id="info"></h1>
</body>
</html>

```

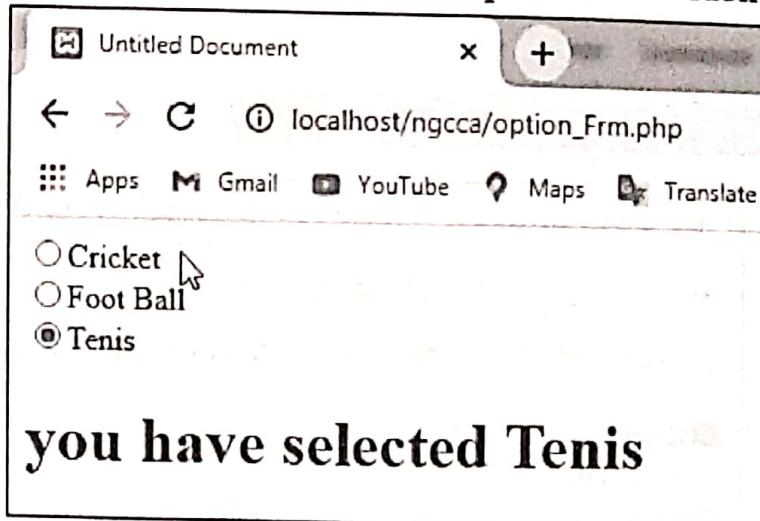


Fig 3.7 AJAX call on Radio/Option button

➤ Checkbox.php

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
<?php
$ri=$_GET['checkselect'];
$str=explode(" ",$ri);
for($i=0;$i<count($str);$i++)
{
    echo $str[$i]."<br>";
}
if($r=="")
{
    echo "you have not selected any item";
}
else
{
    echo "you have selected <span style=color:red;>".$r."</span>";
}
?>
```

```
</body>  
</html>
```

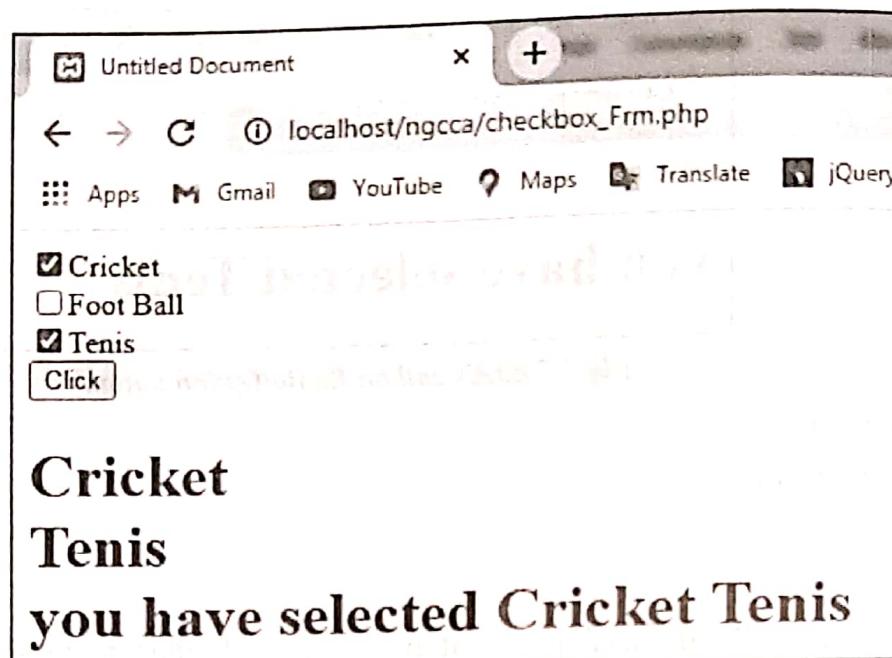


Fig. 3.8 AJAX calls on Checkbox.

6. Write a program to validate a blank field and also validate the length of the data entered (i.e. minimum length of 5).

