

**CS7612 WEB TECHNOLOGY LAB**

**Y Manaswini 2016503555**

**Date – 14/12/18**

**Experiment – 1**

# **SOCKET PROGRAMMING**

**PROGRAM – 1****AIM :**

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

**PROGRAM :**

```
import java.io.*;
import java.net.*;
import java.util.*;
public class Ping
{
    public static void main(String[] args) throws IOException
    {
        Scanner sc = new Scanner(System.in);
        int n = 5;

        for (int i = 0; i < n; i++)
        {
            System.out.print("Enter IP Address : ");
            String ip = sc.next();
            InetAddress inet = InetAddress.getByName(ip);
            System.out.println("Sending ping request to... " + inet);

            if (inet.isReachable(5000))
                System.out.println("Host Reachable ");
            else
                System.out.println("Host NOT Reachable ");
        }
    }
}
```

**OUTPUT:**

```
Enter IP Address : 192.168.0.1
Sending ping request to... /192.168.0.1
Host Reachable
Enter IP Address : 192.168.0.101
Sending ping request to... /192.168.0.101
Host NOT Reachable
```

**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 :****Client.java:**

```
import java.io.*;
import java.net.*;

public class DateTimeClient
{
    public static void main(String args[]) throws Exception
    {
        Socket soc=new
Socket(InetAddress.getLocalHost(),9876);

        System.out.println("Connection established\n");

        BufferedReader in=new BufferedReader(new
InputStreamReader(soc.getInputStream()  ));
        System.out.println(in.readLine());
    }
}
```

**Server.java:**

```
import java.net.*;
import java.io.*;
import java.util.*;

class DateTimeServer
{
    public static void main(String args[]) throws Exception
    {
        ServerSocket s=new ServerSocket(9876);
        System.out.println("Server running on port 9876...\"");
        System.out.println("Waiting For Connection ...");

        Socket soc=s.accept();
        System.out.println("Client Connected");

        DataOutputStream out=new
DataOutputStream(soc.getOutputStream());
    }
}
```

```
        out.writeBytes("Server Date: " + (new
Date()).toString() + "\n");
        System.out.println("Data Sent");
        System.out.println("Server Disconnecting");
        out.close();
        soc.close();
    }
}
```

**OUTPUT:**

Server	Client
Server running on port 9876	Connection established
Waiting For Connection ...	
Client Connected	Server Date: Sun Feb 24
Data Sent	12:49:52 IST 2019
Server Disconnecting	

**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 :****Client.java:**

```
import java.io.*;
import java.util.*;
import java.net.*;

public class PingPongClient
{
    public static void main(String args[])
    {
        try
        {
            Socket socket = new Socket("localhost", 9876);
            System.out.println("Connected to server");

            DataInputStream in=new
                DataInputStream(socket.getInputStream());
            DataOutputStream out=new
                DataOutputStream(socket.getOutputStream());

            Scanner sc = new Scanner(System.in);

            while(true)
            {
                System.out.print("Enter message for server :
");
                String msg = sc.nextLine();

                out.writeUTF(msg);
                if(msg.equals("Ping"))
                    System.out.println("Message Received : "
                        + in.readUTF());

                if(msg.equals("exit"))
                    break;
            }
            System.out.println("Server disconnected");
        }
    }
}
```

```

        out.close();
        socket.close();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}
}

```

**Server.java:**

```

import java.io.*;
import java.net.*;

public class PingPongServer
{
    public static void main(String args[])
    {
        try
        {
            ServerSocket server = new ServerSocket(9876);
            System.out.println("Server running in port
                                9876...");
            System.out.println("Waiting for client to
                                connect...");

            Socket socket=server.accept();
            System.out.println("Client connected\n\n");

            DataInputStream in=new DataInputStream(new
                BufferedInputStream(socket.getInputStream()));
            DataOutputStream out=new
                DataOutputStream(socket.getOutputStream());

            while(true)
            {
                String msg = in.readUTF();
                if(msg.equals("Ping"))
                {
                    System.out.println("Replies for Ping
                                message");
                    out.writeUTF("Pong");
                }
                if(msg.equals("exit"))
                    break;
            }
        }
        catch(Exception e)
    }
}

```

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

**OUTPUT:**

Server	Client
Server running in port 9876...	Connected to server
Waiting for client to connect...	Enter message for server : Hello
Client connected	Enter message for server : Ping
Message Recieved : Hello	Message Recieved : Pong
	Enter message for server : exit
Replies for Ping message	
Server exiting	

**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. Write both client and server PROGRAMs demonstrating this.

**PROGRAM :****Client.java:**

```

import java.io.*;
import java.util.*;
import java.net.*;

public class UpperChatClient
{
    public static void main(String args[])
    {
        try
        {
            Socket socket = new Socket("localhost", 9876);
            System.out.println("Connected to server");

            DataInputStream in=new
                DataInputStream(socket.getInputStream());
            DataOutputStream out=new
                DataOutputStream(socket.getOutputStream());

            Scanner sc = new Scanner(System.in);

            while(true)
            {
                System.out.print("Enter message for server :
                    ");
                String msg = sc.nextLine();

                out.writeUTF(msg);

                System.out.println("Message Received : " +
                    in.readUTF());

                if(msg.equals("exit"))
                    break;
            }
            System.out.println("Server disconnected");
        }
    }
}

```

```

        out.close();
        socket.close();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}
}

```

**Server.java:**

```

import java.io.*;
import java.net.*;

public class UpperChatServer
{
    public static void main(String args[])
    {
        try
        {
            ServerSocket server = new ServerSocket(9876);
            System.out.println("Server running in port 9876...");
            System.out.println("Waiting for client to
                               connect...");

            Socket socket=server.accept();
            System.out.println("Client connected\n\n");

            DataInputStream in=new DataInputStream(new
                BufferedInputStream(socket.getInputStream()));
            DataOutputStream out=new
                DataOutputStream(socket.getOutputStream());

            while(true)
            {
                String msg = in.readUTF();

                System.out.println("Message received : "+msg);
                out.writeUTF(msg.toUpperCase());

                if(msg.equals("exit"))
                    break;
            }
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}

```

```
    }  
}
```

**OUTPUT:**

Server	Client
Server running in port 9876...	Connected to server
Waiting for client to connect...	Enter message for server : Hello
Client connected	Message Recieved : HELLO
Message recieved : Hello	Enter message for server : Voila
Message recieved : Voila	Message Recieved : VOILA
Message recieved : I'm so cool	Enter message for server : I'm so cool
Message recieved : exit	Message Recieved : I'M SO COOL
	Enter message for server : exit
	Message Recieved : EXIT
	Server disconnected

**RESULT :**

Thus the PROGRAM is executed and output is obtained.

**PROGRAM — 5****AIM:**

To implement Broadcasting messages in your lab.

**PROGRAM :****BroadcastServer.java**

```

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();
        }
    }
}

```

**BroadcastClient.java**

```

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);  
            //System.out.println(buf.toString());  
            if(s1.equals("exit"))  
                break;  
            System.out.println(s1);  
        }  
        ds.close();  
    }  
    catch(Exception e){  
        e.printStackTrace();  
    }  
}
```

## **OUTPUT :**

Server	Client
Enter the message to be broadcasted..	Hello World Ping
Hello World	
Enter the message to be broadcasted..	
Ping	
Enter the message to be broadcasted..	
exit	

## **RESULT :**

Thus the PROGRAM is executed and output is obtained.

**Experiment - 2**

# **REMOTE METHOD INVOCATION ( RMI )**

**PROGRAM – 1****AIM :**

To write a Java PROGRAM to impleent Client Server communication using RPC.

**PROGRAM :****Client.java:**

```

import java.io.*;
import java.net.*;

class client1 {
    public static void main(String[] args) throws Exception {
        Socket sock = new Socket("127.0.0.1", 3000);
        BufferedReader keyRead = new BufferedReader(new
                InputStreamReader(System.in));
        OutputStream ostream = sock.getOutputStream();
        PrintWriter pwrite = new PrintWriter(ostream, true);
        InputStream istream = sock.getInputStream();
        BufferedReader receiveRead = new BufferedReader(new
                InputStreamReader(istream));
        System.out.println("Client ready, type and press Enter
key");
        String receiveMessage, sendMessage, temp;
        while (true) {
            System.out.println("\nEnter operation to
                                perform(add,sub,mul,div)....");
            temp = keyRead.readLine();
            sendMessage = temp.toLowerCase();
            pwrite.println(sendMessage);
            System.out.println("Enter first parameter : ");
            sendMessage = keyRead.readLine();
            pwrite.println(sendMessage);
            System.out.println("Enter second parameter : ");
            sendMessage = keyRead.readLine();
            pwrite.println(sendMessage);

            System.out.flush();
            if ((receiveMessage = receiveRead.readLine()) != null)
                System.out.println(receiveMessage);
        }
    }
}

```

**Server.java:**

```

import java.io.*;
import java.net.*;

class server1 {
    public static void main(String[] args) throws Exception {
        ServerSocket sersock = new ServerSocket(3000);
        System.out.println("Server ready");
        Socket sock = sersock.accept();
        BufferedReader keyRead = new BufferedReader(new
            InputStreamReader(System.in));
        OutputStream ostream = sock.getOutputStream();
        PrintWriter pwrite = new PrintWriter(ostream, true);
        InputStream istream = sock.getInputStream();
        BufferedReader receiveRead = new BufferedReader(new
            InputStreamReader(istream));
        String receiveMessage, sendMessage, fun;
        int a, b, c;
        while (true) {
            fun = receiveRead.readLine();
            if (fun != null)
                System.out.println("Operation : " + fun);
            a = Integer.parseInt(receiveRead.readLine());
            System.out.println("Parameter 1 : " + a);
            b = Integer.parseInt(receiveRead.readLine());
            if (fun.compareTo("add") == 0) {
                c = a + b;
                System.out.println("Addition = " + c);
                pwrite.println("Addition = " + c);
            }
            if (fun.compareTo("sub") == 0) {
                c = a - b;
                System.out.println("Substraction = " + c);
                pwrite.println("Substraction = " + c);
            }
            if (fun.compareTo("mul") == 0) {
                c = a * b;
                System.out.println("Multiplication = " + c);
                pwrite.println("Multiplication = " + c);
            }
            if (fun.compareTo("div") == 0) {
                c = a / b;
                System.out.println("Division = " + c);
                pwrite.println("Division = " + c);
            }
        }
    }
}

```

**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 :****Client.java:**

```

import java.io.*;
import java.rmi.*;

public class client2 {
    public static void main(String args[]) throws Exception {
        try {
            String s = "rmi://" + args[0] + "/abc";

            serverint f = (serverint) Naming.lookup(s);

            DataInputStream m = new
                DataInputStream(System.in);

            int n1 = Integer.parseInt(m.readLine());
            int n2 = Integer.parseInt(m.readLine());

            System.out.println("the addition is : " +
                               f.add(n1, n2));
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

```

**Server.java:**

```

import java.net.*;
import java.rmi.*;

public class serverf
{
    public static void main(String args[])
    {
        try
        {
            serverimpl m = new serverimpl();
            Naming.rebind("abc", m);
        }
        catch (Exception e)
        {

```

```
        System.out.println("Exception" + e);
    }
}
```

## Implementation.java:

```
import java.rmi.*;
import java.rmi.server.*;

public class serverimpl extends UnicastRemoteObject
implements serverint
{
    public serverimpl()throws Exception
    {
    }
    public int fact(int n)
    {
        int i,c=1;
        for(i=1;i<=n;i++)
            c=i*c;
        return c;
    }
    public int add(int a,int b)
    {
        return (a+b);
    }
}
```

## Interface.java:

```
import java.rmi.*;  
  
public interface serverint extends Remote {  
    int fact(int n) throws Exception;  
    int add(int a, int b) throws Exception;  
}
```

### **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 :****Client.java:**

```

import java.io.*;
import java.rmi.*;

public class client2 {
    public static void main(String args[]) throws Exception {
        try {
            String s = "rmi://" + args[0] + "/abc";
            serverint f = (serverint) Naming.lookup(s);
            DataInputStream m = new
                DataInputStream(System.in);

            System.out.print("Enter the number of factorial
                             to be found : ");
            int n1 = Integer.parseInt(m.readLine());

            System.out.println("the factorial of "+n1+" is" +
                               f.fact(n1));
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

```

**Server.java:**

```

import java.net.*;
import java.rmi.*;

public class serverf

{
    public static void main(String args[])
    {
        try
        {
            serverimpl m = new serverimpl();
            Naming.rebind("abc", m);
        }
        catch (Exception e)
        {

```

```
        System.out.println("Exception" + e);
    }
}
```

## Implementation.java:

```
import java.rmi.*;
import java.rmi.server.*;

public class serverimpl extends UnicastRemoteObject
implements serverint
{
    public serverimpl()throws Exception
    {

    }

    public int fact(int n)
    {
        int i,c=1;
        for(i=1;i<=n;i++)
            c=i*c;
        return c;
    }

    public int add(int a,int b)
    {
        return (a+b);
    }
}
```

## Interface.java:

```
import java.rmi.*;  
  
public interface serverint extends Remote {  
    int fact(int n) throws Exception;  
  
    int add(int a, int b) throws Exception;  
}
```

## **OUTPUT**

5  
The factorial is 120

## RESULT

Thus, the PROGRAM is executed and output is obtained.

**PROGRAM — 4****AIM :**

To create RMI to perform arithmetic operations using RMI.

**PROGRAM :****Client.java:**

```

import java.rmi.Naming;
import java.rmi.RemoteException;
import java.net.MalformedURLException;
import java.rmi.NotBoundException;

public class clientC {

    public static void main(String[] args)
    {
        try {
            servercmd c = (servercmd)
                Naming.lookup("rmi://localhost/CalculatorService");
            System.out.println("Calculator");
            System.out.print("SUBTRACTION 4-2=" + c.sub(4, 2));
            System.out.print("ADDITION 4+2 =" + c.add(4, 2));
            System.out.print("MULTIPLICATION 4*2=" + c.mul(4, 2));
            System.out.println("DIVISION 4/2=" + c.div(4, 2));
        }
        catch (MalformedURLException murle) {
            System.out.println();
            System.out.println("MalformedURLException");
            System.out.println(murle);
        }
        catch (RemoteException re) {
            System.out.println();
            System.out.println("RemoteException");
            System.out.println(re);
        }
        catch (NotBoundException nbe) {
            System.out.println();
            System.out.println("NotBoundException");
            System.out.println(nbe);
        }
        catch (java.lang.ArithmetricException ae) {
            System.out.println();

            System.out.println("java.lang.ArithmetricException");
            System.out.println(ae);
        }
    }
}

```

```

    }
}
```

**Server.java:**

```

import java.rmi.Naming;

public class serverC {

    public serverC() {
        try {
            servercmd c = new servercmdimpl();
            Naming.rebind("rmi://localhost:1099/CalculatorService",
                c);
        } catch (Exception e) {
            System.out.println("Trouble: " + e);
        }
    }

    public static void main(String args[]) {
        new serverC();
    }
}
```

**Implementation.java:**

```

public class servercmdimpl extends
java.rmi.server.UnicastRemoteObject implements servercmd {

    public servercmdimpl() throws java.rmi.RemoteException {
        super();
    }

    public long add(long a, long b) throws
                    java.rmi.RemoteException {
        return a + b;
    }

    public long sub(long a, long b) throws
                    java.rmi.RemoteException {
        return a - b;
    }

    public long mul(long a, long b) throws
                    java.rmi.RemoteException {
        return a * b;
    }
}
```

```
public long div(long a, long b) throws
        java.rmi.RemoteException {
    return a / b;
}
}
```

**Interface.java:**

```
public interface servercmd
    extends java.rmi.Remote {
    public long add(long a, long b)
        throws java.rmi.RemoteException;

    public long sub(long a, long b)
        throws java.rmi.RemoteException;

    public long mul(long a, long b)
        throws java.rmi.RemoteException;

    public long div(long a, long b)
        throws java.rmi.RemoteException;
}
```

**OUTPUT:**

```
Calculator
SUBTRACTION 4-2 = 2
ADDITION 4+2 = 6
MULTIPLICATION 4*2= 8
DIVISION 4/2= 2
```

**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 :****Client.java:**

```

import java.io.*;
import java.net.*;
import java.util.*;
class Client
{
    public static void main(String args[])
    {
        try
        {
            DatagramSocket client=new DatagramSocket();
            InetAddress
                addr=InetAddress.getByName("127.0.0.1");

            byte[] sendbyte=new byte[1024];
            byte[] receivebyte=new byte[1024];
            BufferedReader in=new BufferedReader(new
                InputStreamReader(System.in));
            while(true)
            {
                System.out.print("Enter your choice : 1.
                    DNS\t 2. Reverse DNS\t 3. Exit\n--\b\b");
                int n =
                    Integer.parseInt(System.console().readLine());
                if(n==1)
                {
                    sendbyte =
                        Integer.toString(n).getBytes();
                    DatagramPacket sender=new
                    DatagramPacket(sendbyte,sendbyte.length,
                        addr,1309);
                    client.send(sender);
                    System.out.print("Enter the DOMAIN
                        NAME:");
                    String str=in.readLine();
                    sendbyte=str.getBytes();
                    sender=new
                    DatagramPacket(sendbyte,sendbyte.length,addr,1309);
                    client.send(sender);
                }
            }
        }
    }
}

