Dharmit Shah
75 FYIT

Practical 1 : XML

a Design a DTD, corresponding XML document and display it in browser using CSS

Theory:- 1) Define the styles rules for the text element such as font-size, color, font-weight, etc.
2) Define each element either as a block, inline or list element, using the display property of CSS
3) Identify the titles and bold them

Code :-

filename: employee.dtd

```
<!ELEMENT employee(firstname,lastname,email)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname(#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

filename: employee.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="employeefile.css"?>
<!DOCTYPE employee SYSTEM "employee.dtd">
<employee>
<firstname>Dharmit</firstname>
<lastname> Shah </lastname>
<email> dharmit@gmail.com</email>
<employee>
```

filename : employeefile.css

```
employee{
        font-size : 25px;
        font-weight : bold;
        color : blue;
}
```

Output :

Dharmit Shah dharmit@gmail.com

b Design an XML document and display it in browser using XSL.

Code:-

Creating student.xml as:-

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<?xml-stylesheet type = "text/xsl" href = "rule.xsl" ?>
<student>
 <s>
  <name> Dharmit Shah </name>
  <branch> CSE </branch>
  <age> 18 </age>
  <city> mumbai </city>
 </s>
 <s>
  <name> Simran Agarwal </name>
  <branch> CSE </branch>
  <age> 23 </age>
  <city> Agra </city>
 </s>
</student>
```

Creating rule.xsl as:-

```
<th> Name </th>
<th> Branch </th>
<th> Age </th>
<th> City </th>
</tr>
<xsl: for-each select = "student's">
<tr>
```

```
<td> <xsl:value-of select="name" /> </td>
<td> <xsl:value-of select="branch" /> </td>
<td> <xsl:value-of select="age" /> </td>
<td> <xsl:value-of select="city" /> </td>
</tr>
</xsl:for-each
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Output

Students basic Details

Name	Branch	Age	City
Dhanmit Shah	CSE	18	Mumbai
Simran Agrawal	CSE	23	Agra

c Design XML schema and corresponding XML document

Theory:
A xml schema is used to define the structure of an XML document

<xs: element name = "employee"> : It defines the element name employee

<xs: complex type): It defines that the element 'employee' is complex type

<xs: sequence> : It defines that the complex type is a sequence of elements

<xs: element name> It defines that 'firstname' is of string type

Code:-
employee.xsd

```
<?xml version = "1.0" ?>
<xs: schema xmlns:xs = "http://www.w3.org/2001/xmlschema">
<xs: element name = "employee">
<xs: complex type>
<xs: sequence>
<xs: element name = "Firstname" type = "xs: string" />
<xs: element name = "Lastname" type: "string" />
<xs: element name = "email" type: "xs: string" />
</xs: sequence>
</xs: complex type>
</xs: element>
```

</xs: schema>

employee.xml

```
<?xml version = "1.0" ?>
<employee xmlns:xs: = "http://www.w3.org/2001/xmlschema"
 xsi:schema = "employee.xsd">

   <firstname> Dharmit </firstname>
   <lastname> Shah </lastname>
   <email> dharmitshah@gmail.com </email>
</employee>
```

Output :-

Dharmit
Shah
dharmitshah@gmail.com

2 Basic PHP - I

a) Write a PHP Program to accept a number from the user and print it factorial

Theory: The factorial of a number n is defined by the product of all digits from 1 to n (including 1 and n)

- It is denoted by n! and is calculated only for positive integers
- Factorial of 0 is always 1.

The simplest way to find the factorial of a number is by using a loop.

