Experiment No: 1 **Date: 14/12/2019**

SOCKET PROGRAMMING IN JAVA

PROGRAM 1:

Aim:

To implement simple ping pong application between two or more different machines.

Program:

```
import java.io.*;
import java.net.*;
class Ping{
    public static void sendPingRequest(String ipAddress)
throws UnknownHostException,IOException{
        InetAddress pingRequest =
InetAddress.getByName(ipAddress);
        System.out.println("Sending ping request to "+
ipAddress);
        if(pingRequest.isReachable(80)){
            System.out.println("Host Reachable");
        }
        else{
            System.out.println("Unable to connect to host");
        }
    }
    public static void main(String[] args) throws
UnknownHostException,IOException{
        String ipAddress = "127.0.0.1";
        sendPingRequest(ipAddress);
        ipAddress = "133.192.31.42";
        sendPingRequest(ipAddress);
        ipAddress = "145.154.42.58";
        sendPingRequest(ipAddress);
    }
}
```

Output:

```
Sending ping request to 127.0.0.1
Host Reachable
Sending ping request to 133.192.31.42
Unable to connect to host
Sending ping request to 145.154.42.58
Unable to connect to host
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 2:

Aim:
To implement date and time display from client to server using Sockets.

Program:

Server Program

```
import java.util.*;
import java.io.*;
import java.net.*;
import java.lang.*;
import java.time.LocalDate;
class Pg2Server{
    public static void main(String[] args) throws Exception{
        ServerSocket ss = new ServerSocket(3000);
        Socket soc = ss.accept();
        DataInputStream in = new
DataInputStream(soc.getInputStream());
        DataOutputStream out = new
DataOutputStream(soc.getOutputStream());
        System.out.println(in.readUTF());
        System.out.println("Request granted");
        out.writeUTF("Date : " + java.time.LocalDate.now() +"
Time : " + java.time.LocalTime.now());
        in.close();
        out.close();
        soc.close();
        ss.close();

    }
}
```

Client Program

```
import java.util.*;
import java.io.*;
import java.lang.*;
import java.net.*;
class Pg2Client{
    public static void main(String[] args) throws Exception{
        Socket soc = new
Socket(InetAddress.getLocalHost(),3000);
        DataOutputStream out = new
DataOutputStream(soc.getOutputStream());
        DataInputStream in = new
DataInputStream(soc.getInputStream());
        out.writeUTF("Request to Server");
```

```
        String readMessage = in.readUTF();
        System.out.println(readMessage);
        System.out.println("Recieved from Server");
        in.close();
        out.close();
        soc.close();

    }
}
```

Output:

```
Server
Request to Server
Request granted
Client
Date: 2019-02-24 Time: 01:41:55.323
Received from Server
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 3:

Aim:

To write a ping pong client and server application. When a client sends a ping message to the server, the server will respond with a pong message. Other messages sent by the client can be safely dropped by the client.

Program:

Server Program

```
import java.util.*;
import java.io.*;
import java.lang.*;
import java.net.*;
class Pg3Server{
    public static void main(String[] args) throws Exception{
        ServerSocket server = new ServerSocket(3000);
        Socket socket = server.accept();
        DataInputStream in = new
DataInputStream(socket.getInputStream());
        DataOutputStream out = new
DataOutputStream(socket.getOutputStream());
        while(true){
            String receiveData = in.readUTF();
            String sendData = "";
```

```
            if(receiveData.equals("PING") ){
                System.out.println(receiveData + " received");
                System.out.println("PONG sent.");
                sendData = "PONG";
                out.writeUTF(sendData);
            }
            if(receiveData.equals("EXIT")){
                break;
            }
            out.writeUTF(sendData);
        }
        in.close();
        out.close();
        socket.close();
        server.close();
    }
}
```

Client Program

```
import java.util.*;
import java.io.*;
import java.lang.*;
import java.net.*;
class Pg3Client{
    public static void main(String[] args) throws Exception{
        Socket socket = new Socket("127.0.0.1",3000);
        DataInputStream in = new
DataInputStream(socket.getInputStream());
        DataOutputStream out = new
DataOutputStream(socket.getOutputStream());
        Scanner sc = new Scanner(System.in);
        while(true){
            System.out.println("Enter the data\nType EXIT to
exit");
            String sendData = sc.next();
            out.writeUTF(sendData);
            if(sendData.equals("EXIT")){
                break;
            }
            String receiveData = in.readUTF();
            if(receiveData.length() != 0 ){
                System.out.println(receiveData);
            }
        }
        sc.close();
        in.close();
        out.close();
        socket.close();
    }
}
```

Output:

```
Server
PING received
PONG sent.
Client
Enter the data
Type EXIT to exit
PING
PONG
Enter the data
Type EXIT to exit
EXIT
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 4:

Aim:

To write a socket based Java server program that responds to client messages as follows: When it receives a message from client, it simply converts the message into all uppercase letters and sends back the same to the client.

Program:

Server Program

```
import java.util.*;
import java.io.*;
import java.lang.*;
import java.net.*;
class Pg4Server{
    public static void main(String[] args) throws Exception{
        DatagramSocket server = new DatagramSocket(3000);
        while(true){
            byte[] sendbyte=new byte[1024];
            byte[] receivebyte=new byte[1024];
            DatagramPacket receiver=new
DatagramPacket(receivebyte,receivebyte.length);
            server.receive(receiver);
            String data = new
String(receiver.getData()).trim();
            if(data.equals("Exit")){
                break;
            }
```

```
            data = data.toUpperCase();
            System.out.println(data);
            InetAddress addr=receiver.getAddress();
            int port=receiver.getPort();
            sendbyte = data.getBytes();
            DatagramPacket sender = new
DatagramPacket(sendbyte,sendbyte.length,addr,port);
            server.send(sender);
        }
    }
}
```

Client Program

```
import java.util.*;
import java.io.*;
import java.lang.*;
import java.net.*;
class Pg4Client{
    public static void main(String[] args) throws Exception{

        DatagramSocket client=new DatagramSocket();
        InetAddress addr=InetAddress.getByName("127.0.0.1");
        Scanner s = new Scanner(System.in);
        int port = 3000;
        while(true){
            byte[] sendbyte=new byte[1024];
            byte[] receivebyte=new byte[1024];
            System.out.println("Enter the string");
            String data;
            data = s.next();
            sendbyte = data.getBytes();
            DatagramPacket sender = new
DatagramPacket(sendbyte,sendbyte.length,addr,port);
            client.send(sender);
            if(data.equals("Exit")){
                break;
            }
            DatagramPacket receiver=new
DatagramPacket(receivebyte,receivebyte.length);
            client.receive(receiver);
            data = new String(receiver.getData()).trim();
            System.out.println("Uppercase string");
            System.out.println(data);
        }
    }
}
```

Output:

```
Server
WELCOME
```

```
Client
Enter the string
welcome
Uppercase string
WELCOME
Enter the string
Exit
```

Result:
Thus, the program is executed and output is obtained.

PROGRAM 5:

Aim:
To write a client and server application to broadcast messages.

Program:

Server Program:

```
import java.io.*;
import java.net.*;
import java.util.*;
class broadcastServer{
    public static void main(String[] args){
        try{
            DatagramSocket ds=new DatagramSocket();
            String msg;
            DatagramPacket dp;
            InetAddress ip = InetAddress.getByName(
"255.255.255.255" );
            Scanner sc=new Scanner(System.in);
            byte[] buf;
            while(true){
                System.out.println("Enter the message to be
broadcasted..\n");
                msg=sc.nextLine();
                buf=msg.getBytes();
                dp=new DatagramPacket(buf,buf.length,ip,5000);
                ds.send(dp);
                if(msg.equals("exit"))
                    break;
            }
            ds.close();
        }
        catch(Exception e){
```

```
            e.printStackTrace();
        }
    }
}
```

Client Program:

```
import java.net.*;
import java.io.*;
import java.util.*;
class broadcastClient{
    public static String toString(byte[] arr){
        String s = "";
        for (int i=0; i<arr.length && arr[i] != 0; s +=
(char)arr[i++]);
        return s;
    }
    public static void main(String[] args){
        try{
            DatagramSocket ds=new DatagramSocket(5000);
            String s1;byte[] buf;
            DatagramPacket dp;
            while(true){
                buf=new byte[1024];
                dp=new DatagramPacket(buf,buf.length);
                ds.receive(dp);
                s1=toString(buf);
                if(s1.equals("exit"))
                    break;
                System.out.println(s1);
            }
            ds.close();
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

Output:

Server:

```
Enter the message to be broadcasted..
Hello
Enter the message to be broadcasted..
Welcome
Enter the message to be broadcasted..
Exit
```

Client:
Hello
Welcome
Exit

Result:
Thus the program is executed and output is obtained.

Experiment No: 2 **Date: 21/12/2018**

REMOTE METHOD INVOCATION

PROGRAM 1

Aim:
To write a Java program to impliment Client Server communication using RPC.

Program:

Server Program

```
import java.io.*;
import java.net.*;
import java.util.*;
class Pg1Server{
    public static void main(String[] args) throws Exception {
        ServerSocket server = new ServerSocket(3000);
        System.out.println("Server ready");
        Socket socket = server.accept();
        Scanner sc = new Scanner(System.in);
        OutputStream ostream = socket.getOutputStream();
        PrintWriter pwrite = new PrintWriter(ostream, true);
        InputStream istream = socket.getInputStream();
        BufferedReader receiveRead = new BufferedReader(new
InputStreamReader(istream));
        String receiveMessage, sendMessage;
        while(true) {
            receiveMessage = receiveRead.readLine();
            if(!receiveMessage.equals("EXIT"))
                System.out.println(receiveMessage);
            else{
                break;
            }
            sendMessage = sc.next();
            pwrite.println(sendMessage);
            System.out.flush();
        }
    }
}
```

Client Program

```
import java.io.*;
import java.net.*;
import java.util.*;
class Pg1Client
    {
    public static void main(String[] args) throws Exception{
        Socket socket = new Socket("127.0.0.1", 3000);
        Scanner sc = new Scanner(System.in);
        OutputStream ostream = socket.getOutputStream();
        PrintWriter pwrite = new PrintWriter(ostream, true);
```

```
        InputStream istream = socket.getInputStream();
        BufferedReader receiveRead = new BufferedReader(new
InputStreamReader(istream));
        System.out.println("Client ready, type and press Enter
key and EXIT to exit");
        String receiveMessage, sendMessage;
        while(true) {
            sendMessage = sc.next();
            pwrite.println(sendMessage);
            if(sendMessage.equals("EXIT")){
                break;
            }
            System.out.flush();
            if((receiveMessage = receiveRead.readLine()) !=
null)
                System.out.println(receiveMessage);
        }
    }
}
```

Output:

```
Server
welcome
Client
Client ready, type and press Enter key and EXIT to exit
welcome
EXIT
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 2

Aim:

To implement Remote command execution using RMI.

Program:

Interface Program

```
import java.rmi.*;
public interface inface2 extends Remote{
    public void CMD(String x) throws Exception;
}
```

Implementation Program

```
import java.net.InetAddress;
import java.rmi.*;
```

```
import java.rmi.server.*;
public class impl2 extends UnicastRemoteObject implements
inface2{
    public impl2()throws Exception{
        super();
    }
    public void CMD(String cmd) throws Exception{
        Runtime r = Runtime.getRuntime();
        Process p = r.exec(cmd);
        System.out.println("The "+ cmd + " has been
executed");
    }
}
```

Server Program

```
import java.util.*;
import java.io.*;
import java.net.*;
import java.rmi.*;
import java.lang.*;
public class Pg2Server{
    public static void main(String[] args) throws Exception{
        impl2 x = new impl2();
        Naming.rebind("rmi://localhost:5000/x",x);
    }
}
```

Client Program

```
import java.util.*;
import java.io.*;
import java.net.*;
import java.rmi.*;
import java.lang.*;
public class Pg2Client{
    public static void main(String[] args) throws Exception{
        String path = "rmi://localhost:5000/x";
        inface2 find = (inface2) Naming.lookup(path);
        Scanner sc = new Scanner(System.in);
        String n = sc.next();
        find.CMD(n);
    }
}
```

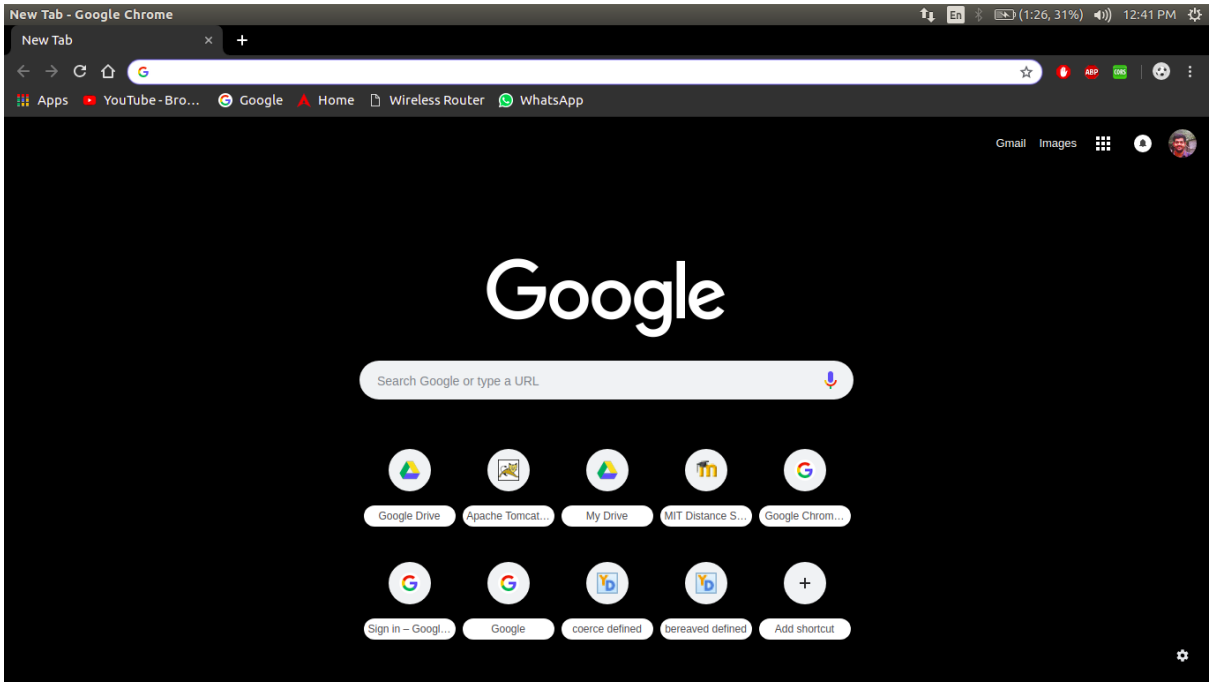
Output:

Server

```
The google-chrome has been executed
```

Client

google-chrome



Result:

Thus, the program is executed and output is obtained.

PROGRAM 3

Aim:

To create RMI to calculate factorial of given number.

Program:

Interface Program

```
import java.rmi.*;
public interface inface3 extends Remote{
    public int fact(int x) throws RemoteException;
}
```

Implementation Program

```
import java.rmi.*;
import java.rmi.server.*;
public class impl3 extends UnicastRemoteObject implements
inface3{
    public impl3()throws Exception{
        super();
    }
    public int fact(int x){
        int ans = 1;
```

```
        for(int i =1;i<=x;i++)
            ans*=i;
        return ans;
    }
}
```

Server Program

```
import java.util.*;
import java.io.*;
import java.net.*;
import java.rmi.*;
import java.lang.*;
public class Pg3Server{
    public static void main(String[] args) throws Exception{
        impl3 x = new impl3();
        Naming.rebind("rmi://localhost:5000/x",x);
    }
}
```

Client Program

```
import java.util.*;
import java.io.*;
import java.net.*;
import java.rmi.*;
import java.lang.*;
public class Pg3Client{
    public static void main(String[] args) throws Exception{
        String path = "rmi://localhost:5000/x";
        inface3 find = (inface3) Naming.lookup(path);
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int ans = find.fact(n);
        System.out.println("The factorial is "+ ans);
    }
}
```

Output:

```
6
The factorial is 720
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 4

Aim:

To create RMI to perform arithmetic operations using RMI.

