WEB TECHNOLOGY LABORATORY MANUAL

15CSL77

Prof. Ganesh Pai
P. A. College of Engineering, Mangaluru

2018

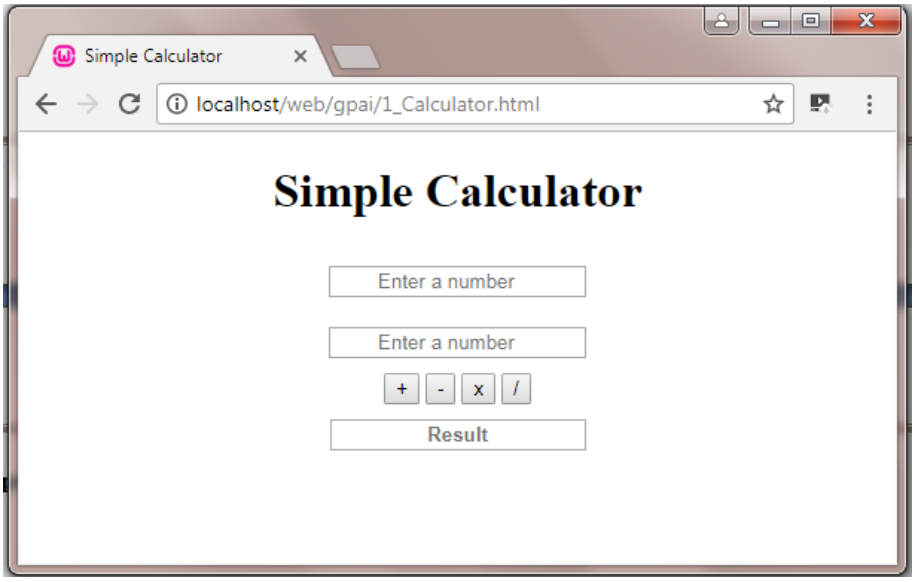
Table of Contents

1. Write a JavaScript to design a simple calculator to perform the following operations: *sum, product, difference and quotient.*

Filename: 1_Calculator.html

```
<!DOCTYPE html>
<html>
  <head>
        <title>Simple Calculator</title>
        <style>
              * {    text-align: center;       }
              input#result {      font-weight: bold;        }
              input        {      margin:10px;              }
        </style>
  </head>
  <body>
        <h1>Simple Calculator</h1>
        <input type="number" id="a" placeholder="Enter a number" autofocus/><br>
        <input type="number" id="b" placeholder="Enter a number"/><br>

        <button onclick="result.value = parseFloat(a.value) + parseFloat(b.value)">
         +</button>
        <button onclick="result.value = a.value - b.value">-</button>
        <button onclick="result.value = a.value * b.value">x</button>
        <button onclick="result.value = a.value / b.value">/</button><br>

        <input type="text" id="result" placeholder="Result" readonly/><br>
  </body>
</html>
```

Output:

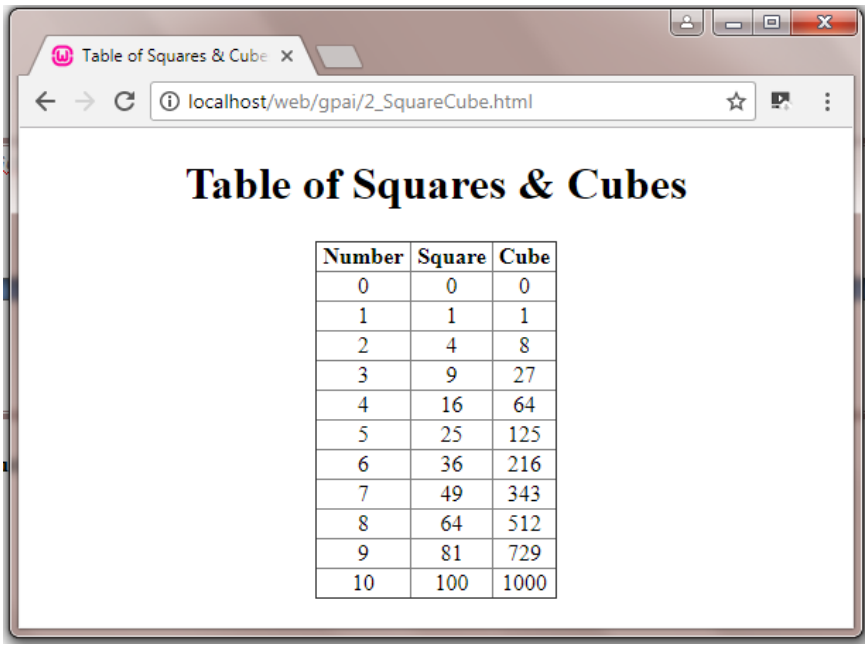


2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.

