

# Socket Programming In Java

**Program 1****AIM :**

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

**PROGRAM :**

```
import java.io.*;
import java.net.*;
class ping
{
    public static void sendPingRequest(String ipAddress)
        throws UnknownHostException, IOException
    {
        InetAddress geek = InetAddress.getByName(ipAddress);
        System.out.println("Sending Ping Request to " +
ipAddress);
        if (geek.isReachable(5000))
            System.out.println("Host is reachable");
        else
            System.out.println("Sorry ! We can't reach to this
host");
    }
    public static void main(String[] args)
        throws UnknownHostException, IOException
    {
        String ipAddress = "127.0.0.1";
        sendPingRequest(ipAddress);

        ipAddress = "192.168.117.70";
        sendPingRequest(ipAddress);
    }
}
```

**OUTPUT:**

```
Sending Ping Request to 127.0.0.1
Host is reachable
Sending Ping Request to 192.168.117.70
Sorry ! We can't reach to this host
```

**RESULT:**

Thus the program is executed and output is obtained

**Program 2****AIM :**

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

**PROGRAM :**

Server:

```
import java.net.*;
import java.io.*;
import java.util.*;
import java.time.format.DateTimeFormatter;
import java.time.LocalDateTime;
public class Server
{
    private Socket socket = null;
    private ServerSocket server = null;
    private DataInputStream in = null;
    private DataOutputStream out = null;
    public Server(int port)
    {
        try
        {
            server = new ServerSocket(port);
            System.out.println("Server started");
            System.out.println("Waiting for a client ...");
            socket = server.accept();
            System.out.println("Client accepted");
            in = new DataInputStream(new
BufferedInputStream(socket.getInputStream()));
            out = new
DataOutputStream(socket.getOutputStream());
            String line = "";
            line = in.readUTF();
            Date d=new Date();
            out.writeUTF(d.toString());
            System.out.println("Closing connection");
            socket.close();
            in.close();
        }
        catch(IOException i)
        {
```

```
        System.out.println(i);
    }
}
public static void main(String args[])
{
Server server = new Server(5000);
}
}
```

**Client:**

```
import java.net.*;
import java.io.*;
import java.util.*;
public class Client
{
    private Socket socket = null;
    private DataInputStream input = null;
    private DataOutputStream out = null;
    public Client(String address, int port)
    {
        try
        {
            socket = new Socket(address, port);
            System.out.println("Connected");
            input = new DataInputStream(new
BufferedInputStream(socket.getInputStream()));
            out = new
DataOutputStream(socket.getOutputStream());
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        String line = "", r = "";
        Scanner sc = new Scanner(System.in);
try{

    line = sc.next();
    out.writeUTF(line);
    r = input.readUTF();
    System.out.println(r);
}
}
```

```
        catch(Exception e)
        {
            System.out.println(e);
        }

        try
        {
            input.close();
            out.close();
            socket.close();
        }
        catch(IOException i)
        {
            System.out.println(i);
        }
    }
public static void main(String args[])
{
    Client client = new Client("127.0.0.1", 5000);
}
}
```

**OUTPUT:**

Server:  
Server started  
Waiting for a client ...  
Client accepted  
Closing connection

Client:  
Connected  
date and time  
Wed Feb 13 23:00:28 IST 2019

**RESULT:**

Thus the program is executed and output is obtained.

**Program 3****AIM:**

To write a ping pong client and server application. When a client sends a ping message to the server, the server will

respond with a pong message. Other messages sent by the client can be safely dropped by the client.

**PROGRAM :**

Server:

```
import java.net.*;
import java.io.*;
import java.util.*;
public class p3server
{
    private Socket socket = null;
    private ServerSocket server = null;
    private DataInputStream in = null;
    private DataOutputStream out = null;
    public p3server(int port)
    {
        try
        {
            server = new ServerSocket(port);
            System.out.println("Server started");
            System.out.println("Waiting for a client ...");
            socket = server.accept();
            System.out.println("Client accepted");
            in = new DataInputStream(new
BufferedInputStream(socket.getInputStream()));
            out = new
DataOutputStream(socket.getOutputStream());
            String line = "";
            Scanner sc=new Scanner(System.in);
            String r="";
            while (!line.equals("Over"))
            {
                try
                {
                    line = in.readUTF();
                    System.out.println(line);
                    r="pong";
                    out.writeUTF(r);
                }
                catch(IOException i)
                {
                    System.out.println(i);
                }
            }
        }
    }
}
```

```

        }
    }
    System.out.println("Closing connection");
    socket.close();
    in.close();
}
catch(IOException i)
{
    System.out.println(i);
}
}
public static void main(String args[])
{
    p3server server = new p3server(5000);
}
}

```

**Client:**

```

import java.net.*;
import java.io.*;
import java.util.*;
public class client
{
    private Socket socket = null;
    private DataInputStream input = null;
    private DataOutputStream out = null;
    public p3client(String address, int port)
    {
        try
        {
            socket = new Socket(address, port);
            System.out.println("Connected");
            input = new DataInputStream(new
BufferedInputStream(socket.getInputStream()));
            out = new
DataOutputStream(socket.getOutputStream());
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        String line = "", r = "";

```

```
Scanner sc = new Scanner(System.in);
while (!line.equals("Over"))
{
    try
    {
        line = sc.next();
        if(!line.equals("ping"))continue;
        out.writeUTF(line);
        r="";
        r = input.readUTF();
        System.out.println(r);
    }
    catch(IOException i)
    {
        System.out.println(i);
    }
}
try
{
    input.close();
    out.close();
    socket.close();
}
catch(IOException i)
{
    System.out.println(i);
}
}
public static void main(String args[])
{
    p3client client = new p3client("127.0.0.1", 5000);
}
```

**OUTPUT:**

Server:  
Server started  
Waiting for a client ...  
Client accepted  
ping  
ping

**Client:**  
Connected  
ping  
pong  
madhu  
ping  
pong

**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 :****Server:**

```
import java.net.*;  
import java.io.*;  
import java.util.*;  
public class p4server  
{  
    private Socket socket = null;  
    private ServerSocket server = null;  
    private DataInputStream in = null;  
    private DataOutputStream out = null;  
    public p4server(int port)  
    {  
        try  
        {  
            server = new ServerSocket(port);  
            System.out.println("Server started");  
            System.out.println("Waiting for a client ...");  
            socket = server.accept();  
            System.out.println("Client accepted");  
            in = new DataInputStream(new  
BufferedInputStream(socket.getInputStream()));
```

```

        out = new
DataOutputStream(socket.getOutputStream());
        String line = "";
Scanner sc=new Scanner(System.in);
String r="";
        while (!line.equals("Over"))
{
    try
{
    line = in.readUTF();
    System.out.println(line);
    out.writeUTF(line.toUpperCase());
}
catch(IOException i)
{
    System.out.println(i);
}
System.out.println("Closing connection");
socket.close();
in.close();
}
catch(IOException i)
{
    System.out.println(i);
}
}
public static void main(String args[])
{
    p4server server = new p4server(5000);
}
}

```

**Client:**

```

import java.net.*;
import java.io.*;
import java.util.*;
public class p4client
{
    private Socket socket = null;
    private DataInputStream input = null;
    private DataOutputStream out = null;

```

```
public p4client(String address, int port)
{
    try
    {
        socket = new Socket(address, port);
        System.out.println("Connected");
        input = new DataInputStream(new
BufferedInputStream(socket.getInputStream()));
        out = new
DataOutputStream(socket.getOutputStream());
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    String line = "", r = "";
    Scanner sc = new Scanner(System.in);
    while (!line.equals("Over"))
    {
        try
        {
            line = sc.next();
            out.writeUTF(line);
            r="";
            r = input.readUTF();
            System.out.println("UPPERCASE VERSION:" + r);
        }
        catch(IOException i)
        {
            System.out.println(i);
        }
    }
    try
    {
        input.close();
        out.close();
        socket.close();
    }
    catch(IOException i)
    {
        System.out.println(i);
    }
}
```

```
    }
    public static void main(String args[])
    {
        p4client client = new p4client("127.0.0.1", 5000);
    }
}
```

**OUTPUT:**

Server:  
Server started  
Waiting for a client ...  
Client accepted  
madhu  
Client:  
Connected  
madhu  
UPPERCASE VERSION:MADHU

**RESULT:**

Thus the program is executed and output is obtained.

**Program 5****AIM:**

To implement Broadcasting messages in your lab.