7. Write a program to validate an Email ID using regular expression and by using DOM.

```
<!DOCTYPE html>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<title>Untitled Document</title>  
<style type="text/css">  
.error  
{  
color:red;  
font-weight:bold;  
}  
</style>  
<script type="text/javascript" language="javascript">
```

```

function validate()
{
    //fullname validation/
    var fullname = document.getElementById('fname').value;
    //var len=fullname.length;
    if(fullname=="")
    {
        document.getElementById('msg1').innerHTML="Please Enter
First Name";
        document.getElementById('fname').style.outlineColor="red";
        document.getElementById('fname').focus();
        return false;
    }
    else
    {
        document.getElementById('msg1').innerHTML="";
    }
    //Last name Validation
    var lnm=document.getElementById("lname").value;
    if(lnm=="")
    {
        document.getElementById("msglnm").innerHTML="Please Enter Last
Name";
        return false;
    }
    else
    {
        document.getElementById("msglnm").innerHTML="";
    }

    //picode validation
    var pin=document.getElementById("pin").value;
    var pinpattern=/^([0-9]{6})$/;
    if(!pinpattern.test(pin))

```

```
{  
    document.getElementById("msgpin").innerHTML="Please enter valid  
    pincode no";  
    document.getElementById("msgpin").focus();  
    return false;  
}  
else  
{  
    document.getElementById("msgpin").innerHTML="";  
}  
  
//Contact validation/  
var patern=/^[\+][0-9\s]{4}[0-9\s]{6}[0-9]{5}$/;  
  
var contactno = document.getElementById("cno").value;  
if(!patern.test(contactno))  
{  
    document.getElementById('msg2').innerHTML="Please Enter Proper  
    Contactno";  
    document.getElementById('cno').style.outlineColor="red";  
    document.getElementById('cno').focus();  
    return false;  
}  
else  
{  
    document.getElementById('msg2').innerHTML="";  
}  
/*var len=contactno.length;  
alert(len);  
alert(contactno);  
if(len==0)  
{  
    document.getElementById('msg2').innerHTML="Please Enter Contactno";  
    document.getElementById('cno').style.outlineColor="red";  
}
```

```
document.getElementById('cno').focus();
return false;
}/*
/*if(contactno=="")
{
    document.getElementById('msg2').innerHTML="Please Enter
Contactno";
    document.getElementById('cno').style.outlineColor="red";
    document.getElementById('cno').focus();
    return false;
}
//var len=contactno.length
if(len==10)
{
    document.getElementById('msg2').innerHTML="";
}
else
{
    document.getElementById('msg2').innerHTML="Please Enter Ten Digit
Number";
    document.getElementById('cno').style.outlineColor="red";
    document.getElementById('cno').focus();
    return false;
}
if(!Number(contactno))
{
    document.getElementById('msg2').innerHTML="Please Enter Numeric value
only";
    document.getElementById('cno').style.outlineColor="red";
    document.getElementById('cno').focus();
    return false;
}*/
```

```
//Email Validation
var emailpattern=/^[_a-zA-Z0-9]+@[a-zA-Z]+\.[a-zA-Z]{2,4}$/;
var emailID = document.getElementById('email').value;

if(!emailpattern.test(emailID))
{
    document.getElementById('msg3').innerHTML="Please enter correct email ID";
    document.getElementById('email').style.outline="green";
    document.getElementById('email').focus();
    return false;
}
else
{
    document.getElementById('msg3').innerHTML="";
}

/*var atpos = emailID.indexOf("@");
//alert(atpos);
var dotpos = emailID.indexOf(".");
//alert(dotpos);
if (atpos < 1 || ( dotpos - atpos < 2 ))
{
    //alert("Please enter correct email ID")
    document.getElementById('msg3').innerHTML="Please enter correct email ID";
    document.getElementById('email').style.outline="green";
    document.getElementById('email').focus();
    return false;
}
else
{
    document.getElementById('msg3').innerHTML="";
}
```

```
}*/
```

```
//radiobutton validation
```

```
var gr = document.getElementsByName('g');
```

```
//alert(gr);
```

```
var lan=gr.length;
```

```
//alert(lan);
```

```
var cnt=0;
```

```
var i;
```

```
//alert("counter"+cnt);
```

```
//alert("lan"+lan);
```

```
for(i=0;i<lan;i++)
```

```
{
```

```
//alert("counter"+cnt);
```

```
if(gr[i].checked==true)
```

```
{
```

```
//alert("counter"+cnt);
```

```
//var g = gr[i].value;
```

```
//alert(g)
```

```
document.getElementById('msg4').innerHTML="";
```

```
cnt=1;
```

```
//alert(cnt);
```

```
break;
```

```
}
```

```
}
```

```
if(cnt==0)
```

```
{
```

```
//alert("counter"+cnt);
```

```
document.getElementById('msg4').innerHTML="Please Select atleast one  
Gender";
```

```
return false;
```

```
}
```

```
//checkbox validation
var ch = document.getElementsByName('c');
//alert(ch);
var len=ch.length;
var cnt=0;
for(var i=0;i<len;i++)
{
    if(ch[i].checked==true)
    {
        /*var val = new Array()
        val[i]=ch[i].value;
        alert(val[i]);*/
        document.getElementById('msg5').innerHTML="";
        cnt=1;
        //alert("counter"+cnt);
        //break;
    }
}
if(cnt==0)
{
    //alert("counter"+cnt);
    document.getElementById("msg5").innerHTML="Please Select atleast one option";
    return false;
}

//combobox validation
var sel=document.getElementById("clr").value;
//alert(sel);
if(sel=="")
{
    document.getElementById('msg6').innerHTML="Please Select atleast one option";
    return false;
```

```
        }
    else
    {
        document.getElementById('msg6').innerHTML="";
    }

//multiple selection in listbox validation
//combobox validation
var sel=document.getElementById("v").value;
//alert(sel);
if(sel=="")
{
    document.getElementById('msg7').innerHTML="Please Select atlist one option";
    return false;
}
else
{
    document.getElementById('msg7').innerHTML="";
}

//Date validation
var db=document.getElementById("db").value;
//alert(db);
var date= new Date();
var dd=date.getDate();
//alert(dd);
var mm=date.getMonth()+1;
//alert(mm);
var yy=date.getFullYear();
//alert(yy);
if(dd<=9)
{
    dd="0"+dd;
}
if(mm<=9)
```

```

    {
        mm="0"+mm;
    }
    var db2= yy+"-"+mm+"-"+dd; //converting dd to mm
    //alert(db2);

    if(db=="")
    {
        document.getElementById('msg8').innerHTML="Please Enter Date of Birth";
        document.getElementById('db').style.outlineColor="red";
        document.getElementById('db').focus();
        return false; //return false to prevent form submission
    }
    //alert(db+"#####"+db2);
    if(db>db2)
    {
        document.getElementById('msg8').innerHTML="Date of Birth is not Greater than Current Date";
        document.getElementById('db').style.outlineColor="red";
        document.getElementById('db').focus();
        return false;
    }
    else
    {
        document.getElementById('msg8').innerHTML="";
    }

//range validation
var age=document.getElementById("rg").value;
//alert(age);
if(age==0)
{
    document.getElementById('msg9').innerHTML="Your Age must be Greater than Zero";
}

