AIM:

1. Write a PHP program to find the largest value among the three inputted values using if-else construct.

PROGRAM CODE:

```
<?php
$n=0;
session_start();
print "<form method='POST' action=">";
print "Enter 3 numbers:";
print "<input type='text' name='t1'>";
print "<input type='text' name='t2'>";
print "<input type='text' name='t3'>";
print "<input type='submit' name='s1'>";
print "</form>";

if(isset($_POST['s1']))
{
  $n1=$_POST['t1'];
  $n2=$_POST['t2'];
  $n3=$_POST['t3'];
  if( $n1>$n2 && $n1>$n3)
  {
    print "<br> The greatest number is $n1";
  }
  else if($n2>$n3)
  {
   print "<br> The greatest number is $n2";
  }
  else
  {
   print "<br> The greatest number is $n3";
  }
}
?>
```

OUTPUT:

Enter 3 numbers: 982 457 1257 Submit

The greatest number is 1257

AIM:

2. Write a PHP program to demonstrate while, do while and for each loop.

PROGRAM CODE:

```
<?php
$n=0;
session_start();
print "<form method='POST' action=">";
print "<input type='text' name='t1' placeholder='Enter number of elements'>";
print "<input type='submit' name='s1'>";
print "</form>";

if(isset($_POST['s1']))
{
  $n=$_POST['t1'];
  $_SESSION['n']=$n;
  print "<form method='POST' action=">";
  for($i=0;$i<$n;$i++)
  {
    print "<input type='text' name='txt$i' placeholder='Enter number'>";
  }
  print "<input type='submit' name='s2'>";
}

$ar=array();
$pal=array();
if(isset($_POST['s2']))
{
  print "<br><br>Entered second button definition...<br><br>";
  $n=$_SESSION['n'];
  for($i=0;$i<$n;$i++)
  {
    $ar[$i]=$_POST['txt'.$i];
  }

  // Process array to find palindromes
  foreach($ar as $a)
  {
    $temp=$a;
    $sum=0;

    while($temp>0)
    {
      $r=$temp%10;
      $sum=$sum*10+$r;
      $temp=(int)($temp / 10);
    }
    if($a==$sum)
    {
      $pal[]=$a;
```

```
  }
 }

 // Output original and palindrome arrays
 print "<br> The original array:";
 print_r($ar);
 print "<br> The palindrome array:";
 print_r($pal);

 $i = 0;
 print "<br><br>Using while loop to print original array:";
 while ($i < count($ar)) {
  print "<br>Element " . ($i + 1) . ": " . $ar[$i];
  $i++;
 }

 $i = 0;
 print "<br><br>Using do while loop to print palindrome array:";
 if (count($pal) > 0) {
  do {
   print "<br>Palindrome " . ($i + 1) . ": " . $pal[$i];
   $i++;
  } while ($i < count($pal));
 } else {
  print "<br>No palindromes found.";
 }

 print "<br><br>Using foreach loop to print original array with indices:";
 foreach ($ar as $index => $value) {
  print "<br>Element " . ($index + 1) . ": " . $value;
 }
}
?>
```

OUTPUT:

Enter number of elements Submit

Entered second button definition...

The original array:Array ([0] => 121 [1] => 345 [2] => 232 [3] => 2412 [4] => 2390)
The palindrome array:Array ([0] => 121 [1] => 232)

Using while loop to print original array:
Element 1: 121
Element 2: 345
Element 3: 232
Element 4: 2412
Element 5: 2390

Using do while loop to print palindrome array:
Palindrome 1: 121
Palindrome 2: 232

Using foreach loop to print original array with indices:
Element 1: 121
Element 2: 345
Element 3: 232
Element 4: 2412
Element 5: 2390

AIM:

3. Write a PHP program to demonstrate various types of arrays.

PROGRAM CODE:

```
<?php
// Indexed Array
$indexedArray = array("Apple", "Banana", "Cherry");
echo "<h3>Indexed Array</h3>";
echo "Elements in the indexed array:<br>";
for ($i = 0; $i < count($indexedArray); $i++) {
   echo "Element " . $i . ": " . $indexedArray[$i] . "<br>";
}

// Associative Array
$associativeArray = array("first" => "Red", "second" => "Green", "third" => "Blue");
echo "<h3>Associative Array</h3>";
echo "Elements in the associative array:<br>";
foreach ($associativeArray as $key => $value) {
   echo "Key: " . $key . ", Value: " . $value . "<br>";
}

// Multidimensional Array
$multiArray = array(
   array("Name" => "John", "Age" => 25, "City" => "New York"),
   array("Name" => "Jane", "Age" => 30, "City" => "Los Angeles"),
   array("Name" => "Paul", "Age" => 28, "City" => "Chicago")
);
echo "<h3>Multidimensional Array</h3>";
echo "Elements in the multidimensional array:<br>";
foreach ($multiArray as $index => $subArray) {
   echo "Record " . ($index + 1) . ":<br>";
   foreach ($subArray as $key => $value) {
      echo "   " . $key . ": " . $value . "<br>";
   }
}
?>
```

OUTPUT:

Indexed Array

Elements in the indexed array:
Element 0: Apple
Element 1: Banana
Element 2: Cherry

Associative Array

Elements in the associative array:
Key: first, Value: Red
Key: second, Value: Green
Key: third, Value: Blue

Multidimensional Array

Elements in the multidimensional array:
Record 1:
  Name: John
  Age: 25
  City: New York
Record 2:
  Name: Jane
  Age: 30
  City: Los Angeles
Record 3:
  Name: Paul
  Age: 28
  City: Chicago

AIM:

4. Write a PHP program to sort an array using functions.

PROGRAM CODE:

```
<?php
$n=0;
session_start();
print "<form method='POST' action=">";
print "Enter limit:";
print "<input type='text' name='t1'>";
print "<input type='submit' name='s1'>";
print "</form>";

if(isset($_POST['s1']))
{
  $n=$_POST['t1'];
  $_SESSION['n']=$n;
  print "<form method='post' action=">";
  for($i=0;$i<$n;$i++)
  {
    print "<input type='text' name='txt$i'>";
  }
  print "<input type='submit' name='s2'>";

}
$as=array();
if(isset($_POST['s2']))
{
  $n=$_SESSION['n'];
  for($i=0;$i<$n;$i++)
  {
   $ar[$i]=$_POST['txt'.$i];
  }
  print "<br><br>The array values are...<br><br> ";
  print_r($ar);
  sort($ar);
  print "<br><br>The sorted array values are...<br><br> ";
  print_r($ar);
  rsort($ar);
  print "<br><br>The reverse sorted array values are...<br><br> ";
  print_r($ar);

}
print "</form>";
?>
```

OUTPUT:

Enter limit: Submit

The array values are...

Array ([0] => 1234 [1] => 43245 [2] => 423 [3] => 235 [4] => 6794)

The sorted array values are...

Array ([0] => 235 [1] => 423 [2] => 1234 [3] => 6794 [4] => 43245)

The reverse sorted array values are...