Program:

Interface Program

```
import java.rmi.*;
public interface inface4 extends Remote{
    public int add(int x,int y) throws RemoteException;
    public int sub(int x,int y) throws RemoteException;
    public int mul(int x,int y) throws RemoteException;
    public int div(int x,int y) throws RemoteException;
    public int mod(int x,int y) throws RemoteException;
}
```

Implementation Program

```
import java.rmi.*;
import java.rmi.server.*;
public class impl4 extends UnicastRemoteObject implements
inface4{
    public impl4()throws Exception{
        super();
    }
    public int add(int x,int y){
        return (x+y);
    }
    public int sub(int x,int y){
        return (x-y);
    }
    public int div(int x,int y){
        return (x/y);
    }
    public int mul(int x,int y){
        return (x*y);
    }
    public int mod(int x,int y){
        return (x%y);
    }
}
```

Server Program

```
import java.util.*;
import java.io.*;
import java.net.*;
import java.rmi.*;
import java.lang.*;
public class Pg4Server{
    public static void main(String[] args) throws Exception{
        impl4 x = new impl4();
```

```
        Naming.rebind("rmi://localhost:3000/x",x);
    }
}
```

Client Program

```
import java.util.*;
import java.io.*;
import java.net.*;
import java.rmi.*;
import java.lang.*;
public class Pg4Client{
    public static void main(String[] args) throws Exception{
        String path = "rmi://localhost:3000/x";
        inface4 find = (inface4) Naming.lookup(path);
        System.out.println("Enter the 2 numbers");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int m = sc.nextInt();
        int addAns = find.add(n,m);
        int subAns = find.sub(n,m);
        int mulAns = find.mul(n,m);
        int divAns = find.div(n,m);
        int modAns = find.mod(n,m);
        System.out.println("The addition answer is "+ addAns);
        System.out.println("The subtraction answer is "+
subAns);
        System.out.println("The multiplication answer is "+
mulAns);
        System.out.println("The division answer is "+ divAns);
        System.out.println("The modulus answer is "+ modAns);
    }
}
```

Output:

```
Enter the 2 numbers
5
7
The addition answer is 12
The subtraction answer is -2
The multiplication answer is 35
The division answer is 0
The modulus answer is 5
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 5

Aim:

Implement Domain name server: It converts IP address for given textual name.

Program:

Interface Program

```
import java.rmi.*;
public interface inface5 extends Remote{
    public String DNS(String x) throws Exception;
}
```

Implementation Program

```
import java.net.InetAddress;
import java.rmi.*;
import java.rmi.server.*;
public class impl5 extends UnicastRemoteObject implements
inface5{
    public impl5()throws Exception{
        super();
    }
    public String DNS(String x){
        String hostname ="";
        try{
            InetAddress address = InetAddress.getByName(x);
            hostname = address.getHostAddress();
        }
        catch(Exception e){
            System.out.println(e);
        }
        return hostname;
    }
}
```

Server Program

```
import java.util.*;
import java.io.*;
import java.net.*;
import java.rmi.*;
import java.lang.*;
public class Pg5Server{
    public static void main(String[] args) throws Exception{
        impl5 x = new impl5();
        Naming.rebind("rmi://localhost:5000/x",x);
    }
}
```

Client Program

```
import java.util.*;
import java.io.*;
import java.net.*;
import java.rmi.*;
import java.lang.*;
public class Pg5Client{
    public static void main(String[] args) throws Exception{
        String path = "rmi://localhost:5000/x";
        inface5 find = (inface5) Naming.lookup(path);
        Scanner sc = new Scanner(System.in);
        String n = sc.next();
        String ans = find.DNS(n);
        System.out.println("The address is "+ ans);
    }
}
```

Output:

```
google.com
The address is 172.217.26.206
```

Result:

Thus, the program is executed and output is obtained.

Experiment No: 3 **Date: 28/12/2018**

CLIENT SIDE SCRIPTING

PROGRAM 1:

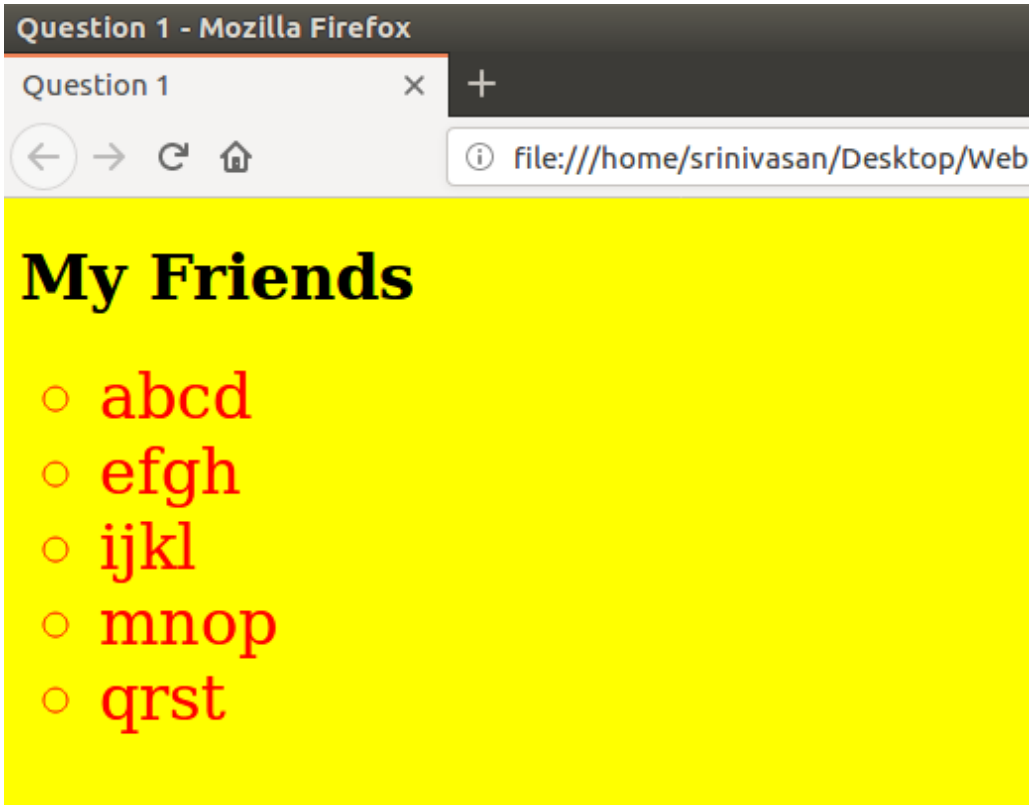
Aim:

To create a Web page HTML CSS that holds a bulleted list (plain circle) of the names of your friends.

Program:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Question 1</title>
        <style>
            ol.list {
                list-style-type:circle;
            }
            body{
                background-color:yellow;
                color: red;
            }
            h1{
                color:black;
            }
            li{
                font-size: 2em;
            }
        </style>
    </head>
    <body>
        <p class="main">
            <h1 class="friends">My Friends</h1>
            <ol class="list">
                <li>abcd</li>
                <li>efgh</li>
                <li>ijkl</li>
                <li>mnop</li>
                <li>qrst</li>
            </ol>
        </p>
    </body>
</html>
```

Output:



Result:
Thus, the program is executed and output is obtained.

PROGRAM 2:

Aim:
To create a web page in HTML CSS to display the maximum and minimum temperature of 5 cities using table.

Program:

HTML Program

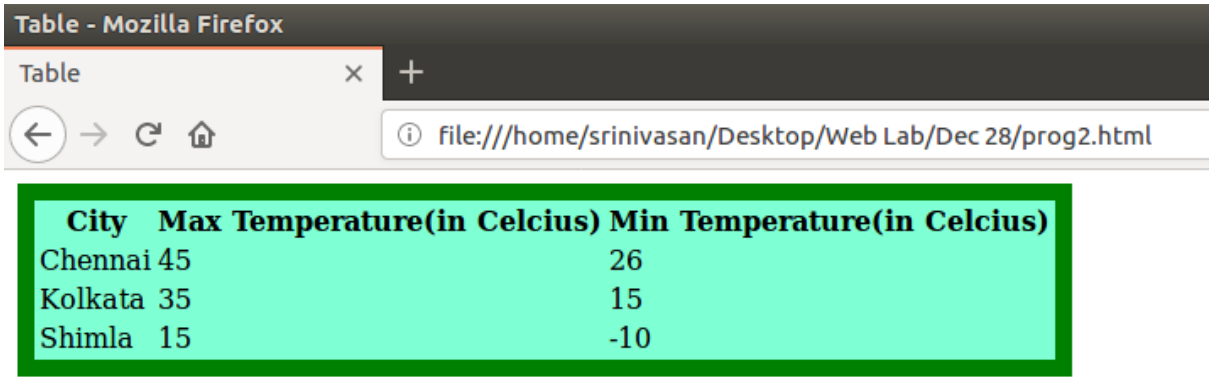
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Table</title>
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <link rel="stylesheet" type="text/css" media="screen"
href="prog2.css" />
</head>
<body>
```

```
    <table class = "table">
        <tr>
            <th>City</th>
            <th>Max Temperature(in Celcius)</th>
            <th>Min Temperature(in Celcius)</th>
        </tr>
        <tr>
            <td>Chennai</td>
            <td>45</td>
            <td>26</td>
        </tr>
        <tr>
            <td>Kolkata</td>
            <td>35</td>
            <td>15</td>
        </tr>
        <tr>
            <td>Shimla</td>
            <td>15</td>
            <td>-10</td>
        </tr>
    </table>
</body>
</html>
```

CSS Program

```
.table {
    background-color: aquamarine;
    border: 10px solid green;
}
```

Output:



Result:

Thus, the program is executed and output is obtained.

PROGRAM 3:

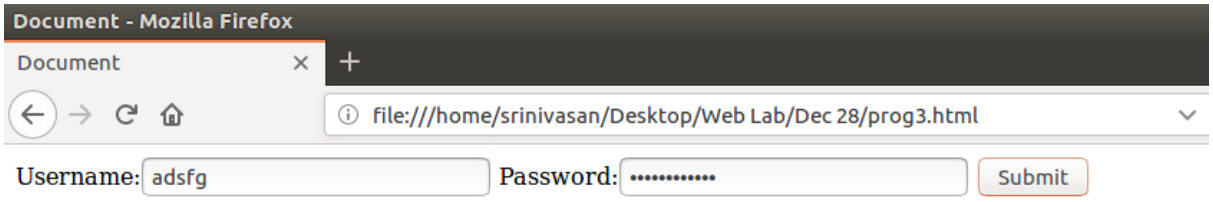
Aim:

To design a web page in HTML that accepts username and password. Opens a new window when the password corresponds to a particular value is set by the developer.

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <!-- <link rel="stylesheet" href="prog3.css"> -->
</head>
<body>
    Username:<input type="text", required, id="ip1">
    Password:<input type="password", required, id="ip2">
    <button type="button", onclick="check()">Submit</button>
</body>
<script>
    function check()
    {
        var uname = document.getElementById("ip1");
        var pass = document.getElementById("ip2");
        var k = pass.value;
        if( k === "abcd")
        {
            window.open("https://www.google.com")
        }
    }
</script>
</html>
```

Output:



Result:

Thus, the program is executed and output is obtained.

PROGRAM 4:

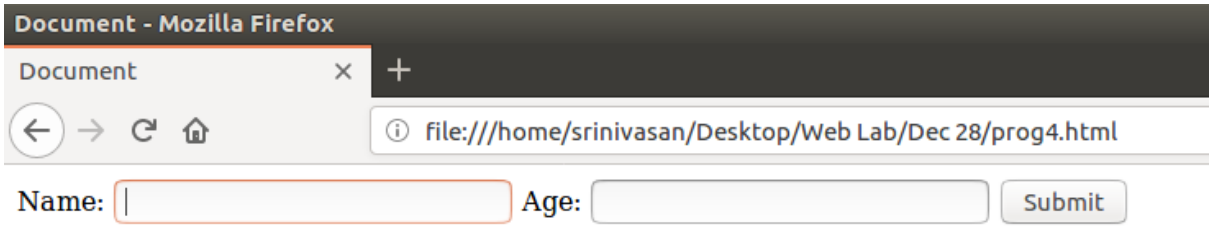
Aim:

To design a web page in HTML that consists of 2 text boxes. When the page is first loaded set the focus to the first text box. The user should not be allowed to leave the box unless enters a value in it.

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    Name: <input type="text", id = "ip1", onblur="check()",
autofocus>
    Age: <input type="text", id = "ip2">
    <button>Submit</button>
</body>
<script>
    function check() {
        var box = document.getElementById("ip1").value;
        if(box == "")
            document.getElementById("ip1").focus();
    }
</script>
</html>
```

Output:



Result:

Thus, the program is executed and output is obtained.

PROGRAM 5:

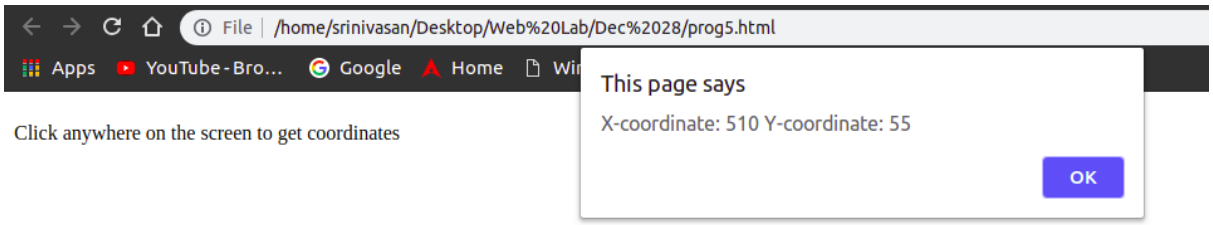
Aim:

To display an alert box to alert the x and y co-ordinates of the cursor in JavaScript.

Program:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Question 5</title>
        <style>
            body{
                position: fixed;
                width:100%;
                height:100%;
            }
        </style>
    </head>
    <body onclick="dispCoord()">
        <p>Click anywhere on the screen to get coordinates</p>
    </body>
    <script>
        function dispCoord(){
            var x = event.clientX;
            var y = event.clientY;
            alert("X-coordinate: "+x +" Y-coordinate: "+y);
        }
    </script>
</html>
```

Output:



Result:

Thus, the program is executed and output is obtained.

PROGRAM 6:

Aim:

To design a simple arithmetic calculator in JavaScript.

Program:

```
<html>
  <head>
      <title>Calculator</title>
  </head>
  <body>
    <h3>Simple Calculator</h3>
    <br />
    <style>
      #calc {
        width: 300px;
        height: 250px;
      }
      #btn {
        width: 100%;
        height: 40px;
        font-size: 20px;
      }
    </style>
    <form Name="calc">
      <table id="calc" border="2">
        <tr>
          <td colspan="5">
            <input
              id="btn"
              name="display"
              onkeypress="return event.charCode >= 48 &&
event.charCode <= 57"
              type="text"
            />
          </td>
          <td style="display:none"><input name="M"
type="number" /></td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="MC"
              OnClick="calc.M.value=''"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="0"
              OnClick="calc.display.value+='0'"
            />
```

```
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="1"
              OnClick="calc.display.value+='1'"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="2"
              OnClick="calc.display.value+='2'"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="+"
              OnClick="calc.display.value+='+'"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="MS"
              OnClick="calc.M.value=calc.display.value"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="3"
              OnClick="calc.display.value+='3'"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="4"
              OnClick="calc.display.value+='4'"
            />
          </td>
```

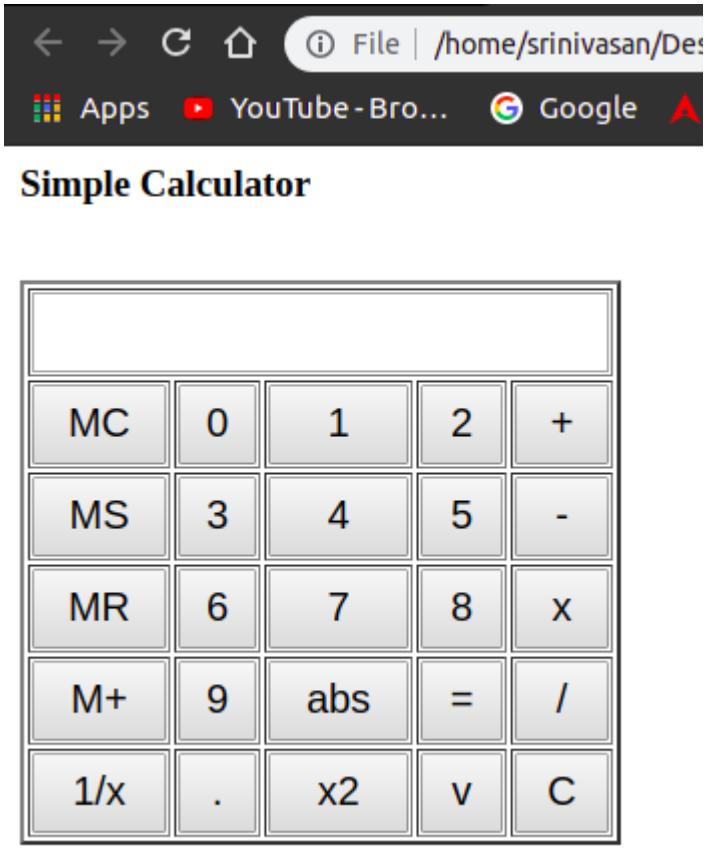
```
          <td>
            <input
              id="btn"
              type="button"
              value="5"
              OnClick="calc.display.value+='5'"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="-"
              OnClick="calc.display.value+='-'"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="MR"
              OnClick="calc.display.value=calc.M.value"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="6"
              OnClick="calc.display.value+='6'"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="7"
              OnClick="calc.display.value+='7'"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="8"
              OnClick="calc.display.value+='8'"
            />
          </td>
          <td>
```

```
            <input
              id="btn"
              type="button"
              value="x"
              OnClick="calc.display.value+='*'"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="M+"

OnClick="calc.M.value=(Number(calc.M.value))+(Number(calc.disp
lay.value))"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="9"
              OnClick="calc.display.value+='9'"
            />
          </td>
          <td>
            <input
            value="abs"
              id="btn"
              type="button"

OnClick="calc.display.value=(Math.abs(calc.display.value))"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="="

OnClick="calc.display.value=eval(calc.display.value)"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="/"
              OnClick="calc.display.value+='/'"
```

```
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="1/x"

OnClick="calc.display.value=1/calc.display.value"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="."
              OnClick="calc.display.value+='.'"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="x2"

OnClick="calc.display.value=Math.pow(calc.display.value,2)"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="v"

OnClick="calc.display.value=Math.sqrt(calc.display.value)"
            />
          </td>
          <td>
            <input
              id="btn"
              type="button"
              value="C"
              OnClick="calc.display.value=''"
            />
          </td>
        </tr>
      </table>
    </form>
  </body>
```

```
</html>
```

Output:



Result:

Thus, the program is executed and output is obtained.