Filename: 2_SquareCube.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Table of Squares & Cubes</title>
        <style type="text/css">
            * {   text-align: center; }
            table {
                  margin : auto;      border-collapse : collapse;
                  width  : 30%;
               }
        </style>
    </head>
    <body>
         <h1>Table of Squares & Cubes</h1>
         <table border='1'>
             <tr> <th>Number</th> <th>Square</th> <th>Cube</th> </tr>
             <script>
                for(var i = 0; i <= 10; i++)
                    document.write("<tr><td>" + i + "</td><td>" + (i*i) +
                                   "</td><td>" + (i*i*i) + "</td></tr>");
            </script>
        </table>
     </body>
</html>
```

Output:



3. Write a JavaScript code that displays text “TEXT-GROWING” with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays “TEXT-SHRINKING” in BLUE color. Then the font size decreases to 5pt.

Filename: 3_GrowingText.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Growing Shrinking Text</title>
        <script>
             var size = 12, growing = true;

             function controlText()
             {
                 if(growing && size < 50)
                     message.style.fontSize = ++size + "pt";
                 else if(!growing && size > 5)
                     message.style.fontSize = --size + "pt";
                 else
                     clearInterval();

                 if(size == 50)
                 {
                     message.style.color = "blue";
                     message.innerHTML = "TEXT-SHRINKING";
                     growing = false;
                  }
             }
        </script>
        <style>
            #message {
                color:red;
                text-align:center;
                padding-top:150px;
            }
        </style>
    </head>

    <body onload = "setInterval(controlText, 100);">
        <div id="message">TEXT-GROWING</div>
    </body>
</html>
```

Output:


Growing Shrinking Text
localhost/web/gpai/3_GrowingText.html
TEXT-GROWING

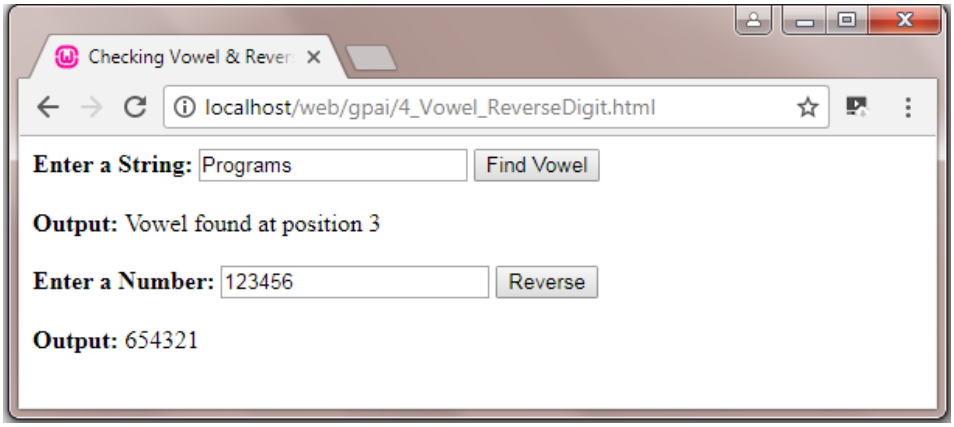

Growing Shrinking Text
localhost/web/gpai/3_GrowingText.html
TEXT-SHRINKING

4. Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems:
   a) Parameter: A string
      Output : The position in the string of the left-most vowel
   b) Parameter: A number
      Output : The number with its digits in the reverse order

Filename: 4_Vowel_ReverseDigit.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Checking Vowel & Reversing Number</title>
        <script>
            function findvowel(string)
            {
                var index = string.search(/[aeiou]/i);
                strOutput.innerText = (index >= 0)? ("Vowel found at position " +
                                                     (index + 1)) : "No vowel found.";
            }

            function reverse(n)
            {
                var reverseNo = 0;
                for(; n != 0; n = Math.floor(n / 10))
                    reverseNo = reverseNo * 10 + n % 10;
                numOutput.innerText = reverseNo;
            }
        </script>
        <style type="text/css">
            label    {    font-weight: bold;    }
            div[id]  {    display: inline;      }
        </style>
    </head>
    <body>
        <div>
           <label for="str">Enter a String:</label> <input type="text" id="str"  autofocus>
            <button onclick = "findvowel(str.value);">Find Vowel</button>
            <br><br>
            <label>Output:</label> <div id="strOutput"></div>
        </div>
        <br>
        <div>
            <label for="number">Enter a Number:</label> <input type="number" id="number">
            <button onclick = "reverse(number.value);">Reverse</button>
            <br><br>
            <label>Output:</label> <div id="numOutput"></div>
        </div>
    </body>
</html>
```

Output:


Checking Vowel & Rever
localhost/web/gpai/4_Vowel_ReverseDigit.html
Enter a String: Programs
Find Vowel
Output: Vowel found at position 3
Enter a Number: 123456
Reverse
Output: 654321

5. Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the College, Branch, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.

Filename: 5_StudentInfo.xml