Array ([0] => 43245 [1] => 6794 [2] => 1234 [3] => 423 [4] => 235)

AIM:

5. Write a PHP program to find prime numbers in an array using functions.

PROGRAM CODE:

```
<?php

function isPrime($num) {
   if ($num <= 1) {
      return false;
   }
   for ($i = 2; $i <= sqrt($num); $i++) {
      if ($num % $i == 0) {
         return false;
      }
   }
   return true;
}

function findPrimeNumbers($arr) {
   $primeNumbers = array();
   foreach ($arr as $number) {
      if (is_numeric($number) && isPrime($number)) {
         $primeNumbers[] = $number;
      }
   }
   return $primeNumbers;
}

session_start();
print "<form method='POST' action=">";
print "Enter limit:";
print "<input type='text' name='t1'>";
print "<input type='submit' name='s1'>";
print "</form>";

if (isset($_POST['s1'])) {
   $n = $_POST['t1'];
   $_SESSION['n'] = $n;

   print "<form method='post' action=">";
   for ($i = 0; $i < $n; $i++) {
      print "<input type='text' name='txt$i'>";
   }
   print "<input type='submit' name='s2'>";
   print "</form>";
}

if (isset($_POST['s2'])) {
   $n = $_SESSION['n'];
   $ar = array();
```

```
    for ($i = 0; $i < $n; $i++) {
        $ar[$i] = $_POST['txt'.$i];
    }

    $primeNumbers = findPrimeNumbers($ar);

    echo "<br>Prime numbers in the array: ";
    foreach ($primeNumbers as $prime) {
        echo $prime . " ";
    }
}

?>
```

OUTPUT:

Enter limit: Submit

2	61	23	9087	54321	Submit

Enter limit: Submit

Prime numbers in the array: 2 61 23

AIM:

6. Write a PHP program to find the Fibonacci series based on a specified range using functions.

PROGRAM CODE:

```
<?php
session_start();
function fibonacciSeries($n) {
   $n1 = 0;
   $n2 = 1;
   for ($i = 0; $i < $n; $i++) {
      print $n1 . " ";
      $n3 = $n1 + $n2;
      $n1 = $n2;
      $n2 = $n3;
   }
}

print "<form method='POST' action=">";
print "Enter the number of terms for Fibonacci series: ";
print "<input type='text' name='t1'>";
print "<input type='submit' name='s1'>";
print "</form>";

if (isset($_POST['s1']))
{
   $n = $_POST['t1'];
   if (is_numeric($n) && $n > 0)
   {
      fibonacciSeries((int)$n);
   }
   else
   {
      echo "Please enter a positive integer.";
   }
}
?>
```

OUTPUT:

Enter the number of terms for Fibonacci series: 10 Submit

0 1 1 2 3 5 8 13 21 34

AIM:

7. Write a PHP program to find the greatest age from the ages of 2 persons and display it using functions. The function should accept the age of 2 persons as arguments and return the largest value of age.

PROGRAM CODE:

```
<?php
function findGreatestAge($age1, $age2)
{
   if ($age1 > $age2)
   {
      return $age1;
   }
   else
   {
      return $age2;
   }
}

print "<form method='POST' action=">";
print "Enter age of person 1: <input type='text' name='age1'><br>";
print "Enter age of person 2: <input type='text' name='age2'><br>";
print "<input type='submit' name='submit' value='Find Greatest Age'>";
print "</form>";

// Check if form is submitted
if (isset($_POST['submit']))
{
   $age1 = $_POST['age1'];
   $age2 = $_POST['age2'];

   if (is_numeric($age1) && $age1 > 0 && is_numeric($age2) && $age2 > 0)
   {
      $greatestAge = findGreatestAge((int)$age1, (int)$age2);
      echo "The greatest age between the two persons is: " . $greatestAge;
   }
   else
   {
      echo "Please enter valid positive ages for both persons.";
   }
}
?>
```

OUTPUT:

Enter age of person 1: 12
Enter age of person 2: 56
Find Greatest Age

The greatest age between the two persons is: 56

AIM:

8. Write a PHP program to demonstrate default parameters in functions.

PROGRAM CODE:

```
<?php
function calculateArea($length, $width = null)
{
   if ($width === null || $width == 0)
   {
      $width = $length;
   }
   return $length * $width;
}

print "<form method='POST' action=">";
print "<br>Enter Length: ";
print "<input type='text' name='t1'>";
print "<br>Enter Width (leave empty for square): ";
print "<input type='text' name='t2'>";
print "<input type='submit' name='s1'>";
print "</form>";

if (isset($_POST['s1']))
{
   $l = $_POST['t1'];
   $b = $_POST['t2'];

   if (is_numeric($l) && $l > 0)
   {
      if (empty($b) || $b == 0)
      {
         $ar = calculateArea((int)$l);
         echo "<br>The area of the square with length $l is: $ar<br>";
      }
      else if (is_numeric($b) && $b > 0)
      {
         $ar = calculateArea((int)$l, (int)$b);
         echo "<br>The area of the rectangle with length $l and width $b is: $ar<br>";
      }
      else
      {
         echo "Please enter a positive number for the width.";
      }
   }
   else
   {
      echo "Please enter a positive number for the length.";
   }
}
?>
```

OUTPUT:

Enter Length: 5
Enter Width (leave empty for square): Submit

The area of the square with length 5 is: 25

Enter Length: 20
Enter Width (leave empty for square): 13 Submit

The area of the rectangle with length 20 and width 13 is: 260

AIM:

9. Write a PHP program to demonstrate single inheritance.

PROGRAM CODE:

```
<?php
class Person
{
   protected $name;

   public function __construct($name)
   {
      $this->name = $name;
   }

   public function getName()
   {
      return $this->name;
   }
}


class DisplayPerson extends Person
{
   public function display()
   {
      echo "The name is: " . $this->getName();
   }
}

print "<form method='POST' action=">";
print "Enter Name: ";
print "<input type='text' name='name'>";
print "<input type='submit' name='submit' value='Submit'>";
print "</form>";

if (isset($_POST['submit']))
{
   $name = $_POST['name'];
   if (!empty($name))
   {
      $person = new DisplayPerson($name);
      $person->display();
   }
   else
   {
      echo "Please enter a name.";
   }
}
?>
```

OUTPUT:

Enter Name: Joana Submit

The name is: Joana

AIM:

10. Write a PHP program to demonstrate multilevel inheritance with class members inherited from another file.

PROGRAM CODE:

mlbase.php

```
<?php
class A
{
 public $a,$b,$c;

 function getab($a,$b)
{
 $this->a=$a;
 $this->b=$b;
}

 function calcabc()
{
 $this->c=$this->a+$this->b;
 echo "$this->c";
}

}

class B extends A
{
 public $x,$y,$z;

 function getxy($x,$y)
{
 $this->x=$x;
 $this->y=$y;
}

 function calcxyz()
{
 $this->z=$this->x*$this->y;
 echo "$this->z";
}

}

?>
```

mlmain.php

```
<form method='POST' action=' '>
Enter a: <input type='text' name='a'><br><br>
Enter b: <input type='text' name='b'><br><br>
Enter x: <input type='text' name='x'><br><br>
```

```
Enter y: <input type='text' name='y'><br><br>
Enter p: <input type='text' name='p'><br><br>
Enter q: <input type='text' name='q'><br><br><br><br>
<input type='submit' name='s' value='Submit'>
</form>

<?php
include "mlbase.php";

class C extends B
{
 public $p,$q,$r;

 function getpq($p,$q)
{
 $this->p=$p;
 $this->q=$q;
}

 function calcpqr()
{
 $this->r=$this->p%$this->q;
 echo "$this->r";
}

}


$obj=new C();

if(isset($_POST['s']))
{
$av=$_POST['a'];
$bv=$_POST['b'];
$obj->getab($av,$bv);

$xv=$_POST['x'];
$yv=$_POST['y'];
$obj->getxy($xv,$yv);

$pv=$_POST['p'];
$qv=$_POST['q'];
$obj->getpq($pv,$qv);

echo "<br><br><b>***Addition***</b><br><br>";
$obj->calcabc();

echo "<br><br><b>***Multiplication***</b><br><br>";
$obj->calcxyz();
```


```
echo "<br><br><b>***Modulus***</b><br><br>";
$obj->calcpqr();
}
?>
```