```

```

        DatagramPacket receiver=new
        DatagramPacket(receivebyte,receivebyte.length);
            client.receive(receiver);
            String s=new String(receiver.getData());
            System.out.println("IP address :
                                "+s.trim());
        }
        if(n==2)
        {
            sendbyte =
                Integer.toString(n).getBytes();
            DatagramPacket sender=new
            DatagramPacket(sendbyte,sendbyte.length,addr,1309);
            client.send(sender);

            System.out.println("Enter the IP
                                address:");
            String str=in.readLine();
            sendbyte=str.getBytes();
            sender=new
            DatagramPacket(sendbyte,sendbyte.length,addr,1309);
            client.send(sender);
            DatagramPacket receiver=new
            DatagramPacket(receivebyte,receivebyte.length);
            client.receive(receiver);
            String s=new String(receiver.getData());
            System.out.println("DOMAIN NAME :
                                "+s.trim());
        }
        if(n == 3)
            break;
    }
    client.close();
}
catch(Exception e)
{
    System.out.println(e);
}
}
}

```

**Server.java:**

```

import java.io.*;
import java.net.*;
import java.util.*;

class Server
{

```

```
public static void main(String args[])
{
    try
    {
        DatagramSocket server=new DatagramSocket(1309);
        while(true)
        {
            byte[] sendbyte=new byte[1024];
            byte[] receivebyte=new byte[1024];
            DatagramPacket receiver=new
                DatagramPacket(receivebyte,receivebyte.length);
            server.receive(receiver);
            int n=Integer.parseInt(new
                String(receiver.getData()).trim());
            System.out.println(n);

            server.receive(receiver);
            String str=new String(receiver.getData());
            String s=str.trim();
            System.out.println(s);
            InetAddress addr=receiver.getAddress();
            int port=receiver.getPort();

            if(n == 1)
            {
                InetAddress address;
                address = InetAddress.getByName(s);

                sendbyte=address.getHostAddress().getBytes();
                DatagramPacket sender=new
                    DatagramPacket(sendbyte,sendbyte.length,addr,port);
                server.send(sender);
            }
            if(n == 2)
            {
                InetAddress ia = InetAddress.getByName(s);
                sendbyte=ia.getHostName().getBytes();
                DatagramPacket sender=new
                    DatagramPacket(sendbyte,sendbyte.length,addr,port);
                server.send(sender);
            }
        }
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}
```

**DNS.java:**

```
import java.rmi.*;
public interface DNS extends Remote
{
    public int find(String x) throws RemoteException;
}
```

**DNSRemote.java:**

```
import java.rmi.*;
import java.rmi.server.*;
import java.io.*;
import java.net.*;
import java.util.*;

public class DNSRemote extends UnicastRemoteObject implements
DNS
{

    DNSRemote() throws RemoteException
    {
        super();
    }
    public int find(String x){
        try
        {
            InetAddress ad =
                java.net.InetAddress.getByName(x);
            String address = ad.getHostAddress();
            System.out.println(address);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        return 0;
    }
}
```

**OUTPUT:**

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

**RESULT :**

Thus, the PROGRAM is executed and output is obtained.

**Experiment – 3**

# **CLIENT SIDE SCRIPTING**

**PROGRAM – 1****AIM :**

Create a Web page HTML CSS that holds a bulleted list of the names of your friends. Make sure that the bullets are in plain circle.

**PROGRAM :****prog1.html**

```
<!DOCTYPE html>
<html>
    <head>
        <title>List</title>
        <link rel="stylesheet" href="prog1.css">
    </head>
    <body class = "body">
        Friends List
        <ul type = "circle">
            <li>Sid</li>
            <li>Manas</li>
            <li>Srini</li>
            <li>Safi</li>
            <li>John</li>
        </ul>
    </body>
</html>
```

**prog1.css**

```
.body {
    align-self: auto;
    background: palegreen;
}
```

**OUTPUT :****RESULT :**

Thus the PROGRAM is executed and output is obtained.

**PROGRAM – 2****AIM :**

Create a web page in HTML CSS to display the maximum and minimum temperature of 5 cities using table.

**PROGRAM :****prog2.html**

```
<!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</th>
            <th>Min Temperature</th>
        </tr>

        <tr>
            <td>Chennai</td>
            <td>45</td>
            <td>26</td>
        </tr>
        <tr>
            <td>Kharagpur</td>
            <td>35</td>
            <td>15</td>
        </tr>
        <tr>
            <td>Peshawar</td>
            <td>10</td>
            <td>-10</td>
        </tr>
    </table>
</body>
</html>
```

**prog2.css**

```
.table {  
    background-color: paleturquoise;  
    border: 1px solid green;  
}
```

**OUTPUT :**

City	Max Temperature	Min Temperature
Chennai	45	26
Kharagpur	35	15
Peshawar	10	-10

**RESULT :**

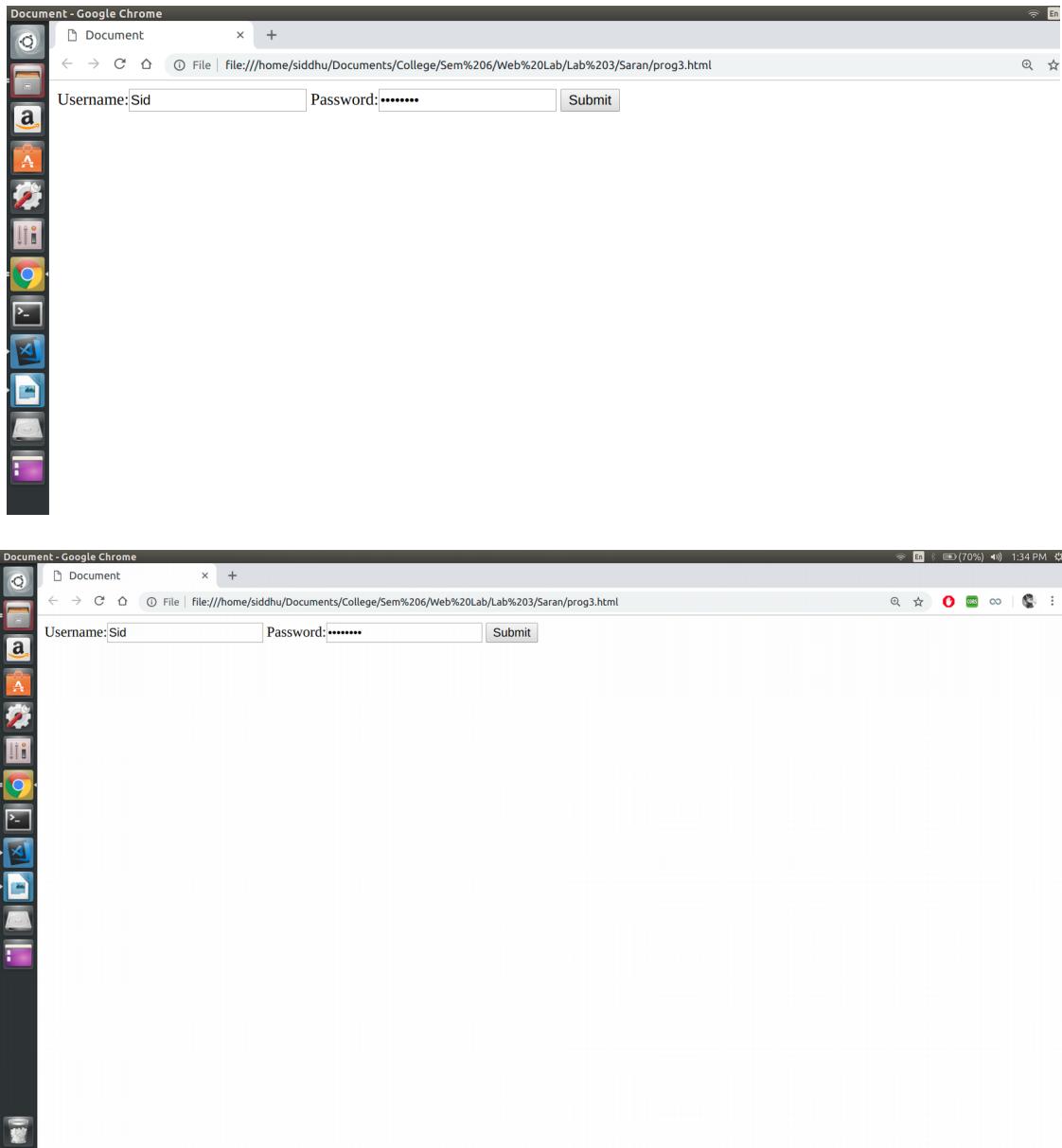
Thus the PROGRAM is executed and output is obtained

**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 :****prog3.html**

```
<!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="inp1">
    Password:<input type="password", required, id="inp2">
    <button type="button", onclick="check()">Submit</button>
</body>
<script>
    function check()
    {
        var uname = document.getElementById("inp1");
        var pass = document.getElementById("inp2");
        var k = pass.value;
        if( k === "password")
        {
            window.open("https://www.google.com/")
        }
    }
</script>
</html>
```

**OUTPUT :****RESULT :**

Thus the PROGRAM is executed and output is obtained

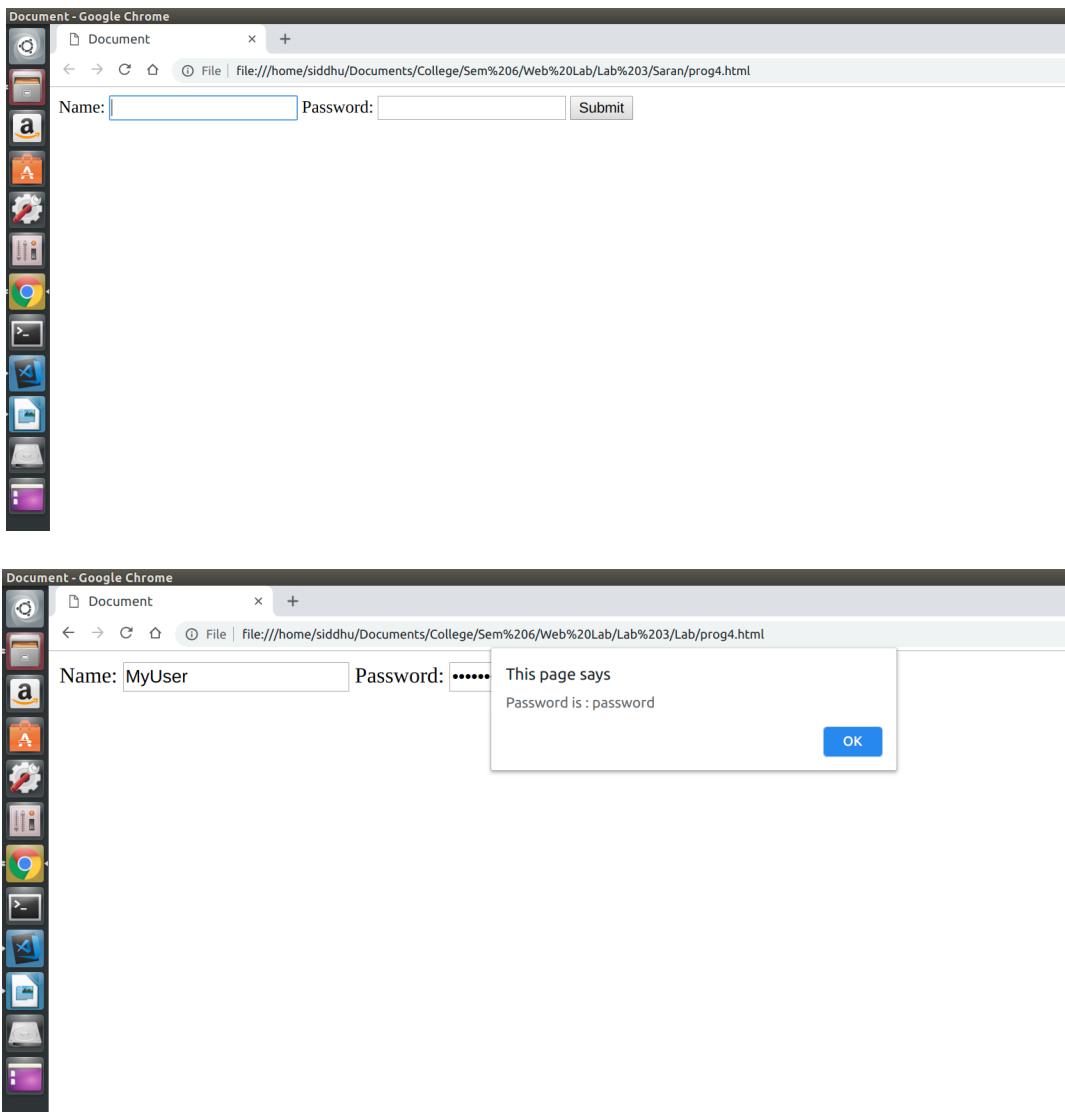
**PROGRAM — 4****AIM :**

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 :****prog4.html**

```
<!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>
        Password: <input type="password" , id="ip2">
        <button onclick="submitFunc()">Submit</button>
    </body>
    <script>
        function check() {
            var box = document.getElementById("ip1").value;
            if (box == "") {
                document.getElementById("ip1").focus();
            }
        }
        function submitFunc(){
            alert("Password is : " +
                document.getElementById("ip2").value);
        }
    </script>
</html>
```

**OUTPUT :****RESULT :**

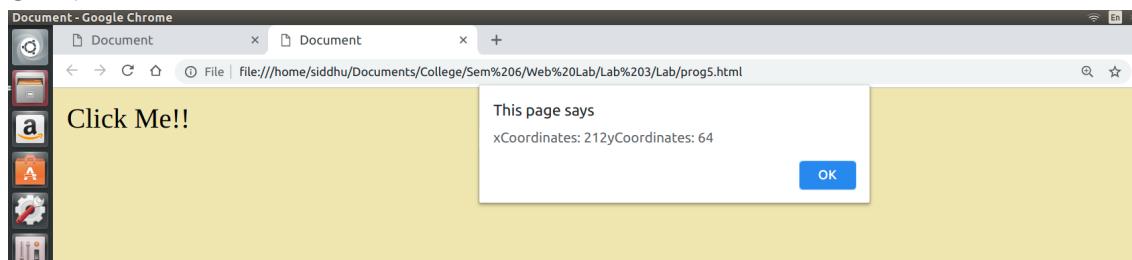
Thus the PROGRAM is executed and output is obtained

**PROGRAM – 5****AIM :**

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

**PROGRAM :****prog5.html**

```
<!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 onclick="fetchCoordinates()", class="body">
    <p class = "para" >Click Me!!</p>
</body>
<style>
    .body{
        width: 100%;
        min-height: 100%;
        background-color: aqua;
    }
</style>
<script>
    function fetchCoordinates() {
        console.log("heloo");
        var cX = event.clientX;
        var cY = event.clientY;
        alert("xCoordinates: " + cX + "yCoordinates: " + cY)
    }
</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 :****prog6.html**

```
<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.charCodeAt >= 48 &&
                          event.charCodeAt <= 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>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="C"
              OnClick="calc.M.value=''"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="7"
              OnClick="calc.M.value+=7"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="8"
              OnClick="calc.M.value+=8"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="9"
              OnClick="calc.M.value+=9"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="4"
              OnClick="calc.M.value+=4"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="5"
              OnClick="calc.M.value+=5"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="6"
              OnClick="calc.M.value+=6"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="1"
              OnClick="calc.M.value+=1"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="2"
              OnClick="calc.M.value+=2"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="3"
              OnClick="calc.M.value+=3"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="0"
              OnClick="calc.M.value+=0"
            />
          </td>
        </tr>
        <tr>
          <td>
            <input
              id="btn"
              type="button"
              value="."
              OnClick="calc.M.value+='.'
            />
          </td>
        </tr>
        <tr>
          <td colspan="5" style="text-align: center;">
            <input
              id="btn"
              type="button"
              value="="
              OnClick="calc.M.value+=eval(calc.M.value)"
            />
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```
>
</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+= calc.M.value"
/>
</td>
</tr>
```