**PROGRAM :**

Server:

```
import java.io.*;
import java.net.*;
import java.util.*;
class broadcastServer{
    public static void main(String[] args){
        try{
            DatagramSocket ds=new DatagramSocket();
            String msg;
            DatagramPacket dp;
            InetAddress ip = InetAddress.getByName("255.255.255.255");
            Scanner sc=new Scanner(System.in);
            byte[ ] buf;
            while(true){
```

```

        System.out.println("Enter the message to
be broadcasted..\n");
        msg=sc.nextLine();
        buf=msg.getBytes();
        dp=new
DatagramPacket(buf,buf.length,ip,5000);
        ds.send(dp);
        if(msg.equals("exit"))
            break;
    }
    ds.close();
}
catch(Exception e){
    e.printStackTrace();
}
}
}

```

**Client:**

```

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

```

```
        System.out.println(s1);
    }
    ds.close();
}
catch(Exception e){
    e.printStackTrace();
}
}
```

**OUTPUT:****Server:**

```
Enter the message to be broadcasted..
hello
```

**RESULT:**

```
Thus the program is executed and output is obtained.
```

# Remote Method Interface

**Program 1****AIM :**

To write a Java program to implement Client Server communication using RPC.

**PROGRAM :****Server:**

```
import java.io.*;
import java.net.*;
class ser
{
    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("Data: "+fun);
            System.out.flush();
        }
    }
}
```

**Client:**

```
import java.io.*;
import java.net.*;
class cli
{
    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 input");  
        temp = keyRead.readLine();  
        sendMessage=temp.toLowerCase();  
        pwrite.println(sendMessage);  
        System.out.flush();  
    }  
}  
}
```

**OUTPUT:**

Server:  
Server ready  
Data: hello

**Client:**

Client ready, type and press Enter key  
Enter input  
hello

**RESULT:**

Thus the program is executed and output is obtained.

**Program 2****AIM :**

To implement Remote command execution using RMI.

**PROGRAM :**

Command.java

```
import java.rmi.*;  
  
public interface Command extends Remote {  
    public void execute(String cmd) throws RemoteException;  
}
```

CommandRemote.java

```
import java.rmi.*;  
import java.rmi.server.*;  
import java.io.*;  
import java.net.*;  
import java.net.UnknownHostException;  
  
public class CommandRemote extends UnicastRemoteObject  
implements Command{  
    CommandRemote() throws RemoteException {  
        super();  
    }  
  
    public void execute(String cmd)  
    {  
        try  
        {  
            Runtime run = Runtime.getRuntime();  
            Process p = run.exec(cmd);  
            System.out.println(p);  
            System.out.println("This " + cmd + " has been  
executed.");  
        }  
        catch(Exception e) {  
            System.out.println(e);  
        }  
    }  
}
```

MyServer.java

```
import java.rmi.*;  
import java.rmi.registry.*;  
  
public class MyServer {  
    public static void main(String[] args) {
```

```
try
{
    Command stub = new CommandRemote();
    Naming.rebind("rmi://localhost:5001/command",
stub);
}
catch(Exception e)
{
    System.out.println(e);
}
}
```

```
MyClient.java
import java.rmi.*;
public class MyClient{
    public static void main(String args[]){
        try{
            Command
stub=(Command)Naming.lookup("rmi://localhost:5001/command");
            stub.execute("google-chrome");
        }
        catch(Exception e){}
    }
}
```

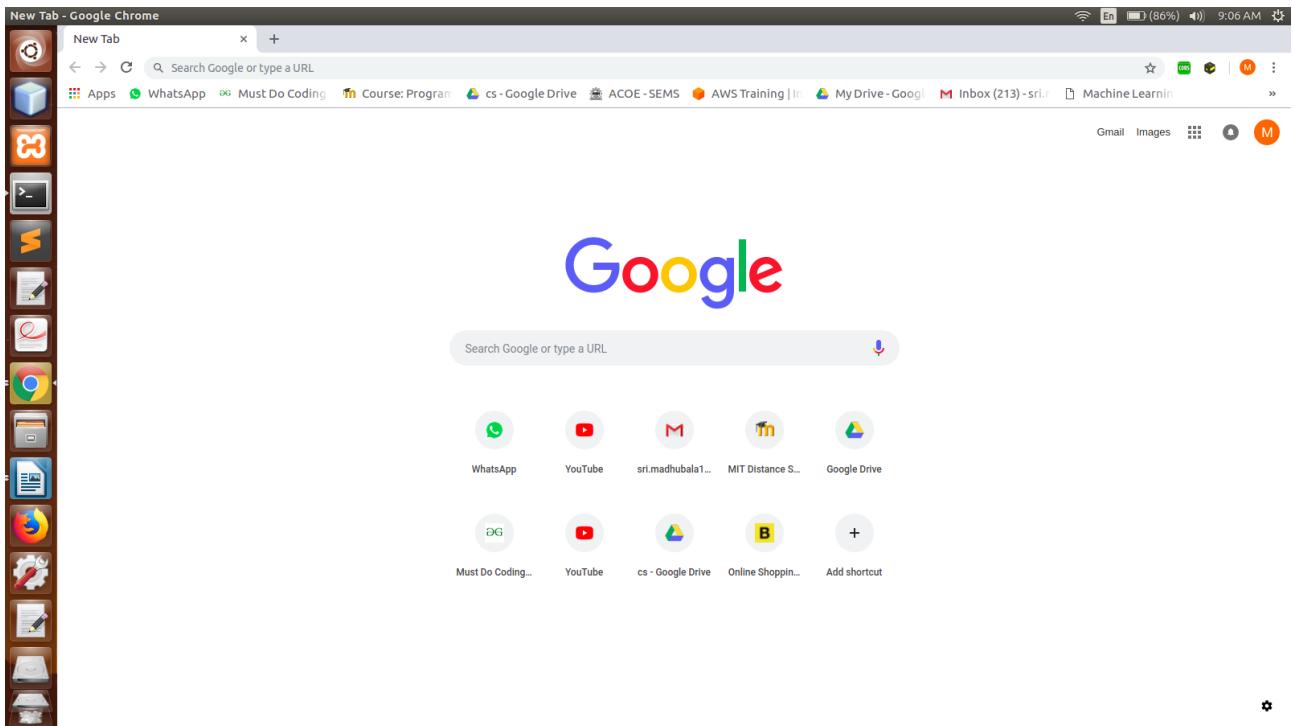
**OUTPUT:**

Server:  
google-chrome

Client:  
Google chrome opened

**RESULT:**

Thus the program is executed and output is obtained.

**Program 3****AIM :**

To create RMI to calculate factorial of given number.

**PROGRAM :**

Factorial.java

```
import java.rmi.*;
public interface Factorial extends Remote{
```

```
    public int add(int x) throws RemoteException;
}
```

FactorialRemote.java

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

```
public class FactorialRemote extends UnicastRemoteObject
implements Adder{
```

```
    FactorialRemote() throws RemoteException{
```

```
    super();
}

public int add(int x){
    int p=1;for(int i=1;i<=x;i++)p*=i;return p;}
}
```

```
MyServer.java
import java.rmi.*;
import java.rmi.registry.*;

public class MyServer{

public static void main(String args[]){
try{

    Factorail stub=new FactorialRemote();
    Naming.rebind("rmi://localhost:5000/sonoo",stub);

}catch(Exception e){System.out.println(e);}
}

}

MyClient.java
import java.rmi.*;
import java.util.*;
public class MyClient{

public static void main(String args[]){
try{

    Factorial stub =
    (Factorial)Naming.lookup("rmi://localhost:5000/sonoo");
    System.out.println("enter number for factorial");
    Scanner sc=new Scanner(System.in);
    int k=sc.nextInt();
    System.out.println(stub.add(k));
}
catch(Exception e){System.out.println(e);}
}
}
```

**OUTPUT:**

```
enter number for factorial
3
6
madhu
```

**RESULT:**

Thus the program is executed and output is obtained.

**Program 4****AIM :**

To create RMI to perform arithmetic operations using RMI.

**PROGRAM :**

Arithmetic.java

```
import java.rmi.*;

public interface Arithmetic extends Remote {
    public int add(int x, int y) throws RemoteException;
    public int sub(int x, int y) throws RemoteException;
    public int mul(int x, int y) throws RemoteException;
    public int div(int x, int y) throws RemoteException;
    public int mod(int x, int y) throws RemoteException;
}
```

ArithmeticRemote.java

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

public class ArithmeticRemote extends UnicastRemoteObject
implements Arithmetic{
    ArithmeticRemote() throws RemoteException {
        super();
    }

    public int add(int x, int y)
    {
        return x+y;
    }
}
```

```
public int sub(int x, int y)
{
    return x-y;
}
public int mul(int x, int y)
{
    return x*y;
}
public int div(int x, int y)
{
    return x/y;
}
public int mod(int x, int y)
{
    return x%y;
}
}
MyServer.java
import java.rmi.*;
import java.rmi.registry.*;

public class MyServer {
    public static void main(String[] args) {
        try
        {
            Arithmetic stub = new ArithmeticRemote();
            Naming.rebind("rmi://localhost:5001/arithmetic",
stub);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

MyClient.java
import java.rmi.*;
public class MyClient{
    public static void main(String args[]){
        try{
```