```
<?xml version = "1.0" ?>
<?xml-stylesheet type = "text/css" href = "5_StudentInfoX.css" ?>

<students>
   <student>
       <usn>4PA15CS001</usn>
       <name>Suresh</name>
       <branch>CSE</branch>
       <college>PACE</college>
       <yoj>2015</yoj>
       <email>suresh@pace.edu.in</email>
   </student>
   <student>
       <usn>4PA15CS002</usn>
       <name>James</name>
       <branch>CSE</branch>
       <college>PACE</college>
       <yoj>2015</yoj>
       <email>james@pace.edu.in</email>
   </student>
   <student>
       <usn>4PA15CS003</usn>
       <name>Donald</name>
       <branch>CSE</branch>
       <college>PACE</college>
       <yoj>2015</yoj>
       <email>donald@pace.edu.in</email>
   </student>
</students>
```

Filename: 5_StudentInfo1.css

```
/* CSS Document */
student  {
    display: block;            border-radius: 25px;
    width: 400px;              margin: 15px;
    border: #000 thin dashed;  padding: 10px;
    color: #0000CC;
    background: linear-gradient(red, yellow, orange);
}

name, yoj, email {
    display:block;       padding-left:20px;
}
```

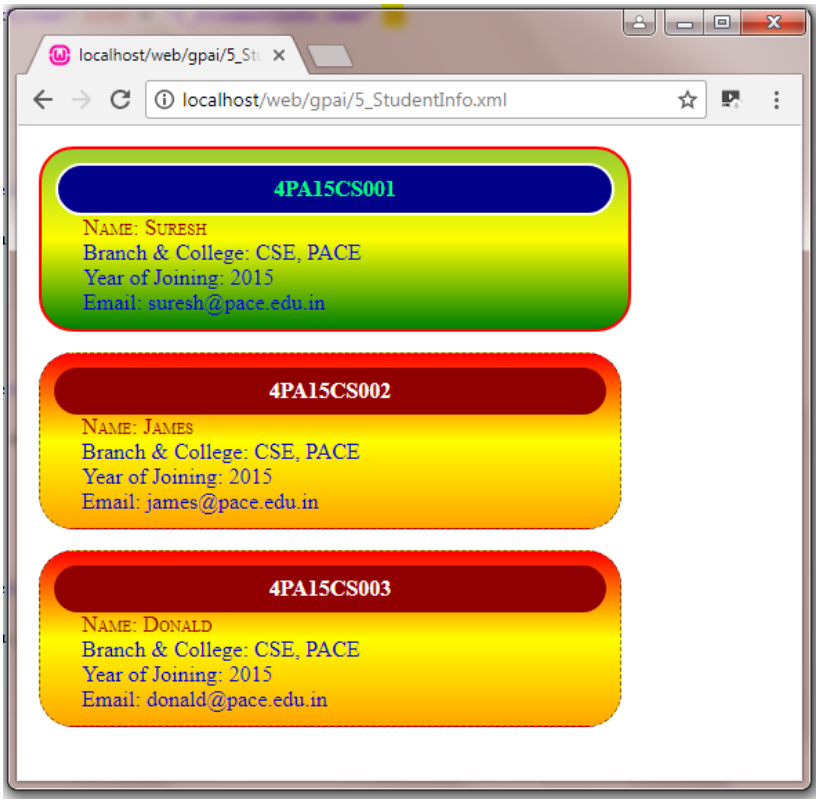
```
usn {
    display: block;         border-radius: 25px;
    width: auto;            padding: 8px;
    background: #900000;    color: #FFFFFF;
    text-align: center;     font: bold 12pt Times;
}
name {
    color:#990000;       font-variant:small-caps;
}
branch {
    padding-left:20px;
}

name::before   { content: "Name: "             }
branch::before { content: "Branch & College: " }
branch::after  { content: ", "                 }
yoj::before    { content: "Year of Joining: "  }
email::before  { content: "Email: "            }

student:hover{
    background: linear-gradient(yellowgreen, yellow, green);
    width: 405px;      border: 2px solid red;
}