```
        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>
```

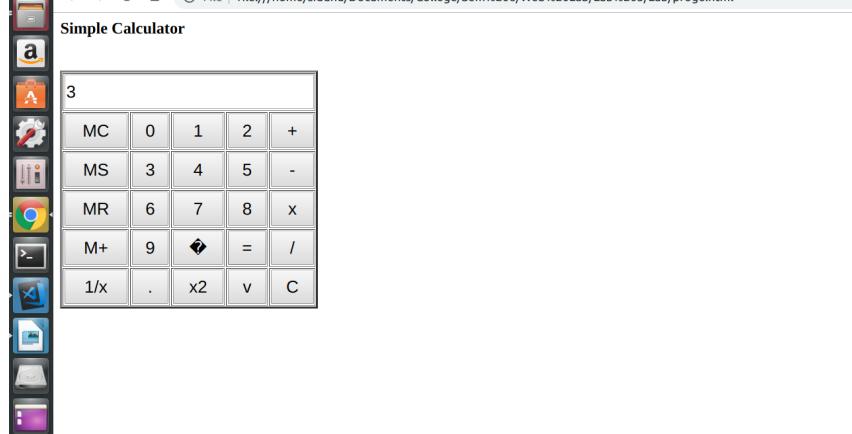
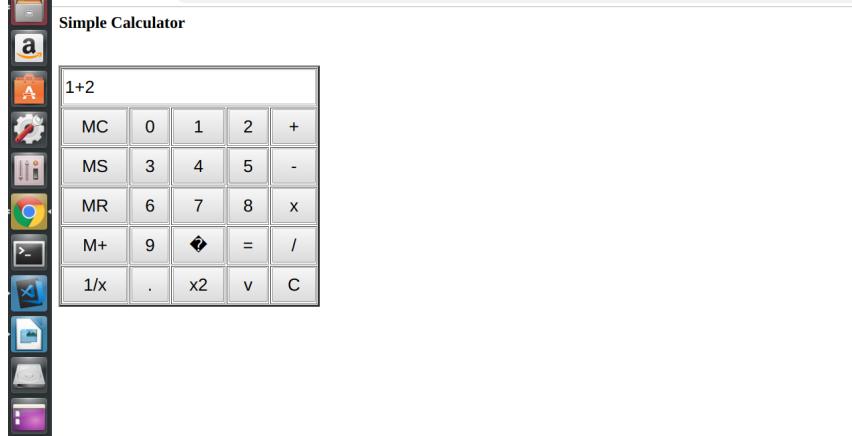
```

        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.display.value))"
        />
    </td>
    <td>
        <input
            id="btn"
            type="button"
            value="9"
            OnClick="calc.display.value+='9'"
        />
    </td>
    <td>
        <input
            id="btn"
            type="button"
            value=""
            OnClick="calc.display.value=(calc.display.value==Math.abs(calc.dis-
                play.value)?-(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 :****prog7.html**

```
<!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 :**



**RESULT :**

Thus the PROGRAM is executed and output is obtained

**PROGRAM — 8****AIM :**

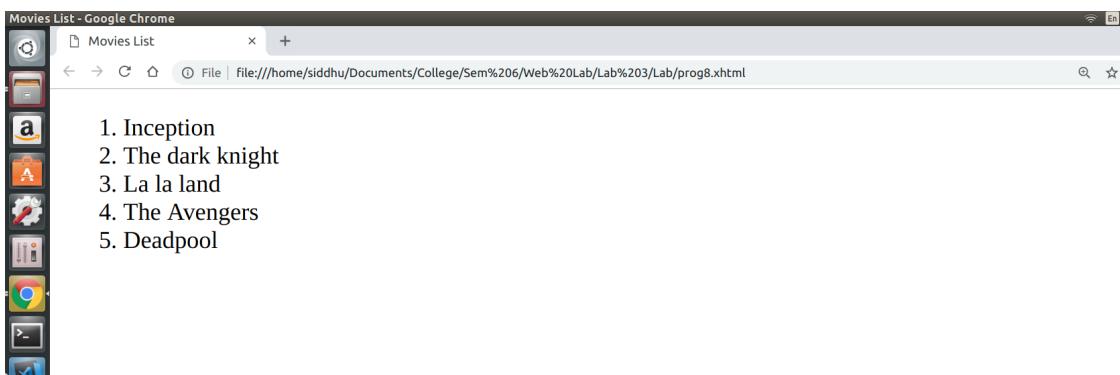
To create, test and validate an XHTML document that describes an ordered list of 5 movies.

**PROGRAM :****prog8.html**

```
<!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>
            Movies List
        </title>
    </head>

    <body>
        <ol>
            <li>Inception</li>
            <li>The dark knight</li>
            <li>La la land</li>
            <li>The Avengers</li>
            <li>Deadpool</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

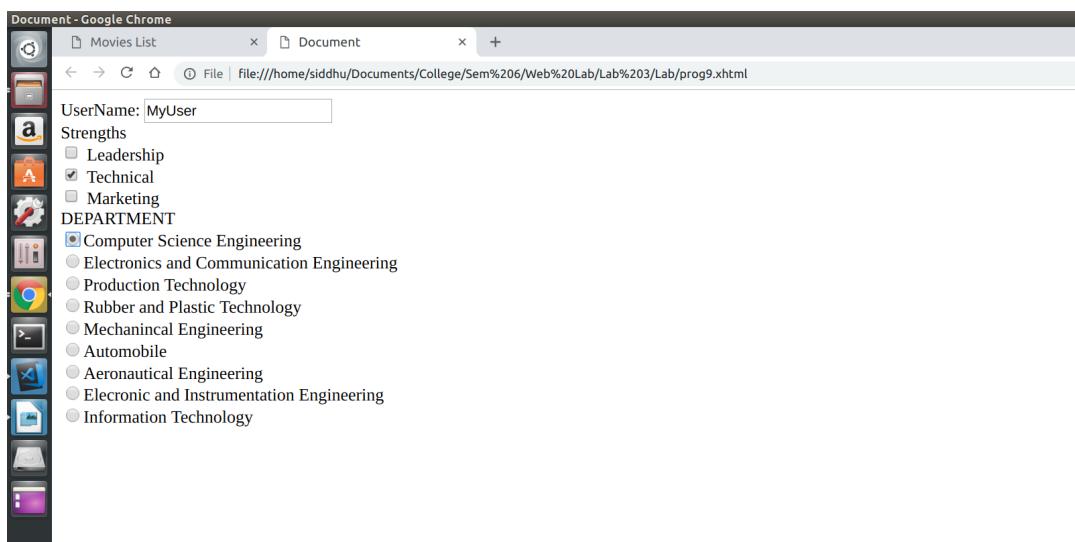
**PROGRAM :****prog9.html**

```
<!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 />
    Strengths <br />
    <form>
        <input type="checkbox" name="gp2" id="cb1" />
    Leadership<br />
        <input type="checkbox" name="gp2" id="cb2" /> Technical<br />
        <input type="checkbox" name="gp2" id="cb4" /> Marketing
    <br />
    </form>
    DEPARTMENT <br />
    <form>
        <input type="radio" name="gp1" id="rb1"
    checked="checked" />Computer Science Engineering<br />
        <input type="radio" name="gp1" id="rb2" />Electronics and
    Communication Engineering<br />
        <input type="radio" name="gp1" id="rb3" />Production
    Technology<br />
        <input type="radio" name="gp1" id="rb4" />Rubber and
    Plastic Technology<br />
        <input type="radio" name="gp1" id="rb5" />Mechanical
    Engineering<br />
        <input type="radio" name="gp1" id="rb6" />Automobile<br />
        <input type="radio" name="gp1" id="rb7" />Aeronautical
    Engineering<br />
```

```
<input type="radio" name="gp1" id="rb8" />Electronic and  
Instrumentation Engineering<br />  
<input type="radio" name="gp1" id="rb9" />Information  
Technology<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 :****index.html**

```
<!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>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="style.css" />
    <title>Computer Technology</title>
</head>
<body>
    <div id="container">
        <div id="mainpic">
            <h1 style="color:blue">Department of <span
class="lightblue">Computer Technology</span></h1>
        </div>
        <div id="menu">
            <ul>
                <li class="menuitem"><a href="#">Home</a></li>
                <li class="menuitem"><a href="#">About</a></li>
                <li class="menuitem"><a href="http://www.mitindia.edu/en/ct-
faculty">Faculty List</a></li>
                    <li class="menuitem"><a
href="http://www.ct.mitindia.edu/studentslist.html">Student List</a></li>
                        <li class="menuitem"><a href="#">Services</a></li>
                        <li class="menuitem"><a href="#">Contact</a></li>
                    </ul>
                </div>
            </div>
```

```
<div id="content">

<p>&nbsp;</p>

<p>&nbsp;</p>

<h1>ABOUT US</h1>

<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>

<p>&nbsp;</p>

<h2>Placement</h2>

<p>The placement session that took place was highly successful and many students bagged great offers from various leading mncs</p>

<p>&nbsp;</p>

<h2>Contact US</h2>

<p>Department of Computer Technology, <br />Madras Institute of Technology, <br />MIT Road, Radha Nagar, Chromepet, <br />Chennai, Tamil Nadu 600044, India. <br />Tel:+91 44 2251 6231/32</p>

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>&nbsp;</p>
```

```
<p>&nbsp;</p>
<p>&nbsp;</p>
<div id="footer">
    <h3><a href="http://www.bryantsmith.com">florida web
        design</a></h3>
</div>
</div>
</body>
</html>
```

**style.css**

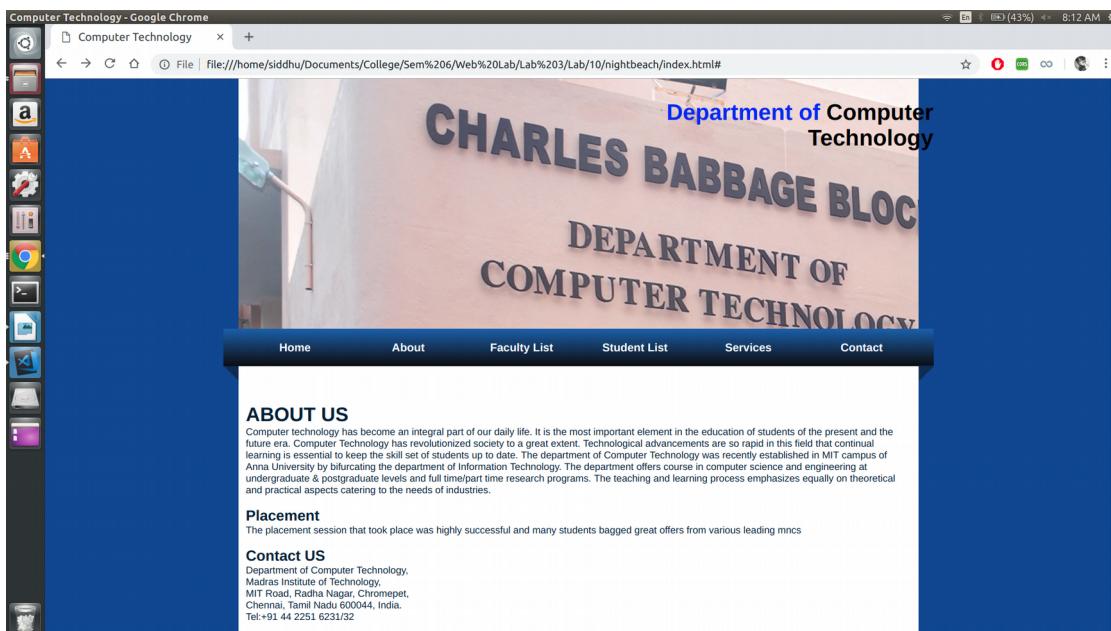
```
body {
    margin: 0;
    padding: 0;
    text-align: left;
    font: 12px Arial, Helvetica, sans-serif;
    font-size: 13px;
    color: #061C37;
    background: #1E4297;
    background-repeat:repeat-x;
}
*
{
    margin: 0 auto 0 auto;
    text-align:left;}
#container
{
    display: block;
    height:auto;
    position: relative;
    width: 940px;
}
#mainpic h1
{
    position:absolute;
    text-align:right;
    font-size:30px;
    color:#FFF;
    left:550px;
    top:100px;
}
```

```
#mainpic h2
{
position:absolute;
text-align:right;
color:#E1E7F7;
left:550px;
top:130px;
}
#mainpic
{
background-image:url(images/main.jpg);
background-repeat:no-repeat;
width:900px;
height:402px;
}
.lightblue
{
color:black;
}
#menu
{
background-image:url(images/menu.png);
background-repeat:no-repeat;
width:940px;
height:69px;
float:left;
clear:both;
}
#content
{
width:880px;
height:auto;
background-color:#FFF;
padding-left:10px;
padding-right:10px;
padding-bottom:5px;
}
#footer h3 a,#footer h3 a:visited
{
display:inline;
text-align:center;
font-size:12px;
text-decoration:none;
color:#7198E1;
}
#menu ul {
list-style: none;
padding: 0px;
margin-left:auto;
```

```

        width:900px;
    }
    #menu li {
        list-style: none;
        padding: 0px;
        display: inline;
    }
    #menu a {
        float: left;
        width: 150px;
        height: 40px;
        display: block;
        text-align: center;
        text-decoration: none;
        color: #ffffff;
        font-weight: bold;
        padding-top: 17px;
        font-size: 15px;
    }
    html, body {
        text-align: center;
    }

```

**OUTPUT :****RESULT :**

Thus the PROGRAM is executed and output is obtained.

**CS7612 WEB TECHNOLOGY LAB**

**Y Manaswini 2016503555**

**Date – 04/01/19**

**Experiment – 4**

**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:**

```
d a b c d e
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) {
            # code...
            $mid=count($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);
        print_r($map);

        arsort($map);
        $a2 = array_keys($map);
        for($x1=0;$x1<count($a2);$x1++)
        {
            print $a2[$x1];
        }

        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:**

```
Array
(
    [a] => 5
    [b] => 4
    [c] => 1
)
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]." ");
    }
?
>
```

**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 an 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."\n";
            }
        }
        return "false";
    }
    $a1 ="1 2 34 12 4455";
    print $a1."\n" ;
    $a2 = compute($a1);
    print $a2 . "\n";
    $a1 ="1 2 34 12 345";
    print $a1."\n" ;
    $a2 = compute($a1);
    print $a2 . "\n";
?
>
```

**OUTPUT:**

```
1 2 34 12 4455
4455
1 2 34 12 345
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)
    {
        # code...
        $filecontents = file_get_contents($fname);
        echo $filecontents;
        $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 => $value) {
            # code...
            if($value > $max)
            {
                $max = $value;
                $word = $key;
            }
        }
        return $word;
    }

    $fname = "prog1_6.txt";
    $file = fopen($fname,"r") or die("Unable to open file");
    echo findMax($file,$fname);
?>
```

**OUTPUT:**

Input File :  
harry potter lilly:potter:james:potter

OUTPUT :  
potter

**RESULT:**

Thus the PROGRAM is executed and OUTPUT is obtained

**PROGRAM 1.7:****AIM :**

To write a PHP script that take in an string containing words that are delimited on the left by spaces and on the right 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 :**

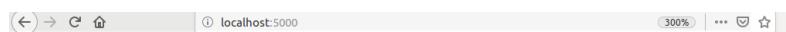
To write an XHTML document that includes an anchor tag that calls a PHP document. Also write the PHP document, which returns a which returns a randomly chosen different greetings. The greetings must be stored as constant strings in the script.

**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:**

```
<!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:**

Thank you.

**RESULT:**

Thus the PROGRAM is executed and OUTPUT is obtained.

**PROGRAM 3:****AIM :**

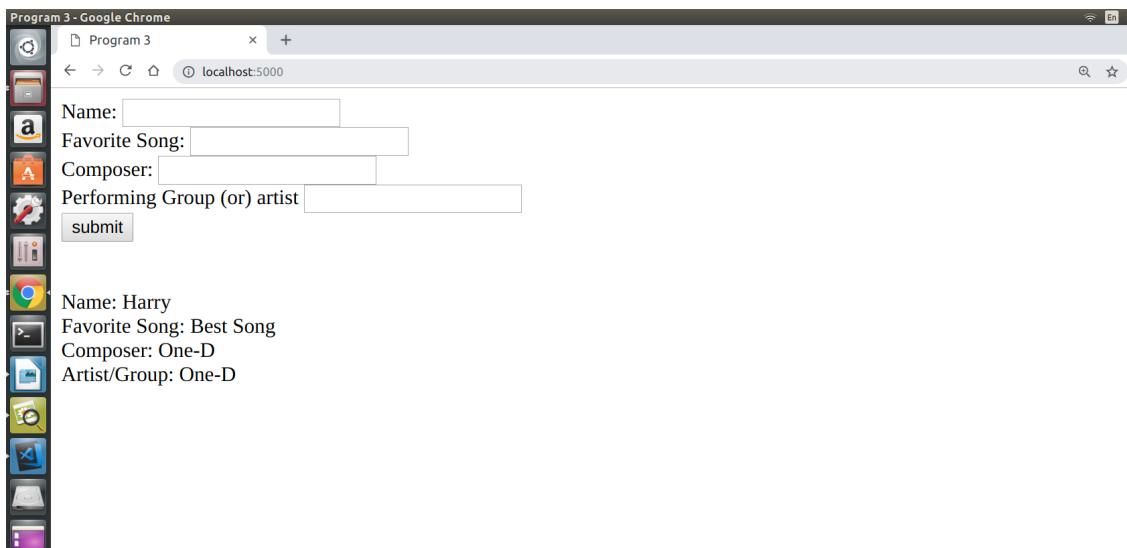
To write an XHTML document to create a form that collects favorite popular songs, including the name of the song, the composer, and the performing artist or group. The document must call one PHP script where the form is submitted and another to request a current list of survey RESULTS.