PROGRAM 7:

Aim:

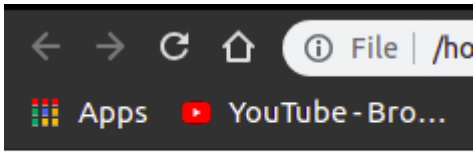
To design a webpage to display a digital clock in JavaScript.

Program:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Page Title</title>
```

```
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <link rel="stylesheet" type="text/css" media="screen"
href="main.css" />
    <script src="main.js"></script>
</head>
<body onload="startTime()">
    <div id="txt"></div>
</body>

<script>
    function startTime() {
        var today = new Date();
        var h = today.getHours();
        var m = today.getMinutes();
        var s = today.getSeconds();
        m = checkTime(m);
        s = checkTime(s);
        document.getElementById('txt').innerHTML = h + ":" + m
+ ":" + s;
        var t = setTimeout(startTime,500);
    }
    function checkTime(i) {
        if(i<10)
        {
            i = "0" + i;
        }
        return i;
    }
</script>
</html>
```

Output:



12:32:31

Result:

Thus, the program is executed and output is obtained.

PROGRAM 8:

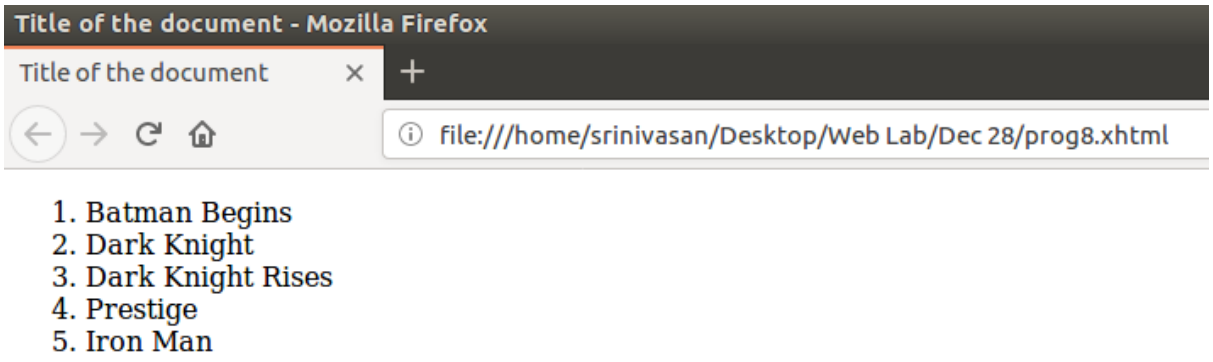
Aim:

To create, test and validate an XHTML document that describes an ordered list of 5 movies.

Program:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>
        Title of the document
    </title>
</head>
<body>
    <ol>
        <li>Batman Begins</li>
        <li>Dark Knight</li>
        <li>Dark Knight Rises</li>
        <li>Prestige</li>
        <li>Iron Man</li>
    </ol>
</body>
</html>
```

Output:



Result:

Thus, the program is executed and output is obtained.

PROGRAM 9:

Aim:

To create, test and validate an XHTML document that has a form with:

(i) A textbox to collect the user names.

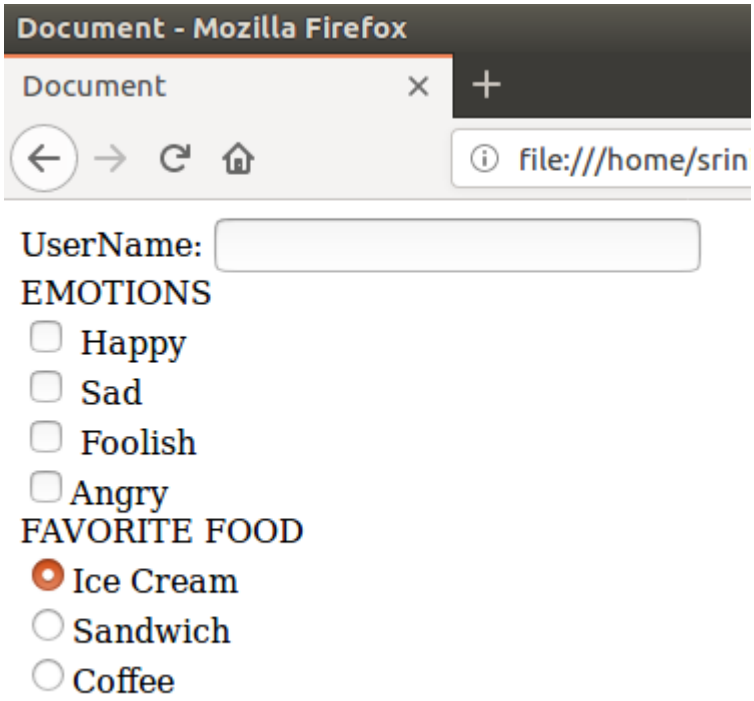
(ii) Four check boxes.

(iii) A collection of 3 radio buttons

Program:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <title>Document</title>
</head>
<body>
    UserName: <input type="text" /> <br />
    EMOTIONS <br />
    <form>
    <input type="checkbox" name="gp2" id="cb1" /> Happy<br>
    <input type="checkbox" name="gp2" id="cb2" /> Sad<br>
    <input type="checkbox" name="gp2" id="cb3" /> Foolish<br>
    <input type="checkbox" name="gp2" id="cb4" />Angry <br>
</form>
    FAVORITE FOOD <br />
    <form>
        <input type="radio" name="gp1" id="rb1" checked =
"checked"/>Ice Cream<br>
        <input type="radio" name="gp1" id="rb2" />Sandwich<br>
        <input type="radio" name="gp1" id="rb3" />Coffee<br>
    </form>
</body>
</html>
```

Output:



Result:
Thus, the program is executed and output is obtained.

PROGRAM 10:

Aim:
To create a dynamic website of Department of Computer Technology in HTML /XHTML /JavaScript

Program:

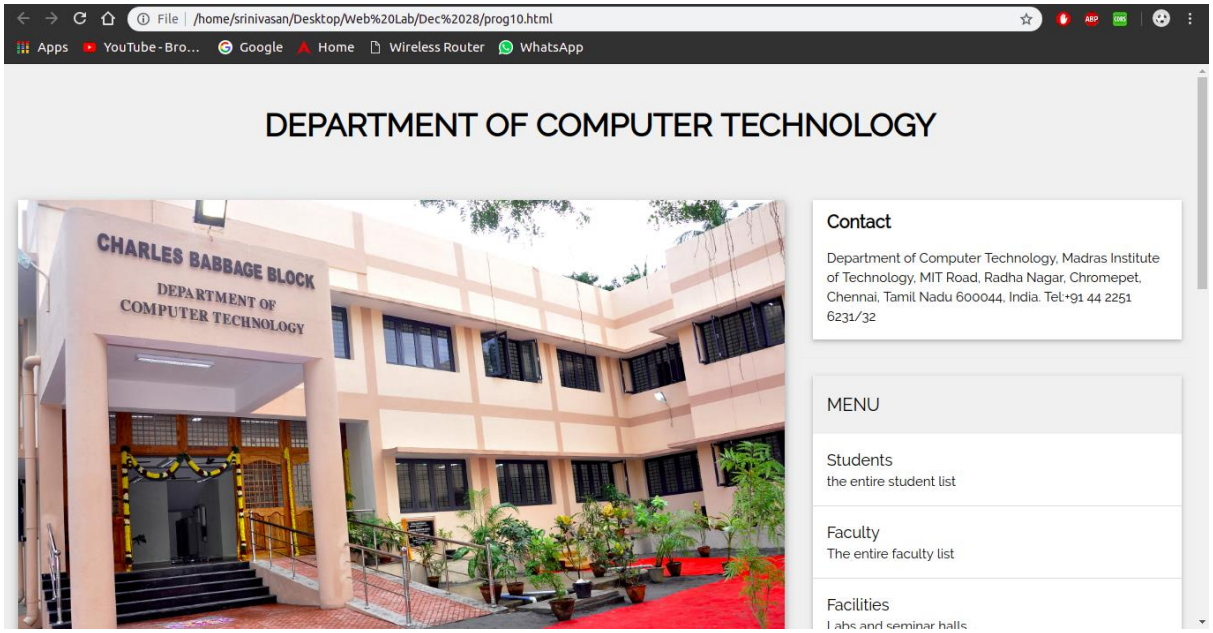
```
<!DOCTYPE html>
<html>
<title>W3.CSS Template</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Raleway">
<style>
body,h1,h2,h3,h4,h5 {font-family: "Raleway", sans-serif}
</style>
<body class="w3-light-grey">
<div class="w3-content" style="max-width:1400px">
```

```
<header class="w3-container w3-center w3-padding-32">
  <h1><b>DEPARTMENT OF COMPUTER TECHNOLOGY</b></h1>
</header>
<div class="w3-row">
<div class="w3-col l8 s12">
  <div class="w3-card-4 w3-margin w3-white">
    <img src="deptold.jpg" alt="Nature" style="width:100%">
    <div class="w3-container">
      <h3><b>Placements of CT</b></h3>
      <h5>Title description, <span class="w3-opacity">April 7,
2014</span></h5>
    </div>
    <div class="w3-container">
      <p>The placement session that took place was highly
successful and many students bagged great offers from various
leading mncs</p>
      <div class="w3-row">
        <div class="w3-col m8 s12">
          <p><button class="w3-button w3-padding-large w3-
white w3-border" onclick="alert('Page currently under
construction! Sorry for the inconvenience')"><b>READ MORE
»</b></button></p>
        </div>
        <div class="w3-col m4 w3-hide-small">
          <p><span class="w3-padding-large w3-
right"><b>Comments  </b> <span class="w3-
tag">0</span></span></p>
        </div>
      </div>
    </div>
  </div>
  <hr>
  <div class="w3-card-4 w3-margin w3-white">
  <img src="ct_dept.jpg" alt="Norway" style="width:100%">
    <div class="w3-container">
      <h3><b>ABOUT US</b></h3>
      <h5>CT, MIT</h5>
    </div>
    <div class="w3-container">
      <p>Computer technology has become an integral part of
our daily life. It is the most important element in the
education of students of the present and the future era.
Computer Technology has revolutionized society to a great
extent. Technological advancements are so rapid in this field
that continual learning is essential to keep the skill set of
students up to date. The department of Computer Technology was
recently established in MIT campus of Anna University by
bifurcating the department of Information Technology. The
department offers course in computer science and engineering
at undergraduate & postgraduate levels and full time/part time
research programs. The teaching and learning process
```

```
emphasizes equally on theoretical and practical aspects
catering to the needs of industries.</p>
      <div class="w3-row">
        <div class="w3-col m8 s12">
          <p><button class="w3-button w3-padding-large w3-
white w3-border" onclick="alert('Page currently under
construction! Sorry for the inconvenience')"><b>READ MORE
»</b></button></p>
        </div>
        <div class="w3-col m4 w3-hide-small">
          <p><span class="w3-padding-large w3-
right"><b>Comments  </b> <span class="w3-
badge">2</span></span></p>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="w3-col l4">
  <div class="w3-card w3-margin w3-margin-top">
    <div class="w3-container w3-white">
      <h4><b>Contact</b></h4>
      <p>Department of Computer Technology,
            Madras Institute of Technology,
            MIT Road, Radha Nagar,
            Chromepet, Chennai,
            Tamil Nadu 600044,
            India.
            Tel:+91 44 2251 6231/32</p>
    </div>
  </div><hr>
  <div class="w3-card w3-margin">
    <div class="w3-container w3-padding">
      <h4>MENU</h4>
    </div>
    <ul class="w3-ul w3-hoverable w3-white">
      <li class="w3-padding-16"
onclick="window.open('http://www.ct.mitindia.edu/studentslist.
html')">
        <span class="w3-large">Students</span><br>
        <span>the entire student list</span>
      </li>
      <li class="w3-padding-16">
        <span class="w3-large">Faculty</span><br>
        <span>The entire faculty list</span>
      </li>
      <li class="w3-padding-16"
onclick="window.open('http://www.ct.mitindia.edu/infrastructur
e.html')">
        <span class="w3-large">Facilities</span><br>
        <span>Labs and seminar halls</span>
```

```
      </li>
      <li class="w3-padding-16 w3-hide-medium w3-hide-small"
onclick="window.open('http://www.ct.mitindia.edu/vision.html')
">
        <span class="w3-large">Vision and Mission</span><br>
        <span>the motto of CT MIT</span>
      </li>
    </ul>
  </div>
  <hr>
</div>
</div><br>
</div>
<footer class="w3-container w3-dark-grey w3-padding-32 w3-
margin-top">
  <button class="w3-button w3-black w3-disabled w3-padding-
large w3-margin-bottom">Previous</button>
  <button class="w3-button w3-black w3-padding-large w3-
margin-bottom" onclick="alert('Page currently under
construction! Sorry for the inconvenience')">Next »</button>
  <p>Powered by <a
href="https://www.w3schools.com/w3css/default.asp"
target="_blank">w3.css</a></p>
 </footer>
 </body>
</html>
```

Output:



Result:

Thus, the program is executed and output is obtained.

Experiment No: 4 **Date: 04/01/2019**

PHP

PROGRAM 1.1:

Aim:

To write a PHP script that take in an array of strings and returns the list of unique strings in the parameter array

Program:

```
<?php
function compute($a1){
    $a2 = array_unique($a1);
    $a3 = array_values($a2);
      return $a3;
}
$a1 = array("a","b","c","a","d","e");
$a2 = compute($a1);
print "The unique elements are\n";
for($x1=0;$x1<count($a2);$x1++){
      print $a2[$x1]." ";
}
?>
```

Output:

```
The unique elements are
a b c d e
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 1.2:

Aim:

To write a PHP script that take in an array of numbers and returns average and median of parameter array.

Program:

```
<?php
function Average($numbers)
{
    # code...
    $k = array_sum($numbers);
    $n = count($numbers);
    $ans = $k/$n;
    return $ans;
}
function Median($numbers)
{
    if (count($numbers)%2 === 0) {
        $mid=count($numbers)/2;
        return (($numbers[$mid-1]+$numbers[$mid])/2);
    }
    else {
        $mid=(count($numbers)-1)/2;
```

```
        return $numbers[$mid];
    }
}
$numbers=array(1,2,3,4,5);
echo ("AVERAGE: " .Average($numbers)."\n");
echo ("MEDIAN: ".Median($numbers)."\n");
?>
```

Output:

```
AVERAGE: 3
MEDIAN: 3
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 1.3:

Aim:

To write a PHP script that take in an array of strings and returns the list of three strings that occur most frequently in parameter array.

Program:

```
<?php
function compute($a1){
      $map = array_count_values($a1);
      arsort($map);
      $a2 = array_keys($map);
      return $a2;
}
$a1 = array("a","b","c","a","b","a","b","a","b","a");
$a2 = compute($a1);
print "The top 3 elements are\n";
for($x1=0;$x1<3;$x1++){
      print $a2[$x1];
      print " ";
}
?>
```

Output:

```
The top 3 elements are
a b c
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 1.4:

Aim:

To write a PHP script that take in an array of numbers (pass by value) and two arrays (pass by reference). The first pass by reference must have numbers less than zero and second must have numbers greater than 0.