student:hover usn{
    background: #000088;    color: SpringGreen;
    border: 2px solid white;
}
```

OR

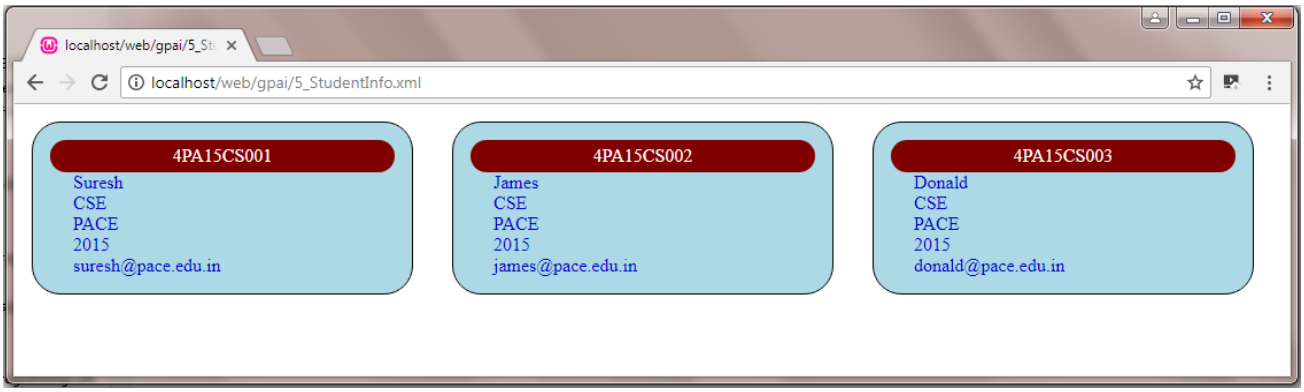
Filename: 5_StudentInfo2.css

```
/* CSS Document */
student  {
     display: inline-block;     border-radius: 25px;
     width: 300px;              margin: 15px;
     border: thin solid black;  padding: 15px;
     color: #0000CC;            background: lightblue;
}
name, branch, college, yoj, email {
    display:block;       padding-left:20px;
}
usn {
    display: block;         border-radius: 25px;
    padding: 5px;           text-align: center;
    background: maroon;     color: white;
}
student:hover{
    background: pink;
}
```

Output:



With 5_StudentInfo1.css



With 5_StudentInfo2.css

5. Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the College, Branch, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.

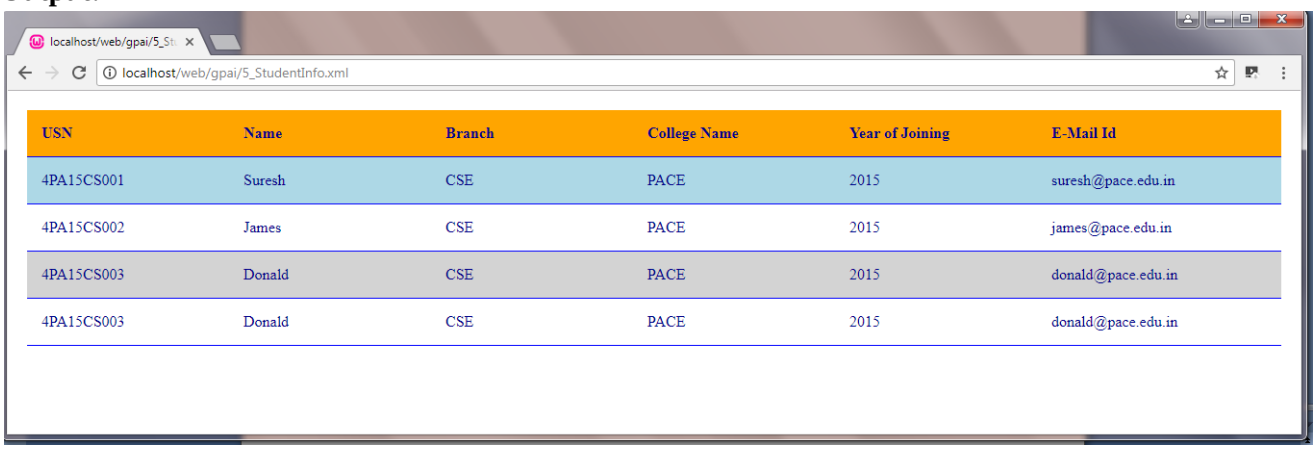
Filename: 5_StudentInfo.xml

```
<?xml version = "1.0" ?>
<?xml-stylesheet type = "text/css" href = "5_StudentInfo.css" ?>