```

```
return false;  
}  
else  
{  
    document.getElementById('msg9').innerHTML="";  
}  
}  
  
function getval()  
{  
    var age=document.getElementById("rg").value;  
    document.getElementById("ans").innerHTML=age;  
}  
/*function can()  
{  
    window.open("../1.html");  
}*/  
</script>  
</head>  
<body>  
<form name="registration" action="#" method="post">  
<table>  
    <tr>  
        <td>  
            First Name  
        </td>  
        <td>  
            <input type="text" name="fname" id="fname" value="" /><span id="msg1" class="error">for ex. Maulik Patel</span>  
        </td>  
    </tr>  
    <tr>  
        <td>  
            Last Name  
        </td>
```

```
<td>
<input type="text" name="lname" id="lname" value="" /><br /><span id="msglnm"
class="error"></span>
</td>
</tr>
<tr>
<td>
    Pincode
</td>
<td>
<input type="text" name="pin" id="pin" value="" placeholder="123456"/><br /><span
id="msgpin" class="error"></span>
</td>
</tr>
<tr>
<td>
    Contactno
</td>
<td>
<input type="text" name="cno" id="cno" value="" placeholder="+091 12345
12345"/><span id="msg2" class="error"></span>
</td>
</tr>
<tr>
<td>
    E-mail
</td>
<td>
<input type="text" name="email" id="email" value="" placeholder="abcd@abcd.com"/><span id="msg3" class="error"></span>
</td>
</tr>
<tr>
<td>
    Gender