Program:

```
<?php
function filter($a, &$p, &$n)
{
    $num = count($a);
    for ($i=0; $i < $num; $i++) {
        if($a[$i] > 0)
        {
            array_push($p,$a[$i]);
        }
        elseif($a[$i] < 0)
        {
            array_push($n,$a[$i]);
        }
    }
}
$numbers=array(-1,2,-3,4,-5);
$pos = array();
$neg = array();
filter($numbers,$pos,$neg);
echo "Positive array:\n";
for ($i=0; $i < count($pos); $i++) {
    echo($pos[$i]." ");
}
echo "\n";
echo "Negative array:\n";
for ($i=0; $i < count($neg); $i++) {
    echo($neg[$i]." ");
}
echo "\n";
?>
```

Output:

```
Positive array:
2 4
Negative array:
-1 -3 -5
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 1.5:

Aim:
To write a PHP script that take in a string of numbers separated by spaces and returns first four-digit number in the string, else return none.

Program:

```
<?php
function compute($a1){
	$a3 = explode(' ',$a1);
	$a2 = array_values($a3);
	for($x1=0;$x1<count($a2);$x1++){
		if(strlen($a2[$x1]) == 4){
			$ans = $a2[$x1];
			return $ans;
		}
	}
	return "false";
}
$a1 ="1 2 34 12 3455";
$a2 = compute($a1);
print $a2 . "\n";
$a1 ="1 2 34 12 345";
$a2 = compute($a1);
print $a2 . "\n";
?>
```

Output:
```
3455
false
```

Result:
Thus, the program is executed and output is obtained.

PROGRAM 1.6:

Aim:
To write a PHP script that take in a file variable of a file of text where the words are separated by spaces or colons and returns the word that appears most often in the file.

Program:

```
<?php
```

```
function findMax($file,$fname)
{
    $filecontents = file_get_contents($fname);
    $words = preg_split('/[\s|:]+/', $filecontents, -1,
PREG_SPLIT_NO_EMPTY);
    $arr = array_count_values($words);
    print_r($words);
    $max = 0;
    $word = "";
    foreach ($arr as $key => $value) {
        if($value > $max)
        {
            $max = $value;
            $word = $key;
        }
    }
    return $word;
}
$fname = "prog1f.txt";
$file = fopen($fname,"r") or die("Unable to open file");
echo findMax($file,$fname);
?>
```

Output:

```
Input file – prog1f.txt
rahul saran mahesh saran:srini:madhu:saran rahul
saran
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 1.7:

Aim:

To write a PHP script that take in a string containing words that are delimited on the left by spaces and on the left with spaces, commas, periods or question marks and returns three most common words in the string that has 3 or more letters

Program:

```
<?php
function compute($a1){
      $a3 = preg_split('/[\s]+|\.|\,/',$a1);
    $a2 = array_values($a3);
    $ff = array();
    for($x=0;$x<count($a2);$x++){
```

```
        if(strlen($a2[$x])>3){
            array_push($ff,$a2[$x]);
        }
    }
    $map = array_count_values($ff);
      arsort($map);
      $ans = array_keys($map);
      return $ans;
}
$a1 =" hello,  there welcome. a vdgdhfghj, welcome.";
$a2 = compute($a1);
print "The top 3 elements are\n";
for($x1=0;$x1<3;$x1++){
      print $a2[$x1];
      print " ";
}
?>
```

Output:

```
The top 3 elements are
welcome hello there
```

Result:

Thus, the program is executed and output is obtained.

PROGRAM 2:

Aim:

To write an XHTML document that includes an anchor tag that calls a PHP document. Also write the PHP document, which returns a randomly chosen greeting of five different greetings. The greetings must be stored as constant strings in the script. A random number between 0 and 4 can be computed by a random function.

Program:

PHP Program

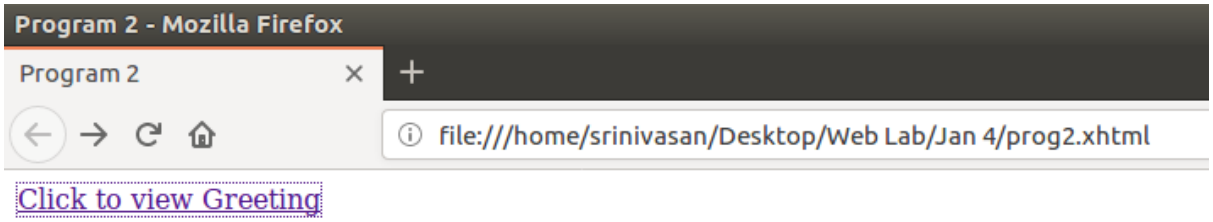
```
<?php
function greetingCall(){
    define("GREETING1","Have a nice day.");
    define("GREETING2","Thank you.");
    define("GREETING3","Welcome.");
    define("GREETING4","Nice meeting you.");
    define("GREETING5","Hope to see you again.");
    $greet =
array(GREETING1,GREETING2,GREETING3,GREETING4,GREETING5);
    $index = rand(0,4);
    echo "<div>".$greet[$index]."</div>";
```

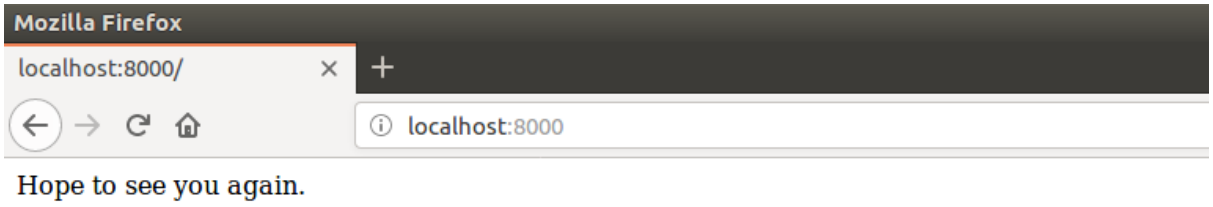
```
}
greetingCall();
?>
```

XHTML Program

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>
        Program 2
    </title>
</head>
<body>
    <a href="http://localhost:8000">Click to view Greeting</a>
</body>
</html>
```

Output:





Result:

Thus, the program is executed and output is obtained.

PROGRAM 3:

Aim:

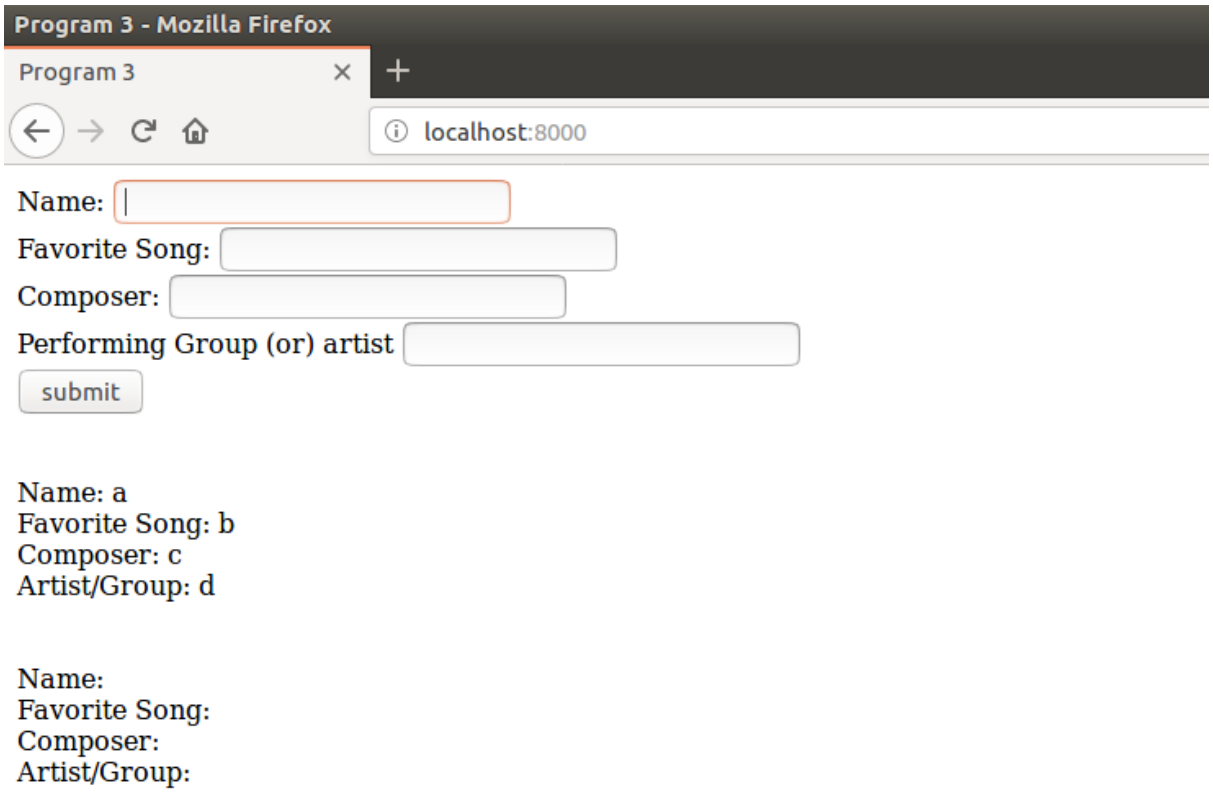
To write an XHTML document to create a form that collects favourite popular songs, including the name of the song, the composer, and the performing artist or group. The document

must call one PHP script where the form is submitted and another to request a current list of survey results.

Program:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>
        Program 3
    </title>
</head>
<body>
    <?php
        $display = "";
        $content = $artist = $name = $song = $composer = "";
        if ($_SERVER["REQUEST_METHOD"] == "POST")
        {
            $file = fopen("prog3.txt","r");
            $content = "Name: " . $_POST["name"] . "\n" .
"Favorite Song: " . $_POST["song"] . "\n" . "Composer: " .
$_POST["composer"] . "\n" . "Artist/Group: " .
$_POST["artist"] . "\n\n";
            $myfile = file_put_contents('prog3.txt',
$content.PHP_EOL , FILE_APPEND | LOCK_EX);
            $display = file('prog3.txt');
        }
    ?>
    <form method="post" action="<?php
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
        Name: <input type="text" name="name" id="name"
value="<?php echo $name;?>"/><br />
        Favorite Song: <input type="text" name="song"
id="song" value="<?php echo $song;?>"/><br />
        Composer: <input type="text" name="composer"
id="composer" value="<?php echo $composer;?>"/><br />
        Performing Group (or) artist <input type="text"
name="artist" id="artist" value="<?php echo $artist;?>"/><br >
        <input type="submit" value="submit" /><br />
        <br />
        <br />
        <div><?php
            for ($i=0; $i < count($display); $i++) {
               echo $display[$i] . "<br>";
            }
        ?></div>
    </form>
</body>
</html>
```

Output:



Result:

Thus, the program is executed and output is obtained.

PROGRAM 4:

Aim:

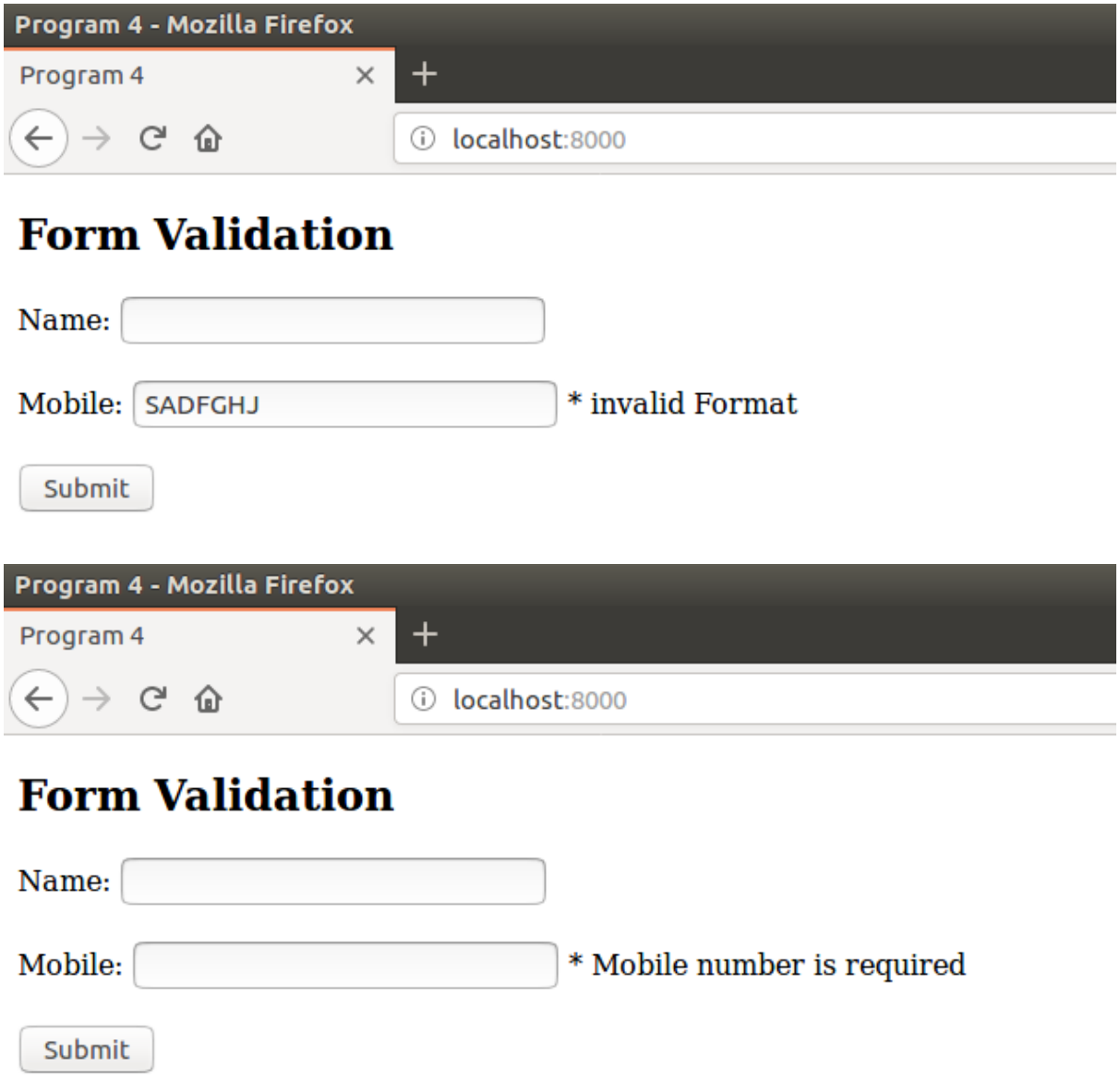
To write an XHTML document to create a form that collects favourite popular songs, including the name of the song, the composer, and the performing artist or group. The document must call one PHP script where the form is submitted and another to request a current list of survey results.

Program:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>
        Program 4
    </title>
</head>
```

```
<body>
    <?php
    $mobileErr = "";
    $mobile = "";
    $name = "";
    if ($_SERVER["REQUEST_METHOD"] == "POST")
    {
        if(empty($_POST["mobile"])) {
            $mobileErr = "Mobile number is required";
        } else {
            $mobile = test_input($_POST["mobile"]);
            if(!preg_match("/\+\d{2}-\d{4}-\d{6}/", $mobile))
            {
                $mobileErr = "invalid Format";
            }
        }
    }
    function test_input($data) {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }
    ?>
    <h2>Form Validation</h2>
    <form method="post" action="<?php
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
        Name: <input type="text" name="name" value="<?php echo
$name;?>" /><br><br>
        Mobile: <input type="text" name="mobile" value="<?php
echo $mobile;?>" />
        <span class="error">* <?php echo $mobileErr;?></span>
        <br /><br />
        <input type="submit" name="submit" value="Submit" />
    </form>
</body>
</html>
```

Output:



Result:
Thus, the program is executed and output is obtained.

PROGRAM 5:

Aim:
To modify the PHP script from Exercise 2 to count the number of visitors and display the numbers for each visitor.

Program:

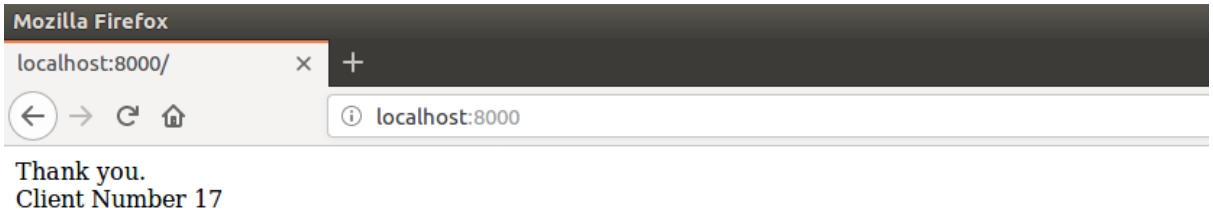
PHP Program

```
<?php
function greetingCall(){
    define("GREETING1","Have a nice day.");
    define("GREETING2","Thank you.");
    define("GREETING3","Welcome.");
    define("GREETING4","Nice meeting you.");
    define("GREETING5","Hope to see you again.");
    $greet =
array(GREETING1,GREETING2,GREETING3,GREETING4,GREETING5);
    $index = rand(0,4);
    $fp = file_get_contents("prog5.txt");
    $val = (int)$fp;
    $val+=1;
    $fp = (string)$val;
    file_put_contents("prog5.txt",$fp);
    print "<div>".$greet[$index]."</div>";
    print "<div> Client Number ".$val."</div>";
}
greetingCall();
?>
```

XHTML Program

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>
        Program 5
    </title>
</head>
<body>
    <a href="http://localhost:8000">Click to view Greeting</a>
</body>
</html>
```

Output:



Result:

Thus, the program is executed and output is obtained.

PROGRAM 6:

Aim:

To implement the following modules using Server-Side Scripting (PHP)

(i) Gathering form data.
(ii) Querying the database.
(iii) Response generation.
(iv) Session management.
(v) Use MySQL or JDBS or Oracle.