<students>
   <columns>
         <col>USN</col>
         <col>Name</col>
         <col>Branch</col>
         <col>College Name</col>
         <col>Year of Joining</col>
         <col>E-Mail Id</col>
   </columns>
   <student>
         <usn>4PA15CS001</usn>
         <name>Suresh</name>
         <branch>CSE</branch>
         <college>PACE</college>
         <yoj>2015</yoj>
         <email>suresh@pace.edu.in</email>
   </student>
   <student>
         <usn>4PA15CS002</usn>
         <name>James</name>
         <branch>CSE</branch>
         <college>PACE</college>
         <yoj>2015</yoj>
         <email>james@pace.edu.in</email>
   </student>
   <student>
         <usn>4PA15CS003</usn>
         <name>Donald</name>
         <branch>CSE</branch>
         <college>PACE</college>
         <yoj>2015</yoj>
         <email>donald@pace.edu.in</email>
   </student>
   <student>
         <usn>4PA15CS003</usn>
         <name>Donald</name>
         <branch>CSE</branch>
         <college>PACE</college>
         <yoj>2015</yoj>
         <email>donald@pace.edu.in</email>
   </student>
</students>
```

Filename: 5_StudentInfo.css

```
/* CSS Document */
students {
	margin:20px;
}

student, columns {
	display: grid;
	grid-template-columns: repeat(auto-fit, 200px);
	grid-gap: 10px;
    border-bottom: 1px solid blue;
	color: darkblue;
	padding: 15px;
}

columns {
	background: orange;
	font-weight:bold;
}

student:nth-child(even) {
	background: lightgrey;
}

student:hover {
    background: lightblue;
}
```

Output:

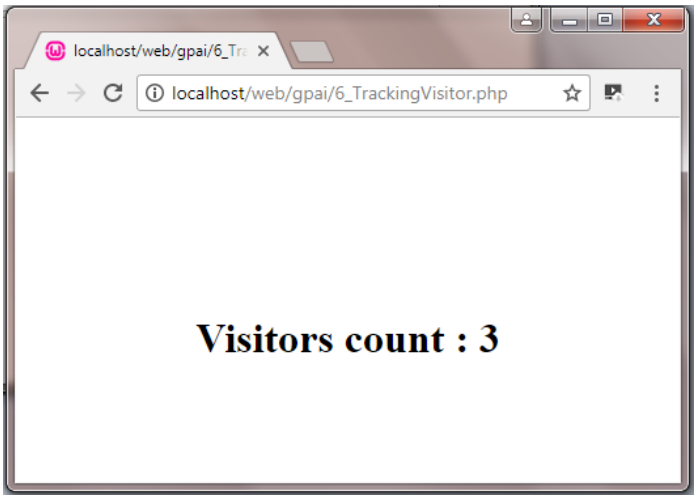


6. Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.

Filename: 6_TrackingVisitor.php

```
<?php
//Track webpage visitors count
session_start();

$filename = "counter.txt";
!file_exists($filename) && file_put_contents($filename, "0");

if(!isset($_SESSION['count']))
{
     $_SESSION['count'] = file_get_contents($filename) + 1;
     file_put_contents($filename, $_SESSION['count']);
}
else
     $_SESSION['count'] = file_get_contents($filename);
?>

<h1 style='text-align:center; margin-top:150px'>
     Visitors count : <?php echo $_SESSION['count'] ?>
</h1>
```

Output:

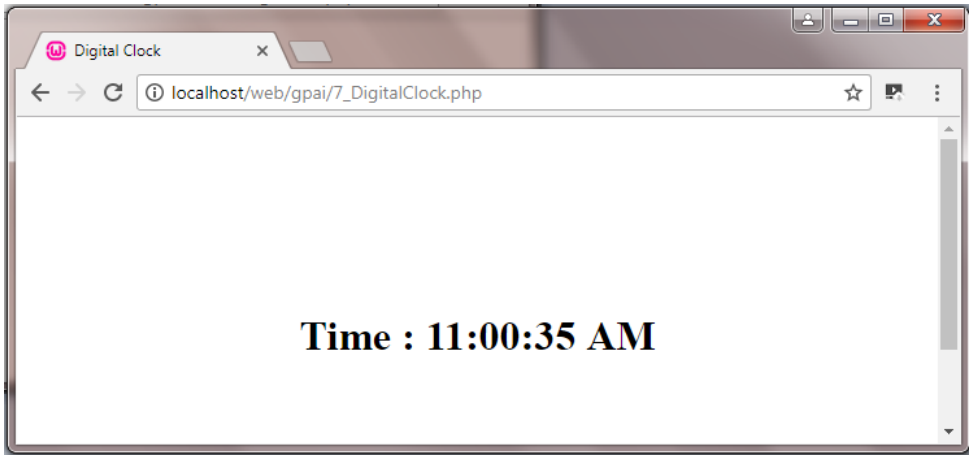


7. Write a PHP program to display a digital clock which displays the current time of the server.

Filename: 7_DigitalClock.php