Code:

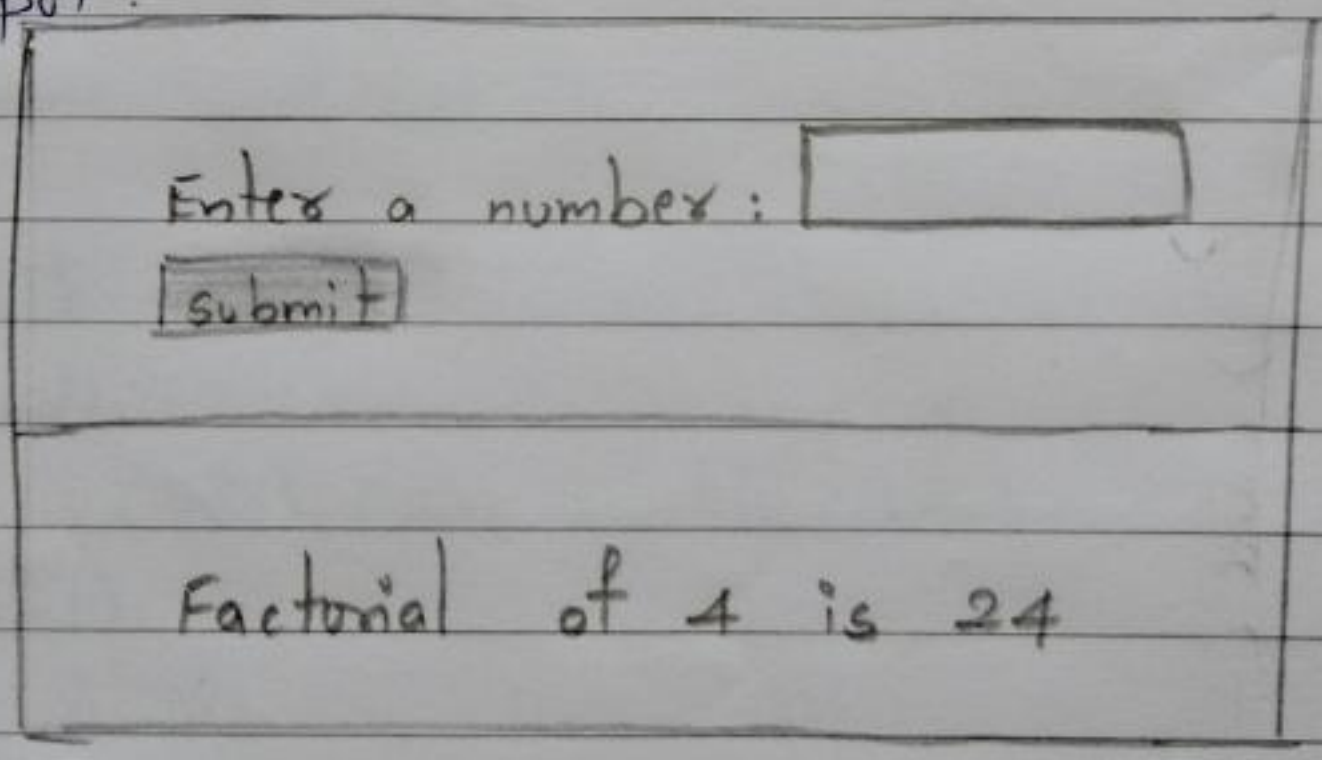
filename: input.html

```
<html>
<body>
    <form method="post" action="fact.php">
        Enter a number: <input type="text" name="n1"> <br>
        <input type="submit" value="Factorial">
    </form>
</body>
</html>
```

factfilename : fact.php

```
<?php
        $n1 = (int) $_POST['n1'];
        $fact = 1;
        for ($i=1; $i<=$n1 ; $i++)
        {
                $fact = $fact* $i;
        }
        echo "Factorial of ".$n1." is:".$fact;
?>
```

Output:



b Write a PHP program to accept number from the user and print whether it is prime or not.

Theory:- A number which is only divisible by 1 and itself is called prime number. Numbers 2, 3, 5, 7, 11, 13, 17, etc are prime numbers.

2 is the only even prime number

It is natural number greater than 1 and so 0 and 1 not prime number.