Program:

```
<html>
<head>
    <style>
        body{
            display: inline-block;
            width:100%;
            height:100%;
            text-align:center;
        }
    </style>
</head>
    <body>
        <?php
            session_start();
        ?>
        <?php
        $regno = "";
        $pwd = "";
        $regErr = "";
        $pwdErr = "";
        $name = "";
        $address = "";
        $mobile = "";
        $time = $_SERVER['REQUEST_TIME'];
        $timeout_duration = 60;
        if (isset($_SESSION['LAST_ACTIVITY']) && ($time -
$_SESSION['LAST_ACTIVITY']) > $timeout_duration) {
            session_unset();
            session_destroy();
            session_start();
        }
        if ($_SERVER["REQUEST_METHOD"] == "POST")
        {
            if(empty($_POST["regno"])) {
                $regErr = "Student Registeration number is
required";
            }
            if(empty($_POST["pwd"])) {
```

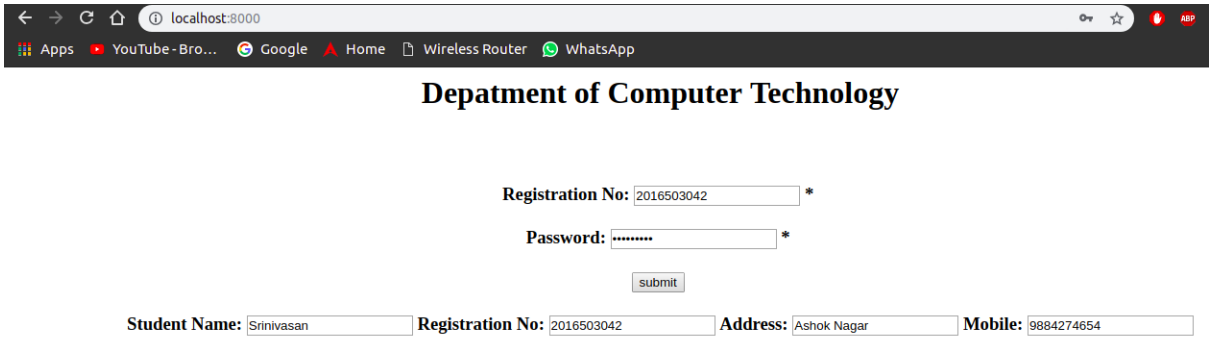
```
                $pwdErr = "Password is required";
            }
            if(isset($_POST["reg"]) && isset($_POST["pwd"])){
                $servername = "localhost";
                $username = "root";
                $password = "srini1998";
                $dbname = "Base1";
                $conn = new
mysqli($servername,$username,$password,$dbname);
                if($conn -> connect_error){
                    die("Connection failed: ".$conn-
>connect_error);
                }
                $regno = $_POST['regno'];
                $pwd = $_POST['pwd'];

                $sql = "SELECT name,address,mobile,id from
Tab1 where id = '$regno' and password = '$pwd';";
                $result = $conn->query($sql);
                if($result->num_rows > 0){
                    while($row = $result->fetch_assoc()){
                        $name = $row["name"];
                        $address = $row["address"];
                        $mobile = $row["mobile"];
                        $regno = $row["id"];
                    }
                }
                else{
                    print "Data mismatch. Please try again.";
                    $regno = "";
                    $pwd = "";
                }

                $conn ->close();
            }
            $_SESSION['LAST_ACTIVITY'] = $time;
        }
        ?>
        <div><h1>Depatment of Computer Technology</h1></div>
        <br>
        <br>
        <form class = "form1" method="post" action="<?php
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
            <h3>
            Registeration No: <input type="text" name="regno"
id ="regno" value="<?php echo $regno;?>" />
            <span class="error">* <?php echo $regErr;?></span>
            <br /><br />
            Password: <input type="password" name="pwd" id
="pwd" value="<?php echo $pwd;?>" />
            <span class="error">* <?php echo $pwdErr;?></span>
```

```
            <br /><br />
            <input type="submit" name="submit" value="submit">
            <br>
            <br>
            <div>
                Student Name: <input type="text" name="name"
id  = "name" value="<?php echo $name;?>" />
                Registeration No: <input type="text"
name="reg" id  = "reg" value="<?php echo $regno;?>" />
                Address: <input type="text" name="address" id
= "address" value="<?php echo $address;?>" />
                Student Name: <input type="text" name="mobile"
id  = "mobile" value="<?php echo $mobile;?>" />
            </div>
            </h3>
        </form>
    </body>
</html>
```

DB Schema

Field	Type	Null	Key	Default	Extra
name	varchar(30)	YES		NULL	
id	varchar(10)	YES		NULL	
address	varchar(30)	YES		NULL	
mobile	varchar(10)	YES		NULL	
password	varchar(20)	YES		NULL	

Output:



Result:

Thus, the program is executed and output is obtained.

Experiment No: 5 **Date: 11/01/2019**

PYTHON

PROGRAM 1:

Aim:
To create a new program called HelloWorld.py. This file should be used to write your ”Hello World !” program.

Program:
```
print("Hello World")
```

Output:
```
Hello World
```

Result:
Thus the program is executed and output is obtained.

PROGRAM 2:

Aim:
To write a function reverse to reverse a list without using the reverse function.

Program:

```
def reverse_list(a):
     b = list()
     for i in range(0,len(a)):
          b.append(a[len(a)-i-1])
     return b

a = [1,2,3,4,5]
b = reverse_list(a)
print(b)
```

Output:
```
[5,4,3,2,1]
```

Result:
Thus the program is executed and output is obtained.

PROGRAM 3:

Aim:
To write a method fact that takes a number from the user and prints the factorial.

Program:

```
from math import factorial
def fact(n):
return factorial(n)

a = 6
print(fact(6))
```

Output:
720

Result:
Thus the program is executed and output is obtained.

PROGRAM 4:

Aim:
To write a GUI for the expression calculator using tk.

Program:

```
from tkinter import *

expression = ""

def press(num):
     global expression
     expression = expression + str(num)
     equation.set(expression)


def equalpress():
     try:
          global expression
          total = str(eval(expression))
          equation.set(total)
          expression = ""
     except:
          equation.set(" error ")
          expression = ""


def clear():
     global expression
```

```
    expression = ""
    equation.set("")


if __name__ == "__main__":
    gui = Tk()
    gui.configure(background = "light green")
    gui.title("Simple Calculator")
    gui.geometry("265x125")
    equation = StringVar()
    expression_field = Entry(gui, textvariable = equation)
    expression_field.grid(columnspan=4, ipadx=70)
    equation.set('Enter expression')

    button1 = Button(gui, text=' 1 ', fg='black', bg='red',
    command=lambda: press(1), height=1, width=7)
    button1.grid(row=2, column=0)
    button2 = Button(gui, text=' 2 ', fg='black', bg='red',
    command=lambda: press(2), height=1, width=7)
    button2.grid(row=2, column=1)
    button3 = Button(gui, text=' 3 ', fg='black', bg='red',
    command=lambda: press(3), height=1, width=7)
    button3.grid(row=2, column=2)
    button4 = Button(gui, text=' 4 ', fg='black', bg='red',
    command=lambda: press(4), height=1, width=7)
    button4.grid(row=3, column=0)
    button5 = Button(gui, text=' 5 ', fg='black', bg='red',
    command=lambda: press(5), height=1, width=7)
    button5.grid(row=3, column=1)
    button6 = Button(gui, text=' 6 ', fg='black', bg='red',
    command=lambda: press(6), height=1, width=7)
    button6.grid(row=3, column=2)
    button7 = Button(gui, text=' 7 ', fg='black', bg='red',
    command=lambda: press(7), height=1, width=7)
    button7.grid(row=4, column=0)
    button8 = Button(gui, text=' 8 ', fg='black', bg='red',
    command=lambda: press(8), height=1, width=7)
    button8.grid(row=4, column=1)
    button9 = Button(gui, text=' 9 ', fg='black', bg='red',
    command=lambda: press(9), height=1, width=7)
    button9.grid(row=4, column=2)
    button0 = Button(gui, text=' 0 ', fg='black', bg='red',
    command=lambda: press(0), height=1, width=7)
    button0.grid(row=5, column=0)
    plus = Button(gui, text=' + ', fg='black', bg='red',
    command=lambda: press("+"), height=1, width=7)
    plus.grid(row=2, column=3)
```

```
    minus = Button(gui, text=' - ', fg='black', bg='red',
    command=lambda: press("-"), height=1, width=7)
    minus.grid(row=3, column=3)
    multiply = Button(gui, text=' * ', fg='black', bg='red',
    command=lambda: press("*"), height=1, width=7)
    multiply.grid(row=4, column=3)
    divide = Button(gui, text=' / ', fg='black', bg='red',
    command=lambda: press("/"), height=1, width=7)
    divide.grid(row=5, column=3)
    equal = Button(gui, text=' = ', fg='black', bg='red',
    command=equalpress, height=1, width=7)
    equal.grid(row=5, column=2)
    clear = Button(gui, text='Clear', fg='black', bg='red',
    command=clear, height=1, width=7)
    clear.grid(row=5, column='1')

    gui.mainloop()
```

Output:

Result:
Thus the program is executed and output is obtained.

PROGRAM 5:

Aim:
To write a procesure to install packages requests, flask and expore them using pip.

Program:

Prog5Server.py:

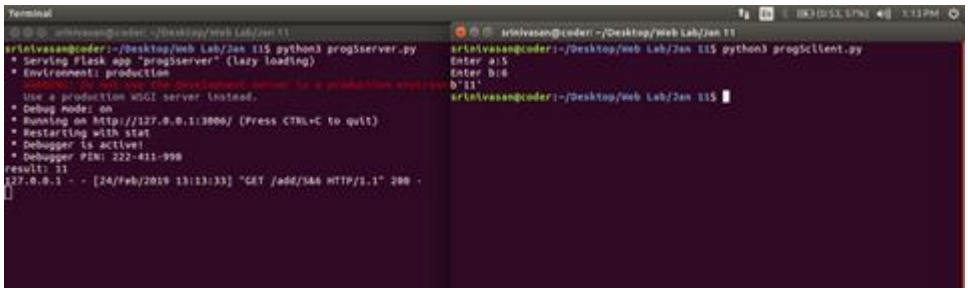
```
from flask import Flask

app = Flask(__name__,static_url_path = "")
@app.route('/add/<int:x>&<int:y>', methods = ['GET'])
def add(x,y):
    print("result:",str(x+y))
    return str(x+y)

if __name__ == '__main__':
```

```
    app.run(host='127.0.0.1',port = 3006,debug = True)
```

Prog5Client.py:

```
import httplib2

http = httplib2.Http()
a = input("Enter a:")
b = input("Enter b:")
operation = "/add/"+str(a)+"&"+str(b)
url = "http://localhost:3006"+operation
(response_headers, content) = http.request(url, method="GET")
print(str(content))
```

Output:



Result:
Thus the program is executed and output is obtained.

PROGRAM 6:

Aim:
To write a script that imports Requests and fetches the content from a page.

Program:

```
from bs4 import BeautifulSoup
import requests

def scrapeSite(url,out):
    r = requests.get(url)
    txt = r.text
    htmlCode = BeautifulSoup(txt,'html.parser')
    if htmlCode.h1:
        ans = htmlCode.find_all('h1')
        for i in ans:
```

```
                out.write(str(i.string) + "\n")
     if htmlCode.p:
          ans = htmlCode.find_all('p')
          for i in ans:
               out.write(str(i.string) + "\n")

try:
     url = input("Enter a url:")
     out = open("out6.txt","w")
     scrapeSite(url,out)
     out.close()
except IOError as e:
     print(e)
```

Output:

```
python prog6.py
Enter a url:https://www.geeksforgeeks.org/c-program-for-tower-
of-hanoi/
```

Out.txt

```
Program for Tower of Hanoi
None
Approach:
Examples:
None
None
None
None
None
None
None
None
None
None
None
None
None
None
None
None
Output:
None
None
None
None
None
None
```

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.
None
None

Result:
Thus the program is executed and output is obtained.

PROGRAM 7:

Aim:
To write a simple script that serves a simple HTTP response and a simple HTML page using Python.

Program:

```
import webbrowser
from urllib.request import urlopen

f = open("helloworld.html","w")
res = urlopen("https://www.google.com")
print(res.info())
res.close()

message = """<html>
<head></head>
<body><p>"""+ str(res.info()) + """</p></body>
</html>"""

f.write(message)
f.close()

webbrowser.open_new_tab('helloworld.html')
```

Output:

HTTP Response:
```
Date: Sun, 24 Feb 2019 06:54:36 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Server: gws
```

```
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Set-Cookie: 1P_JAR=2019-02-24-06; expires=Tue, 26-Mar-2019
06:54:36 GMT; path=/; domain=.google.com
Set-Cookie: NID=160=FHyqmm8VstO2NgnWS9DknWzCyanGq6qQF6GyP6-
FHzJkFWLVKHBjQzCxUx4MRhzAiFnZ5iBGc5Z6ceSo6pqWK3LiIZgxWwQuSMhka
22InLU402eT-
K2JGswFISh48tny7hS_P1HDYN6qqzMAEp0ymDDCaDDEgRNQ8pBlByUlTpM;
expires=Mon, 26-Aug-2019 06:54:36 GMT; path=/;
domain=.google.com; HttpOnly
Alt-Svc: quic=":443"; ma=2592000; v="44,43,39"
Accept-Ranges: none
Vary: Accept-Encoding
Connection: close
```

HTML Page:

Result:
Thus the program is executed and output is obtained.

PROGRAM 8:

Aim:

To implement the following modules using Server Side Scripting (Python)
(i) Gathering form data.
(ii) Querying the database.
(iii) Response generation.
(iv) Session management.
(v) Use MySQL or JDBS or Oracle .

Program:

Prog8.py:

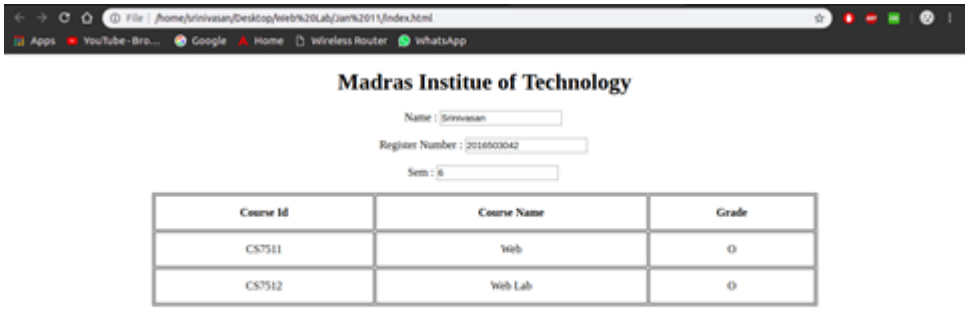
```
import mysql.connector
from flask import Flask, request

app = Flask(__name__,static_url_path = "")
@app.route('/result/<string:x>&<string:y>', methods = ['GET'])
```

```
def result(x,y):
     try:
          db = mysql.connector.connect(host="localhost",
     user="root", password="srini1998", database="Base1")
          print(db)
          cursor = db.cursor()
          sql = "SELECT * from Tab2 where id ='"+x+"' and
     password='"+y+"';"
          print(sql)
          cursor.execute(sql)
          result = cursor.fetchall()
          print(result)
          if len(result) == 0:
               return "Please Verify Credentials"
          ans = ""
          for a in result:
               for b in range(0,len(a)-1):
                    ans=ans+"#"+a[b]
                    return ans
     except mysql.connector.Error as e:
          print(e)

if __name__ == '__main__':
     app.run(host='127.0.0.1',port=3005,debug = True)
```

Database Schema:

Field	Type	Null	Key	Default	Extra
name	varchar(15)	YES		NULL	
id	varchar(15)	YES		NULL	
sem	varchar(2)	YES		NULL	
course_1_id	varchar(6)	YES		NULL	
course_1_name	varchar(20)	YES		NULL	
grade_1	varchar(3)	YES		NULL	
course_2_id	varchar(6)	YES		NULL	
course_2_name	varchar(20)	YES		NULL	
grade_2	varchar(3)	YES		NULL	
password	varchar(20)	YES		NULL	

Output:



Result:

Thus the program is executed and output is obtained.

Experiment No: 6 **Date: 08/02/2019**

JAVA SERVLET

PROGRAM 1:

Aim:

To create a servlet program that makes Ordered list of four random numbers

Program:

Prog1.java:

```
import java.io.*;
import java.util.Random;
import javax.servlet.*;
import javax.servlet.http.*;

public class Prog1 extends HttpServlet {
     protected void doGet(HttpServletRequest req,
     HttpServletResponse res) throws IOException {
          res.setContentType("text/html");
          PrintWriter out = res.getWriter();
          Random rand = new Random();
          out.println("<html>");
          out.println("<head>");
          out.println("<title>Random numbers</title>");
          out.println("</head>");
          out.println("<body>");
          out.println("<ol>");
          out.println("<li>" +
          Integer.toString(rand.nextInt(100) + 1) + "</li>");
          out.println("<li>" +
          Integer.toString(rand.nextInt(100) + 1) + "</li>");
          out.println("<li>" +
          Integer.toString(rand.nextInt(100) + 1) + "</li>");
          out.println("<li>" +
          Integer.toString(rand.nextInt(100) + 1) + "</li>");
          out.println("</ol>");
          out.println("</body>");
          out.println("</html>");
          out.close();
     }
}
```

Deployment:

```
<servlet>
    <servlet-name>Prog1</servlet-name>
    <servlet-class>Prog1</servlet-class>
```

```
</servlet>

<servlet-mapping>
    <servlet-name>Prog1</servlet-name>
    <url-pattern>/Prog1</url-pattern>
</servlet-mapping>
```

Output:

Result:
Thus the program is executed and output is obtained.