```

```

</td>
<td>
    <input type="radio" name="g" value="male" />Male
    <input type="radio" name="g" value="female" />Female-->
<tr>
<td>Color</td>

```

```

<td>
<select name="clr" id="clr">
<option value="">Please Select</option>
<option value="red">red</option>
<option value="green">green</option>
<option value="blue">blue</option>
</select><span id="msg6" class="error"></span>
</td>
</tr>
<tr>
<td>Vehicle</td>
<td>
<select name="v" id="v" size="3" multiple="multiple">
<option value="">Please Select</option>
<option value="Cycle">Cycle</option>
<option value="car">car</option>
<option value="scooter">scooter</option>
<option value="Riksha">Riksha</option>
</select><span id="msg7" class="error"></span>
</td>
</tr>
<tr>
<td>Date of Birth</td>
<td><input type="date" id="db" value="" />
<span id="msg8" class="error"></span>
</td>
</tr>
<tr>
<td>Age</td>
<td><input type="range" id="rg" value="0" min="0" max="100" step="1" onmousemove="getval(); ">
<span id="msg9" class="error"></span>
<span id="ans"></span>
</td>

```

```

</tr>
<tr>
<td>&nbsp;
</td>
<td>
<input type="submit" name="submit" value="Ok" onclick="return(validate());" />
<input type="button" name="cancel" value="cancel" onclick="can()"/>
</td>
</tr>

</table>
</form>
</body>
</html>

```

Untitled Document

File | D:/html%20practical/javascript/validation.html#

Apps Gmail YouTube Maps Translate jQuery Introduction...

First Name	<input type="text" value="Maulik"/>
Last Name	<input type="text" value="Patel"/>
Pincode	<input type="text" value="380"/> Please enter valid pincode no
Contactno	<input type="text" value="+091 12345 12345"/>
E-mail	<input type="text" value="abcd@abcd.com"/>
Gender	<input type="radio"/> Male <input type="radio"/> Female
Sports	<input type="checkbox"/> Cricket <input type="checkbox"/> FootBall <input type="checkbox"/> TableTenis
Color	<input type="text" value="Please Select"/>
Vehicle	<input type="text" value="Please Select"/> Cycle car
Dateof Birth	<input type="text" value="dd/mm/yyyy"/> <input type="button" value=""/>
Age	<input type="text" value="0"/>
<input type="button" value="Ok"/> <input type="button" value="cancel"/>	

Untitled Document

File | D:/html%20practical/javascript/validation.html#

Apps Gmail YouTube Maps Translate jQuery Introduction...

First Name	<input type="text"/>	Please Enter First Name
Last Name	<input type="text" value="Patel"/>	
Pincode	<input type="text" value="380050"/>	
Contactno	<input type="text" value="+091 94275 20202"/>	
E-mail	<input type="text" value="maulik7001@gmail.com"/>	
Gender	<input type="radio"/> Male <input type="radio"/> Female	
Sports	<input type="checkbox"/> Cricket <input type="checkbox"/> FootBall <input type="checkbox"/> TableTenis	
Color	<input type="text" value="Please Select"/>	
Vehicle	<input type="text" value="Please Select"/> Cycle car	
Dateof Birth	<input type="text" value="dd/mm/yyyy"/> <input type="button" value=""/>	
Age	<input type="text" value="0"/>	
<input type="button" value="Ok"/> <input type="button" value="cancel"/>		

The figure consists of two side-by-side screenshots of a web browser window. Both windows have the title 'Untitled Document' and the URL 'D:/html%20practical/javascript/validation.html#'. The browser interface includes standard navigation buttons (back, forward, search), tabs (File, Apps, Gmail, YouTube, Maps, Translate, jQuery Introduction), and a toolbar.