**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:**

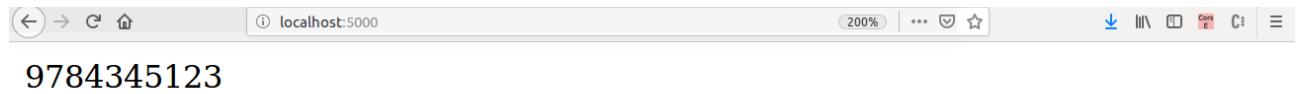
To write an XHTML document to provide a form that collects names and telephone numbers. The mobile numbers must be in the format +dd-dddd-ddddd. Write a PHP script that checks the submitted mobile number to be sure that it confirms to the required format and returns a response indicating whether the number was correct.

**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:**

## Form Validation

Name:

Mobile:  \* invalid Format

**RESULT:**

Thus the PROGRAM is executed and OUTPUT is obtained

**PROGRAM 5:****AIM :**

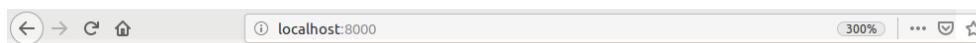
To modify the PHP script from Exercise 2 to count the number of visitors and display the numbers for each visitor.

**PROGRAM:**

```
<?php
    function greetingCall(){
        define("GREETING1","Have a nice day.");
        define("GREETING2","Thank you.");
        define("GREETING3","Welcome.");
        define("GREETING4","Nice meeting you.");
        $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;
        print "<div>".$greet[$index]."</div>";
        print "<div> Client Number ".$val."</div>";
    } greetingCall();
?>
```

**XHTML:**

```
<!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:**

Thank you.  
Client Number 22

**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 Registration number is
                    required";
            }
        }
    ?>
</body>
</html>
```

```

        }
        if(empty($_POST[ "pwd"])){
            $pwdErr = "Password is required";
        }

        if(isset($_POST[ "reg" ]) && isset($_POST[ "pwd"])){
            $servername = "localhost";
            $username = "root";
            $password = "0611";
            $dbname = "test";

            $conn = new mysqli($servername,$username,
                               $password,$dbname);
            if($conn -> connect_error){
                die("Connection failed: ".$conn->connect_error);
            }

            $regno = $_POST[ 'regno' ];
            $pwd = $_POST[ 'pwd' ];

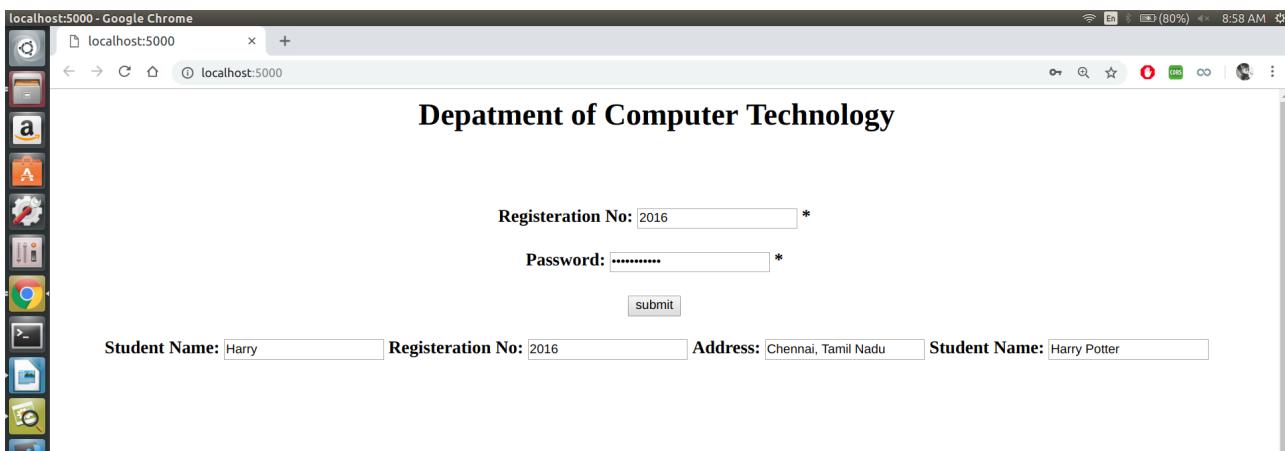
            $sql = "SELECT name,address,mobile,id from Tab1
                    where id = '$regno' and password =
                    '$pwd';";
            $result = $conn->query($sql);
            if($result->num_rows > 0){
                while($row = $result->fetch_assoc()){
                    $name = $row[ "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>Department of Computer Technology</h1></div>
<br>
<br>
```

```

<form class = "form1" method="post" action=<?php
    htmlspecialchars($_SERVER["PHP_SELF"]);?>>
<h3>
Registration 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;?>" />
    Registration 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>

```

**OUTPUT :****RESULT:**

Thus the PROGRAM is executed and OUTPUT is obtained.

**CS7612 WEB TECHNOLOGY LAB**

**Y Manaswini 2016503555**

**Date – 11/01/19**

**Experiment – 5**

# **PYTHON PROGRAMMING**

**Program – 1**

**AIM :**

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

**PROGRAM :**

**p1.py**

```
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 :****p2.py**

```
li = [1, 2, 3, 4, 5]
print('Original List : ', li)
print('Reversed List : ', li[::-1])
```

**OUTPUT:**

```
('Original List : ', [1, 2, 3, 4, 5])
('Reversed List : ', [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 :****p3.py**

```
def fact(n):
    fact = 1
    for i in range(1, n+1):
        fact = fact*i
    return fact

n = int(input('Enter n :'))
print(str(n)+'! is : '+str(fact(n)))
```

**OUTPUT:**

```
Enter n :5
5! is : 120
```

**RESULT :**

Thus the program is executed and output is obtained.

**Program – 4****AIM :**

To write a GUI for the expression calculator using tk.

**PROGRAM :****P4.py**

```
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 blue")
```

```
gui.title("Simple Calculator")

gui.geometry("300x125")

equation = StringVar()

expression_field = Entry(gui, textvariable=equation)

expression_field.grid(columnspan=4, ipadx=70)

equation.set('enter your expression')

button1 = Button(gui, text=' 1 ', fg='black', bg='pink',
                 command=lambda: press(1), height=1, width=7)
button1.grid(row=4, column=0)

button2 = Button(gui, text=' 2 ', fg='black', bg='pink',
                 command=lambda: press(2), height=1, width=7)
button2.grid(row=4, column=1)

button3 = Button(gui, text=' 3 ', fg='black', bg='pink',
                 command=lambda: press(3), height=1, width=7)
button3.grid(row=4, column=2)

button4 = Button(gui, text=' 4 ', fg='black', bg='pink',
                 command=lambda: press(4), height=1, width=7)
button4.grid(row=3, column=0)

button5 = Button(gui, text=' 5 ', fg='black', bg='pink',
                 command=lambda: press(5), height=1, width=7)
button5.grid(row=3, column=1)

button6 = Button(gui, text=' 6 ', fg='black', bg='pink',
                 command=lambda: press(6), height=1, width=7)
button6.grid(row=3, column=2)

button8 = Button(gui, text=' 8 ', fg='black', bg='pink',
                 command=lambda: press(8), height=1, width=7)
button8.grid(row=1, column=1)

button9 = Button(gui, text=' 9 ', fg='black', bg='pink',
                 command=lambda: press(9), height=1, width=7)
button9.grid(row=1, column=2)

button0 = Button(gui, text=' 0 ', fg='black', bg='pink',
                 command=lambda: press(0), height=1, width=7)
button0.grid(row=5, column=0)

plus = Button(gui, text=' + ', fg='black', bg='pink',
```

```

        command=lambda: press("+"), height=1, width=7)
plus.grid(row=1, column=3)

multiply = Button(gui, text=' * ', fg='black', bg='pink',
                  command=lambda: press("*"), height=1,
                  width=7)
multiply.grid(row=4, column=3)

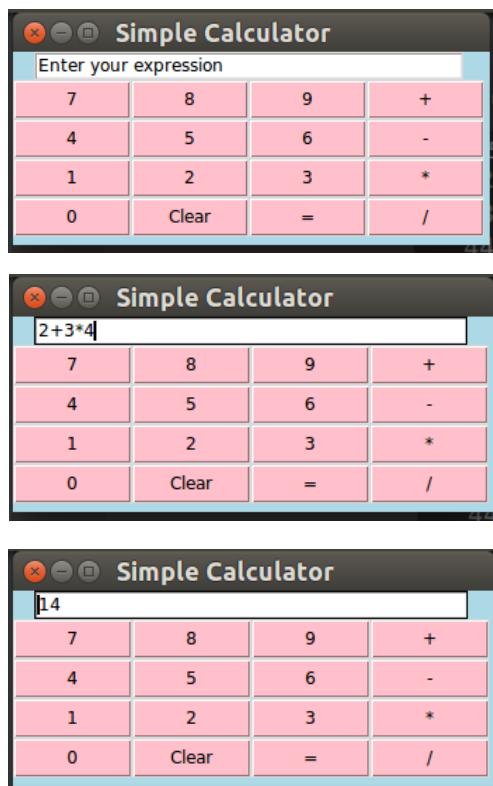
divide = Button(gui, text=' / ', fg='black', bg='pink',
                  command=lambda: press("/"), height=1, width=7)
divide.grid(row=5, column=3)

equal = Button(gui, text=' = ', fg='black', bg='pink',
                  command=equalpress, height=1, width=7)
equal.grid(row=5, column=2)

clear = Button(gui, text='Clear', fg='black', bg='pink',
                  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 :**

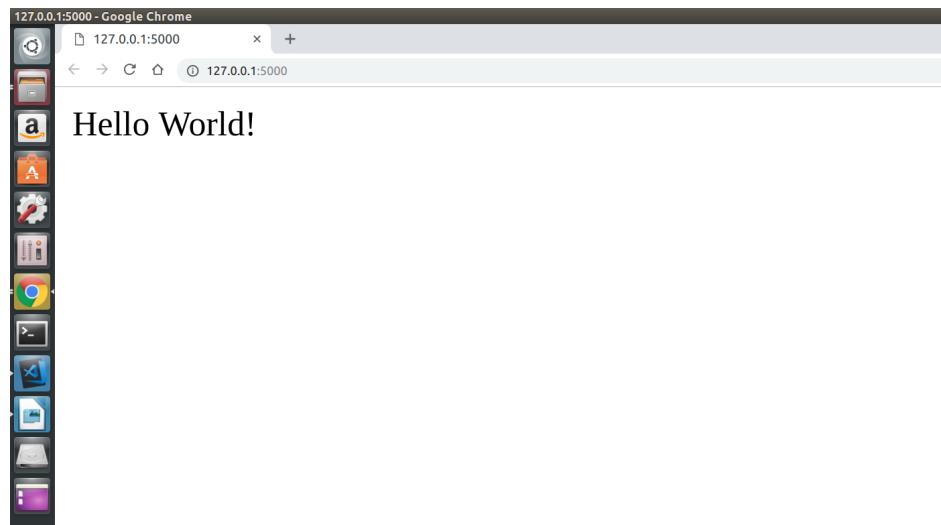
To write a procedure to install packages requests, flask and explore them using pip.

**PROGRAM :****P5.py**

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return "Hello World!"

if __name__ == '__main__':
    app.run()
```

**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 :****p6.py**

```
import requests
from bs4 import BeautifulSoup

try:
    out = open('output6.txt', 'w')
    page = requests.get('https://www.nytimes.com/')
    soup = BeautifulSoup(page.text, 'html.parser')

    headlineItems = soup.find_all('title')
    for item in headlineItems:
        print(item.string)
        out.write(item.string+'\n')
    print()
    out.write('\n')

    headlineItems = soup.find_all('article')
    for item in headlineItems:
        print(item.h2.string)
        out.write(item.h2.string + '\n')

except ValueError:
    print('Value Error')
except IOError:
    print('IOError')
```

**OUTPUT:**

Breaking News, World News & Multimedia - The New York Times

Listen to 'Still Processing'  
In 'The Daily' Newsletter  
Listen to 'The Argument'  
The Enigmatic Figure at the Heart of Mueller's Inquiry  
Manafort 'Repeatedly and Brazenly' Broke Law, Prosecutors Say  
in Sentencing Memo  
Some Aid From Brazil Pierces Venezuela's Blockade  
Inside the Rise and Fall of a Multimillion-Dollar Airbnb  
Scheme

Health Care and Insurance Industries Mobilize to Kill  
'Medicare for All'

Liberal Group Is Seeking Its Next Ocasio-Cortez  
Who Will Win Best Picture, Actor and Actress? Here Are Our  
Predictions

Fill out your Academy Award ballot.

The Oscars are providing a stage for Participant Media's  
comeback story.

11 of Our Best Weekend Reads

Did you stay up to date this week? Take our news quiz.

It's Not That Men Don't Know What Consent Is

Finding What's 'Oddly Satisfying' on the Internet

Everything Is War and Nothing Is True

How America Learned to Stop Worrying and Love Deficits and  
Debt

My Daily Routine: Jasper Newman, 8 Months Old

The Green New Deal Is Better Than Our Climate Nightmare

Why Does Obama Scold Black Boys?

Waiting for the Trump-Kim Nobel Peace Prize

Why Celibacy Matters

Joe Biden's Family Web

Why the Priesthood Needs Women

A Multimillion-Dollar Payday, at the Carwash

Two Art Collectors Who Caught Each Other's Eye

The Week in Books

**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 :****p7.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)

@app.route('/sub/<int:x>&<int:y>', methods = ['GET'])
def sub(x,y):
    print("result:",str(x-y))
    return str(x-y)

@app.route('/mul/<int:x>&<int:y>', methods = ['GET'])
def mul(x,y):
    print("result:",str(x*y))
    return str(x*y)

@app.route('/div/<int:x>&<int:y>', methods = ['GET'])
def div(x,y):
    if y == 0:
        return 'Divide by Zero not permitted'
    print("result:",str(x/y))
    return str(x/y)

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

**index.html**

```
<html>

<head>
    <title>Simple ReSTful Calculator</title>
</head>
```

```

<body>
    <h2>Simple ReSTful Calculator</h2>
    <form>
        Variable-X: <input type="text" id="x"><br />
        Variable-Y: <input type="text" id="y"><br />
        Output: <input type="text" id="op" readonly><br />

        <input type="button" value="ADD" name="add"
               onclick="rest_add()">
        <input type="button" value="SUB" name="sub"
               onclick="rest_sub()">
        <input type="button" value="MUL" name="mul"
               onclick="rest_mul()">
        <input type="button" value="DIV" name="div"
               onclick="rest_div()">
    </form>
    <script type="text/javascript">
        function rest_add() {
            var a =
                parseInt(document.getElementById("x").value);
            var b =
                parseInt(document.getElementById("y").value);
            var url = "/add/" + a + "&" + b;
            var xmlhttp = new XMLHttpRequest();
            xmlhttp.open("GET", url, false);
            xmlhttp.send(null);
            document.getElementById("op").value =
                xmlhttp.responseText;
        }

        function rest_sub() {
            var a =
                parseInt(document.getElementById("x").value);
            var b =
                parseInt(document.getElementById("y").value);
            var url = "/sub/" + a + "&" + b;
            var xmlhttp = new XMLHttpRequest();
            xmlhttp.open("GET", url, false);
            xmlhttp.send(null);
            document.getElementById("op").value =
                xmlhttp.responseText;
        }

        function rest_mul() {
            var a =
                parseInt(document.getElementById("x").value);
            var b =
                parseInt(document.getElementById("y").value);
            var url = "/mul/" + a + "&" + b;
        }
    </script>

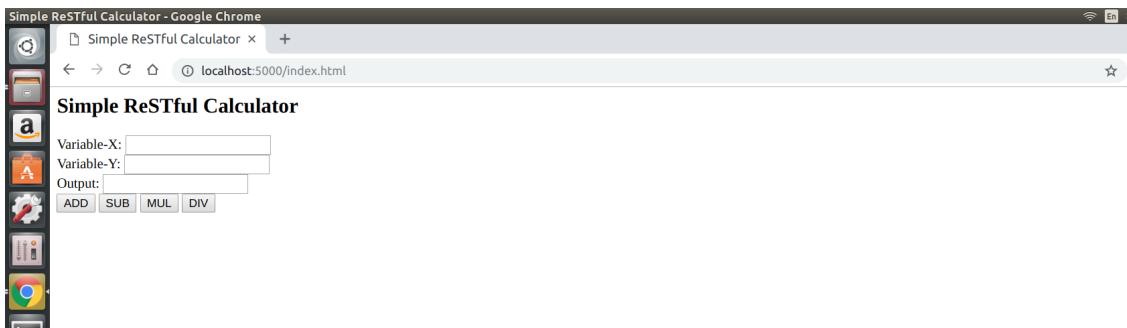
```