PROGRAM 2:

Aim:
To create a servlet program that uses a loop to output an HTML table with 20 Rows and 3 columns.

Program:

Prog2.java:

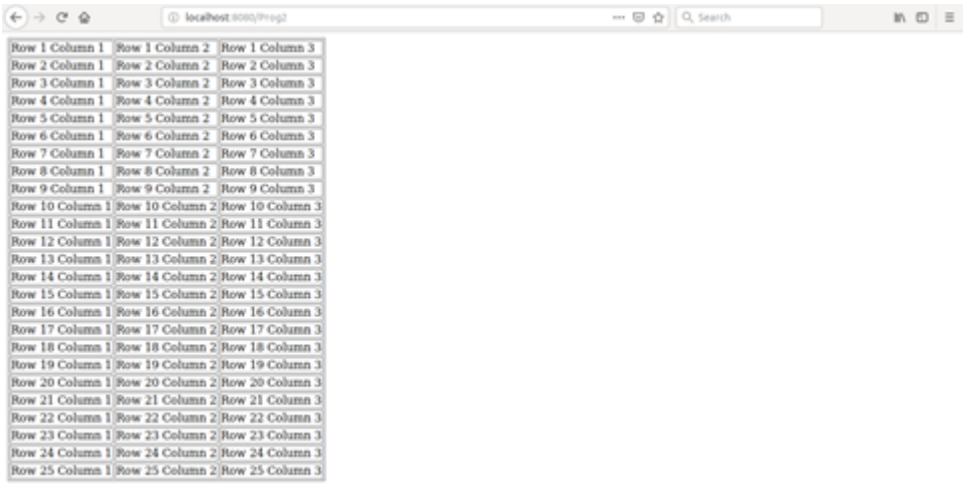
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

public class Prog2 extends HttpServlet {
     public void doGet(HttpServletRequest req,
     HttpServletResponse res) throws IOException {
          res.setContentType("text/html");
          PrintWriter out = res.getWriter();
          out.println("<html>");
          out.println("<head>");
          out.println("<title> Servlet 2 </title>");
          out.println("</head>");
          out.println("<body>");
          out.println("<table border='1'>");
```

```
            for (int i = 1; i < 26; i++) {
                 out.println("<tr>");
                 for (int j = 1; j < 4; j++) {
                      out.println("<td>");
                      out.println("Row " + i + " Column " + j);
                      out.println("</td>");
                 }
                 out.println("</tr>");
            }
            out.println("</table>");
            out.println("</body>");
            out.println("</html>");
      }
}
```

Deployment:

```
<servlet>
    <servlet-name>Prog2</servlet-name>
    <servlet-class>Prog2</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>Prog2</servlet-name>
    <url-pattern>/Prog2</url-pattern>
</servlet-mapping>
```

Output:



Row 1 Column 1	Row 1 Column 2	Row 1 Column 3
Row 2 Column 1	Row 2 Column 2	Row 2 Column 3
Row 3 Column 1	Row 3 Column 2	Row 3 Column 3
Row 4 Column 1	Row 4 Column 2	Row 4 Column 3
Row 5 Column 1	Row 5 Column 2	Row 5 Column 3
Row 6 Column 1	Row 6 Column 2	Row 6 Column 3
Row 7 Column 1	Row 7 Column 2	Row 7 Column 3
Row 8 Column 1	Row 8 Column 2	Row 8 Column 3
Row 9 Column 1	Row 9 Column 2	Row 9 Column 3
Row 10 Column 1	Row 10 Column 2	Row 10 Column 3
Row 11 Column 1	Row 11 Column 2	Row 11 Column 3
Row 12 Column 1	Row 12 Column 2	Row 12 Column 3
Row 13 Column 1	Row 13 Column 2	Row 13 Column 3
Row 14 Column 1	Row 14 Column 2	Row 14 Column 3
Row 15 Column 1	Row 15 Column 2	Row 15 Column 3
Row 16 Column 1	Row 16 Column 2	Row 16 Column 3
Row 17 Column 1	Row 17 Column 2	Row 17 Column 3
Row 18 Column 1	Row 18 Column 2	Row 18 Column 3
Row 19 Column 1	Row 19 Column 2	Row 19 Column 3
Row 20 Column 1	Row 20 Column 2	Row 20 Column 3
Row 21 Column 1	Row 21 Column 2	Row 21 Column 3
Row 22 Column 1	Row 22 Column 2	Row 22 Column 3
Row 23 Column 1	Row 23 Column 2	Row 23 Column 3
Row 24 Column 1	Row 24 Column 2	Row 24 Column 3
Row 25 Column 1	Row 25 Column 2	Row 25 Column 3

Result:
Thus the program is executed and output is obtained.

PROGRAM 3a:

Aim:

To create a servlet program to ake a registration form that collects a Name, Register Number, and email address. Send the data to the servlet that displays it.

Program:

Prog3a.java:

```
import java.io.*;
import java.util.Random;
import javax.servlet.*;
import javax.servlet.http.*;

public class Prog3a extends HttpServlet {
     protected void doGet(HttpServletRequest req,
     HttpServletResponse res) throws IOException {
          res.setContentType("text/html");
          PrintWriter out = res.getWriter();
          out.println("<html>");
          out.println("<head>");
          out.println("<title>Form Display</title>");
          out.println("</head>");
          out.println("<body>");
          out.println("Name: " + req.getParameter("name"));
          out.println("Registration number: " +
          req.getParameter("reg_no"));
          out.println("Email_id: "+
          req.getParameter("email"));
          out.println("</body>");
          out.println("</html>");
          out.close();
     }
}
```

Prog3a.html:

```
<html>
   <body>
      <form action = "Prog3a" method = "GET">
         Name: <input type = "text" name = "name">
         <br />
         Registration Number: <input type = "text" name =
"reg_no">
         <br />
         Email Id: <input type = "text" name = "email" />
         <input type = "submit" value = "Submit" />
```

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

Deployment:

```
<servlet>
     <servlet-name>Prog3a</servlet-name>
     <servlet-class>Prog3a</servlet-class>
</servlet>
<servlet-mapping>
     <servlet-name>Prog3a</servlet-name>
     <url-pattern>/Prog3a</url-pattern>
</servlet-mapping>
```

Output:

Result:
Thus the program is executed and output is obtained.

PROGRAM 4:

Aim:
To use session tracking to rdo the servlet that says "Welcome Guest" to first-time visitors (with browsing session) and "Welcome back" to repeat visitors

Program:

Prog4.java:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

public class Prog4 extends HttpServlet {
     private int hitCount;
     private HttpSession session;
     public void init() {
          hitCount = 0;
```

```
            session = null;
      }
      public void doGet(HttpServletRequest req,
      HttpServletResponse res) throws IOException {
            hitCount++;
            res.setContentType("text/html");
            PrintWriter out = res.getWriter();
            if (session == null) {
                  out.println("Welcome Guest");
                  session = req.getSession();
            } else {
                  out.println("Welcome Back");
            }
      }
}
```

Deployment:

```
<servlet>
    <servlet-name>Prog4</servlet-name>
    <servlet-class>Prog4</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>Prog4</servlet-name>
    <url-pattern>/Prog4</url-pattern>
</servlet-mapping>
```

Output:

First Time:

During same session:

Result:
Thus the program is executed and output is obtained.

PROGRAM 5:

Aim:
To write a servlet that displays the values of Name, Register Number, and emailAddress request parameters. If a parameter is missing and the client is a irst-time visitor, have the servlet list "Unknown for the missing values. If a parameter

is missing and the client is a repeat visitor, have the servlet use previously entered values for the missing values

Program:

Prog5.java:

```
import java.io.*;
import java.util.Random;
import javax.servlet.*;
import javax.servlet.http.*;

public class Prog5 extends HttpServlet {
     protected void doGet(HttpServletRequest req,
     HttpServletResponse res) throws IOException {
          res.setContentType("text/html");
          PrintWriter out = res.getWriter();
          Cookie name = null, reg_no = null, email = null;
          Cookie[] cookies = null;
          cookies = req.getCookies();
          try {
               if (cookies != null) {
                    for (int i = 0; i < cookies.length; i++) {
                         if(cookies[i].getName().equals("Name"
                         )){
                              name = cookies[i];
                         }
                         if(cookies[i].getName().equals("Reg_N
                         o"))
                              reg_no = cookies[i];
                         if(cookies[i].getName().equals("Email
                         "))
                              email = cookies[i];
                    }
                    if (req.getParameter("name").length() ==
                    0)
                         out.println("Name: " +
                    name.getValue()) ;
                    else {
                         out.println("Name: " +
                         req.getParameter("name"));
                         if
                         (name.getValue().equals("Unknown")) {
                              name.setValue(req.getParameter("
                              name"));
                              res.addCookie(name);
                         }
                    }
```

```
                if (req.getParameter("reg_no").length() ==
                0)
                     out.println("Registration Number: " +
                reg_no.getValue());
                else {
                     out.println("Registration Number: " +
                req.getParameter("reg_no"));
                     if(reg_no.getValue().equals("Unknown"
                     )){
                          reg_no.setValue(req.getParameter
                     ("reg_no"));
                          res.addCookie(reg_no);
                     }
                }
                if (req.getParameter("email").length() ==
                0)
                     out.println("Email: " +
                email.getValue());
                else {
                     out.println("Email: " +
                     req.getParameter("email"));
                     if
                     (email.getValue().equals("Unknown"))
                     {
                          email.setValue(req.getParameter(
                          "email"));
                          res.addCookie(email);
                     }
                }
           }
           else {
                if (req.getParameter("name").length() ==
                0)
                     name = new Cookie("Name", "Unknown");
                else
                     name = new Cookie("Name",
                req.getParameter("name"));
                out.println("Name: " + name.getValue());
                if (req.getParameter("reg_no").length() ==
                0)
                     reg_no = new Cookie("Reg_No",
                "Unknown");
                else
                     reg_no = new Cookie("Reg_No",
                req.getParameter("reg_no"));
```

```
                    out.println("Registration Number: " +
                    reg_no.getValue());
                    if (req.getParameter("email").length()==0)
                         email = new Cookie("Email",
                    "Unknown");
                    else
                         email = new Cookie("Email",
                    req.getParameter("email"));
                    out.println("Email: " + email.getValue());
                    name.setMaxAge(60 * 60 * 24);
                    reg_no.setMaxAge(60 * 60 * 24);
                    email.setMaxAge(60 * 60 * 24);
                    res.addCookie(name);
                    res.addCookie(reg_no);
                    res.addCookie(email);
               }
               out.close();
          } catch (Exception e) {
               System.out.println(e);
          }
     }
}
```

Prog5.html:

```
<html>
   <body>
      <form action = "Prog5" method = "GET">
         Name: <input type = "text" name = "name">
         <br />
         Registration Number: <input type="text"
name="reg_no">
         <br />
         Email Id: <input type = "text" name = "email" />
         <input type = "submit" value = "Submit" />
      </form>
   </body>
</html>
```

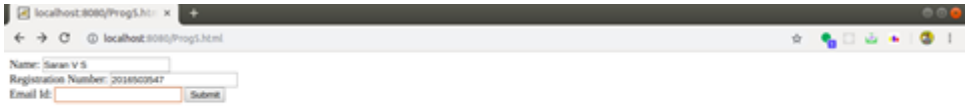
Deployment:

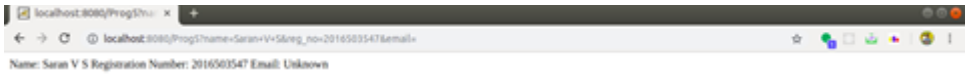
```
<servlet>
    <servlet-name>Prog5</servlet-name>
    <servlet-class>Prog5</servlet-class>
</servlet>

<servlet-mapping>
```

```
    <servlet-name>Prog5</servlet-name>
    <url-pattern>/Prog5</url-pattern>
</servlet-mapping>
```

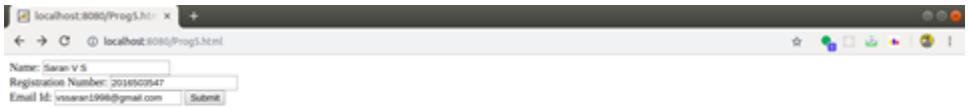
Output:

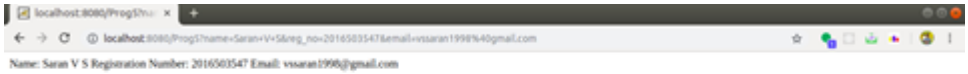
Incomplete Data Before Cookie Creation:



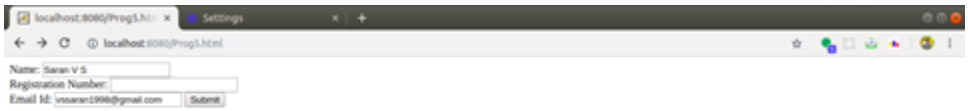


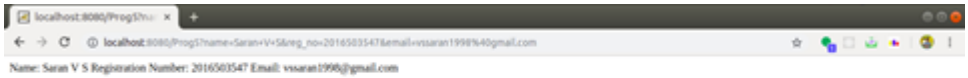
Complete Data for Cookie Creation:





Incomplete Data After cookie creation:





Result:

Thus the program is executed and output is obtained.

PROGRAM 6:

Aim:

To write a servlet program that shows all the request headers. Use a red background and a yellow foreground for Google Chrome users; use a yellow background and a red foreground for Firefox and other users.

Program:

Prog6.java:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

public class Prog6 extends HttpServlet{
     public void doGet(HttpServletRequest req,
     HttpServletResponse res) throws IOException{
          res.setContentType("text/html");
          String userAgent = req.getHeader("user-agent");
          PrintWriter out = res.getWriter();
          String htmlHeader = "<html> <head> <title> Servlet 6
          </title> </head>";
          out.println(htmlHeader);
          String htmlFooter = "</font> </body> </html>";
          if (userAgent.contains("Chrome")){
               out.println("<body bgcolor='red'
          text='yellow'>");
          }
          else{
               out.println("<body bgcolor='yellow'
          text='red'>");
          }
          out.println("<table align=center border=1 > <tr>
          <th> Header </th> <th> Value </th> </tr>");
          Enumeration e = req.getHeaderNames();
          while(e.hasMoreElements()){
               String header = (String)e.nextElement();
               if(header != null ){
                    out.println("<tr>");
                    out.println("<td> " + header + "</td>");
                    out.println("<td> " +
                    req.getHeader(header) + "</td>" );
                    out.println("</td>");
               }
          }
          out.println("</table>");
```

```
            out.println(htmlFooter);
        }
}
```

Deployment:

```
<servlet>
    <servlet-name>Prog6</servlet-name>
    <servlet-class>Prog6</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>Prog6</servlet-name>
    <url-pattern>/Prog6</url-pattern>
</servlet-mapping>
```

Output:

Google Chrome:

Mozilla Firefox:

Result:
Thus the program is executed and output is obtained.

PROGRAM 7:

Aim:
To write a servlet that returns a Bad Request error page(400) unless the user supplies email-id without @ symbol in the form.

Program:

Prog7.java:

```
import java.io.*;
import java.util.Random;
import javax.servlet.*;
import javax.servlet.http.*;

public class Prog7 extends HttpServlet {
```

```
     protected void doGet(HttpServletRequest req,
     HttpServletResponse res) throws IOException,
     ServletException {
          res.setContentType("text/html");
          PrintWriter out = res.getWriter();
          Random rand = new Random();
          String email = req.getParameter("email");
          if (email.indexOf("@") >= 0) {
               out.println("<html>");
               out.println("<head>");
               out.println("<title>Success page</title>");
               out.println("</head>");
               out.println("<body>");
               out.println("<h1>Success</h1>");
               out.println("</body>");
               out.println("</html>");
          } else {
               res.sendError(404, "Bad Request");
          }
          out.close();
     }
}
```

Prog7.html:

```
<html>
   <body>
      <form action = "Prog7" method = "GET">
         Email Id: <input type = "text" name = "email" />
         <input type = "submit" value = "Submit" />
      </form>
   </body>
</html>
```

Deployment:

```
<servlet>
    <servlet-name>Prog7</servlet-name>
    <servlet-class>Prog7</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>Prog7</servlet-name>
    <url-pattern>/Prog7</url-pattern>
</servlet-mapping>
```

Output:

Email Id with @ Symbol:

Email Id without @ Symbol:

Result:

Thus the program is executed and output is obtained.

PROGRAM 8:

Aim:

To write a servlet that redirects a new servlet (named errorServlet) if the user supplies email-id without @ symbol in the form.

Program:

Prog8.java:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
```

```
public class Prog8 extends HttpServlet{
     public void doPost(HttpServletRequest req,
     HttpServletResponse res) throws IOException {
          try {
               res.setContentType("text/html");
               String email = req.getParameter("userName");
               String pass = req.getParameter("passWord");
               if (email.contains("@")) {
                    RequestDispatcher rd =
                    req.getRequestDispatcher("Prog8Success");
                    rd.include(req, res);
               } else {
                    RequestDispatcher rd =
                    req.getRequestDispatcher("Prog8Error");
                    rd.include(req, res);
               }
          } catch (Exception e) {
               e.printStackTrace();
          }
     }
}
```

Prog8Error.java:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

public class Prog8Error extends HttpServlet {
     public void doPost(HttpServletRequest req,
HttpServletResponse res) throws IOException {
          res.setContentType("text/html");
          PrintWriter out = res.getWriter();
          out.println("The e-mail id is invalid.");
          out.println("To retry click the link below");
          out.println("<a href='/Prog8.html'>Login Page</a>");
     }
}
```

Prog8Success.java:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
public class Prog8Success extends HttpServlet {
```

```
     public void doPost(HttpServletRequest req,
     HttpServletResponse res) throws IOException {
          res.setContentType("text/html");
          PrintWriter out = res.getWriter();
          String email = req.getParameter("userName");
          out.println("Welcome " + email);
     }
}
```

Prog8.html:

```
<html>
    <head>
        <title>Prog 8 </title>
    </head>
    <body>
        <form action =  "Prog8" method = "post">
            Email-id:<input type="text"
name="userName"/><br><br>
            Password:<input type="password" name="passWord"/>
<br><br>
            <input type="submit" value="login"/>
        </form>
    </body>
</html>
```

Deployment:

```
<!-- Program 8 - Request Dispatcher -->

<servlet>
    <servlet-name>Prog8</servlet-name>
    <servlet-class>Prog8</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>Prog8</servlet-name>
    <url-pattern>/Prog8</url-pattern>
</servlet-mapping>

<!-- Program 8 Error - Request Dispatcher -->

<servlet>
    <servlet-name>Prog8Error</servlet-name>
    <servlet-class>Prog8Error</servlet-class>
</servlet>

<servlet-mapping>
```

```
    <servlet-name>Prog8Error</servlet-name>
    <url-pattern>/Prog8Error</url-pattern>
</servlet-mapping>

<!-- Program 8 Success - Request Dispatcher -->

<servlet>
    <servlet-name>Prog8Success</servlet-name>
    <servlet-class>Prog8Success</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>Prog8Success</servlet-name>
    <url-pattern>/Prog8Success</url-pattern>
</servlet-mapping>
```

Output:

Email ID with @ Symbol:

Email ID without @ Symbol:

Result:

Thus the program is executed and output is obtained.