Code:-

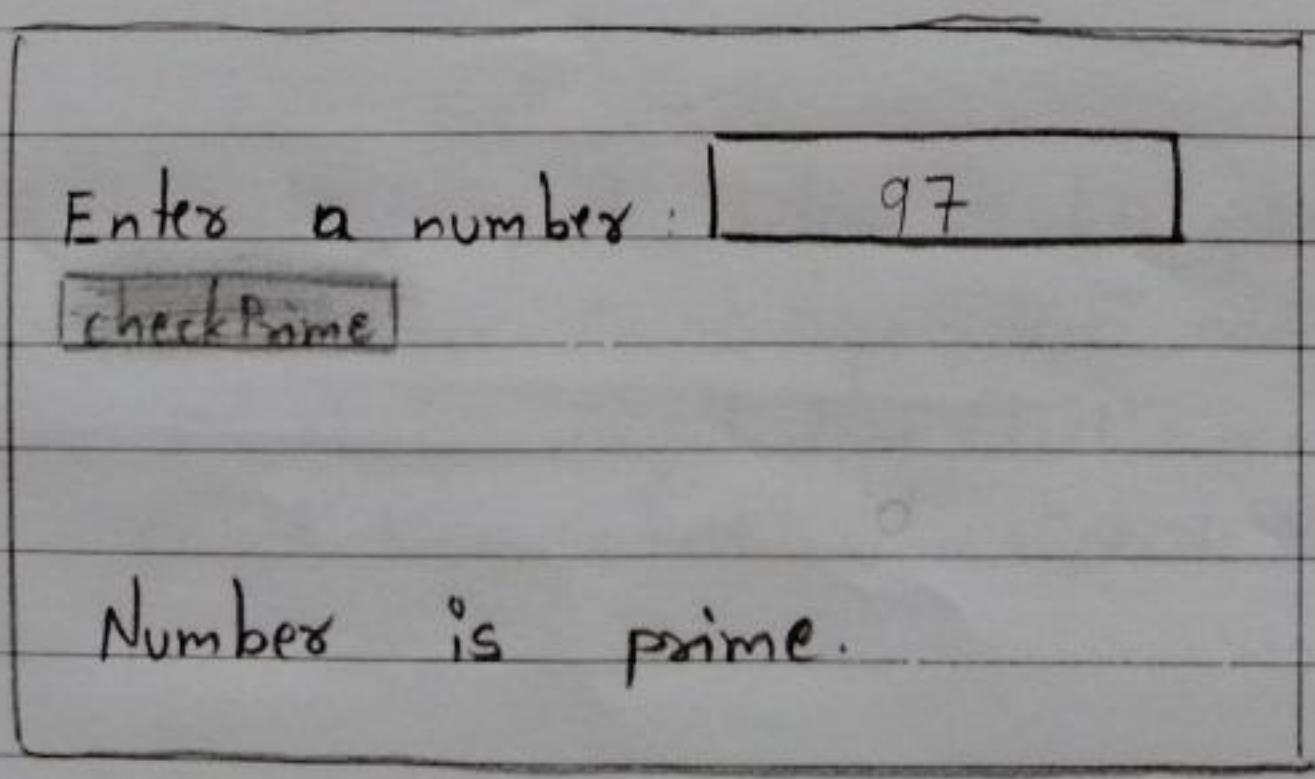
filename: Input.html

```
<html>
<head>
    <title> Prime number </title>
</head>
<body>
    <form method="post" action="checkPrime.php">
        Enter a number: <input type="text" name="n1"> <br>
        <input type="submit" value="check Prime">
    </form>
</body>
</html>
```

checkPrime.php

```
<?php
    $n1 = (int) $_POST['n1'];
$flag = 0;
for ($i = 2 ; $i <= $n1/2 ; $i++)
{
    if ($n1 % $i == 0)
    {
        $flag = 1;
        break;
    }
}
if($flag == 0)
    echo "Number is prime";
else
    echo "Number is not prime";
?>
```

Output:



Q3 PHP Basic - II

a Write a PHP code to find the greater of 2 numbers. Accept the no. from the user.