```
<!-- Digital Clock -->

<?php
    date_default_timezone_set("Asia/Kolkata");
?>

<html>
     <head>
            <title>Digital Clock</title>
            <meta http-equiv="refresh" content="1">
     </head>
     <body>
            <h1 style='text-align:center; margin:150px'>
                  Time : <?php echo date("g:i:s A"); ?>
            </h1>
     </body>
</html>
```

Output:



8. Write the PHP programs to do the following:
    a) Implement simple calculator operations.
    b) Find the transpose of a matrix.
    c) Multiplication of two matrices.
    d) Addition of two matrices.

Filename: 8a_Calculator.php

```
<form method="post">
      <input type="number" step=any name="a" placeholder="Enter a number" required><br>
      <input type="number" step=any name="b" placeholder="Enter a number" required><br>
      <input type="submit" value="Calculate">
</form>
<?php
//Simple Calculator
if(isset($_POST['a']))
{
      $a = $_POST['a'];
      $b = $_POST['b'];
      echo   "<h3>$a + $b = " . ($a + $b) . "</h3>" .
             "<h3>$a - $b = " . ($a - $b) . "</h3>" .
             "<h3>$a * $b = " . ($a * $b) . "</h3>" .
             "<h3>$a / $b = " . ($b == 0? "Division by zero" : ($a / $b)) . "</h3>" .
             "<h3>$a % $b = " . ($b == 0? "Division by zero" : ($a % $b)) . "</h3>";
}
?>
```

Filename: 8b_Matrix.php

```
<?php
include "8b_matlib.php";

$a = array( array(10,20,30),
                  array(4,15,16),
                  array(17,12,9));
$b = array( array(11,12,31),
                  array(20,15,22),
                  array(17,8,6));

echo "<br>Matrix A<br>";
matrix_print($a);

echo "<br>Matrix B<br>";
matrix_print($b);

//Matrix Transpose
echo "<br>Matrix Transpose A<sup>T</sup><br>";
matrix_print(matrix_transpose($a));

//Matrix Multiplication
echo "<br>Matrix Product A x B<br>";
matrix_print(matrix_product($a, $b));
```

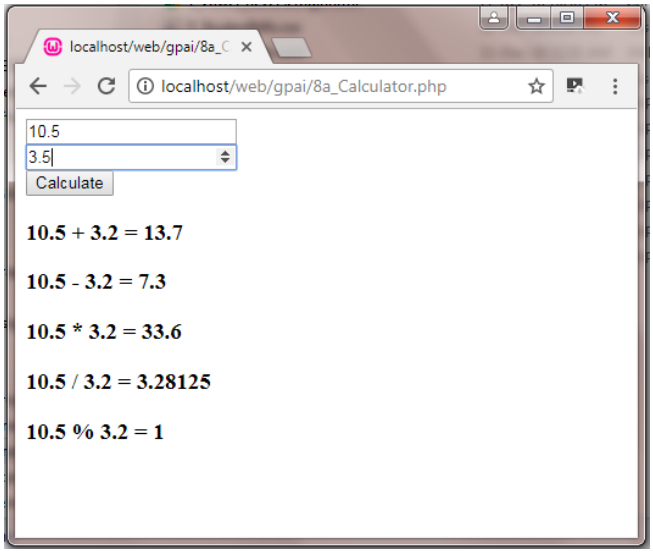
```
//Matrix Additon
echo "<br>Matrix Sum A + B<br>";
matrix_print(matrix_sum($a, $b));
?>
```

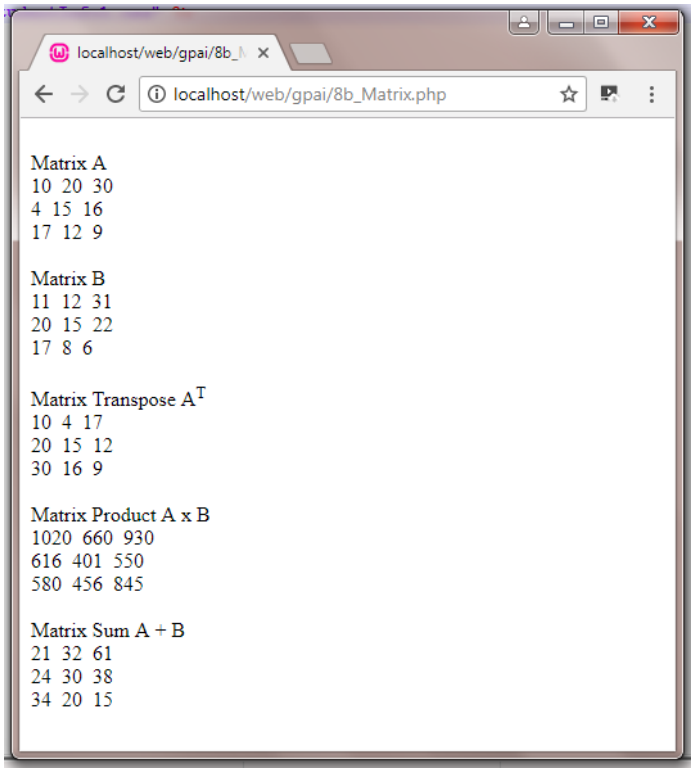
Filename: 8b_matlib.php