Experiment No: 7 **Date: 15/02/2019**

PROGRAMMING WITH AJAX, JQUERY AND JSON

PROGRAM 1:

Aim:
To create a DTD for a catalog of cars, here each car has the cild elements make, model, year, color, engine, number_of_doors, transmission_type, and accessories. The engine element has the child elements number_of_cylinders and fuel_system (carburated or fuel injected). The accessories element has the attributes radio, air_conditioning, power_windows, power_steering, and power_brakes, each of which is required and has the possible values yes and no. Entities must be declared for the names of popular car models.

Program:

```
<!ELEMENT catalog (car)>
<!ELEMENT car (make,model, year, color, engine,
number_of_doors, transmission_type, accessories)>
<!ELEMENT make (#PCDATA)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT color (#PCDATA)>
<!ELEMENT engine (number_of_cylinders,fuel_system)>
<!ELEMENT number_of_cylinders (#PCDATA)>
<!ELEMENT fuel_system (#PCDATA)>
<!ELEMENT number_of_doors (#PCDATA)>
<!ELEMENT transmission_type (#PCDATA)>
<!ELEMENT accessories (#PCDATA)>
<!ATTLIST accessories radio (yes|no)>
<!ATTLIST accessories air_conditioning (yes|no)>
<!ATTLIST accessories power_windows (yes|no)>
<!ATTLIST accessories power_steering (yes|no)>
<!ATTLIST accessories power_brakes (yes|no)>
```

Result:
Thus the program is executed and output is obtained.

PROGRAM 2:

Aim:
To create an XML document with atleast three instances of the car element defined in the DTD of Program 1. Process the document by using the DTD of Program 1, and produce a display of raw XML document.

Program:

```
<?xml version="1.0"?>
<!DOCTYPE catalog SYSTEM "prog1.dtd">
<?xml-stylesheet type="text/xsl" href="prog2.xsl"?>
<catalog>
    <car>
        <year>1998</year>
        <make>Indian</make>
        <model>BMW</model>
        <color>Blue</color>
        <engine>
            <number_of_cylinders>6</number_of_cylinders>
            <fuel_system>fuel_injected</fuel_system>
        </engine>
        <transmission_type>auto</transmission_type>
        <number_of_doors>4</number_of_doors>
        <accessories radio = "yes" air_conditioning = "no"
power_windows = "yes" power_steering = "no" power_brakes =
"yes"></accessories>
    </car>

    <car>
        <make>English</make>
        <model>Benz</model>
        <year>1912</year>
        <color>Black</color>
        <engine>
            <number_of_cylinders>8</number_of_cylinders>
            <fuel_system>carburated</fuel_system>
        </engine>
        <number_of_doors>2</number_of_doors>
        <transmission_type>manual</transmission_type>
        <accessories radio = "yes" air_conditioning = "no"
power_windows = "yes" power_steering = "no" power_brakes =
"yes"></accessories>
    </car>

    <car>
        <make>German</make>
        <model>Volkswagon</model>
        <year>1957</year>
        <color>Red</color>
        <engine>
            <number_of_cylinders>4</number_of_cylinders>
            <fuel_system>carburated</fuel_system>
```

```
        </engine>
        <number_of_doors>6</number_of_doors>
        <transmission_type>auto</transmission_type>
        <accessories radio = "yes" air_conditioning = "no"
power_windows = "yes" power_steering = "no" power_brakes =
"yes"></accessories>
    </car>
</catalog>
```

Output:

Raw Display:

Styled Display:

Result:

Thus the program is executed and output is obtained.

PROGRAM 3:

Aim:

To modify the example application of Program 1 to allow the user to select a make and model of used cars. The make must be

in a menu. When a make is chosen, a meny of models must be displayed. This menu is produced by hardwired data in the original document. When a model is chosen, an AJAX request must be made to get a list of the years and colors of the chosen make and model that are available. Make up a server-resident script to produce the data from an example array or hash.

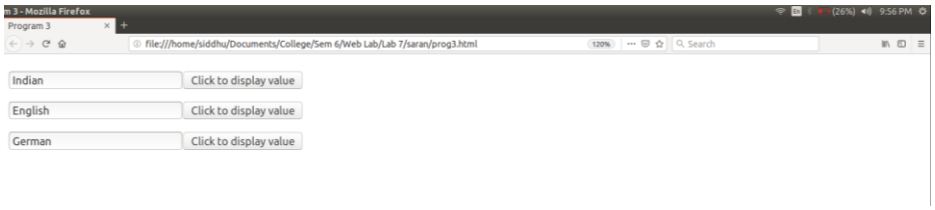
Program:

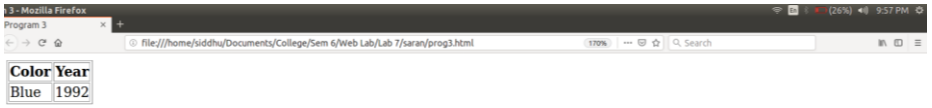
```
<html>
    <head>
        <title> Program 3 </title>
        <style>
            #display{
                visibility: hidden;
            }
        </style>
        <script>
                function getData(){
                    var xml_load1 = new XMLHttpRequest();
                    xml_load1.onreadystatechange =function(){
                    if(this.readyState==4&&this.status==200){
                            getMakeElement(this);
                        }
                    };
                    xml_load1.open("GET","index.xml",true);
                    xml_load1.send();
                }
                function getMakeElement(xml){
                    var xmlDoc = xml.responseXML;
                    var content = "";
                    var x = xmlDoc.getElementsByTagName(
"make");
                    for(i = 0;i < x.length;i++){
                        var name = "button"+i.toString();
                        var id1 = "value"+i.toString();
                        content += "<br> <input type = 'text'"
+ "value='" + x[i].childNodes[0].nodeValue + "' id=" + id1+ "
/><button id="+name+" onclick =
displayUtil('"+x[i].childNodes[0].nodeValue+ "')> Click to
display value </button><br>";
                    }
                    document.getElementById("list").innerHTML
= content;
                }
```

```
                function displayUtil(name){
                     document.getElementById("list").style.visi
                bility="hidden";
                     document.getElementById("list").style.disp
                lay = "none";
                     console.log(name);
                     getContentData(name)
                }

                function getContentData(data){
                     var xml_load1 = new XMLHttpRequest();
                     xml_load1.onreadystatechange =function(){
                         if(this.readyState == 4 && this.status
== 200){
                              loadData(this,data);
                         }
                     };
                     xml_load1.open("GET","index.xml",true);
                     xml_load1.send();
                }

                function loadData(xml,data){
                     var xmlDoc = xml.responseXML;
                     var content = "<table
border=1><tr><th>Color</th><th>Year</th></tr>";
                     var x =
xmlDoc.getElementsByTagName("car");
                     for(var i = 0;i<x.length;i++){

if(x[i].getElementsByTagName("make")[0].childNodes[0].nodeValu
e == data){
                              content+="<tr><td>"+
x[i].getElementsByTagName("color")[0].childNodes[0].nodeValue
+ "</td><td>"+
x[i].getElementsByTagName("year")[0].childNodes[0].nodeValue +
"</td></tr>";
                         }
                     }
                     content+="</table>";
                     document.getElementById("display").innerHT
                ML = content;

                document.getElementById("display").style.visibi
                lity = "visible";
                 }

            </script>
```

```
    </head>
    <body onload = "getData()">
        <div id="list">
        </div>
        <div id = "display">
        </div>
    </body>
</html>
```

Output:





Result:
Thus the program is executed and output is obtained.

PROGRAM 4:

Aim:
To modify the example applicatio of Program 1 to have it provide the addresses of repeat customers, using a hash of names and addresses.

Program:

Prog4.dtd:

```
<!ELEMENT catalog (car,customers)>
<!ELEMENT car
(make,model,year,color,engine,number_of_doors,transmission_typ
e,accessories)>
<!ELEMENT make (#PCDATA)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT color (#PCDATA)>
<!ELEMENT engine (number_of_cylinders,fuel_system)>
<!ELEMENT number_of_cylinders (#PCDATA)>
<!ELEMENT fuel_system (#PCDATA)>
<!ELEMENT number_of_doors (#PCDATA)>
<!ELEMENT transmission_type (#PCDATA)>
<!ELEMENT accessories (#PCDATA)>
<!ATTLIST accessories radio (yes|no)>
<!ATTLIST accessories air_conditioning (yes|no)>
<!ATTLIST accessories power_windows (yes|no)>
<!ATTLIST accessories power_steering (yes|no)>
<!ATTLIST accessories power_brakes (yes|no)>
<!ELEMENT customers (id,name,address,count)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT address (city,state,zipcode)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT catalog (#PCDATA)>
<!ELEMENT catalog (#PCDATA)>
<!ELEMENT count (#PCDATA)>
```

Prog4.xml:

```
<?xml version="1.0"?>
<!DOCTYPE catalog SYSTEM "prog4.dtd">
<catalog>
    <car>
        <year>1998</year>
        <make>Indian</make>
        <model>BMW</model>
        <color>Blue</color>
        <engine>
            <number_of_cylinders>6</number_of_cylinders>
            <fuel_system>fuel_injected</fuel_system>
        </engine>
        <transmission_type>auto</transmission_type>
        <number_of_doors>4</number_of_doors>
```

```
        <accessories radio = "yes" air_conditioning = "no"
power_windows = "yes" power_steering = "no" power_brakes =
"yes"></accessories>
    </car>

    <car>
        <make>English</make>
        <model>Benz</model>
        <year>1912</year>
        <color>Black</color>
        <engine>
            <number_of_cylinders>8</number_of_cylinders>
            <fuel_system>carburated</fuel_system>
        </engine>
        <number_of_doors>2</number_of_doors>
        <transmission_type>manual</transmission_type>
        <accessories radio = "yes" air_conditioning = "no"
power_windows = "yes" power_steering = "no" power_brakes =
"yes"></accessories>
    </car>

    <car>
        <make>German</make>
        <model>Volkswagon</model>
        <year>1957</year>
        <color>Red</color>
        <engine>
            <number_of_cylinders>4</number_of_cylinders>
            <fuel_system>carburated</fuel_system>
        </engine>
        <number_of_doors>6</number_of_doors>
        <transmission_type>auto</transmission_type>
        <accessories radio = "yes" air_conditioning = "no"
power_windows = "yes" power_steering = "no" power_brakes =
"yes"></accessories>
    </car>

    <customer>
        <id>1</id>
        <name>Saran</name>
        <address>
            <city>Chennai</city>
            <state>TN</state>
            <zip>600073</zip>
        </address>
    </customer>
```

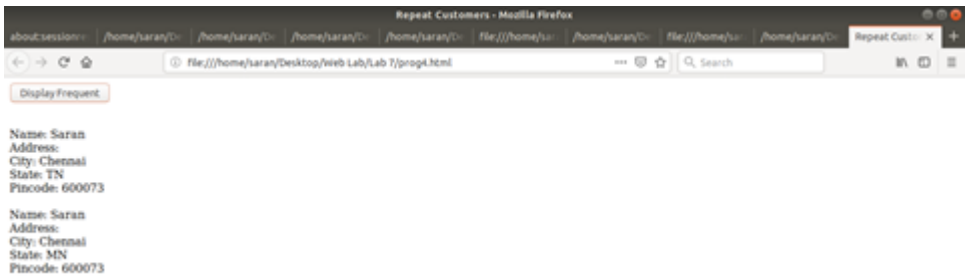
```
    <customer>
        <id>2</id>
        <name>Saran</name>
        <address>
            <city>Chennai</city>
            <state>MN</state>
            <zip>600073</zip>
        </address>
    </customer>
</catalog>
```

Prog4.html:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Repeat Customers</title>
    <meta name="viewport" content="width=device-width,
initial-scale=1">
</head>
<body>
    <input type="button" value="Display Frequent"
onclick=dispData()>
    <br>
    <br>
    <p id="demo"></p>

    <script>
        function dispData() {
            var xhttp = new XMLHttpRequest();
            xhttp.onreadystatechange = function() {
                if(this.readyState == 4 && this.status == 200)
                {
                    dispXML(this);
                }
            };
            xhttp.open("GET","prog4.xml",true);
            xhttp.send();
        }

        function dispXML(xml) {
            var xmlDoc = xml.responseXML;
            var x = xmlDoc.getElementsByTagName("customer");
            var content = "";
            alert(x.length);
```

```
            for (var i = 0; i < x.length; i++) {
                console.log(i);
                content += "Name: " +
x[i].getElementsByTagName("name")[0].childNodes[0].nodeValue +
" <br>";
                content += "Address: <br>"

                var y = x[i].getElementsByTagName("address");

                for (var j = 0; j < y.length; j++) {
                    content += "City: " +
y[j].getElementsByTagName("city")[0].childNodes[0].nodeValue +
" <br>";
                    content += "State: " +
y[j].getElementsByTagName("state")[0].childNodes[0].nodeValue
+ " <br>";
                    content += "Pincode: " +
y[j].getElementsByTagName("zip")[0].childNodes[0].nodeValue +
" <br>";
                }
                content += "<br>"
            }
            document.getElementById("demo").innerHTML =
content

        }
    </script>

</body>
</html>
```

Output:



Result:

Thus the program is executed and output is obtained.

PROGRAM 5:

Aim:

To modify the example application of Program 4 with validate the zip code when it is entered, to ensure that it is a valid zip code for the given city and state. The response document can be a PHP script that looks up the zip code and the city and state in a small table of examples.

Program:

Prog5.html:

```
<html>
    <head>
        <title>Program 5</title>
        <meta charset="utf-8">
        <script>
            function myfunc(){
                var x = document.getElementById("city").value;
                var y =
document.getElementById("state").value;
                var z = document.getElementById("zipcode")
.value;
                console.log(x);
                var req = x + " " + y + " " + z;
                var xml_load1 = new XMLHttpRequest();
                xml_load1.onreadystatechange = function(){
                    if(this.readyState == 4 && this.status ==
200){
                        alert(this.responseText);
                    }
                };

xml_load1.open("GET","prog5.php?val="+req,true);
                xml_load1.send();
            }
        </script>
    </head>
    <body>
        <form>
            City:<input id="city" type="text" value=""
/><br><br>
```

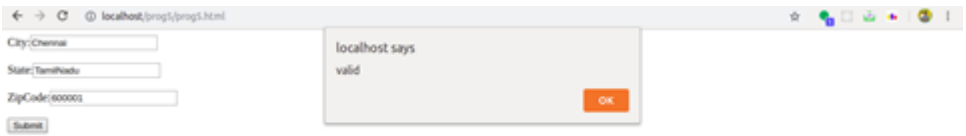
```
            State:<input id="state" type="text" value=""
/><br><br>
            ZipCode:<input id="zipcode" type="text" value=""
/><br><br>
            <input type="button" value="Submit"
onclick=myfunc() />
        </form>
    </body>
</html>
```

Prog5.php:

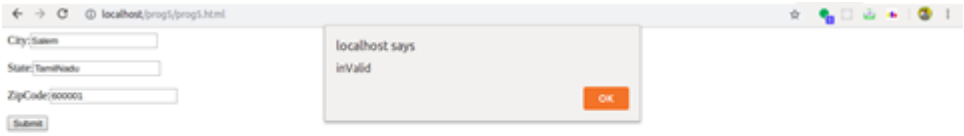
```
<?php