```

        Arithmetic
stub=(Arithmetic)Naming.lookup("rmi://localhost:5001/arithmet
ic");
        System.out.println(stub.add(4,3));
        System.out.println(stub.sub(4,3));
        System.out.println(stub.mul(4,3));
        System.out.println(stub.div(4,3));
        System.out.println(stub.mod(4,3));
    }
    catch(Exception e){}
}
}

```

**OUTPUT:**

```

Enter the 2 numbers
2
3
The addition answer is 5
The subtraction answer is -1
The multiplication answer is 6
The division answer is 0
The modulus answer is 0

```

**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 :**

```

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{
    }

    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;
    }
}
```

## Server.java

```
import java.rmi.*;
import java.rmi.registry.*;

public class Server{

    public static void main(String args[]){
        try{

            DNS stub=new DNSRemote();
            Naming.rebind("rmi://localhost:5000/sonoo",stub);

        }
        catch(Exception e){System.out.println(e);}
    }
}
```

## Client.java

```
import java.rmi.*;
import java.io.*;
import java.io.*;
import java.net.*;
import java.util.*;
public class Client{

public static void main(String args[]){
try{
    Scanner in = new Scanner(System.in);
    DNS
stub=(DNS)Naming.lookup("rmi://localhost:5000/sonoo");
    System.out.println("Enter the DOMAIN NAME :");
    String str=in.next();
    stub.find(str);
}
catch(Exception e){System.out.println(e);}
}
}
```

**OUTPUT:**

Server:  
216.58.197.46

Client:  
Enter the DOMAIN NAME :  
google.com

**RESULT:**

Thus the program is executed and output is obtained.

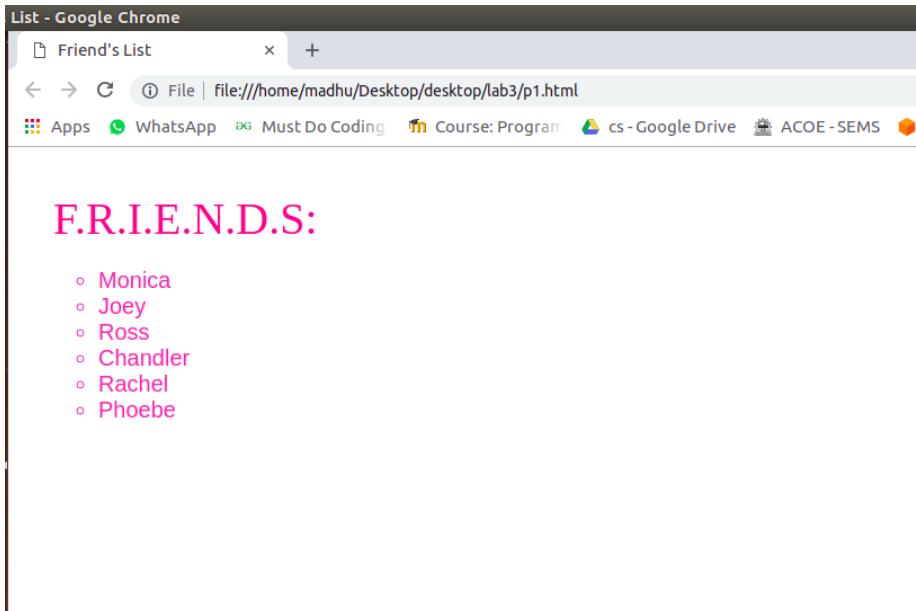
# Client Side Scripting

**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 :**

```
<html>
<meta http-equiv="cache-control" content="no-cache" />
<meta http-equiv="Pragma" content="no-cache" />
<meta http-equiv="Expires" content="-1" />
<head>
<title>
Friend's List
</title>
</head>
<body bgcolor=#888888>
<style>
body{
    font-size: 40px;
    color:DeepPink;
    margin-left:40px;
    margin-top: 40px
}
ul.circle{
    font-size:20px;
    font-family:Arial;
    list-style-type:circle;
    color:HotPink;
}
</style>
F.R.I.E.N.D.S:<br>
<ul class="circle">
<li>Monica
<li>Joey
<li>Ross
<li>Chandler
<li>Rachel
<li>Phoebe
</ul>
</body></html>
```

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

Thus the program is executed and output is obtained

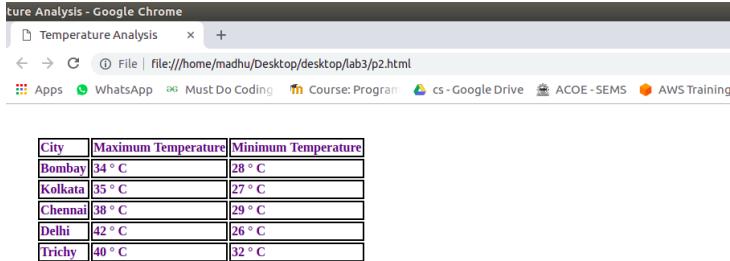
**Program 2****AIM :**

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

**PROGRAM :**

```
<html>
<head>
<meta http-equiv="cache-control" content="no-cache" />
<meta http-equiv="Pragma" content="no-cache" />
<meta http-equiv="Expires" content="-1" />
<title>
Temperature Analysis
</title>
</head>
<body bgcolor=MediumSeaGreen>
<style>
body
{
    font-size: 40px;
```

```
        color:Indigo;
        margin-left:40px;
        margin-top: 40px
    }
    td{
        border:solid 4px black;
    }
</style>
<table>
<tr>
<td>City</td>
<td>Maximum Temperature</td>
<td>Minimum Temperature</td>
</tr>
<tr>
<td>Bombay</td>
<td>34 &#176 C</td>
<td>28 &#176 C</td>
</tr>
<tr>
<td>Kolkata</td>
<td>35 &#176 C</td>
<td>27 &#176 C</td>
</tr>
<tr>
<td>Chennai</td>
<td>38 &#176 C</td>
<td>29 &#176 C</td>
</tr>
<tr>
<td>Delhi</td>
<td>42 &#176 C</td>
<td>26 &#176 C</td>
</tr>
<tr>
<td>Trichy</td>
<td>40 &#176 C</td>
<td>32 &#176 C</td>
</tr>
</table>
</body>
</html>
```

**OUTPUT:**


A screenshot of a Google Chrome window titled "Temperature Analysis - Google Chrome". The address bar shows the file path: "file:///home/madhu/Desktop/Desktop/lab3/p2.html". The tab bar includes "Temperature Analysis", "File | file:///home/madhu/Desktop/Desktop/lab3/p2.html", "Apps", "WhatsApp", "Must Do Coding", "Course: Program", "cs - Google Drive", "ACOE - SEMS", and "AWS Training". Below the title bar is a table with the following data:

City	Maximum Temperature	Minimum Temperature
Bombay	34 ° C	28 ° C
Kolkata	35 ° C	27 ° C
Chennai	38 ° C	29 ° C
Delhi	42 ° C	26 ° C
Trichy	40 ° C	32 ° C

**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 :**

```
<html>
<head>
<title>
LOGIN
</title>
</head>
<style type="text/css">
    body{
        font-size: 30px;
        color:DeepPink;
        margin-left:40px;
        margin-top: 40px
    }
</style>
<body bgcolor="#888888>
Username<input type="text" height="20px"><br>
```

```

<br>
Password<input id="pass" type="Password"><br>
<br>
<button onclick="func()" left-margin="20px">submit</button>
<script>
function func(){
    if(document.getElementById("pass").value=="madhu")
    {
        window.location.href="p3a.html";
    }
    else
    {
        alert("invalid");
    }
}
</script>
</body>
</html>

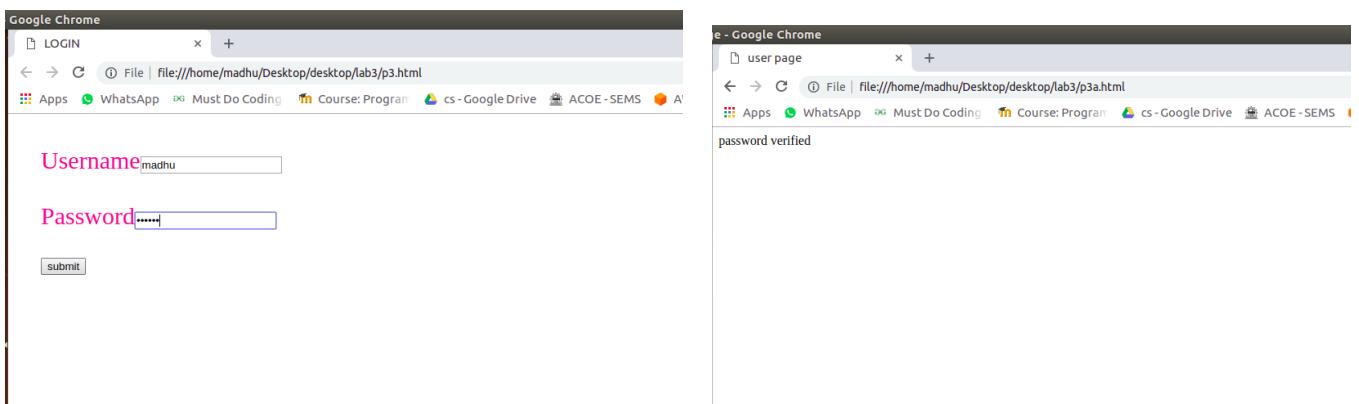
```

**Redirected page:**

```

<html>
<head><title>
user page
</title></head>
<body>
password verified</body>
</html>

```

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

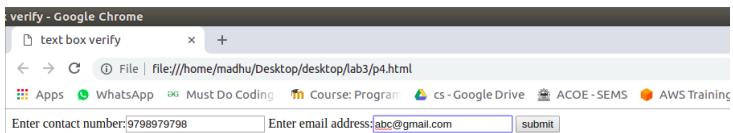
Thus the program is executed and output is obtained

**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 :**