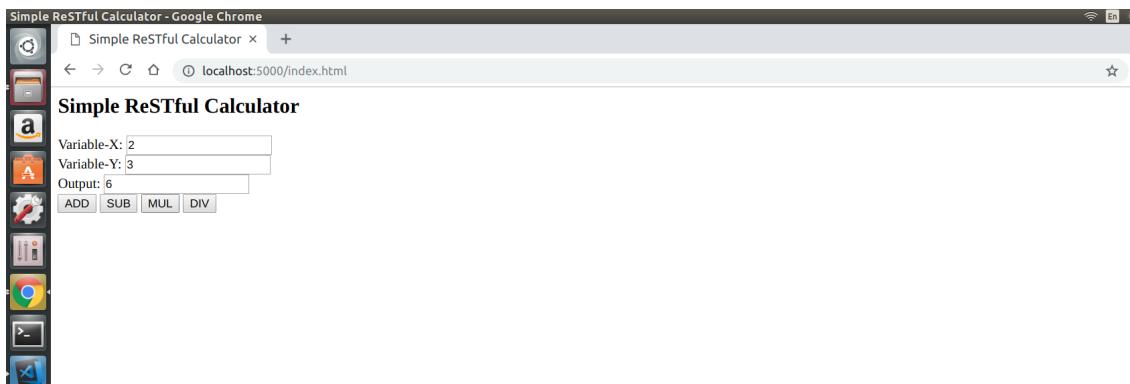
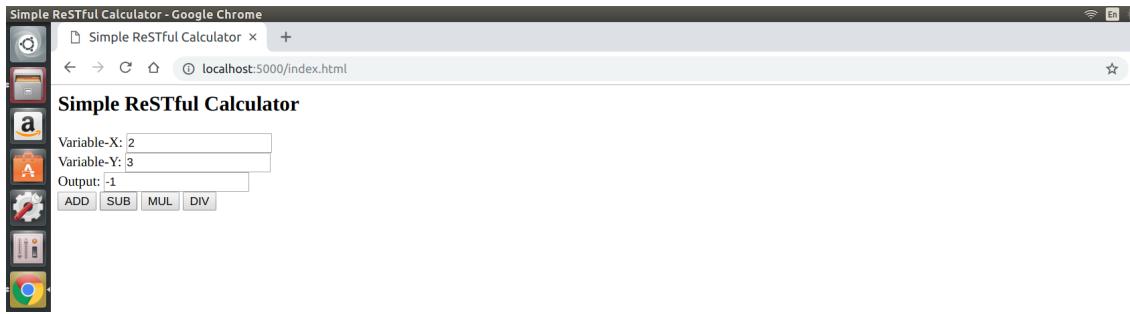
```

        var xmlhttp = new XMLHttpRequest();
        xmlhttp.open("GET", url, false);
        xmlhttp.send(null);
        document.getElementById("op").value =
            xmlhttp.responseText;
    }

    function rest_div() {
        var a =
            parseInt(document.getElementById("x").value);
        var b =
            parseInt(document.getElementById("y").value);
        var url = "/div/" + a + "&" + b;
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.open("GET", url, false);
        xmlhttp.send(null);
        document.getElementById("op").value =
            xmlhttp.responseText;
    }
</script>
</body>
</html>

```

**OUTPUT:**

**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 :**

```
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="Sid
_12345", database="test")
        print(db)
        cursor = db.cursor()

        sql = "SELECT * from student where RegNo ='"+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)):
                ans=ans+"#"+str(a[b])
        return ans

    except Exception as e:
        print('Exception caught', e)

if __name__ == '__main__':
    app.run(host='127.0.0.1',port=5555,debug = True)
```

**OUTPUT :**

The screenshot shows a Google Chrome window titled "Result page - Saf - Google Chrome". The address bar displays "127.0.0.1:5555/index8.html". The main content area has a heading "RESULTS CHECKING". Below it are two input fields: "Roll No : ". Below that is another input field: "Password : ". A "Login" button is centered below the password field. To the left of the browser window is a vertical toolbar with various icons.

The screenshot shows a Google Chrome window titled "Result page - Saf - Google Chrome". The address bar displays "127.0.0.1:5555/index8.html". The main content area has a heading "RESULTS CHECKING". Below it are two input fields: "Name : ". Below that is another input field: "Register Number : ". Below these fields is a table with three rows and three columns. The table has a border and contains the following data:

Course Name	Mark	Grade
Big Data	70	O
NLP	65	O

**RESULT :**

Thus the program is executed and output is obtained.

**Experiment - 6**

# **JAVA SERVLET**

**PROGRAM 1:****AIM:**

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

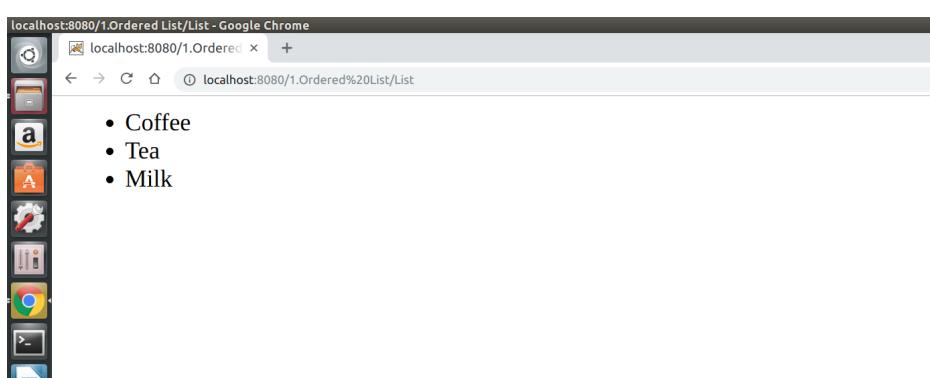
**PROGRAM :****MyServlet.java**

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

public class MyServlet extends HttpServlet
{
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
    throws ServletException, IOException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.print("<ul><li>Coffee</li><li>Tea</li><li>Milk
                  </li></ul>");
    }
}
```

**web.xml**

```
<web-app>
    <servlet>
        <servlet-name>List</servlet-name>
        <servlet-class>MyServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>List</servlet-name>
        <url-pattern>/List</url-pattern>
    </servlet-mapping>
</web-app>
```

**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 :****MyServlet.java**

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

public class MyServlet extends HttpServlet
{
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response) throws
                      ServletException, IOException
    {
        response.setContentType("text/html");
        RequestDispatcher view =
        request.getRequestDispatcher("html/Table.html");
        view.forward(request, response);
    }
}
```

**Table.html**

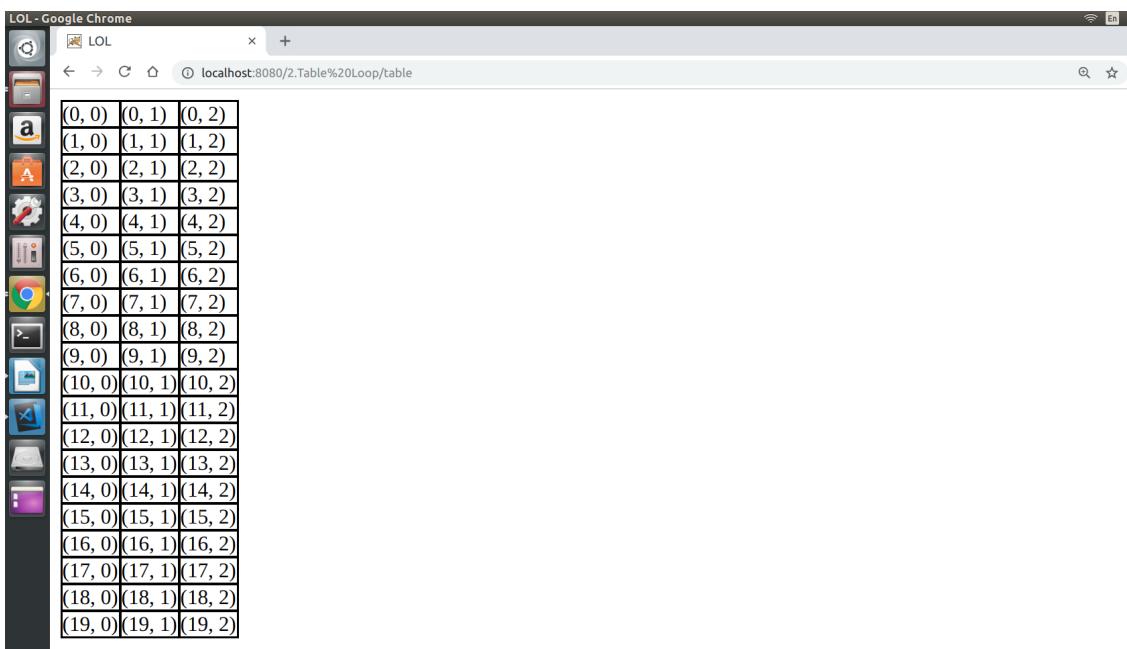
```
<html>
  <head>
    <title>LOL</title>
  </head>
  <style>
    table, th, td {
      border: 1px solid black;
    }
  </style>
  <body>
    <script>
      var mytable = "<table cellpadding=\"0\""
                    cellspacing=\"0\"><tbody><tr>";
      for (var i=0; i<20; i++)
      {
        mytable += "</tr><tr>";
        for(var j=0; j<3; j++)
          mytable += "<td>" + i + ", " + j +
                     "</td>";
      }
    </script>
  </body>
</html>
```

```
mytable += "</tr></tbody></table>";

        document.write(mytable);
    </script>
</body>
</html>
```

**web.xml**

```
<web-app>
    <servlet>
        <servlet-name>HTML Table</servlet-name>
        <servlet-class>MyServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>HTML Table</servlet-name>
        <url-pattern>/table</url-pattern>
    </servlet-mapping>
</web-app>
```

**OUTPUT :****Result:**

Thus the program is executed and output is obtained.

**PROGRAM 3:****AIM :**

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

**PROGRAM :****Form.html**

```
<html>
  <body>
    <form action = "SimpleForm" method = "GET">
      Name: <input type = "text" name = "name">
      <br />
      Reg. No: <input type = "text" name = "reg_no" />
      <br />
      Email: <input type = "text" name = "email" />
      <br />
      <input type = "submit" value = "Submit" />
    </form>
  </body>
</html>
```

**SimpleForm.java**

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

public class SimpleForm extends HttpServlet
{
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
  throws ServletException, IOException
  {
    response.setContentType("text/html");

    PrintWriter out = response.getWriter();
    String title = "Submitted Response";
    String docType = "";
    out.println(docType +
               "<html>\n" +
               "<head><title>" + title + "</title></head>\n" +
               "<body bgcolor = \"#f0f0f0\">\n" +
               "<h3 align = \"center\">" + title + "</h3>\n" +
               "<ul>\n" +
               "  <li><b>Name</b>: " +
               request.getParameter("name") + "\n" +
```

```

        " <li><b>Reg. No</b>: " +
            request.getParameter("reg_no") + "\n" +
        " <li><b>Email</b>: " +
            request.getParameter("email") + "\n" +
        "</ul>\n" +
        "</body>" +
        "</html>"
    );
}
}
}

```

**web.xml**

```

<web-app>
    <servlet>
        <servlet-name>SimpleForm</servlet-name>
        <servlet-class>SimpleForm</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>SimpleForm</servlet-name>
        <url-pattern>/SimpleForm</url-pattern>
    </servlet-mapping>
</web-app>

```

**OUTPUT :**

The image contains two screenshots of a Google Chrome browser window. The top screenshot shows a form with three input fields: 'Name: Harry Potter', 'Reg. No: 32', and 'Email: harrypotter@hogwarts.com'. A 'Submit' button is below the fields. The bottom screenshot shows the results of the submission with the title 'Submitted Response' and the message: '• Name: Harry Potter  
• Reg. No: 32  
• Email: harrypotter@hogwarts.com'.

**Result:**

Thus the program is executed and output is obtained.

**PROGRAM 4:****AIM :**

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

**PROGRAM :****SessionTrack.java**

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

public class SessionTrack extends HttpServlet
{
    public void doGet(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException
    {
        HttpSession session = request.getSession(true);

        Date createTime = new Date(session.getCreationTime());

        Date lastAccessTime = new
Date(session.getLastAccessedTime());

        String title = "Welcome Back";

        if (session.isNew())
        {
            title = "Welcome to my website";
            session.setAttribute("userID", "ABCD");
        }
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

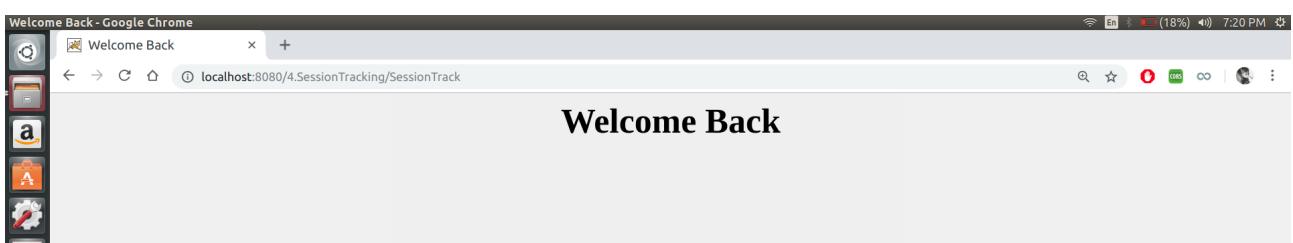
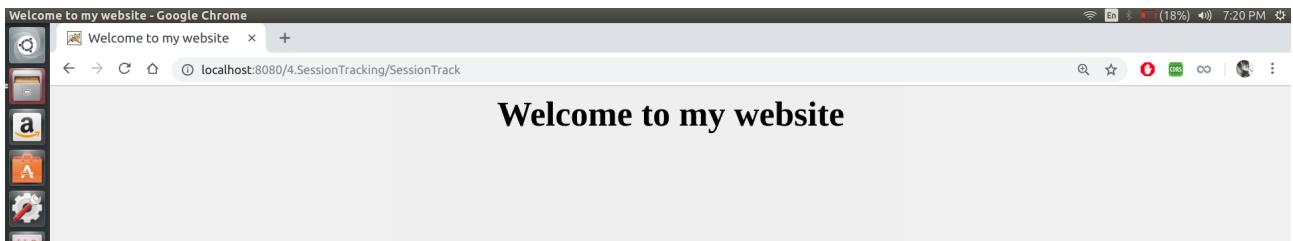
        out.println("<html>\n" +
                    "<head><title>" + title + "</title></head>\n" +

                    "<body bgcolor = \"#f0f0f0\">\n" +
                    "    <h1 align = \"center\">" + title + "</h1>\n" +
                    "</body>" +
                    "</html>"
        );
    }
}
```

**web.xml**

```
<web-app>
    <servlet>
        <servlet-name>SessionTrack</servlet-name>
        <servlet-class>SessionTrack</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>SessionTrack</servlet-name>
        <url-pattern>/SessionTrack</url-pattern>
    </servlet-mapping>
</web-app>
```

**OUTPUT :****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 first-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 :****Form.html**

```
<html>
    <body>
        <form action = "Cookie" method = "GET">
            Name: <input type = "text" name = "name">
            <br />
            Reg. No: <input type = "text" name = "reg_no" />
            <br />
            Email: <input type = "text" name = "email" />
            <br />
            <input type = "submit" value = "Submit" />
        </form>
    </body>
</html>
```

**CookieClass.java**

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

public class CookieClass extends HttpServlet
{
    public void doGet(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        Cookie name = null, reg_no = null, email = null;
        Cookie[] cookies = request.getCookies();
        cookies = request.getCookies();

        if(cookies != null)
        {
            for (int i = 0; i < cookies.length; i++)
                if (cookies[i].getName().equals("name"))
                    name = cookies[i];
                else if (cookies[i].getName().equals("reg_no"))
                    reg_no = cookies[i];
                else if (cookies[i].getName().equals("email"))
                    email = cookies[i];
        }
        else
            name = new Cookie("name", "Unknown");
            reg_no = new Cookie("reg_no", "Unknown");
            email = new Cookie("email", "Unknown");

        out.println("Name: " + name.getValue());
        out.println("Reg. No: " + reg_no.getValue());
        out.println("Email: " + email.getValue());
    }
}
```