Left Screenshot: This shows a form with various input fields. The 'E-mail' field contains 'maulik@gmail.com' and has a red border, indicating it is invalid. A tooltip message 'Please enter correct email ID' is displayed below the field. Other fields include 'First Name' (Maulik), 'Last Name' (Patel), 'Pincode' (330050), 'Contactno' (+091 94275 20202), 'Gender' (radio buttons for Male and Female), 'Sports' (checkboxes for Cricket, FootBall, TableTenis, with a note 'Please Select at least one opt'), 'Color' (dropdown menu 'Please Select'), 'Vehicle' (dropdown menu 'Please Select' with options Cycle and car), 'Dateof Birth' (date input field dd/mm/yyyy with a red border), and 'Age' (slider set to 0). Buttons 'Ok' and 'cancel' are at the bottom.

Right Screenshot: This shows the same form after some changes. The 'E-mail' field now contains 'maulik7001@gmail.com'. The 'Dateof Birth' field is now set to '05/05/2021' and also has a red border, with a tooltip 'Date of Birth is not Greater than Current Date'. All other fields and their values remain the same as in the left screenshot.

Fig. 3.9 JavaScript Validation for emptyfield,email and length.

- 8. Write a program that checks a particular stuid already exists in the student(stuid,stu_name,mob,country) table or not. If stuid exists then display a message "User Already Exist. Try another stuid". If it does not exists then add the data in the student table.Implement using AJAX.**

➤ **std_id_varify.php**

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
<script type="text/javascript" language="javascript">
function cancell1()
{
    document.location.href="std_id_varify.php";
    return false;
}
</script>
<script type="text/javascript" language="javascript">
```

```

function varifystdid()
{
    var xmlhttp;
    if(window.XMLHttpRequest)
    {
        xmlhttp=new XMLHttpRequest();
    }
    var fname=document.getElementById("fname").value;
    var contact=document.getElementById("contact").value;
    var email=document.getElementById("email").value;
    var address=document.getElementById("address").value;
    xmlhttp.open('GET','validate_id.php?fname='+fname+'&contact='+contact+'&email='+
    +email+'&address='+address,true);
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            document.getElementById("info").innerHTML=xmlhttp.responseText;
        }
    }
    xmlhttp.send();
}

function resetdata()
{
    document.getElementById("fname").value="";
    document.getElementById("contact").value="";
    document.getElementById("email").value="";
    document.getElementById("address").value="";
}

</script>
</head>
<body>
<input type="hidden" value="Add" name="action"/>

```

```
<table>
  <tr>
    <td>Full Name</td>
    <td><input type="text" name="fname" id="fname" value="" /></td>
  </tr>
  <tr>
    <td>Contact No.</td>
    <td><input type="text" name="contact" id="contact" value="" /></td>
  </tr>
  <tr>
    <td>Email</td>
    <td><input type="text" id="email" name="email" value="" /></td>
  </tr>
  <tr>
    <td>Address</td>
    <td><textarea name="address" id="address"></textarea></td>
  </tr>
  <tr>
    <td><input type="submit" name="submit" value="Submit"
      onClick="varifystdid();"/></td>
    <td><input type="reset" value="reset" name="reset" onClick="resetdata();"/></td>
    <td><input type="button" value="cancel" name="cancel" onclick="cancel1();"/></td>
  </tr>
</table>
<span id="info" style="color:red;font-size:24px;"></span>
</body>
</html>
```

➤ **validate_id.php**

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
```

```
<body>
<?php
include("conection.php");
$name=$_GET["fname"];
$contact=$_GET["contact"];
$email=$_GET["email"];
$address=$_GET["address"];
$select="select Email from student where Email='$email'";
$res=mysqli_query($con,$select);
$row=mysqli_num_rows($res);
if($row>0)
{
    echo "$email Allready Exist";
}
else
{
    $q="insert into student(Stud_id,FullName,Contactno,Email,Address) values
    ('','$name','','$contact','','$email','','$address')";
    $inq=mysqli_query($con,$q);
    if($inq)
    {
        header("Location:index.php?msg=Record Inserted Successfully...!");
    }
}
?>
</body>
</html>
```