Theory: There are three methods to add two numbers.

- Adding in simple code in PHP
- Adding in form in PHP
- Adding without using arithemetic operator (+)

Code:-

Filename: input.html

```
<html>
<head>
    <title>Greater of two no.s </title>
</head>
<body>
    <form method="post action." check.php">
        1st Number: <input type."text" name."n1"><br>
        2nd Number: <input type="text" name = "n2"><br>
        <input type"="submit" value ="check">
    <form>
</body>
</html>
```

check.php

```
<?php
    $n1 = (int) $_POST['n1'];
    $n2 = (int) $_POST['n2'];
    if ($n1 > $n2)
        echo $n1." is greater than ".$n2;
    else if ($n2 > $n1)
        echo $n2." is greater than ".$n1;
    else
        echo "Both the nos are equal";
?>
```

Output:

1st Number : [ ]

2nd Number : [ ]

check

355 is greater than 332

b Write a PHP program to display the following Binary Pyramid

1

0 1

1 0 1

0 1 0 1

Theory: ,' POST = sends the form data as an http post transaction

2) echo - used to display the output

3) for - used to traverse of time.

4) If - else = It executes one block of code if the specified condition is true and another block of code if the condition is ~~true~~ false

Code:

```
<?php
    for($i=0; $i<4; $i++)
    {
        for($j=0; $j<=$i; $j++)
        {
            if(($i+$j)%2 ==0)
                echo "1";
            else
                echo "0";
        }
        echo "<br>";
    }
?>
```

Q4 String Functions and Array

a Write a PHP Program to demonstrate different string functions

Theory:-

1) A php script starts with <?php and ends with ?>
2) strlen(): Returns the length of a string
3) strchr() : finds the fast accurrence of a string inside another string.
4) str-replace() :- Replace some character in a string
5) str-word_count :- Count the number of words in the string
6) strpos() :- Return the position of the fast occurence of a string inside another
7) strlower() :- convert the string into lower upper letters
8) str_count - count the number of times a substring occurs in a string
9) strupper() :- convert the string into upper letters
10) substr() :- Returns a post of a string
11) strcasecmp() :- Compare two strings (case insensitive)

Code:-

```
<?php
    echo strlen("Hello");
    echo strchr("Hello world!","world");
    echo str_replace("world","Dhamit","Hello world!");
    echo str_word_count("Hello world!")
    echo strpos("I love php, I love php too!","php");
    echo substr_count("Hello World. The world is nice","world");
    echo substr("Hello world",6);
    echo strtolower("Hello WORLD");
    echo strtoupper("Hello WORLD.");
    echo strcmp("Hello world!","Hello world!");
    echo strcasecmp("Hello world!","Hello world!");
?>
```

Output:-

```
5
world!
Hello Dhamit!
2
7
world
Hello world
HELLO WORLD
0
0
```

b Write a PHP program to create one dimensional array.

Theory:-

1) A php script starts with <?php and ends with ?>
2) count() :- return the length of an array
3) indexed arrays: The index can be assigned automatically (index always starts at 0)
4) To loop through and print all the values of an indexed array we can use for loop.
5) Associative array - used to named keys that we assign to them.

Code:

```
<?php
    $cars = array("Volvo", "BMW", "Toyota");
    echo "I like". $cars[0] . ", " . $cars[1] . "and" . $cars[2] .
    "." ;
    echo count($cars);
    for ($x=0; $x < $arrlength ; $x++)
    {
      echo $cars[$x];
      echo "<br>";
    }
    // Associative arrays
    $age = array("Peter" => "35", "Ben" => "37", "Joe" => "43");
    echo "Peter is". $age['Peter'] . "years old.";
?>
```

PTO

Output :

I like Volvo , BMW and Toyota.
3
Volvo
BMW
Toyota
Peter is 35 years old.