```

{
    if (cookies[i].getName().equals("Name"))
        name = cookies[i];
    if (cookies[i].getName().equals("Reg_No"))
        reg_no = cookies[i];
    if (cookies[i].getName().equals("Email"))
        email = cookies[i];
}

if (request.getParameter("name").length() == 0)
    out.println("Name: " + name.getValue()+"<br
                />");
else
{
    out.println("Name: " +
                request.getParameter("name")+"<br />");
    if (name.getValue().equals("Unknown"))
    {
        name.setValue(request.getParameter("name"));
        response.addCookie(name);
    }
}

if (request.getParameter("reg_no").length() == 0)
    out.println("Registration Number: " +
                reg_no.getValue()+"<br />");
else
{
    out.println("Registration Number: " +
                request.getParameter("reg_no")+"<br />");
    if (reg_no.getValue().equals("Unknown"))
    {
        reg_no.setValue(request.getParameter("reg_no"));
        response.addCookie(reg_no);
    }
}

if (request.getParameter("email").length() == 0)
    out.println("Email: " + email.getValue()
                +"<br />");
else
{
    out.println("Email: " +
                request.getParameter("email")+"<br />");
    if (email.getValue().equals("Unknown"))
    {
        email.setValue(request.getParameter("email"));
        response.addCookie(email);
    }
}
}

```

```

        else if(cookies == null)
        {
            if (request.getParameter("name").length() == 0)
                name = new Cookie("Name", "Unknown");
            else
                name = new Cookie("name",
                                   request.getParameter("name"));

            if (request.getParameter("reg_no").length() == 0)
                reg_no = new Cookie("Reg_No", "Unknown");
            else
                reg_no = new Cookie("Reg_No",
                                   request.getParameter("reg_no"));

            if (request.getParameter("email").length() == 0)
                email = new Cookie("Email", "Unknown");
            else
                email = new Cookie("email",
                                   request.getParameter("email"));
            response.addCookie(name);
            response.addCookie(reg_no);
            response.addCookie(email);
        }
    }
}

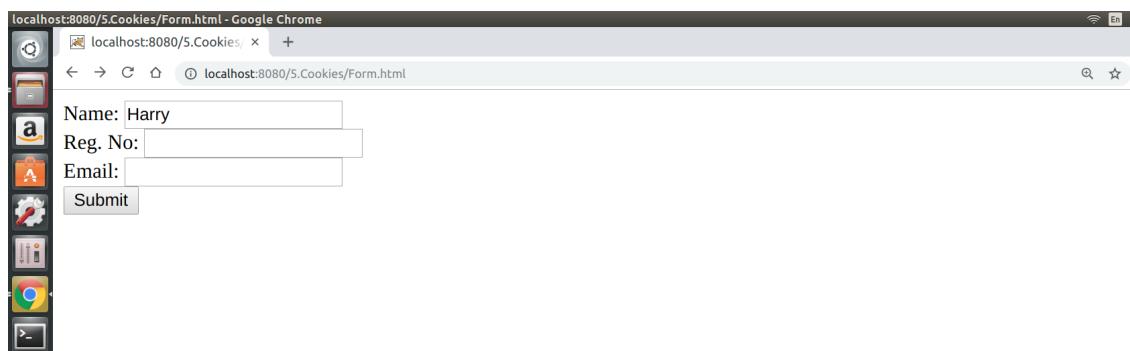
```

**web.xml**

```

<web-app>
    <servlet>
        <servlet-name>Cookie</servlet-name>
        <servlet-class>CookieClass</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>Cookie</servlet-name>
        <url-pattern>/Cookie</url-pattern>
    </servlet-mapping>
</web-app>

```

**OUTPUT :**

The image consists of three vertically stacked screenshots of a Google Chrome browser window, illustrating the interaction with cookies.

- Screenshot 1:** The browser title bar says "localhost:8080/5.Cookies/Cookie?name=Harry&reg\_no=&email=". The main content area displays the text "Name: Harry", "Registration Number: Unknown", and "Email: Unknown".
- Screenshot 2:** The browser title bar says "localhost:8080/5.Cookies/Form.html". The main content area shows a form with fields for Name (containing "Harry"), Reg. No. (containing "25"), and Email (containing "potter@hogwarts.com"). Below the form is a "Submit" button.
- Screenshot 3:** The browser title bar says "localhost:8080/5.Cookies/Cookie?name=&reg\_no=25&email=potter%40hogwarts.com". The main content area displays the text "Name: Harry", "Registration Number: 25", and "Email: potter@hogwarts.com".

**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 :****RequestHeaders.java**

```
import static java.awt.Color.yellow;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class RequestHeader extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet RequestHeader</title>");
            out.println("</head>");
            final String clientBrowser =
getClientBrowser(request);
            if (clientBrowser.equals("IE"))
                out.println("<body bgcolor= 'yellow' "
text='red'>");
            else
                out.println("<body bgcolor = 'red' "
text='yellow'>");
            out.println("<h1>Servlet RequestHeader at " +
request.getContextPath() + "</h1>");
            Enumeration<String> headerNames =
request.getHeaderNames();
            while (headerNames.hasMoreElements()) {
                String headerName = headerNames.nextElement();
                out.print("Header Name: <em>" + headerName);
                String headerValue =
request.getHeader(headerName);
```

```

        out.print("</em>, Header Value: <em>" +
headerValue);
        out.println("</em><br/>");
    }
    out.println("<h3> The client browser is : " +
clientBrowser + "</h3>");

    out.println("</body>");
    out.println("</html>");
}
}

public String getClientBrowser(HttpServletRequest request) {
    final String browserDetails = request.getHeader("User-
Agent");
    final String user = browserDetails.toLowerCase();

    String browser = "";

    if (user.contains("msie")) {
        String substring =
browserDetails.substring(browserDetails.indexOf("MSIE")).split(";")[0];
        browser = substring.split(" ")[0].replace("MSIE",
"IE") + "-" + substring.split(" ")[1];
    } else if (user.contains("safari") &&
user.contains("version")) {
        browser =
(browserDetails.substring(browserDetails.indexOf("Safari")).split(
" ")[0]).split("/")[0] + "-"
+
(browserDetails.substring(browserDetails.indexOf("Version")).split(
" ")[0]).split("/")[1];
    } else if (user.contains("opr") || user.contains("opera"))
{
        if (user.contains("opera"))
            browser =
(browserDetails.substring(browserDetails.indexOf("Opera")).split(
" ")[0]).split("/")[0] + "-"
+
(browserDetails.substring(browserDetails.indexOf("Version")).split(
" ")[0]).split("/")[1];
        else if (user.contains("opr"))
            browser =
((browserDetails.substring(browserDetails.indexOf("OPR")).split(
" ")[0]).replace("/", "-"))
                .replace("OPR", "Opera");
    } else if (user.contains("chrome")) {
        browser =
(browserDetails.substring(browserDetails.indexOf("Chrome")).split(
" ")[0]).replace("/", "-");
    }
}
}

```

```

        } else if ((user.indexOf("mozilla/7.0") > -1) ||
(user.indexOf("netscape6") != -1)
            || (user.indexOf("mozilla/4.7") != -1) ||
(user.indexOf("mozilla/4.78") != -1)
            || (user.indexOf("mozilla/4.08") != -1) ||
(user.indexOf("mozilla/3") != -1)) {
        browser = "Netscape-?";

    } else if (user.contains("firefox")) {
        browser =
(browserDetails.substring(browserDetails.indexOf("Firefox")).split
(" "))[0].replace("/", "-");
    } else if (user.contains("rv")) {
        browser = "IE";
    } else {
        browser = "UnKnown, More-Info: " + browserDetails;
    }

    return browser;
}

@Override
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "Short description";
}
}

```

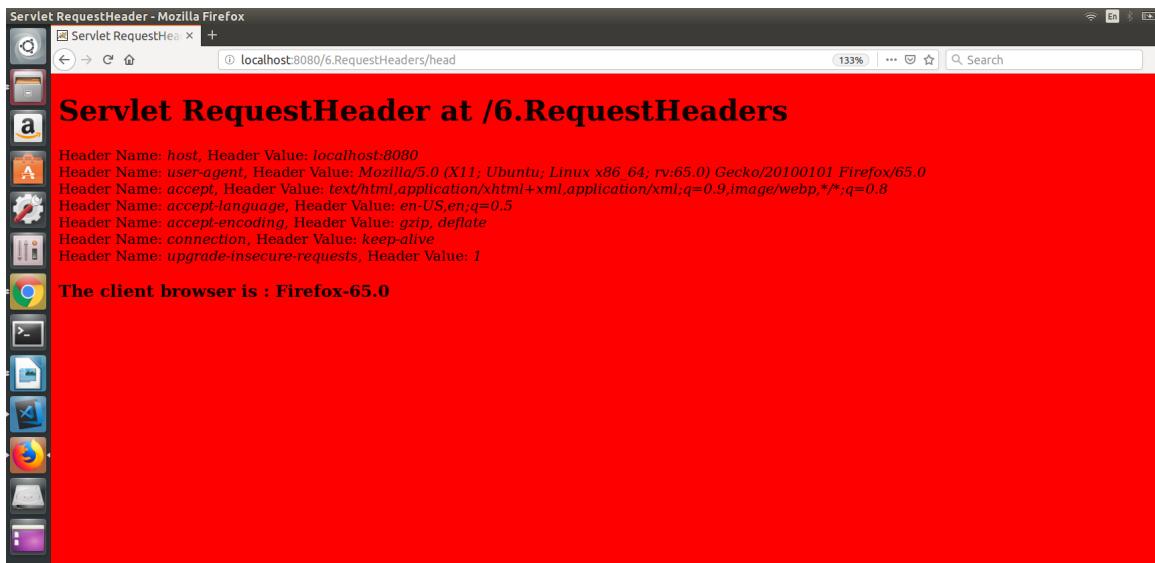
**web.xml**

```

<web-app>
    <servlet>
        <servlet-name>myservlet</servlet-name>
        <servlet-class>RequestHeader</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>myservlet</servlet-name>

```

```
<url-pattern>/head</url-pattern>
</servlet-mapping>
</web-app>
```

**OUTPUT :****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 :****Form.html**

```
<html>
    <body>
        <form action="bad" method="GET">
            Username :<input type="text" name="username"><br>
            Regno :<input type="text" name="regno"><br>
            E-mail :<input type="text" name="email"><br>
            <input type="submit" name="submit" value="SUBMIT">
        </form>
    </body>
</html>
```

**MyServlet.java**

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

public class MyServlet extends HttpServlet {

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response) throws
    ServletException, IOException
    {
        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        String user = request.getParameter("username");
        String regno = request.getParameter("regno");
        String email = request.getParameter("email");

        if (email.contains("@") == false)
            response.sendError(400, "Bad Request");
        else
```

```

    {
        out.println(user + "<br>");
        out.println(regno + "<br>");
        out.println(email + "<br>");

    }
}

```

**web.xml**

```

<web-app>
    <servlet>
        <servlet-name>myservlet</servlet-name>
        <servlet-class>MyServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>myservlet</servlet-name>
        <url-pattern>/bad</url-pattern>
    </servlet-mapping>
</web-app>

```

**OUTPUT :****Sample Ouput 1 :**

The figure consists of two screenshots of a Google Chrome browser window. The top screenshot shows a form with three input fields and one submit button. The bottom screenshot shows the results of the form submission.

**Screenshot 1 (Form):**

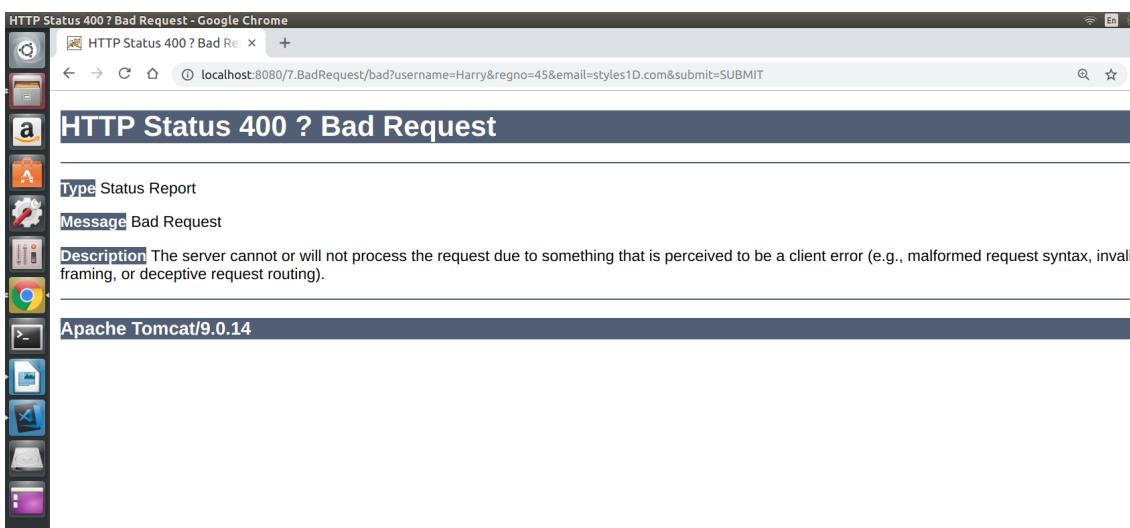
- Username : Harry
- Regno : 45
- E-mail : styles@1D.com
- SUBMIT

**Screenshot 2 (Results):**

- Harry
- 45
- styles@1D.com

## Sample Output 2 :

A screenshot of a Google Chrome browser window. The address bar shows "localhost:8080/7.BadRequest/Form.html". The page contains a form with three input fields: "Username : Harry", "Regno : 45", and "E-mail : styles1D.com", followed by a "SUBMIT" button. The browser's sidebar on the left shows various icons.

**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 :****Form.html**

```
<html>
    <body>
        <form action="pg1" method="GET">
            Username :<input type="text" name="username"><br>
            Regno :<input type="text" name="regno"><br>
            E-mail :<input type="text" name="email"><br>

            <input type="submit" name="submit" value="SUBMIT">
        </form>
    </body>
</html>
```

**Servlet1.java**

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

public class MyServlet extends HttpServlet {

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        String user = request.getParameter("username");
        String regno = request.getParameter("regno");
        String email = request.getParameter("email");

        if (email.contains("@") == false) {
            RequestDispatcher rd =
                request.getRequestDispatcher("/pg2");
            rd.forward(request, response);
        } else {
            out.println(user + "<br>");
            out.println(regno + "<br>");
            out.println(email + "<br>");
        }
    }
}
```

```

        }
    }
}
```

**Servlet2.java**

```

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

public class MyServlet1 extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response) throws ServletException,
                      IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("@ missing");