```
<?php
function matrix_print($matrix) //Print matrix in rows & column form
{
     foreach ($matrix as $array)
         echo implode(" nbsp;", $array) . "<br>";
}

function matrix_sum($a, $b)    //Addition of Matrix a & b
{
     $rows_a = count($a);      $cols_a = count($a[0]);
     $rows_b = count($b);      $cols_b = count($b[0]);
     if ($rows_a != $rows_b || $cols_a != $cols_b)
     {
         echo "Matric order mismatch.";
         return;
     }

     for($i = 0; $i < $rows_a; $i++)
         for($j = 0; $j < $cols_a; $j++)
             $c[$i][$j] = $a[$i][$j] + $b[$i][$j];
     return $c;
}

function matrix_product($a, $b)       //Multiply matrix a & b
{
     $rows_a = count($a);      $cols_a = count($a[0]);
     $rows_b = count($b);      $cols_b = count($b[0]);
     if ($cols_a != $rows_b)
     {
         echo "Matric order mismatch.";
         return;
     }

     for($i = 0; $i < $rows_a; $i++)
     {
         for($j = 0; $j < $cols_b; $j++)
         {
            $c[$i][$j] = 0;
            for($k = 0; $k < $cols_a; $k++)
            {
                $c[$i][$j] += $a[$i][$k] * $b[$k][$j];
            }
         }
     }
```

```
    return $c;
}

function matrix_transpose($a)  //Compute Matrix Transpose
{
    $rows = count($a);  $cols = count($a[0]);
    for($i = 0; $i < $rows; $i++)
        for($j = 0; $j < $cols; $j++)
            $b[$i][$j] = $a[$j][$i];
    return $b;
}
?>
```

Output:



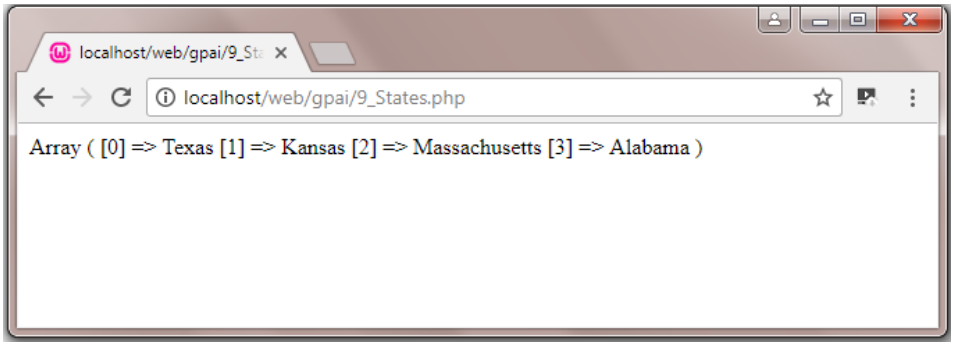


9. Write a PHP program named states.php that declares a variable ***states*** with value "Mississippi Alabama Texas Massachusetts Kansas". Write a PHP program that does the following:
   a) Search for a word in variable states that *ends in* ***xas***. Store this word in element 0 of a list named *statesList*.
   b) Search for a word in states that *begins with* ***k*** and *ends in* ***s***. Perform a case-insensitive comparison. Store this word in element 1 of *statesList*.
   c) Search for a word in states that *begins with* ***M*** and *ends in* ***s***. Store this word in element 2 of the list.
   d) Search for a word in states that *ends in* ***a***. Store this word in element 3 of the list.

Filename: 9_States.php