```
<html>
<head>
<title>
text box verify
</title>
</head>
<body>
Enter contact number:<input type="text" id="q" onblur="func1()" autofocus>
Enter email address:<input type="text">
<input type="submit" value="submit" onclick="func()">
<script>
function func()
{
    alert("hello");
}
function func1()
{
    var x=document.getElementById("q");
    if(x.value=="")
        x.focus();
}
</script>
</body>
</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 coordinates of the cursor in JavaScript.

**PROGRAM :**

```
<html>
<style type="text/css">
    body{
        font-size: 30px;
        color:black;
        margin-left:40px;
        margin-top: 40px
    }

</style>
<body bgcolor="aqua">
<h2 onclick="showCoords(event)">Click this heading to get the
x (horizontal) and y (vertical) coordinates of the mouse
pointer when it was clicked.</h2>
<p id="demo"></p>
<script>
function showCoords(event) {
    var x = event.clientX;
    var y = event.clientY;
    var coords = "X coords: " + x + ", Y coords: " + y;
    alert(coords);
}
</script>
</body>
</html>
```

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

Thus the program is executed and output is obtained

**Program 6****AIM :**

To design a simple arithmetic calculator in JavaScript.

**PROGRAM :**

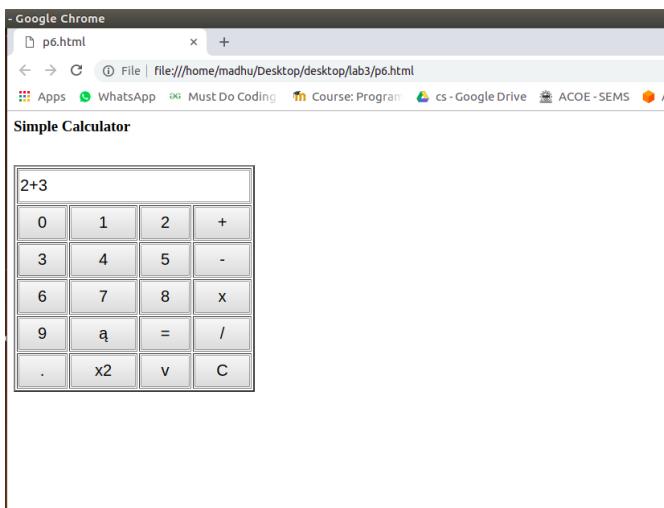
```
<html>
<head></head>
<body>
<h3>Simple Calculator</h3>
<br/>
<style>
#calc{width:300px;height:250px;}
#btn{width:100%;height:40px;font-size:20px;}
</style>
<form Name="calc">
<table id="calc" border=2>
<tr>
<td colspan=4><input id="btn" name="display"
onkeypress="return eventCharCode >= 48 && eventCharCode <=
57" type="text"></td>
<td style="display:none"><input name="M" type="number"></td>
</tr>
<tr>
<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="3"
OnClick="calc.display.value+='3'"></td>
<td><input id="btn" type=button value="4"
OnClick="calc.display.value+='4'"></td>
<td><input id="btn" type=button value="5"
OnClick="calc.display.value+='5'"></td>
<td><input id="btn" type=button value="-"
OnClick="calc.display.value+='-'"></td>
</tr>
<tr>
<td><input id="btn" type=button value="6"
OnClick="calc.display.value+='6'"></td>
<td><input id="btn" type=button value="7"
OnClick="calc.display.value+='7'"></td>
<td><input id="btn" type=button value="8"
OnClick="calc.display.value+='8'"></td>
<td><input id="btn" type=button value="x"
OnClick="calc.display.value+='*'></td>
</tr>
<tr>
<td><input id="btn" type=button value="9"
OnClick="calc.display.value+='9'"></td>
<td><input id="btn" type=button value="q"
OnClick="calc.display.value=(calc.display.value==Math.abs(cal
c.display.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="."
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 :**

```

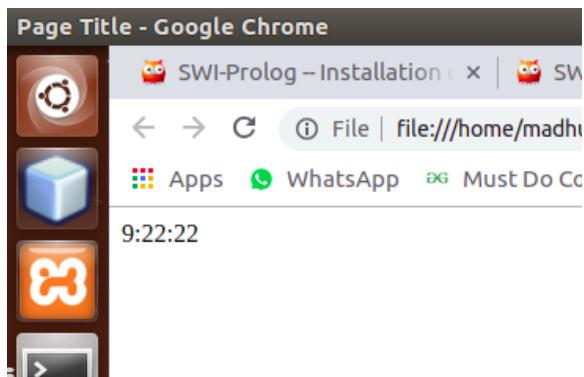
<html>
<head>
<meta http-equiv="cache-control" content="no-cache" />
<meta http-equiv="Pragma" content="no-cache" />

```

```
<meta http-equiv="Expires" content="-1" />
<title>DIGITAL CLOCK</title></head>
<body onload="showtime()">
<style>
.clock {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translateX(-50%) translateY(-50%);
    color: #17D4FE;
    font-size: 60px;
    font-family: Orbitron;
    letter-spacing: 7px;
}

</style>
<script type="text/javascript">
function showTime(){
    var date = new Date();
    var h = date.getHours(); // 0 - 23
    var m = date.getMinutes(); // 0 - 59
    var s = date.getSeconds(); // 0 - 59
    var session = "AM";
    if(h == 0){
        h = 12;
    }
    if(h > 12){
        h = h - 12;
        session = "PM";
    }
    h = (h < 10) ? "0" + h : h;
    m = (m < 10) ? "0" + m : m;
    s = (s < 10) ? "0" + s : s;
    var time = h + ":" + m + ":" + s + " " + session;
    document.getElementById("MyClockDisplay").innerText =
time;
    document.getElementById("MyClockDisplay").textContent =
time;
    setTimeout(showTime, 1000);
}
</script>
```

```
showTime();  
<div id="MyClockDisplay" class="clock"></div>  
</body>  
</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.

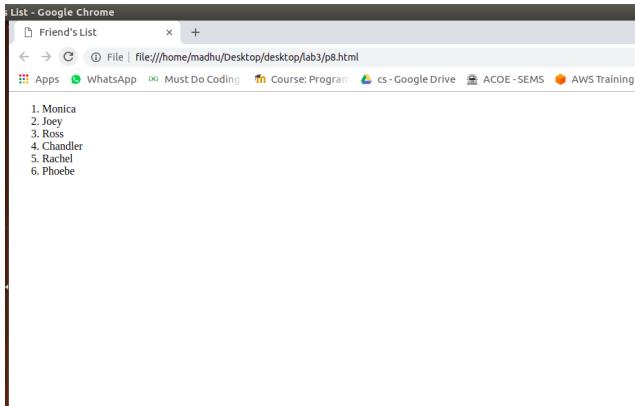
**PROGRAM :**

```
<!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>  
Friend's List  
</title>  
</head>  
<body bgcolor="#888888">  
<ol class="circle">  
<li>Monica</li>  
<li>Joey</li>
```

```

<li>Ross</li>
<li>Chandler</li>
<li>Rachel</li>
<li>Phoebe</li>
</ul>
</body>
</html>

```

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

Thus the program is executed and output is obtained

**Program 9****AIM :**

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

- (i) A textbox to collect the user names.
- (ii) Four check boxes.
- (iii) A collection of 3 radio buttons