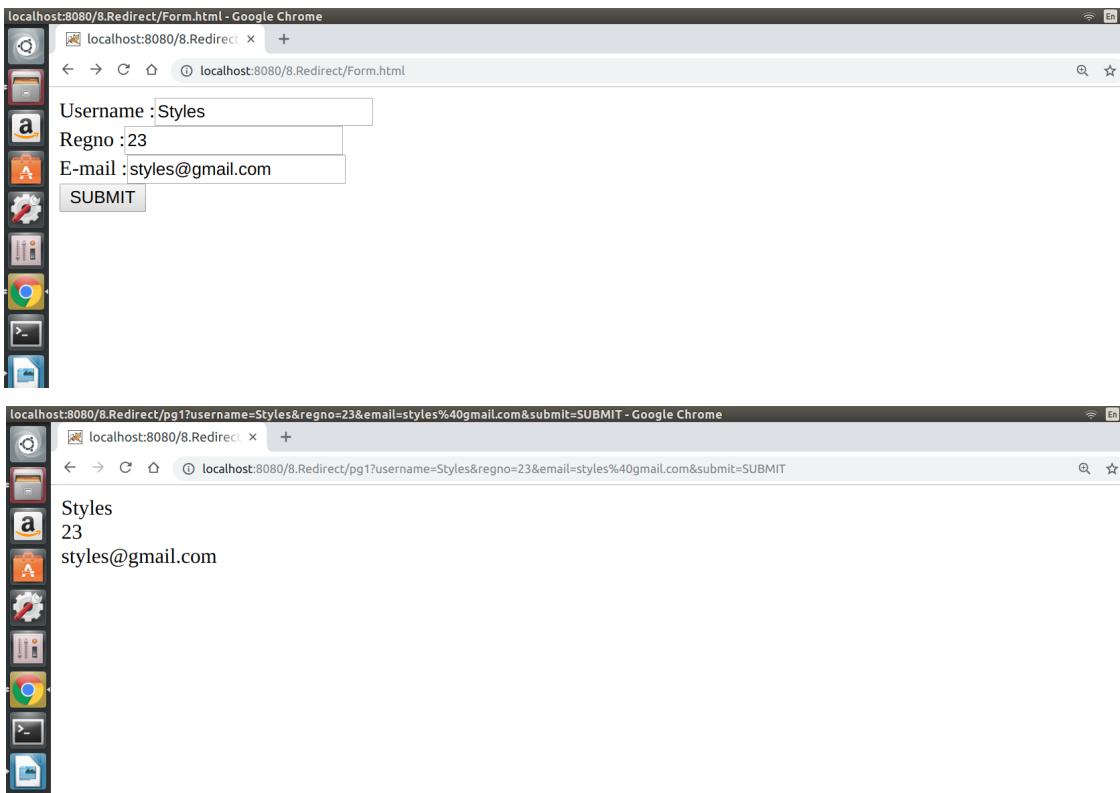
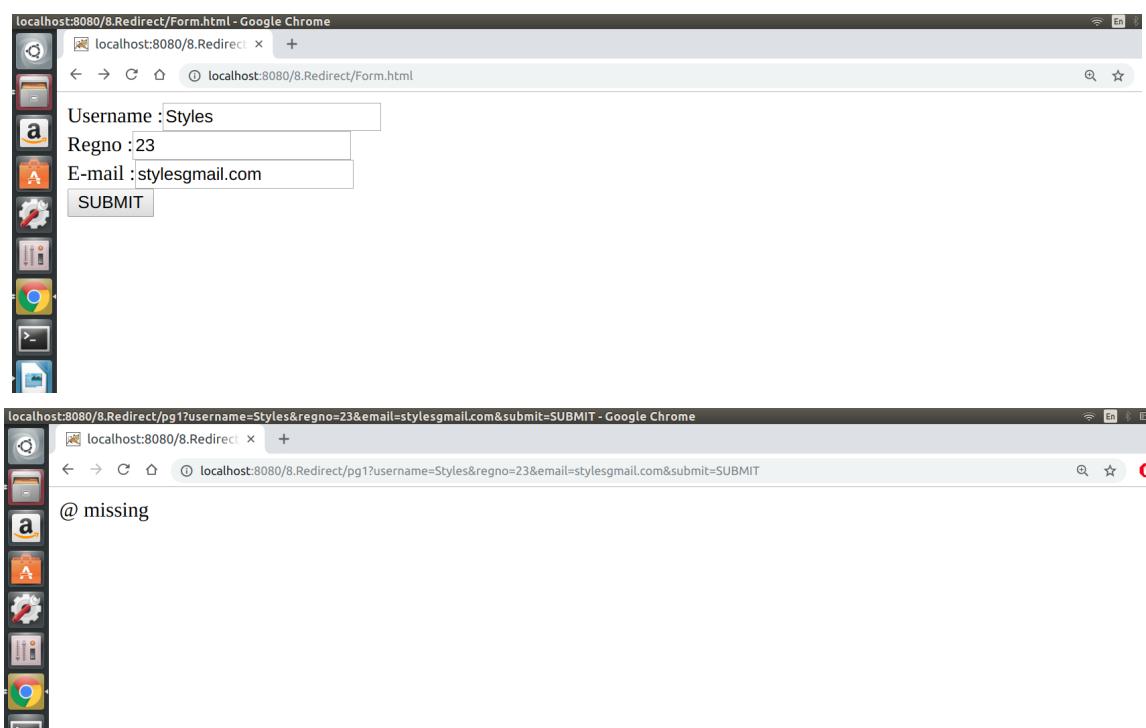
    }
}
```

**web.xml**

```

<web-app>
    <servlet>
        <servlet-name>myservlet</servlet-name>
        <servlet-class>Servlet1</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>myservlet</servlet-name>
        <url-pattern>/pg1</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>myservlet1</servlet-name>
        <servlet-class>Servlet2</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>myservlet1</servlet-name>
        <url-pattern>/pg2</url-pattern>
    </servlet-mapping>
</web-app>
```

**OUTPUT :****Sample Output 1 :****Sample Output 2 :****Result:**

Thus the program is executed and output is obtained.

**CS7612 WEB TECHNOLOGY LAB**

**Y Manaswini 2016503555**

**Date - 15/02/19**

**Experiment - 7**

# **AJAX, JSON, JQUERY**

**PROGRAM 1:****AIM :**

To create a DTD for a catalog of cars, here each car has the child 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 :**

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="Cars.xsl"?>
<!DOCTYPE note [
    <!ELEMENT automobile (car+)>
    <!ELEMENT car
(make,model,year,color,engine,no_of_doors,transmission_type,access
ories)>
        <!ELEMENT engine      (no_of_cylinders,fuel_systems)>
        <!ATTLIST accessories radio CDATA #REQUIRED >
        <!ATTLIST accessories air_condition CDATA #REQUIRED >
        <!ATTLIST accessories power_window CDATA #REQUIRED >
        <!ATTLIST accessories power_steering CDATA #REQUIRED >
        <!ATTLIST accessories power_brake CDATA #REQUIRED >
]>
<catalog>
    <car>
        <make>Maruti</make>
        <model>800</model>
        <year>1999</year>
        <color>Black</color>
        <engine>
            <number_of_cylinders>1</number_of_cylinders>
            <fuel_system>fuel injected</fuel_system>
        </engine>
        <number_of_doors>4</number_of_doors>
        <transmission_type>BB</transmission_type>
        <accessories radio="Yes" air_conditioning="Yes"
                    power_window="No" power_steering="Yes"
                    power_brake="Yes"></accessories>
    </car>
    <car>
        <make>Toyota</make>
        <model>Camry</model>
        <year>2018</year>
        <color>Matte Black</color>
        <engine>
            <number_of_cylinders>1</number_of_cylinders>
```

```

        <fuel_system>fuel injected</fuel_system>
    </engine>
    <number_of_doors>4</number_of_doors>
    <transmission_type>BB</transmission_type>
    <accessories radio="Yes" air_conditioning="Yes"
                power_window="No" power_steering="Yes"
                power_brake="Yes"></accessories>
</car>

<car>
    <make>Toyota</make>
    <model>Corolla</model>
    <year>2016</year>
    <color>Silver</color>
    <engine>
        <number_of_cylinders>1</number_of_cylinders>
        <fuel_system>fuel injected</fuel_system>
    </engine>
    <number_of_doors>4</number_of_doors>
    <transmission_type>BB</transmission_type>
    <accessories radio="Yes" air_conditioning="Yes"
                power_window="No" power_steering="Yes"
                power_brake="Yes"></accessories>
</car>
</catalog>

```

**OUTPUT :**

The XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<catalog>
    - <car>
        <make>Maruti</make>
        <model>800</model>
        <year>1999</year>
        <color>Black</color>
        - <engine>
            <number_of_cylinders>1</number_of_cylinders>
            <fuel_system>fuel injected</fuel_system>
        </engine>
        <number_of_doors>4</number_of_doors>
        <transmission_type>BB</transmission_type>
        <accessories radio="Yes" air_conditioning="Yes"
                    power_window="No" power_steering="Yes" power_brake="Yes"/>
    </car>
    - <car>
        <make>Toyota</make>
        <model>Camry</model>
        <year>2018</year>
        <color>White Black</color>
        - <engine>
            <number_of_cylinders>1</number_of_cylinders>
            <fuel_system>fuel injected</fuel_system>
        </engine>
        <number_of_doors>4</number_of_doors>
        <transmission_type>BB</transmission_type>
        <accessories radio="Yes" air_conditioning="Yes" power_window="No" power_steering="Yes" power_brake="Yes"/>
    </car>
    - <car>
        <make>Toyota</make>
        <model>Corolla</model>
        <year>2016</year>
        <color>Silver</color>
        - <engine>
            <number_of_cylinders>1</number_of_cylinders>
            <fuel_system>fuel injected</fuel_system>
        </engine>
        <number_of_doors>4</number_of_doors>
        <transmission_type>BB</transmission_type>
        <accessories radio="Yes" air_conditioning="Yes" power_window="No" power_steering="Yes" power_brake="Yes"/>
    </car>
    - <car>
        <make>Toyota</make>
        <model>Innova</model>
        <year>2015</year>
        <color>Green</color>
        - <engine>
            <number_of_cylinders>1</number_of_cylinders>
            <fuel_system>fuel injected</fuel_system>
        </engine>
        <number_of_doors>4</number_of_doors>
        <transmission_type>BB</transmission_type>
        <accessories radio="Yes" air_conditioning="Yes" power_window="No" power_steering="Yes" power_brake="Yes"/>
    </car>
</catalog>

```

**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 :****Cars.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="Cars.xsl"?>
<!DOCTYPE note [
    <!ELEMENT automobile (car+)>
    <!ELEMENT car
(make,model,year,color,engine,no_of_doors,transmission_type,access
ories)>
    <!ELEMENT engine      (no_of_cylinders,fuel_systems)>
    <!ATTLIST accessories radio CDATA #REQUIRED >
    <!ATTLIST accessories air_condition CDATA #REQUIRED >
    <!ATTLIST accessories power_window CDATA #REQUIRED >
    <!ATTLIST accessories power_steering CDATA #REQUIRED >
    <!ATTLIST accessories power_brake CDATA #REQUIRED >
]>
<catalog>
    <car>
        <make>Maruti</make>
        <model>800</model>
        <year>1999</year>
        <color>Black</color>
        <engine>
            <number_of_cylinders>1</number_of_cylinders>
            <fuel_system>fuel injected</fuel_system>
        </engine>
        <number_of_doors>4</number_of_doors>
        <transmission_type>BB</transmission_type>
        <accessories radio="Yes" air_conditioning="Yes"
                      power_window="No" power_steering="Yes"
                      power_brake="Yes"></accessories>
    </car>

    <car>
        <make>Toyota</make>
        <model>Camry</model>
        <year>2018</year>
        <color>Matte Black</color>
        <engine>
            <number_of_cylinders>1</number_of_cylinders>
            <fuel_system>fuel injected</fuel_system>
```

```

        </engine>
        <number_of_doors>4</number_of_doors>
        <transmission_type>BB</transmission_type>
        <accessories radio="Yes" air_conditioning="Yes"
                     power_window="No" power_steering="Yes"
                     power_brake="Yes"></accessories>
    </car>

    <car>
        <make>Toyota</make>
        <model>Corolla</model>
        <year>2016</year>
        <color>Silver</color>
        <engine>
            <number_of_cylinders>1</number_of_cylinders>
            <fuel_system>fuel injected</fuel_system>
        </engine>
        <number_of_doors>4</number_of_doors>
        <transmission_type>BB</transmission_type>
        <accessories radio="Yes" air_conditioning="Yes"
                     power_window="No" power_steering="Yes"
                     power_brake="Yes"></accessories>
    </car>

    <car>
        <make>Toyota</make>
        <model>Innova</model>
        <year>2015</year>
        <color>Green</color>
        <engine>
            <number_of_cylinders>1</number_of_cylinders>
            <fuel_system>fuel injected</fuel_system>
        </engine>
        <number_of_doors>4</number_of_doors>
        <transmission_type>BB</transmission_type>
        <accessories radio="Yes" air_conditioning="Yes"
                     power_window="No" power_steering="Yes"
                     power_brake="Yes"></accessories>
    </car>

</catalog>

```

**Cars.xsl**

```

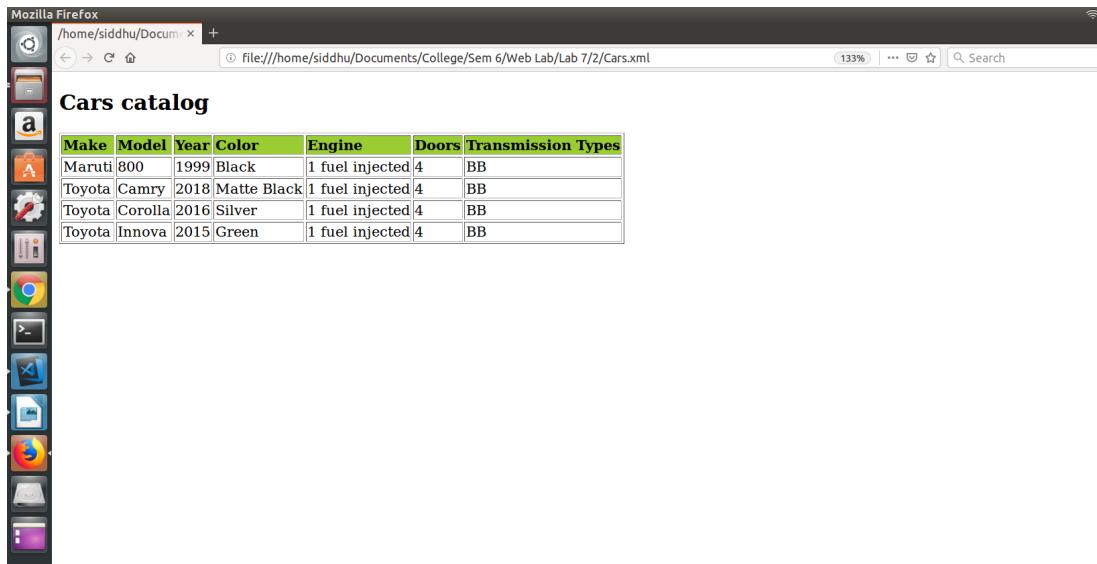
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <body>
        <h2>Cars catalog</h2>

```

```

<table border="1">
    <tr bgcolor="#9acd32">
        <th style="text-align:left">Make</th>
        <th style="text-align:left">Model</th>
        <th style="text-align:left">Year</th>
        <th style="text-align:left">Color</th>
        <th style="text-align:left">Engine</th>
        <th style="text-align:left">Doors</th>
        <th style="text-align:left">Transmission
            Types</th>
        <th style="text-
            align:left">Accessories</th>
        <!-- <th style="text-align:left">-->
    </tr>
    <xsl:for-each select="catalog/car">
        <tr>
            <td>
                <xsl:value-of select="make"/>
            </td>
            <td>
                <xsl:value-of select="model"/>
            </td>
            <td>
                <xsl:value-of select="year"/>
            </td>
            <td>
                <xsl:value-of select="color"/>
            </td>
            <td>
                <xsl:value-of select="engine"/>
            </td>
            <td>
                <xsl:value-of
                    select="number_of_doors"/>
            </td>
            <td>
                <xsl:value-of
                    select="transmission_type"/>
            </td>
            <td>
                <xsl:value-of
                    select="accessories"/>
            </td>
        </tr>
    </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

**OUTPUT :**

A screenshot of a Mozilla Firefox browser window. The address bar shows the URL: file:///home/siddhu/Documents/College/Sem 6/Web Lab/Lab 7/2/Cars.xml. The main content area displays a table titled "Cars catalog". The table has columns: Make, Model, Year, Color, Engine, Doors, and Transmission Types. The data in the table is as follows:

Make	Model	Year	Color	Engine	Doors	Transmission Types
Maruti	800	1999	Black	1 fuel injected	4	BB
Toyota	Camry	2018	Matte Black	1 fuel injected	4	BB
Toyota	Corolla	2016	Silver	1 fuel injected	4	BB
Toyota	Innova	2015	Green	1 fuel injected	4	BB

**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 menu 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;
            }
        </script>
    </head>
    <body>
        <div id="display">
            <ul id="list">
                <li>Ford</li>
                <li>BMW</li>
                <li>VW</li>
            </ul>
        </div>
    </body>
</html>
```

```

        }
        function displayUtil(name){

            document.getElementById("list").style.visibility =
                "hidden";

            document.getElementById("list").style.display =
                "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].nodeValue == 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").innerHTML =
                content;

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

    </script>
</head>
<body onload = "getData()">
```

```

<div id="list">
</div>
<div id = "display">
</div>
</body>
</html>