$address[] = "Chennai TamilNadu 600001";
$address[] = "Salem TamilNadu 600002";
$address[] = "Madurai TamilNadu 600003";

$val = $_REQUEST["val"];
$fl = 0;
for($i=0;$i<sizeof($address);$i++){
    if($address[$i] == $val){
        $fl = 1;
        break;
    }
}
if($fl==0){
    echo 'inValid';
}
else{
    echo 'valid';
}

?>
```

Output:

Valid Case:



Invalid Case:



Result:

Thus the program is executed and output is obtained.

Experiment No: 8 **Date: 22/02/2019**

XML, DTD, PARSERS, XSLT. XPATH, SAX

PROGRAM 1:

Aim:

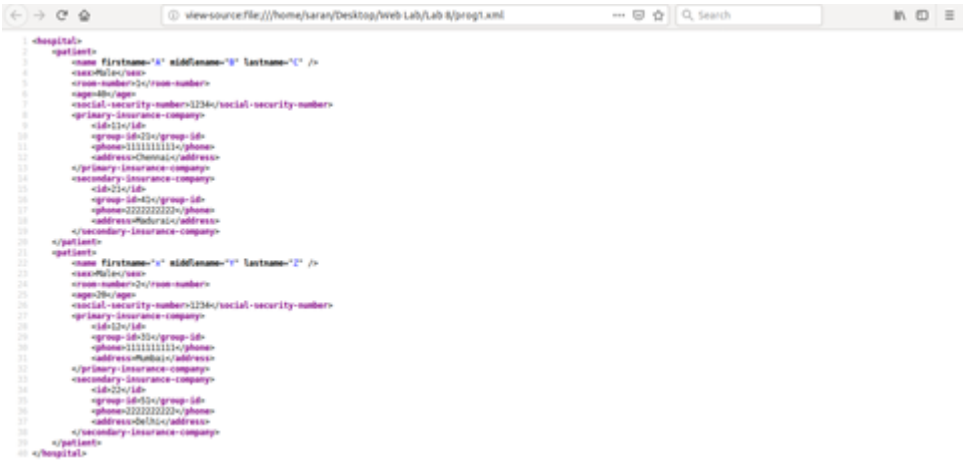
To design an XML document that stores information about patients in a hospital. Information about parents must include their name (in three parts), Social Security Number, age, room number, primary insurance company – including member identification number, group number, phone number, and address – secondary insurance company (wit same parts as the primary insurance company has), known medical problems, and known drug allergies. Both attributes and nested tags must be included. Make up a sample data for at least four patients.

Program:

```
<hospital>
    <patient>
        <name firstname="A" middlename="B" lastname="C" />
        <sex>Male</sex>
        <room-number>1</room-number>
        <age>40</age>
        <social-security-number>1234</social-security-number>
        <primary-insurance-company>
            <id>11</id>
            <group-id>21</group-id>
            <phone>1111111111</phone>
            <address>Chennai</address>
        </primary-insurance-company>
        <secondary-insurance-company>
            <id>21</id>
            <group-id>41</group-id>
            <phone>2222222222</phone>
            <address>Madurai</address>
        </secondary-insurance-company>
    </patient>
    <patient>
        <name firstname="x" middlename="Y" lastname="Z" />
        <sex>Male</sex>
        <room-number>2</room-number>
        <age>20</age>
        <social-security-number>1234</social-security-number>
        <primary-insurance-company>
            <id>12</id>
            <group-id>31</group-id>
            <phone>1111111111</phone>
            <address>Mumbai</address>
        </primary-insurance-company>
```

```
        <secondary-insurance-company>
            <id>22</id>
            <group-id>51</group-id>
            <phone>2222222222</phone>
            <address>Delhi</address>
        </secondary-insurance-company>
    </patient>
</hospital>
```

Output:



Result:
Thus the program is executed and output is obtained.

PROGRAM 2:

Aim:
To write a DTD for the document described in Program 1, but with the following restrictions: name, Social Security number, age, room number, and primary insurance company are required. All the other elements are optional, as are middle names.

Program:

```
<!ELEMENT hospital (patient)>
<!ELEMENT patient (name,sex,room-number,age,social-security-
number,primary-insurance-company,secondary-insurance-company)>
<!ELEMENT name (#PCDATA)>
<!ATTLIST firstname #REQUIRED>
<!ATTLIST middlename >
```

```
<!ATTLIST lastname #REQUIRED>
<!ELEMENT age(#PCDATA) #REQUIRED>
<!ELEMENT sex(#PCDATA) #REQUIRED>
<!ELEMENT room-number (#PCDATA) #REQUIRED>
<!ELEMENT social-security-number(#PCDATA) #REQUIRED>
<!ELEMENT primary-insurance-company (id,group-
id,phone,address) #REQUIRED>
<!ELEMENT  id (#PCDATA>
<!ELEMENT  group-id (#PCDATA>
<!ELEMENT  phone (#PCDATA>
<!ELEMENT  address (#PCDATA>
<!ELEMENT secondary-insurance-company (id,group-
id,phone,address)>
```

Result:
Thus the program is executed and output is obtained.

PROGRAM 3:

Aim:
To create a CSS style sheet for the XML document of program 1 and use it to create a display of the document.

Program:

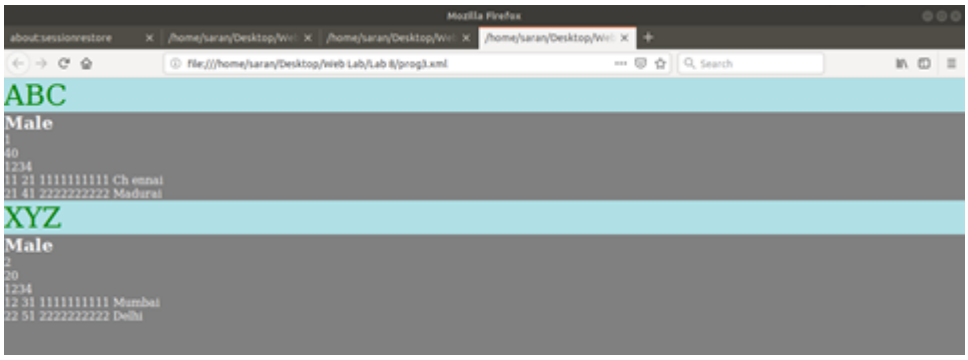
Prog3.css:

```
hospital {
    color: white;
    background-color : gray;
    width: 100%;
}
name {
    color: green;
    font-size : 40px;
    background-color : powderblue;
}
name,sex, room-number, age, social-security-number, primary-
insurance-company, secondary-insurance-company {
    display : block;
}
sex {
    font-size : 25px;
    font-weight : bold;
}
```

Prog3.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="prog3.css" ?>
<hospital>
    <patient>
        <name firstname="A" middlename="B"
lastname="C">ABC</name>
        <sex>Male</sex>
        <room-number>1</room-number>
        <age>40</age>
        <social-security-number>1234</social-security-number>
        <primary-insurance-company>
            <id>11</id>
            <group-id>21</group-id>
            <phone>1111111111</phone>
            <address>Ch ennai</address>
        </primary-insurance-company>
        <secondary-insurance-company>
            <id>21</id>
            <group-id>41</group-id>
            <phone>2222222222</phone>
            <address>Madurai</address>
        </secondary-insurance-company>
    </patient>
    <patient>
        <name firstname="x" middlename="Y"
lastname="Z">XYZ</name>
        <sex>Male</sex>
        <room-number>2</room-number>
        <age>20</age>
        <social-security-number>1234</social-security-number>
        <primary-insurance-company>
            <id>12</id>
            <group-id>31</group-id>
            <phone>1111111111</phone>
            <address>Mumbai</address>
        </primary-insurance-company>
        <secondary-insurance-company>
            <id>22</id>
            <group-id>51</group-id>
            <phone>2222222222</phone>
            <address>Delhi</address>
        </secondary-insurance-company>
    </patient>
</hospital>
```

Output:



Result:
Thus the program is executed and output is obtained.

PROGRAM 4:

Aim:
To create an XSLT stylesheet for one patient element of the XML document of program 1 and use it to create a display of that element.

Program:

Prog4.xsl:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body bgcolor="purple">
  <h2 align="center" style="color:blue font-
family:helvetica">Hospital Patient History</h2>
  <table border="1" align="center">
    <tr bgcolor="green">
      <th style="text-align:left font-size:300%">FName</th>
      <th style="text-align:left font-size:300%">MName</th>
      <th style="text-align:left font-size:300%">LName</th>
      <th style="text-align:left font-size:300%">Sex</th>
      <th style="text-align:left font-size:300%">Room-
number</th>
      <th style="text-align:left font-size:300%">Age</th>
```

```
      <th style="text-align:left font-size:300%">Social
Security Number</th>
      <th style="text-align:left font-size:300%">SIC:ID</th>
      <th style="text-align:left font-
size:300%">SIC:GrpID</th>
      <th style="text-align:left font-
size:300%">SIC:Phone</th>
      <th style="text-align:left font-
size:300%">SIC:Address</th>
      <th style="text-align:left font-size:300%">SIC:ID</th>
      <th style="text-align:left font-
size:300%">SIC:GrpID</th>
      <th style="text-align:left font-
size:300%">SIC:Phone</th>
      <th style="text-align:left font-
size:300%">SIC:Address</th>
    </tr>
    <xsl:for-each select="hospital/patient[name='ABC']">
    <tr>
      <td><xsl:value-of select="name/@firstname"/></td>
      <td><xsl:value-of select="name/@middlename"/></td>
      <td><xsl:value-of select="name/@lastname"/></td>
      <td><xsl:value-of select="sex"/></td>
      <td><xsl:value-of select="room-number"/></td>
      <td><xsl:value-of select="age"/></td>
      <td><xsl:value-of select="social-security-number"/></td>
      <td><xsl:value-of select="primary-insurance-
company/id"/></td>
      <td><xsl:value-of select="primary-insurance-
company/group-id"/></td>
      <td><xsl:value-of select="primary-insurance-
company/phone"/></td>
      <td><xsl:value-of select="primary-insurance-
company/address"/></td>
      <td><xsl:value-of select="secondary-insurance-
company/id"/></td>
      <td><xsl:value-of select="secondary-insurance-
company/group-id"/></td>
      <td><xsl:value-of select="secondary-insurance-
company/phone"/></td>
      <td><xsl:value-of select="secondary-insurance-
company/address"/></td>
    </tr>
    </xsl:for-each>
  </table>
</body>
</html>
```

```
</xsl:template>
</xsl:stylesheet>
```

Prog4.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="prog4.xsl" ?>
<hospital>
    <patient>
        <name firstname="A" middlename="B"
lastname="C">ABC</name>
        <sex>Male</sex>
        <room-number>1</room-number>
        <age>40</age>
        <social-security-number>1234</social-security-number>
        <primary-insurance-company>
            <id>11</id>
            <group-id>21</group-id>
            <phone>1111111111</phone>
            <address>Chennai</address>
        </primary-insurance-company>
        <secondary-insurance-company>
            <id>21</id>
            <group-id>41</group-id>
            <phone>2222222222</phone>
            <address>Madurai</address>
        </secondary-insurance-company>
    </patient>
    <patient>
        <name firstname="x" middlename="Y"
lastname="Z">XYZ</name>
        <sex>Male</sex>
        <room-number>2</room-number>
        <age>20</age>
        <social-security-number>1226</social-security-number>
        <primary-insurance-company>
            <id>12</id>
            <group-id>31</group-id>
            <phone>1111111111</phone>
            <address>Mumbai</address>
        </primary-insurance-company>
        <secondary-insurance-company>
            <id>22</id>
            <group-id>51</group-id>
            <phone>2222222222</phone>
            <address>Delhi</address>
        </secondary-insurance-company>
```

```
    </patient>
</hospital>
```

Output:

Result:
Thus the program is executed and output is obtained.

PROGRAM 5:

Aim:
To modify the XSLT stylesheet of Program 4 so that it formats all the patient elements in the XML document of Program 1 abd use the stylesheet to create a display of the whole document.

Program:

Prog5.xsl:

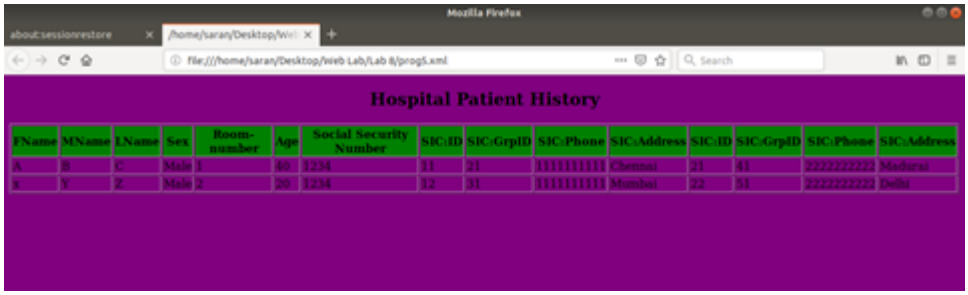
```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body bgcolor="purple">
  <h2 align="center" style="color:blue font-
family:helvetica">Hospital Patient History</h2>
  <table border="1" align="center">
    <tr bgcolor="green">
      <th style="text-align:left font-size:300%">FName</th>
      <th style="text-align:left font-size:300%">MName</th>
      <th style="text-align:left font-size:300%">LName</th>
      <th style="text-align:left font-size:300%">Sex</th>
      <th style="text-align:left font-size:300%">Room-
number</th>
      <th style="text-align:left font-size:300%">Age</th>
      <th style="text-align:left font-size:300%">Social
Security Number</th>
```

```
      <th style="text-align:left font-size:300%">SIC:ID</th>
      <th style="text-align:left font-
size:300%">SIC:GrpID</th>
      <th style="text-align:left font-
size:300%">SIC:Phone</th>
      <th style="text-align:left font-
size:300%">SIC:Address</th>
      <th style="text-align:left font-size:300%">SIC:ID</th>
      <th style="text-align:left font-
size:300%">SIC:GrpID</th>
      <th style="text-align:left font-
size:300%">SIC:Phone</th>
      <th style="text-align:left font-
size:300%">SIC:Address</th>
    </tr>
    <xsl:for-each select="hospital/patient">
    <tr>
      <td><xsl:value-of select="name/@firstname"/></td>
      <td><xsl:value-of select="name/@middlename"/></td>
      <td><xsl:value-of select="name/@lastname"/></td>
      <td><xsl:value-of select="sex"/></td>
      <td><xsl:value-of select="room-number"/></td>
      <td><xsl:value-of select="age"/></td>
      <td><xsl:value-of select="social-security-number"/></td>
      <td><xsl:value-of select="primary-insurance-
company/id"/></td>
      <td><xsl:value-of select="primary-insurance-
company/group-id"/></td>
      <td><xsl:value-of select="primary-insurance-
company/phone"/></td>
      <td><xsl:value-of select="primary-insurance-
company/address"/></td>
      <td><xsl:value-of select="secondary-insurance-
company/id"/></td>
      <td><xsl:value-of select="secondary-insurance-
company/group-id"/></td>
      <td><xsl:value-of select="secondary-insurance-
company/phone"/></td>
      <td><xsl:value-of select="secondary-insurance-
company/address"/></td>
    </tr>
    </xsl:for-each>
  </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Prog5.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="prog5.xsl" ?>
<hospital>
    <patient>
        <name firstname="A" middlename="B"
lastname="C">ABC</name>
        <sex>Male</sex>
        <room-number>1</room-number>
        <age>40</age>
        <social-security-number>1234</social-security-number>
        <primary-insurance-company>
            <id>11</id>
            <group-id>21</group-id>
            <phone>1111111111</phone>
            <address>Chennai</address>
        </primary-insurance-company>
        <secondary-insurance-company>
            <id>21</id>
            <group-id>41</group-id>
            <phone>2222222222</phone>
            <address>Madurai</address>
        </secondary-insurance-company>
    </patient>
    <patient>
        <name firstname="x" middlename="Y"
lastname="Z">XYZ</name>
        <sex>Male</sex>
        <room-number>2</room-number>
        <age>20</age>
        <social-security-number>1234</social-security-number>
        <primary-insurance-company>
            <id>12</id>
            <group-id>31</group-id>
            <phone>1111111111</phone>
            <address>Mumbai</address>
        </primary-insurance-company>
        <secondary-insurance-company>
            <id>22</id>
            <group-id>51</group-id>
            <phone>2222222222</phone>
            <address>Delhi</address>
        </secondary-insurance-company>
    </patient>
</hospital>
```

Output:



Result:

Thus the program is executed and output is obtained.