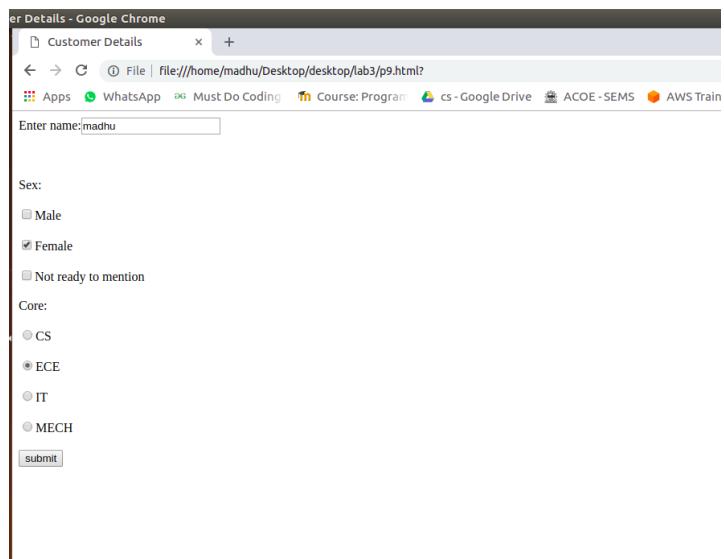
**PROGRAM :**

```

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

```

```
Customer Details
</title>
</head>
<body bgcolor="#888888">
<style>
ul.circle{
    font-size:20px;
    font-family:Arial;
    list-style-type:circle;
}
</style>
<form>
Enter name:<input type="text" required></input>
<br><br>
<br><br>
Sex:<br><br>
<input type="checkbox">Male<br><br>
<input type="checkbox">Female<br><br>
<input type="checkbox">Not ready to mention<br><br>
Core:<br><br>
<input type="radio">CS<br><br>
<input type="radio">ECE<br><br>
<input type="radio">IT<br><br>
<input type="radio">MECH<br><br>
<input type="submit" value="submit"></input>
</body>
</html>
```

**OUTPUT:**

The screenshot shows a Google Chrome window titled "Customer Details - Google Chrome". The address bar displays the URL "file:///home/madhu/Desktop/Desktop/lab3/p9.html?". The page content is as follows:

Enter name:madhu

Sex:

Male  
 Female  
 Not ready to mention

Core:

CS  
 ECE  
 IT  
 MECH

The browser's toolbar includes icons for Apps, WhatsApp, Must Do Coding, Course: Program, cs - Google Drive, ACOE - SEMS, and AWS Train.

**RESULT:**

Thus the program is executed and output is obtained

**Program 10****AIM :**

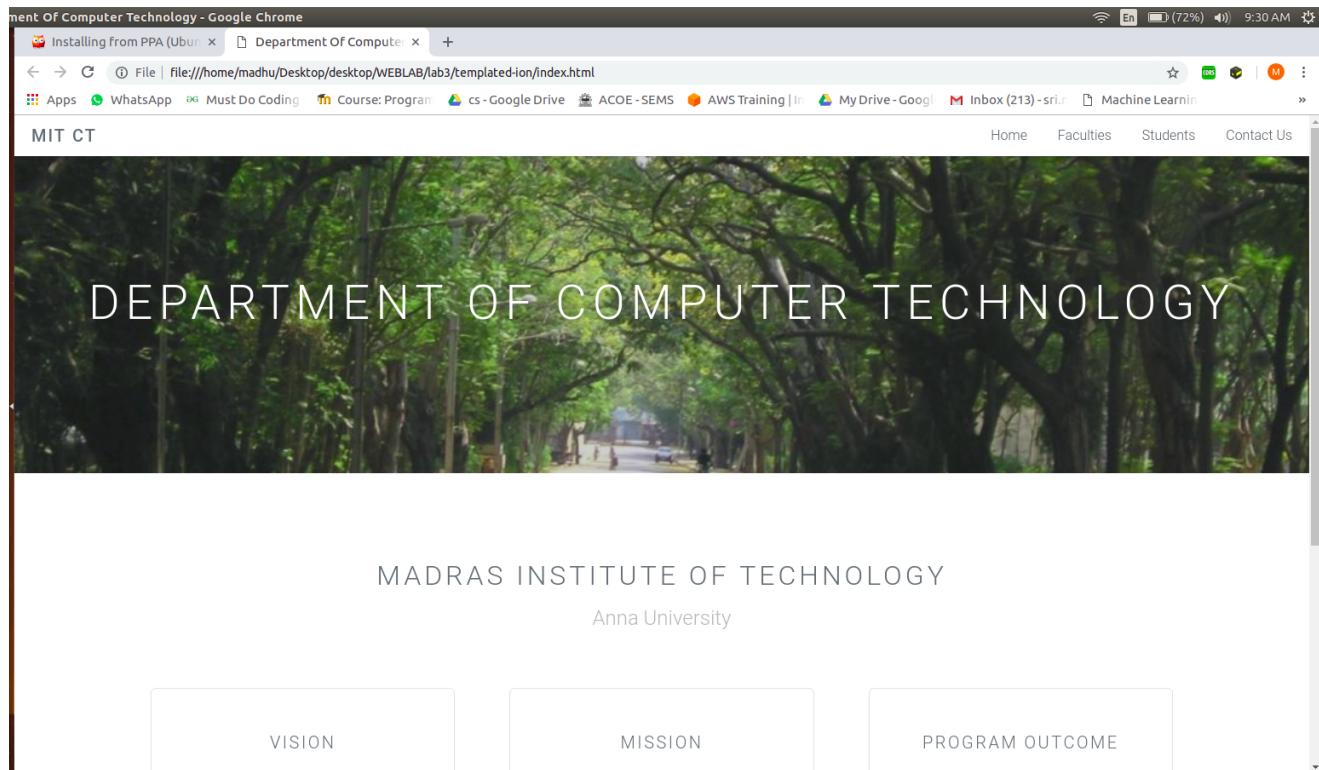
To create a dynamic website of Department of Computer Technology in HTML / XHTML / JavaScript/

**PROGRAM :**

```
index.html
<!DOCTYPE HTML>
<html>
<head>
<title>Department Of Computer Technology</title>
<meta http-equiv="content-type" content="text/html;
charset=utf-8" />
<meta name="description" content="" />
<meta name="keywords" content="" />
<script src="js/jquery.min.js"></script>
<script src="js/skel.min.js"></script>
<script src="js/skel-layers.min.js"></script>
<script src="js/init.js"></script>
<noscript>
<link rel="stylesheet" href="css/skel.css" />
<link rel="stylesheet" href="css/style.css" />
<link rel="stylesheet" href="css/style-xlarge.css" />
</noscript>
</head>
<body id="top">
<header id="header" class="skel-layers-fixed">
<h1><a href="#">MIT CT</a></h1>
<nav id="nav">
<ul>
<li><a href="index.html">Home</a></li>
<li><a href="left-sidebar.html">Faculties</a></li>
<li><a href="right-sidebar.html">Students</a></li>
<li><a href="no-sidebar.html">Contact Us</a></li>
</ul>
</nav>
</header>
<section id="banner">
```

```
<div class="inner">
<h2>Department of Computer Technology</h2>
<ul class="actions">
</ul>
</div>
</section>
<section id="one" class="wrapper style1">
<header class="major">
<h2>Madras Institute of Technology</h2>
<p>Anna University</p>
</header>
<div class="container">
<div class="row">
<div class="4u">
<section class="special box">
<h3>Vision</h3>
<p>To be premier in producing excellent computer professionals through comprehensive educational programmes, promoting students with managerial skills to become competitive entrepreneurs, carrying out research in collaboration with industries.</p>
</section>
</div>
<div class="4u">
<section class="special box">
<h3>Mission</h3>
<p>The department of Computer Technology strives to contribute to the expansion of knowledge in the discipline of Computer Science and Engineering by developing problem solving and analytical skills with sound knowledge in basic sciences and engineering.</p>
</section>
</div>
<div class="4u">
<section class="special box">
<h3>Program Outcome</h3>
<p>An ability to apply basic knowledge of mathematics and science in any engineering field.  
An ability to design and conduct experiments, as well as to analyze and interpret data.</p>
</section>
```

```
</div>
</div>
</div>
</section>
<footer class="text-center">
<p>Copyright © 2018 Computer Technology
| Designed by Madhu and Ramya</p>
</body>
</html>
```

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

Thus the program is executed and output is obtained

# PHP

**Program 1.1:****AIM:**

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

**PROGRAM:**

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

**OUTPUT:**

The unique elements are  
a b c d e

**RESULT:**

Thus, the program is executed and output is obtained.

**Program 1.2:****AIM:**

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

**PROGRAM:**

```
<?php
function Average($numbers)
{
# code...
$k = array_sum($numbers);
$n = count($numbers);
$ans = $k/$n;
```

```

return $ans;
}
function Median($numbers)
{
if (count($numbers)%2 === 0) {
$mid=count($numbers)/2;
return (( $numbers[$mid-1]+$numbers[$mid])/2);
}
else {
$mid=(count($numbers)-1)/2;
return $numbers[$mid];
}
}
$numbers=array(1,2,3,4,5);
echo ("AVERAGE: " .Average($numbers)."\n");
echo ("MEDIAN: ".Median($numbers)."\n");
?>

```

**OUTPUT:**

AVERAGE: 3

MEDIAN: 3

**RESULT:**

Thus, the program is executed and output is obtained.