OUTPUT:

Enter a: 10

Enter b: 20

Enter x: 30

Enter y: 40

Enter p: 50

Enter q: 5

Submit

Addition

30

Multiplication

1200

Modulus

0

AIM:

11. Write a PHP program to demonstrate hierarchical inheritance with class members inherited from another file.

PROGRAM CODE:

hbase.php

```
<?php

class A
{
 public $x,$y,$z;
}

class B extends A
{

 function getxy($x,$y)
{
 $this->x=$x;
 $this->y=$y;
}

 function calcxyz()
{
 $this->z=$this->x*$this->y;
 echo "$this->z";
}

}

?>
```

hmain.php

```
<form method='POST' action=' '>
Enter x: <input type='text' name='x'><br><br>
Enter y: <input type='text' name='y'><br><br><br><br>
<input type='submit' name='s' value='Submit'>
</form>

<?php
include "hbase.php";

class C extends A
{

 function getxy($x,$y)
{
 $this->x=$x;
 $this->y=$y;
}
```

```
 function calcxyz()
{
 $this->z=$this->x%$this->y;
 echo "$this->z";
}

}


$obj1=new B();
$obj2=new C();

if(isset($_POST['s']))
{

$xv=$_POST['x'];
$yv=$_POST['y'];

$obj1->getxy($xv,$yv);
$obj2->getxy($xv,$yv);

echo "<br><br><b>***Multiplication***</b><br><br>";
$obj1->calcxyz();

echo "<br><br><b>***Modulus***</b><br><br>";
$obj2->calcxyz();
}
?>
```

OUTPUT:

Enter x: 20

Enter y: 234

Submit

Multiplication

4680

Modulus

20

AIM:

12. Write a PHP program to demonstrate interfaces.

PROGRAM CODE:

```
<?php

interface CalculatorInterface {
   public function add($num1, $num2);
   public function subtract($num1, $num2);
   public function multiply($num1, $num2);
   public function divide($num1, $num2);
}

class BasicCalculator implements CalculatorInterface
{
   public function add($num1, $num2)
   {
      return $num1 + $num2;
   }

   public function subtract($num1, $num2)
   {
      return $num1 - $num2;
   }

   public function multiply($num1, $num2)
   {
      return $num1 * $num2;
   }

   public function divide($num1, $num2)
   {
      if ($num2 != 0)
      {
         return $num1 / $num2;
      }
      else
      {
         return "Cannot divide by zero!";
      }
   }
}

session_start();
print "<form method='POST' action=">";
print "Enter numbers:";
print "<br><input type='text' name='t1'>";
print "<br><input type='text' name='t2'>";
print "<br><input type='submit' name='s1'>";
print "</form>";
```

```
if (isset($_POST['s1']))
{
   $calculator = new BasicCalculator();
   $num1 = $_POST['t1'];
   $num2 = $_POST['t2'];
   echo "Addition: " . $calculator->add($num1, $num2) . "<br>";
   echo "Subtraction: " . $calculator->subtract($num1, $num2) . "<br>";
   echo "Multiplication: " . $calculator->multiply($num1, $num2) . "<br>";
   echo "Division: " . $calculator->divide($num1, $num2) . "<br>";
}

?>
```

OUTPUT:

Enter numbers:
200
10
Submit

Addition: 210
Subtraction: 190
Multiplication: 2000
Division: 20

AIM:

13. Write a PHP program to demonstrate inheriting interfaces.

PROGRAM CODE:

```
<?php
interface EmployeeInterface
{
   public function getName();
   public function getSalary();
}

interface SalaryInterface
{
   public function increaseSalary();
}

class Employee implements EmployeeInterface,SalaryInterface
{
   protected $name;
   protected $salary;
   public function __construct($name, $salary)
   {
      $this->name = $name;
      $this->salary = $salary;
   }
   public function getName()
   {
      return $this->name;
   }
   public function getSalary()
   {
      return $this->salary;
   }
   public function increaseSalary()
   {
      if ($this->salary < 1000)
      {
         $this->salary *= 1.10;
      }
      elseif ($this->salary > 1000 && $this->salary <= 4000)
      {
         $this->salary *= 1.20;
      }
      elseif ($this->salary > 4000 && $this->salary <= 10000)
      {
         $this->salary *= 1.30;
      }
      elseif ($this->salary > 10000)
      {
         $this->salary *= 1.40;
```

```
        }
    }
}

function displayEmployeeDetails(Employee $employee)
{
    echo "Name: " . $employee->getName() . "<br>";
    echo "Original Salary: $" . $employee->getSalary() . "<br>";
    $employee->increaseSalary();
    echo "New Salary: $" . $employee->getSalary() . "<br><br>";
}


session_start();
print "<form method='POST' action=">";
print "Enter Deatils:<br>";
print "Name:      <input type='text' name='name'><br>";
print "Salary:    <input type='text' name='sal'><br>";
print "<input type='submit' name='s1'>";
print "</form>";

if(isset($_POST['s1']))
{
  $name=$_POST['name'];
  $sal=$_POST['sal'];
  $employee1 = new Employee($name,$sal);
  echo "<h3>Employee Details</h3>";
  displayEmployeeDetails($employee1);

}
?>
```

OUTPUT:

Enter Deatils:
Name: Joana
Salary: 9000
Submit

Employee Details

Name: Joana
Original Salary: $9000
New Salary: $11700

AIM:

14. Write a javascript program to find the largest and the second largest element in an array.

PROGRAM CODE:

```
<!DOCTYPE html>
<html>
<head>
  <title>Find Largest and Second Largest Elements</title>
</head>
<body>
  <form id="inputForm">
    <label for="arrayInput">Enter array elements (comma-separated):</label><br>
    <input type="text" id="arrayInput" name="arrayInput" required><br><br>
    <button type="submit">Find Largest and Second Largest</button>
  </form>
  <div id="resultContainer">
    <h3 id="largestResult"></h3>
    <h3 id="secondLargestResult"></h3>
  </div>
  <script>
    document.getElementById('inputForm').addEventListener('submit', function (event) {
      event.preventDefault();

      const arrayInput = document.getElementById('arrayInput').value;
      const array = arrayInput.split(',').map(Number);
      if (array.length < 2) {
        alert('Please enter at least two numbers separated by commas.');
        return;
      }
      const [largest, secondLargest] = findLargestAndSecondLargest(array);
      document.getElementById('largestResult').innerText = 'Largest Element: ' + largest;
      document.getElementById('secondLargestResult').innerText = 'Second Largest
Element: ' + secondLargest;
    });
    function findLargestAndSecondLargest(arr) {
      let largest = -Infinity;
      let secondLargest = -Infinity;

      for (let i = 0; i < arr.length; i++) {
        if (arr[i] > largest) {
          secondLargest = largest;
          largest = arr[i];
        } else if (arr[i] > secondLargest && arr[i] !== largest) {
          secondLargest = arr[i];
        }
      }
      return [largest, secondLargest];
    }
  </script>
</body></html>
```

OUTPUT:

Enter array elements (comma-separated):
1,2,3,4,5

Find Largest and Second Largest

Largest Element: 5

Second Largest Element: 4

AIM:

15. Write a javascript program to perform arithmetic operations.

PROGRAM CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <h1>Enter 2 numbers</h1>
   <input type="number" name="num1" id="num1" placeholder="Enter first Number">
   <br><br>
   <input type="number" name="num2" id="num2" placeholder="Enter second Number">
   <br>
   <input type="submit" value="Add" name="add" id="add" onclick="addop()">
   <input type="submit" value="Subtract" name="sub" id="sub" onclick="subop()">
   <input type="submit" value="Multiply" name="mul" id="mul" onclick="mulop()">
   <input type="submit" value="Division" name="divop" id="divop" onclick="divop()">
   <input type="submit" value="Modulos" name="modulo" id="modulo"
onclick="moduloop()">
   <h3 id="result"></h3>
   <script>
      function addop() {
         var n1 = parseFloat(document.getElementById('num1').value);
         var n2 = parseFloat(document.getElementById('num2').value);
         var res;
         res = n1 + n2;
         document.getElementById('result').textContent = res;
      }
      function subop() {
         var n1 = parseFloat(document.getElementById('num1').value);
         var n2 = parseFloat(document.getElementById('num2').value);
         var res;
         res = n1 - n2;
         document.getElementById('result').textContent = res;
      }
      function mulop() {
         var n1 = parseFloat(document.getElementById('num1').value);
         var n2 = parseFloat(document.getElementById('num2').value);
         var res;
         res = n1 * n2;
         document.getElementById('result').textContent = res;
      }
      function divop() {
         var n1 = parseFloat(document.getElementById('num1').value);
         var n2 = parseFloat(document.getElementById('num2').value);
```

```
            var res;
            res = n1 / n2;
            document.getElementById('result').textContent = res;
        }
        function moduloop() {
            var n1 = parseFloat(document.getElementById('num1').value);
            var n2 = parseFloat(document.getElementById('num2').value);
            var res;
            res = n1 % n2;
            document.getElementById('result').textContent = res;
        }
    </script>
</body>
</html>
```

OUTPUT:

Enter 2 numbers

2

3

Add | Subtract | Multiply | Division | Modulos

0.6666666666666666

AIM:

16. Write a javascript program to perform reverse of a number.

PROGRAM CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <h1>REVERSE OF A NUMBER</h1>
   <input type="number" name="num1" id="num1" placeholder="Enter a Number">
   <br><br>
   <input type="submit" value="Reverse" onclick="reverse()">
   <h3 id="result"></h3>
   <script>
      function reverse(){
         var num=parseInt(document.getElementById('num1').value);
         var rev=0;
         var n;
         while(num>0){
            n=num%10;
            rev=rev*10+n;
            num=Math.floor(num/10);
         }
         document.getElementById('result').textContent=rev;
      }
   </script>
</body>
</html>
```

OUTPUT:

REVERSE OF A NUMBER

1357

Reverse

7531

AIM:

17. Write a javascript program to perform Fibonacci series operations.

PROGRAM CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <h1>Fibonacci Series</h1>
   <input type="number" name="num1" id="num1" placeholder="Enter Limit">
   <br><br>
   <input type="submit" value="Submit" onclick="fibo_series()">
   <h2 id="result"></h2>
   <script>
   function fibo_series(){
      var num=parseInt(document.getElementById('num1').value);
      var n1=0;
      var n2=1;
      var n3;
      var fiboSeries = [n1, n2];

      for (var i = 2; i < num; i++) {
         n3 = n1 + n2;
         fiboSeries.push(n3);
         n1 = n2;
         n2 = n3;
      }

      document.getElementById('result').textContent=fiboSeries.slice(0,num).join(', ');
   }
</script>
</body>
</html>
```

OUTPUT:

Fibonacci Series

3

Submit

0, 1, 1

AIM:

18. Write a javascript program to demonstrate functions.

PROGRAM CODE:

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Functions Demo</title>
</head>
<body>
  <h1>String Manipulation</h1>
  <form id="stringForm">
    <input type="text" id="userString" placeholder="Enter a string" required>
    <br><br>
    <button type="button" onclick="processString()">Process String</button>
  </form><br>
  <div id="result"></div>
  <script>
    function processString() {
      var inputString = document.getElementById('userString').value;

      function reverseString(str) {
        return str.split('').reverse().join('');
      }
      function countVowels(str) {
        var vowelCount = 0;
        var vowels = 'aeiouAEIOU';
        for (var i = 0; i < str.length; i++) {
          if (vowels.indexOf(str[i]) !== -1) {
            vowelCount++;
          }
        }
        return vowelCount;
      }
      var reversedString = reverseString(inputString);
      var vowelCount = countVowels(inputString);

      var resultDiv = document.getElementById('result');
      resultDiv.innerHTML = `
        Reversed String: ${reversedString} <br>
        Vowel Count: ${vowelCount}
      `;
    }
  </script>
</body>
</html>
```

OUTPUT:

String Manipulation

Hello is this a good time

Process String

Reversed String: emit doog a siht si olleH
Vowel Count: 9

AIM:

19. Write a javascript program to demonstrate event handling.

PROGRAM CODE:

```
<!DOCTYPE html>
<html>
<head>
   <title>Event Handling Demo</title>
   <style>
      #hoverDiv {
         width: 200px;
         height: 200px;
         background-color: lightblue;
         text-align: center;
         line-height: 200px;
         margin-top: 20px;
      }
   </style>
</head>
<body>
   <h1>JavaScript Event Handling</h1>

   <button id="clickButton">Click Me</button>
   <p id="buttonMessage"></p>
   <input type="text" id="inputField" placeholder="Type something...">
   <p id="inputMessage"></p>
   <div id="hoverDiv">Hover over me!</div>
   <p id="hoverMessage"></p>

   <script>
      document.getElementById('clickButton').addEventListener('click', function() {
         document.getElementById('buttonMessage').innerText = 'Button was clicked!';
      });

      document.getElementById('inputField').addEventListener('input', function() {
         const inputValue = document.getElementById('inputField').value;
         document.getElementById('inputMessage').innerText = 'You typed: ' + inputValue;
      });

      document.getElementById('hoverDiv').addEventListener('mouseover', function() {
         document.getElementById('hoverMessage').innerText = 'Mouse is over the div!';
      });

      document.getElementById('hoverDiv').addEventListener('mouseout', function() {
         document.getElementById('hoverMessage').innerText = '';
      });
   </script>
</body>
</html>
```

OUTPUT:


JavaScript Event Handling
Click Me
Button was clicked!
asdw
You typed: asdw
Hover over me!
Mouse is over the div!

AIM:

20. Write a javascript program to perform form validation.

PROGRAM CODE:

```
<!DOCTYPE html>
<html>

<head>
    <title>Form Validation</title>
    <style>
        .error {
            color: red;
        }
    </style>
</head>

<body>
    <h1>Form Validation</h1>

    <form id="registrationForm" onsubmit="return validateForm()">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name">
        <span class="error" id="nameError"></span>
        <br><br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email">
        <span class="error" id="emailError"></span>
        <br><br>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password">
        <span class="error" id="passwordError"></span>
        <br><br>

        <label for="confirm_password">Confirm Password:</label>
        <input type="password" id="confirm_password" name="confirm_password">
        <span class="error" id="confirmPasswordError"></span>
        <br><br>

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

    <script>
        function validateForm() {
            const name = document.getElementById('name').value;
            const email = document.getElementById('email').value;
            const password = document.getElementById('password').value;
            const confirmPassword = document.getElementById('confirm_password').value;
```

```
        let valid = true;

        if (name === "") {
            document.getElementById('nameError').innerText = "Name is required";
            valid = false;
        } else {
            document.getElementById('nameError').innerText = "";
        }

        if (email === "") {
            document.getElementById('emailError').innerText = "Email is required";
            valid = false;
        } else if (!/\S+@\S+\.\S+/.test(email)) {
            document.getElementById('emailError').innerText = "Invalid email format";
            valid = false;
        } else {
            document.getElementById('emailError').innerText = "";
        }

        if (password === "") {
            document.getElementById('passwordError').innerText = "Password is required";
            valid = false;
        } else if (password.length < 8) {
            document.getElementById('passwordError').innerText = "Password must be at least
8 characters long";
            valid = false;
        } else {
            document.getElementById('passwordError').innerText = "";
        }

        if (confirmPassword === "") {
            document.getElementById('confirmPasswordError').innerText = "Confirm
Password is required";
            valid = false;
        } else if (password !== confirmPassword) {
            document.getElementById('confirmPasswordError').innerText = "Passwords do not
match";
            valid = false;
        } else {
            document.getElementById('confirmPasswordError').innerText = "";
        }

        return valid;
    }

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

OUTPUT:

Form Validation

Name: Name is required

Email: Email is required

Password: Password is required

Confirm Password: Confirm Password is required

Submit

Form Validation

Name: Steve

Email: abc@gmai Invalid email format

Password: ••••••• Password must be at least 8 characters long

Confirm Password: •••• Passwords do not match

Submit

AIM:

21. Write a javascript program to demonstrate ajax.

PROGRAM CODE:

```
<!DOCTYPE html>
<html>
<head>
   <title>AJAX Demo</title>
</head>
<body>
   <h1>Fetching Data with AJAX</h1>
   <button id="fetchButton">Fetch Data</button>
   <div id="dataContainer"></div>

   <script>
      document.getElementById('fetchButton').addEventListener('click', () => {
         const xhr = new XMLHttpRequest();
         xhr.open('GET', 'data.txt', true);

         xhr.onload = function () {
            if (this.status === 200) {
               document.getElementById('dataContainer').innerText = this.responseText;
            } else {
               console.error('Error fetching data');
            }
         };
         xhr.onerror = function () {
            console.error('Request error');
         };
         xhr.send();
      });
   </script>
</body>
</html>
```

data.txt

This is the data fetched from the server using AJAX.

OUTPUT:


Fetching Data with AJAX
Fetch Data
This is the data fetched from the server using AJAX.

AIM:

22. Write a PHP program to demonstrate Cookies.

PROGRAM CODE:

```
<?php

  if(isset($_POST['s']))
{
  $name=$_POST['t1'];
  if(strcmp($name,'Administrator')==0)
{
  setcookie("username", "$name", time()+30*24*60*60);

  $msg = "<b>Welcome ".$_COOKIE['username']."!!!<br>"."You have been
registered!!!</b>";
  echo $msg;
}
  else
{
  echo "Welcome Guest!";
}
}
?>

<form method='post' action=' '>
<br><br>Enter the user name: <input type='text' name='t1'><br><br>
<input type='submit' name='s'><br>
</form>
```

OUTPUT:

Welcome Administrator!!!
You have been registered!!!

Enter the user name: Administrator

Submit

AIM:

23. Write a PHP program to demonstrate sessions.

PROGRAM CODE:

```
<?php
  session_start();
  if(isset($_POST['s']))
{
  $name=$_POST['t1'];
  if(isset($_SESSION[$name] ) )
{
    $_SESSION[$name] += 1;
  }
  else
{
    $_SESSION[$name] = 1;
  }

  $msg = "<b>Welcome ".$name."!!!<br>"."You have visited this page
".$_SESSION[$name]." time(s)!!!</b>";
  echo $msg;
}
?>

<form method='post' action=' '>
<br><br>Enter the user name: <input type='text' name='t1'><br><br>
<input type='submit' name='s'><br>
</form>
```

OUTPUT:

Welcome student!!!
You have visited this page 1 time(s)!!!

Enter the user name:

Submit

Welcome student!!!
You have visited this page 2 time(s)!!!

Enter the user name:

Submit

AIM:

24. Write a PHP program to demonstrate file handling.

PROGRAM CODE:

```
<?php
session_start();
?>

<!DOCTYPE html>
<html>
<head>
   <title>File Operations</title>
</head>
<body>
   <form action="" method="post">
      <label for="filename">Filename:</label>
      <input type="text" name='file'>
      <label for="operation">Select Operation:</label>
      <select id="operation" name="operation">
         <option value="create">Create and Write</option>
         <option value="read">Read</option>
         <option value="append">Append</option>
         <option value="delete">Delete</option>
      </select><br><br>

      <label for="content">Content:</label><br>
      <textarea id="content" name="content" rows="4" cols="50"></textarea><br><br>

      <input type="submit" name='s' value="Submit">
   </form>
</body>
</html>

<?php

function createAndWrite($file_path, $content) {
   $file = fopen($file_path, 'w');
   if ($file) {
      fwrite($file, $content);
      fclose($file);
      echo "File created and data written successfully.<br>";
   } else {
      echo "Unable to open file for writing.<br>";
   }
}

function read($file_path) {
   if (file_exists($file_path)) {
      $file = fopen($file_path, 'r');
      if ($file) {
```

```
            $content = fread($file, filesize($file_path));
            fclose($file);
            echo "Data read from file:<br><pre>" . htmlspecialchars($content) . "</pre><br>";
        } else {
            echo "Unable to open file for reading.<br>";
        }
    } else {
        echo "File does not exist.<br>";
    }
}


function append($file_path, $content) {
    $file = fopen($file_path, 'a');
    if ($file) {
        fwrite($file, $content);
        fclose($file);
        echo "Data appended to file successfully.<br>";
    } else {
        echo "Unable to open file for appending.<br>";
    }
}

function delete($file_path) {
    if (file_exists($file_path)) {
        unlink($file_path);
        echo "File deleted successfully.<br>";
    } else {
        echo "File does not exist.<br>";
    }
}

try{
if (isset($_POST['s'])) {
    if(empty($_POST['file']))
        throw new Exception("Filename cannot be empty.");
    $file_path = $_POST['file'];
    $operation = $_POST['operation'];
    $content = isset($_POST['content']) ? $_POST['content'] : '';

    switch ($operation) {
        case 'create':
            createAndWrite($file_path, $content);
            break;
        case 'read':
            read($file_path);
            break;
        case 'append':
            append($file_path, $content);
            break;
```

```
        case 'delete':
            delete($file_path);
            break;
        default:
            echo "Invalid operation selected.<br>";
            break;
    }
}
}
catch(Exception $e){
    echo "Error: " . $e->getMessage() . "<br>";
}
?>
```

OUTPUT:

Filename: test Select Operation: Create and Write

Content:

```
This is a test file
```

Submit

Filename: Select Operation: Create and Write

Content:

Submit

Data read from file:

```
This is a test file
```

Filename: Select Operation: Create and Write

Content:

Submit

File deleted successfully.

AIM:

25. Write a PHP program to upload file (with displaying uploaded file in web page).

PROGRAM CODE:

fupload1.php

```
<html>
<body>
<?php
if(isset($_POST['Submit1']))
{
$filepath = "docs/".$_FILES["file"]["name"];

if(move_uploaded_file($_FILES["file"]["tmp_name"], $filepath))
{
echo "<br>The File ".$_FILES["file"]["name"]." was uploaded in the \"docs\"
folder...!!!<br>";
echo "<br><br>Contents of the file...<br><br>";
echo nl2br(file_get_contents($filepath));
}
else
{
echo "Error !!";
}
}
?>
</body>
</html>
```

filedisp.php

```
<html>
<head>
<title>PHP File Upload</title>
</head>
<body>

<form action="fupload1.php" method="post" enctype="multipart/form-data">
Select file :
<input type="file" name="file"><br/>
<input type="submit" value="Upload" name="Submit1"> <br/>
</form>
```

OUTPUT:

Select file : Choose File WELCOME.txt
Upload

The File WELCOME.txt was uploaded in the "docs" folder...!!!

Contents of the file...

WELCOME JOANA TO FIRST CLASS OF PHP!
Let's code.

AIM:

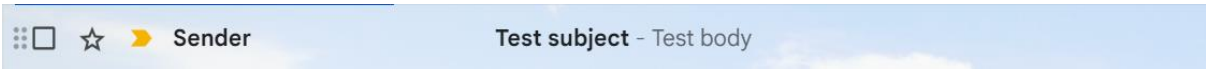
26. Develop a PHP program to send email.

PROGRAM CODE:

```
<?php
// PHPMailer classes into the global namespace
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
// Base files
require 'PHPMailer/src/Exception.php';
require 'PHPMailer/src/PHPMailer.php';
require 'PHPMailer/src/SMTP.php';
// create object of PHPMailer class with boolean parameter which sets/unsets exception.
$mail = new PHPMailer(true);
try {
    $mail->isSMTP(); // using SMTP protocol
    $mail->Host = 'smtp.gmail.com'; // SMTP host as gmail
    $mail->SMTPAuth = true;  // enable smtp authentication
    $mail->Username = 'stevedominicfez@gmail.com';  // sender gmail host
    $mail->Password = 'hobo raar tbal blzd'; // sender gmail host password
    $mail->SMTPSecure = 'tls';  // for encrypted connection
    $mail->Port = 587;   // port for SMTP

    $mail->setFrom('stevedominicfez@gmail.com', "Sender"); // sender's email and name
    $mail->addAddress('imca-355@scmsgroup.org', "Receiver");  // receiver's email and name

    $mail->Subject = 'Test subject';
    $mail->Body    = 'Test body';

    $mail->send();
    echo 'Message has been sent';
} catch (Exception $e) { // handle error.
    echo 'Message could not be sent. Mailer Error: ', $mail->ErrorInfo;
}
?>
```

OUTPUT:

`Message has been sent



AIM:

27. Design a registration form using PHP and validate it.

PROGRAM CODE:

```
<!DOCTYPE html>
<html>
<head>
   <title>Registration Form</title>
   <style>
      .error {color: #FF0000;}
   </style>
</head>
<body>

<?php
$nameErr = $emailErr = $passwordErr = $confirmPasswordErr = "";
$name = $email = $password = $confirm_password = "";

if (isset($_POST['s'])) {
   if (empty($_POST["name"])) {
      $nameErr = "Name is required";
   } else {
      $name = trim($_POST["name"]);
      if (!preg_match("/^[a-zA-Z-' ]*$/",$name)) {
         $nameErr = "Only letters and white space allowed";
      }
   }

   if (empty($_POST["email"])) {
      $emailErr = "Email is required";
   } else {
      $email = trim($_POST["email"]);
      if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
         $emailErr = "Invalid email format";
      }
   }

   if (empty($_POST["password"])) {
      $passwordErr = "Password is required";
   } else {
      $password = trim($_POST["password"]);
      if (strlen($password) < 8) {
         $passwordErr = "Password must be at least 8 characters long";
      }
   }

   if (empty($_POST["confirm_password"])) {
      $confirmPasswordErr = "Confirm Password is required";
```

```
    } else {
        $confirm_password = trim($_POST["confirm_password"]);
        if ($password !== $confirm_password) {
            $confirmPasswordErr = "Passwords do not match";
        }
    }

    if (empty($nameErr) && empty($emailErr) && empty($passwordErr) &&
empty($confirmPasswordErr)) {
        echo "<h3>Registration Successful!</h3>";
    }
}
?>

<h2>Registration Form</h2>
<form method="post" action="">
    Name: <input type="text" name="name">
    <span class="error">* <?php echo $nameErr;?></span>
    <br><br>
    Email: <input type="text" name="email">
    <span class="error">* <?php echo $emailErr;?></span>
    <br><br>
    Password: <input type="password" name="password">
    <span class="error">* <?php echo $passwordErr;?></span>
    <br><br>
    Confirm Password: <input type="password" name="confirm_password">
    <span class="error">* <?php echo $confirmPasswordErr;?></span>
    <br><br>
    <input type="submit" name="s" value="Register">
</form>

</body>
</html>
```

OUTPUT:

Registration Form

Name: * Name is required

Email: * Email is required

Password: * Password is required

Confirm Password: * Confirm Password is required

Register

Registration Form

Name: *

Email: * Invalid email format

Password: * Password must be at least 8 characters long

Confirm Password: * Passwords do not match

Register

AIM:

28. Design a student database application using PHP and MySQL and perform insertion, selection, deletion, updation and search operations in it.

PROGRAM CODE:

```
<html>
<body style='height:1000px;align-items: center'>
   <form method='POST' action=' '>
      <h1 style='margin-top: 25px'>Student details</h1>
      <input type='text' placeholder="Roll No" name='rollno'><br><br>
      <input type='text' placeholder="Name" name='sname'><br><br>
      <input type='number' placeholder="Marks" name='mark'><br><br>
      <input type='submit' name='insert' value='INSERT'>
      <input type='submit' name='select' value='SELECT'>
      <input type='submit' name='delete' value='DELETE'>
      <input type='submit' name='update' value='UPDATE'>
      <input type='submit' name='search' value='SEARCH'>
   </form>
</body>
</html>

<?php
if (isset($_POST['insert'])) {
   $rno = $_POST['rollno'];
   $nam = $_POST['sname'];
   $mark = $_POST['mark'];
   $con = mysqli_connect("localhost", "root", "", "student");
   if (!$con) {
      die("Connection failed: " . mysqli_connect_error());
   }
   $sql = "INSERT INTO student VALUES($rno,'$nam',$mark)";
   try {
      if (mysqli_query($con, $sql)) {
         echo "</br>Inserted Successfully";
      } else {
         echo " </br>Error in inserting";
      }
   } catch (Exception $e) {
      echo "</br>Enter the values in textbox to  insert";
   }
   mysqli_close($con);
}

if (isset($_POST['select'])) {
   $con = mysqli_connect("localhost", "root", "", "student");
   if (!$con) {
      die("Connection failed: " . mysqli_connect_error());
   }
   $sql = "SELECT RollNo,Name,Marks from student";
   try {
```

```
        $result = mysqli_query($con, $sql);
        if (mysqli_num_rows($result) > 0) {
            echo "</br>Selected Details Are<br> ";
            while ($row = mysqli_fetch_assoc($result))
                echo "</br>Rollno:" . $row["RollNo"] . "</br>Name:" . $row["Name"] .
"</br>Marks:" . $row["Marks"] . "<br>";
        } else {
            echo "</br>0 Results";
        }
    } catch (Exception $e) {
        echo "<br>Error in Selecting";
    }
    mysqli_close($con);
}

if (isset($_POST['delete'])) {
    $rno = $_POST['rollno'];

    $con = mysqli_connect("localhost", "root", "", "student");
    if (!$con) {
        die("Connection failed: " . mysqli_connect_error());
    }
    $sql = "DELETE from student where RollNo=$rno";
    try {
        mysqli_query($con, $sql);
        if (mysqli_affected_rows($con))
            echo "</br>Deleted Successfully";
        else
            echo "</br>Deletion Failed";
    } catch (Exception $e) {
        echo "</br>Enter the id of student to be deleted";
    }
    mysqli_close($con);
}

if (isset($_POST['update'])) {
    $rno = $_POST['rollno'];
    $nam = $_POST['sname'];
    $mark = $_POST['mark'];

    $con = mysqli_connect("localhost", "root", "", "student");
    if (!$con)
        die("Connectionfailed" . mysqli_connect_error());
    $sql = "update student set Name='$nam',Marks=$mark where RollNo=$rno";
    try {
        mysqli_query($con, $sql);
        if (mysqli_affected_rows($con))
            echo "</br>Updated Successfully";
        else
            echo "</br>Error in Updating";
```