```
<?php
//String Matching
$states = "Mississippi Alabama Texas Massachusetts Kansas";
$pattern_list = array("/[a-zA-Z]*xas\b/", "/\bk[a-z]*s\b/i", "/\bM[a-z]*s\b/", "/[a-zA-Z]*a\b/");

$i=0;
foreach ($pattern_list as $pattern)
{
     if (preg_match($pattern, $states, $match))
           $statesList[$i] = $match[0];
     $i++;
}

print_r($statesList);
?>
```

Output:



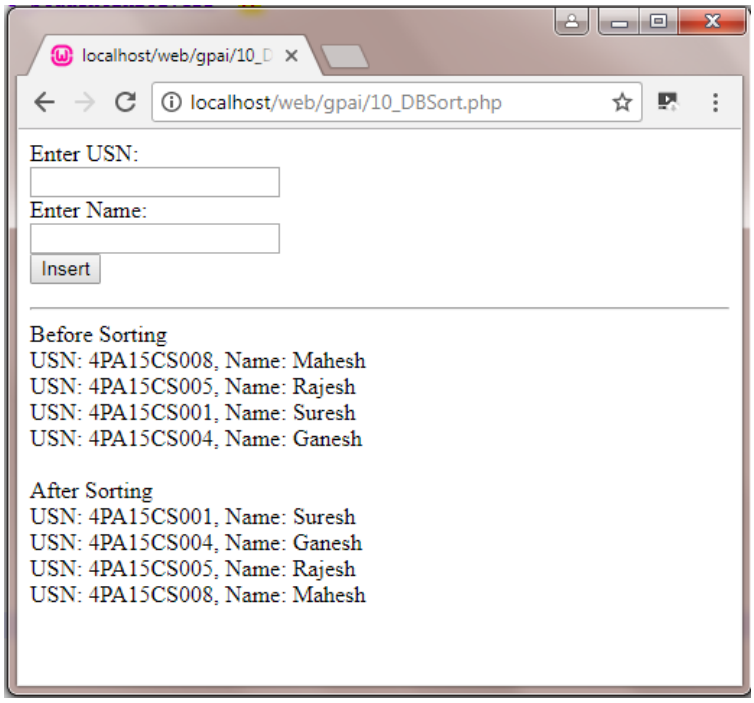
10. Write a PHP program to sort the student records which are stored in the database using selection sort.

Filename: 10_DBSort.php

```
<form method="post">
    Enter USN:<br>
    <input type="text" name="usn" required pattern="^4[pP][aA]1[0-9][a-zA-Z]{2}[0-9]{3}$"><br>
    Enter Name:<br>
    <input type="text" name="name" required pattern="[a-zA-Z][a-zA-Z ]+"><br>
    <input type="submit" value="Insert">
</form>
<hr>

<?php
$con = mysqli_connect("localhost", "root", "", "department");

if(isset($_POST['usn']))
{
    $usn = strtoupper($_POST['usn']);   $name = $_POST['name'];
    mysqli_query($con, "INSERT INTO student VALUES('$usn', '$name')");
}

$result = mysqli_query($con, "SELECT usn, name FROM student");
$rows   = mysqli_fetch_all($result, MYSQLI_ASSOC);
mysqli_close($con);

echo "Before Sorting<br>";
foreach ($rows as $student)
    printf("USN: %s, Name: %s<br>", $student['usn'], $student['name']);

$rowCount = mysqli_num_rows($result);
for($i = 0; $i < $rowCount; $i++)   //Selection Sort
{
    $min = $i;
    for($j = $i + 1; $j < $rowCount; $j++)
        if($rows[$j]["usn"] < $rows[$min]["usn"])
            $min = $j;
    $tmp        = $rows[$i];
    $rows[$i]   = $rows[$min];
    $rows[$min] = $tmp;
}

echo "<br>After Sorting<br>";
foreach ($rows as $student)
    printf("USN: %s, Name: %s<br>", $student['usn'], $student['name']);
?>
```

Note: *Database & Table need to be created manually*

Output:


localhost/web/gpai/10_D
localhost/web/gpai/10_DBSort.php
Enter USN:
Enter Name:
Insert
Before Sorting
USN: 4PA15CS008, Name: Mahesh
USN: 4PA15CS005, Name: Rajesh
USN: 4PA15CS001, Name: Suresh
USN: 4PA15CS004, Name: Ganesh
After Sorting
USN: 4PA15CS001, Name: Suresh
USN: 4PA15CS004, Name: Ganesh
USN: 4PA15CS005, Name: Rajesh
USN: 4PA15CS008, Name: Mahesh