**Program 1.3:****AIM:**

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

**PROGRAM:**

```

<?php
function compute($a1){
$map = array_count_values($a1);
arsort($map);
$a2 = array_keys($map);
return $a2;
}
$a1 = array("a","b","c","a","b","a","b","a","b","a");
$a2 = compute($a1);

```

```
print "The top 3 elements are\n";
for($x1=0;$x1<3;$x1++){
print $a2[$x1];
print " ";
}
?>
```

**OUTPUT:**

The top 3 elements are

a b c

**RESULT:**

Thus, the program is executed and output is obtained.

**Program 1.4:****AIM:**

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

**PROGRAM:**

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

```

for ($i=0; $i < count($pos); $i++) {
echo($pos[$i]." ");
}
echo "\n";
echo "Negative array:\n";
for ($i=0; $i < count($neg); $i++) {
echo($neg[$i]." ");
}
echo "\n";
?>

```

**OUTPUT:**

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

**RESULT:**

Thus, the program is executed and output is obtained.

**Program 1.5:****AIM:**

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

**PROGRAM:**

```

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

```

```

print $a2 . "\n";
$a1 ="1 2 34 12 345";
$a2 = compute($a1);
print $a2 . "\n";
?>

```

**OUTPUT:**

3455  
false

**RESULT:**

Thus, the program is executed and output is obtained.

**Program 1.6:****AIM:**

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

**PROGRAM:**

```

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

```

```
echo findMax($file,$fname);
?>
```

**OUTPUT:**

Input file – proglf.txt  
 rahul saran mahesh saran:srini:madhu:saran rahul  
 saran

**RESULT:**

Thus, the program is executed and output is obtained.

**Program 1.7:****AIM:**

To write a PHP script that take in a string containing words that are delimited on the left by spaces and on the 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 vdgdfghj, welcome." ;
$a2 = compute($a1);
print "The top 3 elements are\n";
for($x1=0;$x1<3;$x1++){
print $a2[$x1];
print " ";
}
```

```
?>
```

**OUTPUT:**

The top 3 elements are  
welcome hello there

**RESULT:**

Thus, the program is executed and output is obtained.

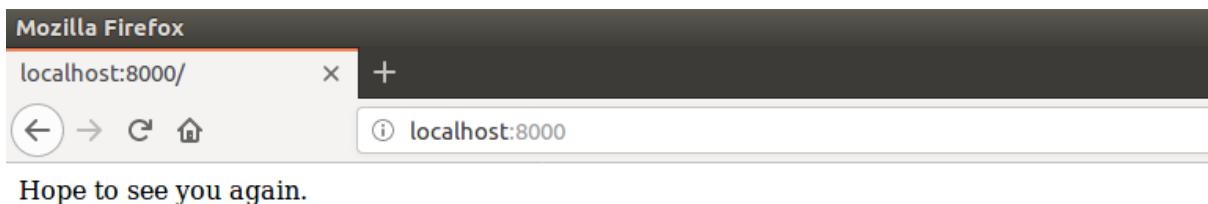
**Program 2:****AIM:**

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

**PROGRAM:**

```
PHP Program
<?php
function greetingCall(){
define("GREETING1","Have a nice day.");
define("GREETING2","Thank you.");
define("GREETING3","Welcome.");
define("GREETING4","Nice meeting you.");
define("GREETING5","Hope to see you again.");
$greet =
array(GREETING1,GREETING2,GREETING3,GREETING4,GREETING5);
$index = rand(0,4);
echo "<div>".$greet[$index]."</div>";
}
greetingCall();
?>
XHTML Program
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>
```

```
Program 2
</title>
</head>
<body>
<a href="http://localhost:8000">Click to view Greeting</a>
</body>
</html>
```

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

Thus, the program is executed and output is obtained.

**Program 3:****AIM:**

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

**PROGRAM:**

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

```
}

?></div>
</form>
</body>
</html>
```

Output:

**OUTPUT:**

Program 3 - Mozilla Firefox

Program 3    +

← → ⌂ ⌄    i localhost:8000

Name:

Favorite Song:

Composer:

Performing Group (or) artist

Name: a  
Favorite Song: b  
Composer: c  
Artist/Group: d

Name:  
Favorite Song:  
Composer:  
Artist/Group:

**RESULT:**

Thus, the program is executed and output is obtained.

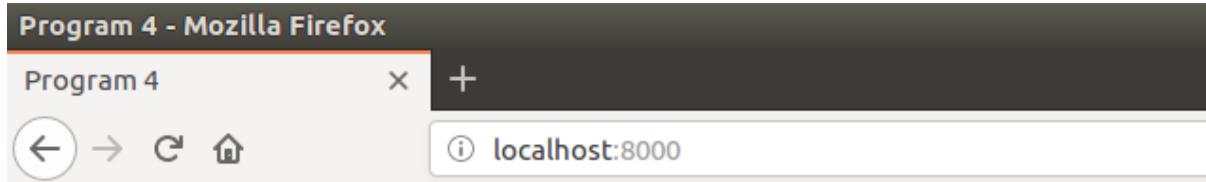
**Program 4:****AIM:**

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

**PROGRAM:**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>
Program 4
</title>
</head>
<body>
<?php
$mobileErr = "";
$mobile = "";
$name = "";
if ($_SERVER[ "REQUEST_METHOD" ] == "POST")
{
if(empty($_POST[ "mobile" ])) {
$mobileErr = "Mobile number is required";
} else {
$mobile = test_input($_POST[ "mobile" ]);
if(!preg_match("/^+\d{2}-\d{4}-\d{6}/", $mobile))
{
$mobileErr = "invalid Format";
}
}
}
function test_input($data) {
$data = trim($data);
$data = stripslashes($data);
$data = htmlspecialchars($data);
return $data;
}
?>
<h2>Form Validation</h2>
<form method="post" action="<?php
htmlspecialchars($_SERVER[ "PHP_SELF" ]);?>">
Name: <input type="text" name="name" value="<?php echo
$name;?>" /><br><br>
Mobile: <input type="text" name="mobile" value="<?php
echo $mobile;?>" />
<span class="error">*<?php echo $mobileErr;?></span>
<br /><br />
```

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

**OUTPUT:**

## 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 Program
<?php
function greetingCall(){
define("GREETING1","Have a nice day.");
define("GREETING2","Thank you.");
define("GREETING3","Welcome.");
define("GREETING4","Nice meeting you.");
define("GREETING5","Hope to see you again.");
$greet =
array(GREETING1,GREETING2,GREETING3,GREETING4,GREETING5);
```

```
$index = rand(0,4);
$fp = file_get_contents("prog5.txt");
$val = (int)$fp;
$val+=1;
$fp = (string)$val;
file_put_contents("prog5.txt",$fp);
print "<div>".$greet[$index]."</div>";
print "<div> Client Number ".$val."</div>";
}
greetingCall();
?>

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

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

Thus, the program is executed and output is obtained.

**Program 6:****AIM:**

To implement the following modules using Server-Side Scripting

(PHP)

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

**PROGRAM:**

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

```
session_start();
}
if ($_SERVER[ "REQUEST_METHOD" ] == "POST")
{
if(empty($_POST[ "regno" ])) {
$regErr = "Student Registration number is
required";
}
if(empty($_POST[ "pwd" ])) {
$pwdErr = "Password is required";
}
if(isset($_POST[ "reg" ]) && isset($_POST[ "pwd" ])){
$servername = "localhost";
$username = "root";
$password = "srini1998";
$dbname = "Basel";
$conn = new
mysqli($servername,$username,$password,$dbname);
if($conn -> connect_error){
die("Connection failed: ".$conn-
>connect_error);
}

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

$sql = "SELECT name,address,mobile,id from
Tab1 where id = '$regno' and password = '$pwd' ;";
$result = $conn->query($sql);
if($result->num_rows > 0){

while($row = $result->fetch_assoc()){
$name = $row[ "name" ];
$address = $row[ "address" ];
$mobile = $row[ "mobile" ];
$regno = $row[ "id" ];
}
}
else{
print "Data mismatch. Please try again.";
$regno = "";
$pwd = "";
}
```

```
}

$conn ->close();
}
$_SESSION[ 'LAST_ACTIVITY' ] = $time;
?
<div><h1>Depatment of Computer Technology</h1></div>
<br>
<br>
<form class = "form1" method="post" action="<?php
htmlspecialchars($_SERVER[ "PHP_SELF" ]);?>">
<h3>
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:**

The screenshot shows a web browser window with the URL `localhost:8000` in the address bar. The title of the page is "Department of Computer Technology". Below the title, there is a login form with two fields: "Registration No:" containing "2016503042" and "Password:" containing ".....". Below the form is a "submit" button. At the bottom of the page, there is a horizontal bar with four input fields: "Student Name: Srinivasan", "Registration No: 2016503042", "Address: Ashok Nagar", and "Mobile: 9884274654".

**DB Schema**

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

**RESULT:**

Thus, the program is executed and output is obtained.

# PYTHON

**Program 1****AIM :**

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

**PROGRAM :**