```
    } catch (Exception $e) {
        echo "</br>Error in Updating";
    }
    mysqli_close($con);
}

if (isset($_POST['search'])) {
    $rno = $_POST['rollno'];
    $nam = $_POST['sname'];
    $mark = $_POST['mark'];
    $con = mysqli_connect("localhost", "root", "", "student");
    if (!$con)
        die("Connection failed" . mysqli_connect_error());
    $sql = "select RollNo,Name,Marks from student where RollNo=$rno";

    try {
        $result = mysqli_query($con, $sql);
        if (mysqli_num_rows($result) > 0) {
            while ($row = mysqli_fetch_assoc($result))
                echo "</br>Details of Roll No: " . $row["RollNo"] . " are  </br>Rollno:" .
$row["RollNo"] . "</br>Name:" . $row["Name"] . "</br>Marks:" . $row["Marks"];
        } else {
            echo "</br>0 Results";
        }
    } catch (Exception $e) {
        echo "<br>Enter id of student to be Searched in text box";
    }
    mysqli_close($con);
}
?>
```

OUTPUT:

Student details

Roll No

Name

Marks

INSERT SELECT DELETE UPDATE SEARCH

Inserted Successfully

Student details

Roll No

Name

Marks

INSERT SELECT DELETE UPDATE SEARCH

Selected Details Are

Rollno:1
Name:Brain
Marks:70

Rollno:2
Name:Ben
Marks:80

Student details

1

Brian

75

INSERT SELECT DELETE UPDATE SEARCH

Updated Successfully

Student details

Roll No

Name

Marks

INSERT SELECT DELETE UPDATE SEARCH

Details of Roll No: 2 are
Rollno:2
Name:Ben
Marks:80

AIM:

29. Design a login database application, which contains a registration page and a login page, using PHP and MySQL and perform insertion, selection, and search operations in it. If the login credentials inputted by the user in the login page has been found as a record while searching in the database table, print “Login details have been found”. Else, print “Login details have not been found”.

PROGRAM CODE:

resgister.php

```
<!DOCTYPE html>
<html>
<head>
   <title>User Registration</title>
</head>
<body>
   <h2>User Registration</h2>
   <form method="POST" action="register.php">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username" required><br><br>

      <label for="password">Password:</label>
      <input type="password" id="password" name="password" required><br><br>

      <input type="submit" name="register" value="Register">
   </form>
   <br>
   <p>Already registered? <a href="login.php">Login here</a></p>
   <?php
   if (isset($_POST['register'])) {
      $username = $_POST['username'];
      $password = $_POST['password'];
      $con = mysqli_connect("localhost", "root", "", "dbuser");
      if (!$con) {
         die("Connection failed: " . mysqli_connect_error());
      }
      $check_username = mysqli_real_escape_string($con, $username);
      $sql_check = "SELECT * FROM users WHERE username='$check_username'";
      $result_check = mysqli_query($con, $sql_check);

      if (mysqli_num_rows($result_check) > 0) {
         echo "<p style='color: red;'>Username already exists. Please choose a different
username.</p>";
      } else {
         $hashed_password = password_hash($password, PASSWORD_DEFAULT);
         $sql_insert = "INSERT INTO users (username, password) VALUES
('$check_username', '$hashed_password')";

         if (mysqli_query($con, $sql_insert)) {
            echo "<p>Registration successful!</p>";
```

```
            } else {
                echo "<p style='color: red;'>Error in registration.</p>";
            }
        }
        mysqli_close($con);
    }
    ?>
</body>
</html>
```

login.php

```
<!DOCTYPE html>
<html>
<head>
    <title>User Login</title>
</head>
<body>
    <h2>User Login</h2>
    <form method="POST" action="login.php">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required><br><br>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br><br>

        <input type="submit" name="login" value="Login">
    </form>
    <br>
    <p>Don't have an account? <a href="register.php">Register here</a></p>

    <?php
    if (isset($_POST['login'])) {
        $username = $_POST['username'];
        $password = $_POST['password'];
        $con = mysqli_connect("localhost", "root", "", "dbuser");
        if (!$con) {
            die("Connection failed: " . mysqli_connect_error());
        }
        $check_username = mysqli_real_escape_string($con, $username);
        $sql = "SELECT * FROM users WHERE username='$check_username'";
        $result = mysqli_query($con, $sql);

        if (mysqli_num_rows($result) > 0) {
            $row = mysqli_fetch_assoc($result);
            if (password_verify($password, $row['password'])) {
                session_start();
                $_SESSION['username'] = $username;
                echo "<p>Login successful! Welcome, $username</p>";
                echo "<p><a href='logout.php'>Logout</a></p>";
            } else {
```

```
            echo "<p style='color: red;'>Invalid username or password.</p>";
        }
    } else {
        echo "<p style='color: red;'>Invalid username or password.</p>";
    }
    mysqli_close($con);
}
?>
</body>
</html>
```

pgrm.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Student Details</title>
</head>
<body style='height:1000px;align-items: center'>
    <form method='POST' action=' '>
        <h1 style='margin-top: 25px'>Student details</h1>
        <input type='text' placeholder="Roll No" name='rollno'><br><br>
        <input type='text' placeholder="Name" name='sname'><br><br>
        <input type='number' placeholder="Marks" name='mark'><br><br>
        <input type='submit' name='insert' value='INSERT'>
        <input type='submit' name='select' value='SELECT'>
        <input type='submit' name='delete' value='DELETE'>
        <input type='submit' name='update' value='UPDATE'>
        <input type='submit' name='search' value='SEARCH'>
        <br><br>
        <input type='submit' name='logout' value='LOGOUT'>
    </form>
</body>
</html>

<?php
session_start();

if (isset($_POST['logout'])) {
    session_unset();
    session_destroy();
    header("Location: login.php");
    exit();
}

if (isset($_POST['insert'])) {
    $rno = $_POST['rollno'];
    $nam = $_POST['sname'];
    $mark = $_POST['mark'];
    $con = mysqli_connect("localhost", "root", "", "dbuser");
    if (!$con) {
```

```
        die("Connection failed: " . mysqli_connect_error());
    }
    $sql = "INSERT INTO student VALUES($rno,'$nam',$mark)";
    try {
        if (mysqli_query($con, $sql)) {
            echo "</br>Inserted Successfully";
        } else {
            echo " </br>Error in inserting";
        }
    } catch (Exception $e) {
        echo "</br>Enter the values in textbox to insert";
    }
    mysqli_close($con);
}

if (isset($_POST['select'])) {
    $con = mysqli_connect("localhost", "root", "", "dbuser");
    if (!$con) {
        die("Connection failed: " . mysqli_connect_error());
    }
    $sql = "SELECT RollNo,Name,Marks from student";
    try {
        $result = mysqli_query($con, $sql);
        if (mysqli_num_rows($result) > 0) {
            echo "</br>Selected Details Are<br> ";
            while ($row = mysqli_fetch_assoc($result))
                echo "</br>Rollno:" . $row["RollNo"] . "</br>Name:" . $row["Name"] .
"</br>Marks:" . $row["Marks"] . "<br>";
        } else {
            echo "</br>0 Results";
        }
    } catch (Exception $e) {
        echo "<br>Error in Selecting";
    }
    mysqli_close($con);
}
if (isset($_POST['delete'])) {
    $rno = $_POST['rollno'];

    $con = mysqli_connect("localhost", "root", "", "dbuser");
    if (!$con) {
        die("Connection failed: " . mysqli_connect_error());
    }
    $sql = "DELETE from student where RollNo=$rno";
    try {
        mysqli_query($con, $sql);
        if (mysqli_affected_rows($con))
            echo "</br>Deleted Successfully";
        else
            echo "</br>Deletion Failed";
```

```
    } catch (Exception $e) {
        echo "</br>Enter the id of student to be deleted";
    }
    mysqli_close($con);
}

if (isset($_POST['update'])) {
    $rno = $_POST['rollno'];
    $nam = $_POST['sname'];
    $mark = $_POST['mark'];

    $con = mysqli_connect("localhost", "root", "", "dbuser");
    if (!$con)
        die("Connection failed" . mysqli_connect_error());
    $sql = "update student set Name='$nam',Marks=$mark where RollNo=$rno";
    try {
        mysqli_query($con, $sql);
        if (mysqli_affected_rows($con))
            echo "</br>Updated Successfully";
        else
            echo "</br>Error in Updating";
    } catch (Exception $e) {
        echo "</br>Error in Updating";
    }
    mysqli_close($con);
}

if (isset($_POST['search'])) {
    $rno = $_POST['rollno'];
    $nam = $_POST['sname'];
    $mark = $_POST['mark'];
    $con = mysqli_connect("localhost", "root", "", "dbuser");
    if (!$con)
        die("Connection failed" . mysqli_connect_error());
    $sql = "select RollNo,Name,Marks from student where RollNo=$rno";

    try {
        $result = mysqli_query($con, $sql);
        if (mysqli_num_rows($result) > 0) {
            while ($row = mysqli_fetch_assoc($result))
                echo "</br>Details of Roll No: " . $row["RollNo"] . " are  </br>Rollno:" .
$row["RollNo"] . "</br>Name:" . $row["Name"] . "</br>Marks:" . $row["Marks"];
        } else {
            echo "</br>0 Results";
        }
    } catch (Exception $e) {
        echo "<br>Enter id of student to be Searched in text box";
    }
    mysqli_close($con);
}?>
```