```

**index.xml**

```

<catalog>
    <car>
        <make>Indian</make>
        <model>BMW</model>
        <color>Blue</color>
        <year>1992</year>
        <engine>
            <number_of_cylinders>6</number_of_cylinders>
            <fuel_system>fuel injected</fuel_system>
        </engine>
        <number_of_doors>4</number_of_doors>
        <transmission_type>automatic</transmission_type>
        <accessories radio = "yes" air_conditioning = "no"
                     power_windows = "yes" power_steering = "no"
                     power_brakes = "yes"></accessories>
    </car>

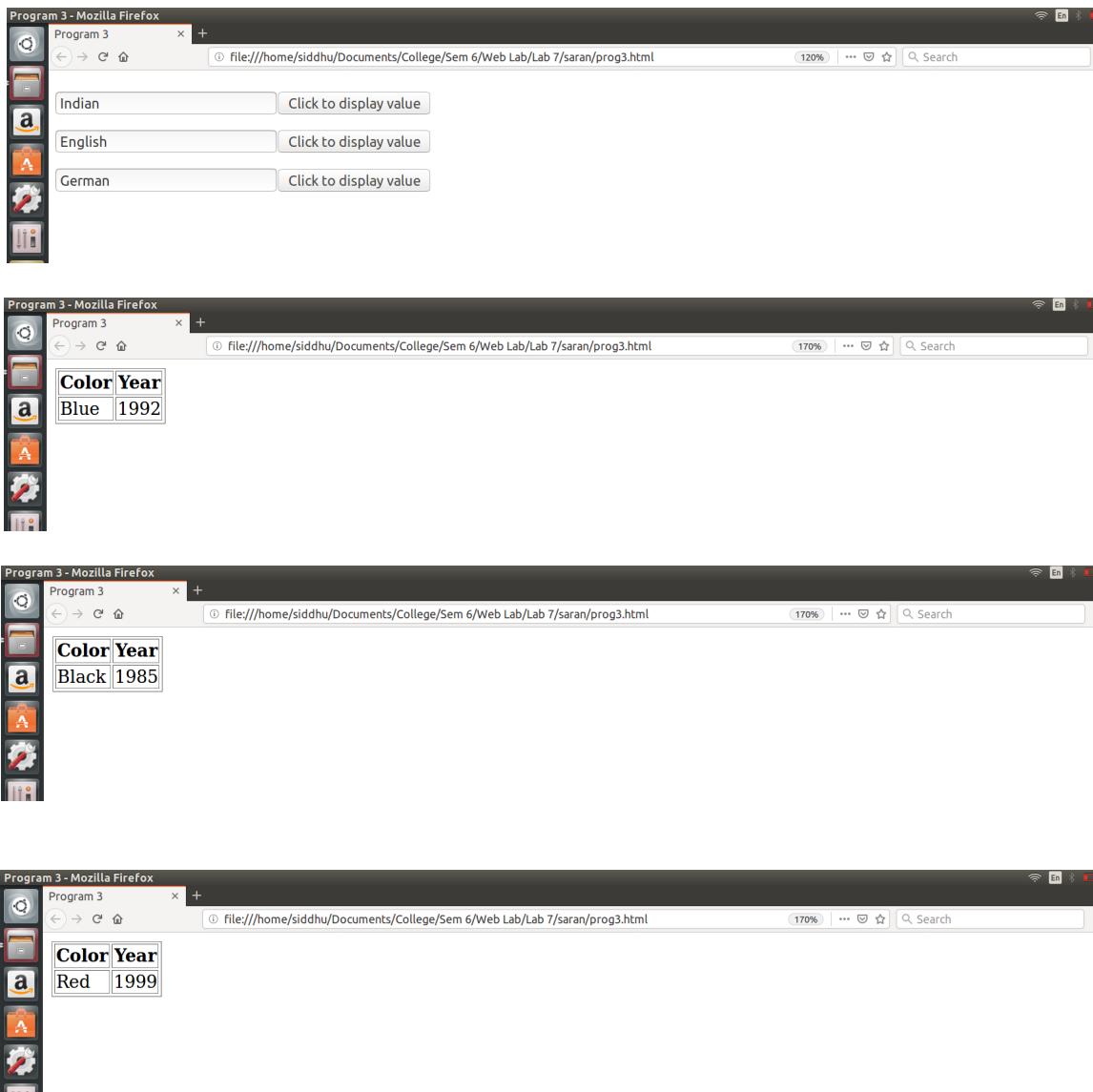
    <car>
        <make>English</make>
        <model>Benz</model>
        <color>Black</color>
        <year>1985</year>
        <engine>
            <number_of_cylinders>8</number_of_cylinders>
            <fuel_system>carburated</fuel_system>
        </engine>
        <number_of_doors>2</number_of_doors>
        <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>Volkswagen</model>
        <color>Red</color>
        <year>1999</year>
        <engine>
            <number_of_cylinders>4</number_of_cylinders>
            <fuel_system>carburated</fuel_system>
        </engine>
        <number_of_doors>6</number_of_doors>
    </car>
</catalog>

```

```

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

```

**OUTPUT :****Result:**

Thus the program is executed and output is obtained.

**PROGRAM 4:****Aim:**

To modify the example application 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_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)>
<!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>

    <customer>
        <id>1</id>
        <name>Harry</name>
        <address>
            <city>Albany</city>
            <state>New York</state>
            <zip>10001</zip>
        </address>
    </customer>

    <customer>
        <id>2</id>
        <name>Lily Potter</name>
        <address>
            <city>Albany</city>
            <state>New York</state>
            <zip>10001</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 inputData() {
            var id = document.getElementById("id");
            var name = document.getElementById("name");
            var city = document.getElementById("city");
            var state = document.getElementById("state");
            var zip = document.getElementById("zip");

            var xhttp = new XMLHttpRequest();
            xhttp.onreadystatechange = function() {
                if(this.readyState == 4 && this.status ==
                    200) {
                    alert(this.responseText);

document.getElementById("demo").innerHTML = this.responseText;
                }
            };
            xhttp.open("GET", "prog4.php?id=" + id, true);
            xhttp.send();
        }
        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");

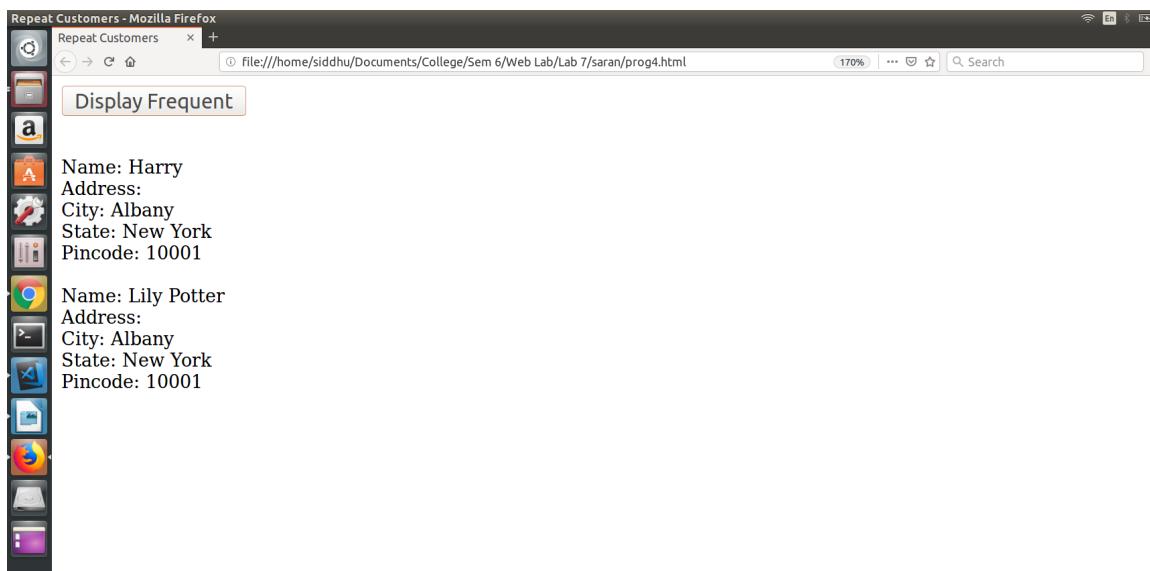
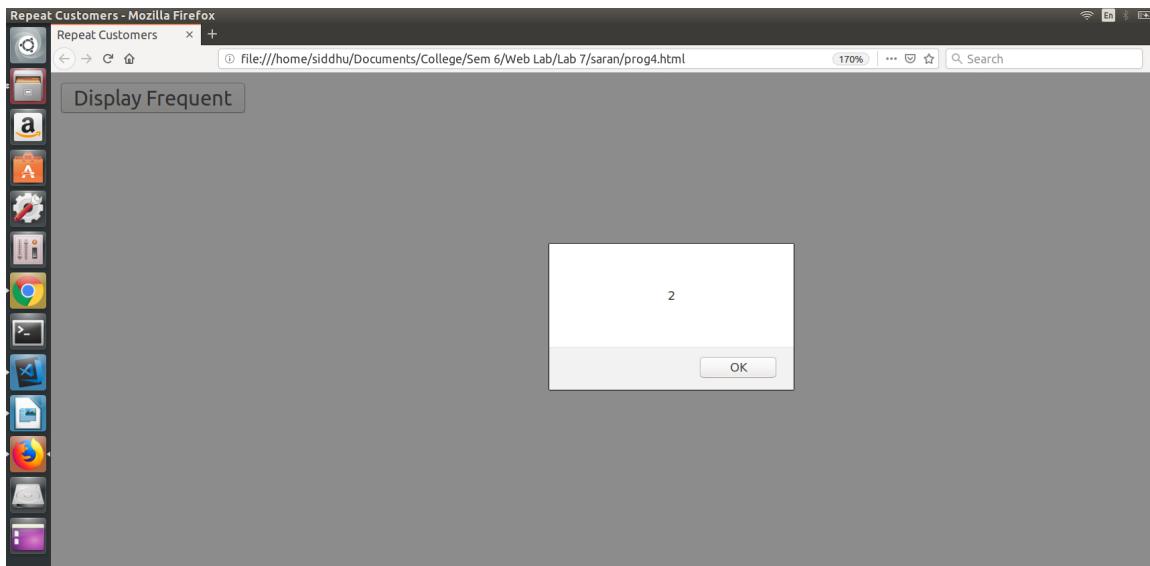
            for (var j = 0; j < y.length; j++) {
                content += "City: " +
                    y[j].getElementsByTagName("city")[0].childNodes[0].nodeValue +
                    " <br>";
                content += "State: " +
                    y[j].getElementsByTagName("state")[0].childNodes[0].nodeValue +
                    " <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**

```

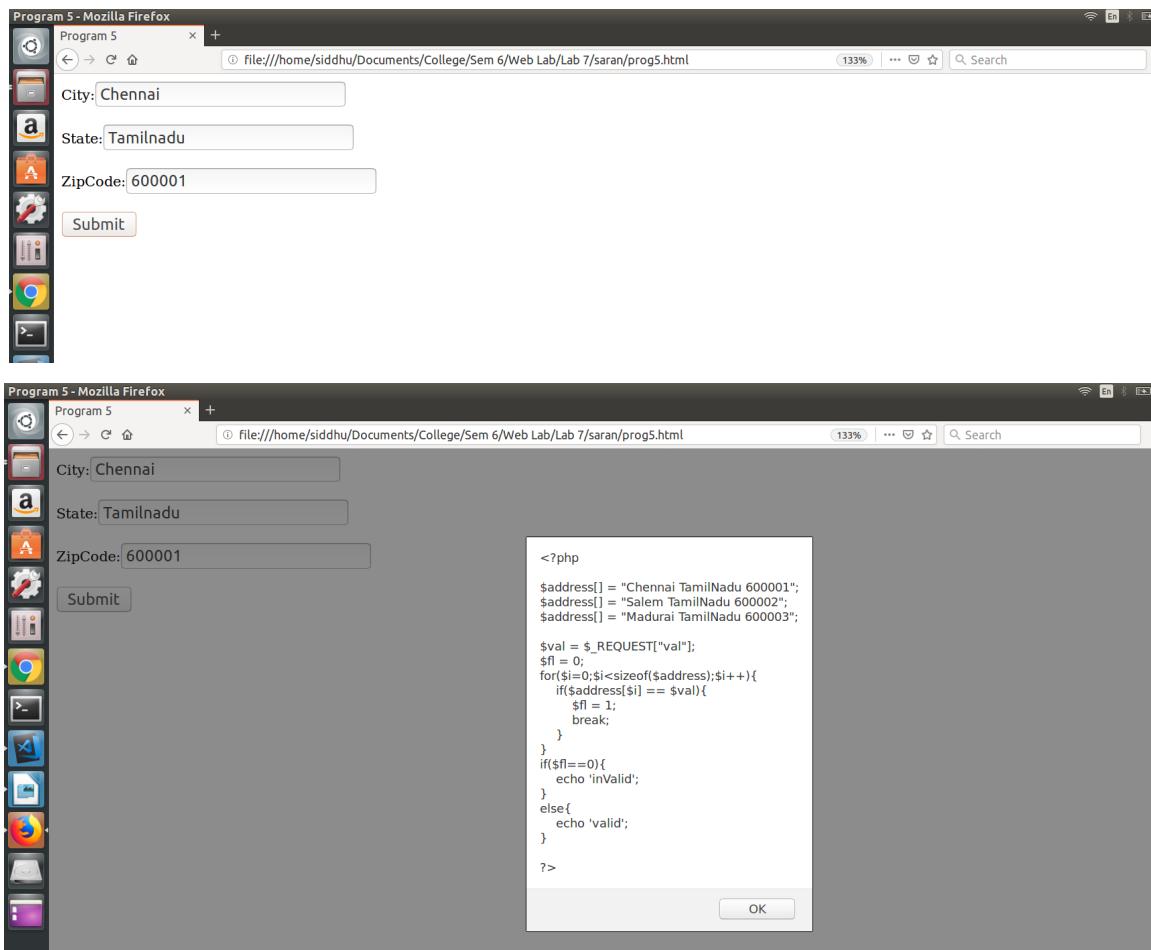
<html>
    <head>
        <title>Program 5</title>
        <meta charset="utf-8">
        <script>
            function myfunc(){
                var x = document.getElementById("city").value;
                var y =
                    document.getElementById("state").value;
                var z =
                    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){
                    }
                };
                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 :****Result:**

Thus the program is executed and output is obtained.

Experiment - 8

# **XML, XSLT, SAX, XPATH, DTD, PARSERS**

**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 (with 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 atleast 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>
    </patient>

```

```

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

</hospital>

```

**OUTPUT :**

```

- <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>

```

**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 thet document.

**PROGRAM :**

```

name,sex, room-number, age, social-security-number{
    color: blue;
    font-family: verdana;
    font-size: 200%;
    display : block;
}
hospital {
    border: 1px solid powderblue;
    display : block;
}
sex{
    color: white;
    background-color : gray;
    width: 200%;
}
primary-insurance-company, secondary-insurance-company{
    font-size : 25px;
    font-weight : bold;
}

```

**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 :**

```

<?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="blue">
    <h2 align="center" style="color:yellow font-
family:helvetica">Patient History</h2>
    <table border="1" align="center">
        <tr bgcolor="green">
            <th style="text-align:left font-size:100%">FName</th>
            <th style="text-align:left font-size:100%">MName</th>
            <th style="text-align:left font-size:100%">LName</th>
            <th style="text-align:left font-size:100%">Sex</th>
            <th style="text-align:left font-size:100%">Room-
number</th>
            <th style="text-align:left font-size:100%">Age</th>
            <th style="text-align:left font-size:100%">Social
Security Number</th>
            <th style="text-align:left font-size:100%">SIC:ID</th>
            <th style="text-align:left font-
size:100%">SIC:GrpID</th>
            <th style="text-align:left font-
size:100%">SIC:Phone</th>
            <th style="text-align:left font-
size:100%">SIC:Address</th>
            <th style="text-align:left font-size:100%">SIC:ID</th>
            <th style="text-align:left font-
size:100%">SIC:GrpID</th>
            <th style="text-align:left font-
size:100%">SIC:Phone</th>
            <th style="text-align:left font-
size:100%">SIC:Address</th>
        </tr>
        <xsl:for-each select="hospital/patient[name='XYZ']">
            <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>
            </tr>
        </xsl:for-each>
    </table>
</body>
</html>

```

```

        <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>
```

**OUTPUT:**

FName	MName	LName	Sex	Room-number	Age	Social Security Number	SIC:ID	SIC:GrpID	SIC:Phone	SIC:Address	SIC:ID	SIC:GrpID	SIC:Phone	SIC:Address
A	B	C	Male	1	30	1234	11	21	1111111111	Chennai	21	11	9999999999	Madurai

**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 :**

```
<?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="blue">
    <h2 align="center" style="color:blue font-
family:helvetica">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>
```

**OUTPUT:**

file:///home/manaswini/Desktop/web/xml/prog5.xml

Patient History															
FName	MName	LName	Sex	Room-number	Age	Social Security Number	SIC:ID	SIC:GrpID	SIC:Phone	SIC:Address	SIC:ID	SIC:GrpID	SIC:Phone	SIC:Address	
A	B	C	Male	1	40	1234	11	21	1111111111	Chennai	21	41	2222222222	Madurai	
x	Y	Z	Male	2	20	1234	12	31	1117711111	Mumbai	22	51	2222222222	Delhi	
S	T	U	Female	4	27	4321	32	39	7777777777	Hyderabad	30	78	9653982534	Chennai	

**Result:**

Thus the program is executed and output is obtained.