```
print("Hello world !")
```

**OUTPUT:**

Hello world!

**RESULT:**

Thus, the program is executed and output is obtained.

**Program 2****AIM :**

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

**PROGRAM :**

```
l=[1,2,3,4]
l1=l[::-1]
print(l1)
```

**OUTPUT:**

[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 :**

```
import math
k=int(input("Enter number"))
print(math.factorial(k))
```

**OUTPUT:**3  
6**RESULT:**

Thus, the program is executed and output is obtained.

**Program 4****AIM :**

To write a GUI for the expression calculator using tk.

**PROGRAM :**

```
from tkinter import *
expression = ""
def press(num):
    global expression
    expression = expression + str(num)
    equation.set(expression)
def equalpress():
    try:
        global expression
        total = str(eval(expression))
        equation.set(total)
        expression = ""
    except:
        equation.set(" error ")
        expression = ""
def clear():
    global expression
    expression = ""
    equation.set("")
if __name__ == "__main__":
    gui = Tk()
    gui.configure(background="light green")
    gui.title("Simple Calculator")
    gui.geometry("265x125")
    equation = StringVar()
    expression_field = Entry(gui, textvariable=equation)
    expression_field.grid(columnspan=4, ipadx=70)
```

```
equation.set('enter your expression')
button1 = Button(gui, text=' 1 ', fg='black',
bg='red',
command=lambda: press(1), height=1, width=7)
button1.grid(row=2, column=0)
button2 = Button(gui, text=' 2 ', fg='black',
bg='red',
command=lambda: press(2), height=1, width=7)
button2.grid(row=2, column=1)

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

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

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

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

button7 = Button(gui, text=' 7 ', fg='black',
bg='red',
command=lambda:
press(7), height=1, width=7)
button7.grid(row=4, column=0)

button8 = Button(gui, text=' 8 ', fg='black',
bg='red',
```

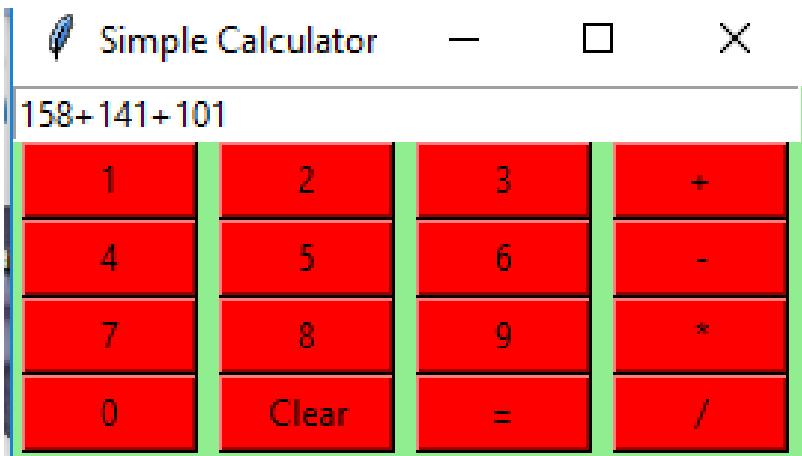
```
                                command=lambda:  
press(8), height=1, width=7)  
    button8.grid(row=4, column=1)  
  
    button9 = Button(gui, text=' 9 ', fg='black',  
bg='red',  
                                command=lambda:  
press(9), height=1, width=7)  
    button9.grid(row=4, column=2)  
  
    button0 = Button(gui, text=' 0 ', fg='black',  
bg='red',  
                                command=lambda:  
press(0), height=1, width=7)  
    button0.grid(row=5, column=0)  
  
    plus = Button(gui, text=' + ', fg='black', bg='red',  
                                command=lambda: press("+"),  
height=1, width=7)  
    plus.grid(row=2, column=3)  
  
    minus = Button(gui, text=' - ', fg='black', bg='red',  
                                command=lambda: press("-"),  
height=1, width=7)  
    minus.grid(row=3, column=3)  
  
    multiply = Button(gui, text=' * ', fg='black',  
bg='red',  
                                command=lambda:  
press("*"), height=1, width=7)  
    multiply.grid(row=4, column=3)  
  
    divide = Button(gui, text=' / ', fg='black',  
bg='red',  
                                command=lambda:  
press("/"), height=1, width=7)  
    divide.grid(row=5, column=3)  
  
    equal = Button(gui, text=' = ', fg='black', bg='red',  
                                command=equalpress, height=1,  
width=7)  
    equal.grid(row=5, column=2)
```

```

        clear = Button(gui, text='Clear', fg='black',
bg='red',
                           command=clear, height=1,
width=7)
        clear.grid(row=5, column='1')

# start the GUI
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 :**

```

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

if __name__ == '__main__':

```

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

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

**OUTPUT:**

The screenshot shows two terminal windows side-by-side. The left terminal window shows the execution of `server.py`, which starts a Flask application on port 3006. The right terminal window shows the execution of `client.py`, which sends a GET request to the server to add two numbers (5 and 6) and prints the result (11).

```
Terminal
srinivasan@coder:~/Desktop/Web Lab/Jan 11$ python3 progserver.py
 * Serving Flask app "progserver" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment
   Use a production WSGI server instead.
 * Debug mode on
 * Running on http://127.0.0.1:3006/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active
 * Debugger PIN: 222-411-998
result: 11
127.0.0.1 - - [24/Feb/2019 13:13:33] "GET /add/5&6 HTTP/1.1" 200 -
[1]
[2]
```

```
Terminal
srinivasan@coder:~/Desktop/Web Lab/Jan 11$ python3 progclient.py
Enter a:5
Enter b:6
b'11'
```

**RESULT:**

Thus, the program is executed and output is obtained.

**Program 6****AIM :**

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

**PROGRAM :**

```
from bs4 import BeautifulSoup
import requests

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

```
for i in ans:  
out.write(str(i.string) + "\n")  
if htmlCode.p:  
ans = htmlCode.find_all('p')  
for i in ans:  
out.write(str(i.string) + "\n")  
  
try:  
url = input("Enter a url:")  
out = open("out6.txt", "w")  
scrapeSite(url, out)  
out.close()  
except IOError as e:  
print(e)
```

OUTPUT:

Terminal:

Enter a

url:[https://en.wikipedia.org/wiki/Separation\\_of\\_content\\_and\\_presentation](https://en.wikipedia.org/wiki/Separation_of_content_and_presentation)

out.txt

Separation of content and presentation

From Wikipedia, the free encyclopedia

Jump to navigationJump to search

An example of CSS Code, which makes up the visual and styling components of a web page.

Separation of content and presentation (or separation of content and style) is the separation of concerns design principle as applied to the authoring and presentation of content. Under this principle, visual and design aspects (presentation and style) are separated from the core material and structure (content) of a document.[1][2][non-primary source needed] A typical analogy used to explain this principle is the distinction between the human skeleton (as the structural component) and human flesh (as the visual component) which makes up the body's appearance. Common applications of this principle are seen in Web design (HTML and CSS)[3][4] and markup language (see LaTeX).

**RESULT:**

Thus, the program is executed and output is obtained.

**Program 7****AIM :**

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

**PROGRAM :**

```
import webbrowser
from urllib.request import urlopen

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

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

f.write(message)
f.close()

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

**OUTPUT:****HTTP Response:**

```
Date: Sun, 24 Feb 2019 06:54:36 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more
info."
Server: gws
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Set-Cookie: 1P_JAR=2019-02-24-06; expires=Tue, 26-Mar-2019
06:54:36 GMT; path=/; domain=.google.com
Set-Cookie: NID=160=FHyqmm8VstO2NgnWS9DknWzCyanGq6qQF6GyP6-
FHzJkFWLVKHBjQzCxUx4MRhzAiFnZ5iBGc5Z6ceSo6pqWK3LiIZgxWwQuSMhk
a22InLU402eT-
K2JGswFISh48tny7hS_P1HDYN6qqzMAEp0ymDDCaDDEgRNQ8pBlByUlTpM;
```

```
expires=Mon, 26-Aug-2019 06:54:36 GMT; path=/;
domain=.google.com; HttpOnly
Alt-Svc: quic=":443"; ma=2592000; v="44,43,39"
Accept-Ranges: none
Vary: Accept-Encoding
Connection: close
```

**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="srini1998", database="Basel")
print(db)
cursor = db.cursor()
sql = "SELECT * from Tab2 where id ='" +x+ "' and
password=' " +y+ " ' ;"
print(sql)
cursor.execute(sql)
result = cursor.fetchall()
print(result)
if len(result) == 0:
return "Please Verify Credentials"
ans = ""
```

```

for a in result:
    for b in range(0,len(a)-1):
        ans=ans+"#" +a[b]
    return ans
except mysql.connector.Error as e:
    print(e)