OUTPUT:

User Registration

Username:

Password:

Register

Already registered? Login here

User Login

Username: test

Password: •••

Login

Don't have an account? Register here

Student details

Roll No

Name

Marks

INSERT SELECT DELETE UPDATE SEARCH

LOGOUT

Inserted Successfully

Student details

2

Priya

90

INSERT SELECT DELETE UPDATE SEARCH

LOGOUT

Updated Successfully

Student details

Roll No

Name

Marks

INSERT SELECT DELETE UPDATE SEARCH

LOGOUT

Details of Roll No: 1 are
Rollno:1
Name:Arun
Marks:89

AIM:

30. Develop a PHP project using Laravel framework.

PROGRAM CODE:

terminal

composer create-project --prefer-dist laravel/laravel inventory-management

.env

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE= inventory_management
DB_USERNAME=root
DB_PASSWORD=
```

Terminal

php artisan make:model Inventory -m

database\migrations\2024_07_14_093506_create_inventories_table.php

```
public function up()
{
    Schema::create('inventories', function (Blueprint $table) {
        $table->id();
        $table->string('product_name');
        $table->integer('quantity');
        $table->date('date');
        $table->timestamps();
    });
}
```

Terminal

php artisan migrate
php artisan make:controller InventoryController

routes\web.php

```
use App\Http\Controllers\InventoryController;

Route::resource('inventories', InventoryController::class);
```

app/Http/Controllers/InventoryController.php

```
namespace App\Http\Controllers;
use App\Models\Inventory;
use Illuminate\Http\Request;
class InventoryController extends Controller
{
    public function index()
    {
        $inventories = Inventory::all();
        return view('inventory.index', compact('inventories'));
    }
public function create()
    {
        return view('inventory.create');
    }
    public function store(Request $request)
    {
        $request->validate([
            'product_name' => 'required',
            'quantity' => 'required|integer',
            'date' => 'required|date',
        ]);
        Inventory::create($request->all());
        return redirect()->route('inventories.index')->with('success', 'Inventory created
successfully.');
    }
    public function edit(Inventory $inventory)
    {
        return view('inventory.edit', compact('inventory'));
    }
    public function update(Request $request, Inventory $inventory)
    {
        $request->validate([
            'product_name' => 'required',
            'quantity' => 'required|integer',
            'date' => 'required|date',
        ]);
        $inventory->update($request->all());
        return redirect()->route('inventories.index')->with('success', 'Inventory updated
successfully.');
    }
    public function destroy(Inventory $inventory)
    {
        $inventory->delete();
        return redirect()->route('inventories.index')->with('success', 'Inventory deleted
successfully.');}}
```

resources\views\inventory\index.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Inventory Management</title>
</head>
<body>
    <h1>Inventory List</h1>
    <a href="{{ route('inventories.create') }}">Add New Inventory</a>
    @if ($message = Session::get('success'))
        <p>{{ $message }}</p>
    @endif
    <table border="1">
        <tr>
            <th>ID</th>
            <th>Product Name</th>
            <th>Quantity</th>
            <th>Date</th>
            <th>Actions</th>
        </tr>
        @foreach ($inventories as $inventory)
        <tr>
            <td>{{ $inventory->id }}</td>
            <td>{{ $inventory->product_name }}</td>
            <td>{{ $inventory->quantity }}</td>
            <td>{{ $inventory->date }}</td>
            <td>
                <a href="{{ route('inventories.edit', $inventory->id) }}">Edit</a>
                <form action="{{ route('inventories.destroy', $inventory->id) }}" method="POST"
style="display:inline-block;">
                    @csrf
                    @method('DELETE')
                    <button type="submit">Delete</button>
                </form>
            </td>
        </tr>
        @endforeach
    </table>
</body>
</html>
```

resources\views\inventory\create.blade.php

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>Add New Inventory</title>
</head>
<body>
    <h1>Add New Inventory</h1>
    @if ($errors->any())
        <div>
            <strong>Whoops!</strong> There were some problems with your input.<br><br>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif
    <form action="{{ route('inventories.store') }}" method="POST">
        @csrf
        <div>
            <label>Product Name:</label>
            <input type="text" name="product_name" required>
        </div>
        <div>
            <label>Quantity:</label>
            <input type="number" name="quantity" required>
        </div>
        <div>
            <label>Date:</label>
            <input type="date" name="date" required>
        </div>
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

resources\views\inventory\edit.blade.php

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>Edit Inventory</title>
</head>
<body>
    <h1>Edit Inventory</h1>
    @if ($errors->any())
        <div>
            <strong>Whoops!</strong> There were some problems with your input.<br><br>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif
    <form action="{{ route('inventories.update', $inventory->id) }}" method="POST">
        @csrf
        @method('PUT')
        <div>
            <label>Product Name:</label>
            <input type="text" name="product_name" value="{{ $inventory->product_name }}"
required>
        </div>
        <div>
            <label>Quantity:</label>
            <input type="number" name="quantity" value="{{ $inventory->quantity }}"
required>
        </div>
        <div>
            <label>Date:</label>
            <input type="date" name="date" value="{{ $inventory->date }}" required>
        </div>
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

terminal

```
php artisan serve
```

OUTPUT:

Inventory Management System

Add New Inventory

ID	Product Name	Quantity	Date	Actions
4	smartphone	50	2024-07-14	Edit Delete
5	TV	30	2024-07-10	Edit Delete

Add New Inventory

Product Name:

smartphone

Quantity:

50

Date:

14 / 07 / 2024

Submit

Edit Inventory

Product Name:

smartphone

Quantity:

29

Date:

14 / 07 / 2024

Submit

Inventory Management System

Add New Inventory

Inventory deleted successfully.

ID	Product Name	Quantity	Date	Actions
4	smartphone	30	2024-07-14	Edit Delete