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

```

**Database Schema:**

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

**OUTPUT:**

Course Id	Course Name	Grade
CS7511	Web	O
CS7512	Web Lab	O

**RESULT:**

Thus, the program is executed and output is obtained.

# JAVA SERVLETS

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

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

**PROGRAM:**

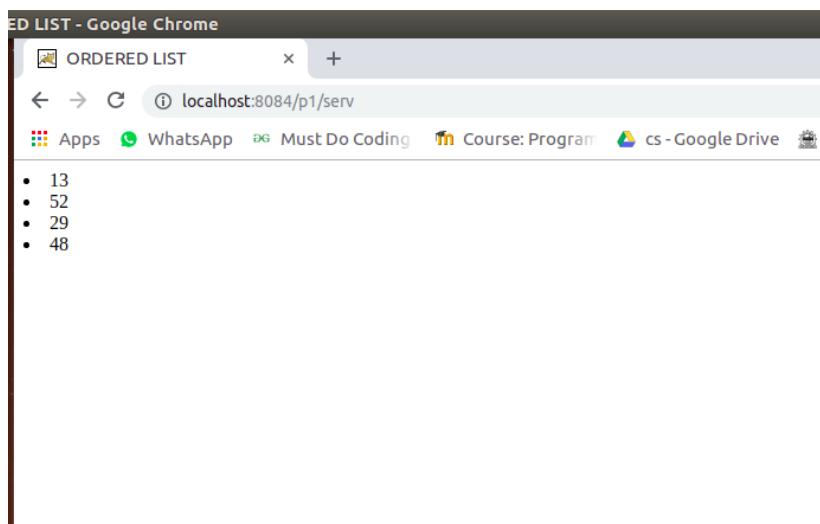
```
server.java
package com.serv;
import java.util.Random;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class serv extends HttpServlet {
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    response.setContentType("text/html");
    Random rand=new Random();
    String title = "ORDERED LIST";
    out.println("<html>");
    out.println("<head>");
    out.println("<title>" + title + "</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<ol>");
    out.println("<li>" +
Integer.toString(rand.nextInt(100) + 1) + "</li>");
    out.println("</ol>");
    out.println("</body>");
    out.println("</html>");
```

```
}

    private void processRequest(HttpServletRequest request,
HttpServletRequest response) {
        throw new UnsupportedOperationException( "Not
supported yet."); //To change body of generated methods,
choose Tools | Templates.
    }

}

index.html
<html>
    <head>
        <title>Program 1</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    </head>
    <body>
        <a href="/p1/serv" >hey</a>
    </body>
</html>
```

**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:**

```
server.java
package com.serv1;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class serv extends HttpServlet {
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    response.setContentType("text/html");
    String title = "ORDERED LIST";
    out.println("<html>");
    out.println("<head>");
    out.println("<title>" + title + "</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<table border='1'>");
    for(int i=1;i<26;i++)
    {
        out.println("<tr>");
        for(int j=1;j<4;j++)
        {
            out.println("<td>");
            out.println("row:" + i + "Column:" + j);
            out.println("</td>");
        }
        out.println("</tr>");
    }
    out.println("</table>");
    out.println("</body>");
    out.println("</html>");
}
```

```

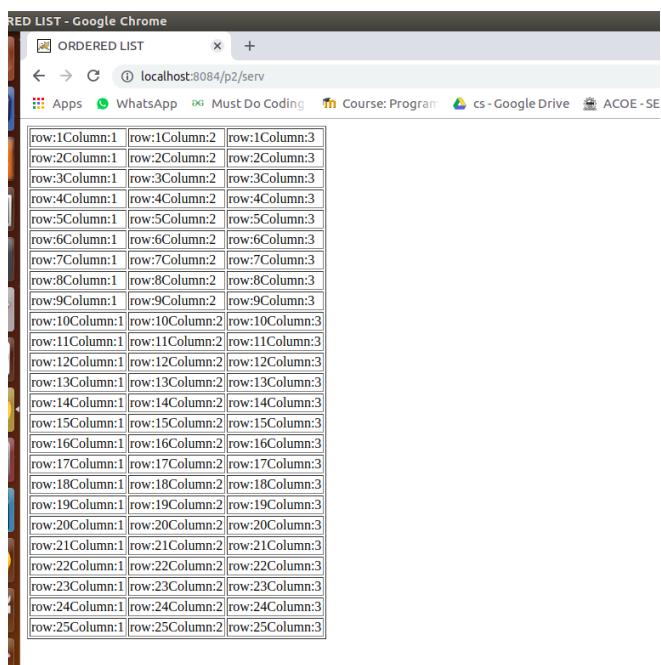
    private void processRequest(HttpServletRequest request,
HttpServletRequest response) {
    throw new UnsupportedOperationException("Not
supported yet."); //To change body of generated methods,
choose Tools | Templates.
}
}

```

```

index.html
<html>
    <head>
        <title>Table</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    </head>
    <body>
        <a href="/p2/serv"> madhu</a>
    </body>
</html>

```

**OUTPUT:**

**RESULT:**

Thus the program is executed and output is obtained.

**PROGRAM 3a****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:**

```
server.java
package com.serv2;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class serv2a extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use
following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Form</title>");
        out.println("</head>");
        out.println("<body>");

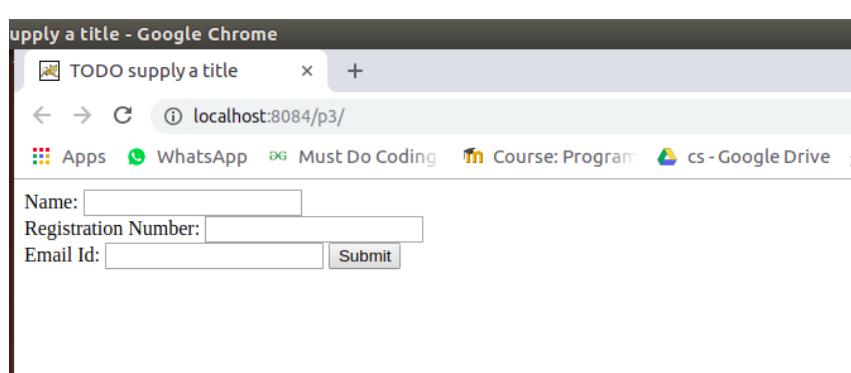
out.println("name:"+request.getParameter("name"));
        out.println("registeration
number:"+request.getParameter("reg_no"));
        out.println("email
id:"+request.getParameter("email"));
        //
out.println("name:"+request.getParameter("name"));

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

```
// <editor-fold defaultstate="collapsed"
desc="HttpServlet methods. Click on the + sign on the left to
edit the code.">
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletRequest response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}
```

index.html

```
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    </head>
    <body>
        <form action = "serv2a" method = "GET">
            Name: <input type = "text" name = "name">
            <br />
            Registration Number: <input type = "text" name =
"reg_no">
            <br />
            Email Id: <input type = "text" name = "email" />
            <input type = "submit" value = "Submit" />
        </form>
    </body>
</html>
```

**OUTPUT:**

**RESULT:**

Thus the program is executed and output is obtained.

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

To modify the servlet program so that it inserts data into database and then selects all the data in the database to create response to the client.

**PROGRAM:**

server.java

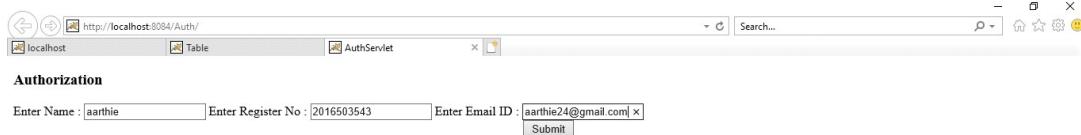
LOGINDAO.java

```
public class LoginDAO {
    String url = "jdbc:mysql://localhost:3306/ramya";
    String username = "root";
    String password = "ramya2898";
    public void addUser(String name, String regno, String
email, HttpServletResponse response) throws
ClassNotFoundException, SQLException, IOException {
        PrintWriter out = response.getWriter();
        out.println("add user");
        String sql = "insert into student1
values(?, ?, ?)";
        response.setContentType("text/html");
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection(url, username, password);
        PreparedStatement st = con.prepareStatement(sql);
        st.setString(1, name);
        st.setString(2, regno);
        st.setString(3, email);
        st.executeUpdate();
        String sql1 = "select * from student1";
        PreparedStatement st1 =
con.prepareStatement(sql1);
        ResultSet rs1 = st1.executeQuery();
        out.print("<html>");
        out.print("<table>");
        out.print("<tr> <th> Name </th> <th> Register No
</th> <th> Email ID </th> </tr>");
        while(rs1.next()) {
            out.print("<tr> <td> " + rs1.getString(1));
            out.print(" <td> " + rs1.getString(2));
            out.print(" <td> " + rs1.getString(3));
            out.print(" <td> " + rs1.getString(4));
            out.print(" </tr>");
        }
        out.print("</table> ");
        out.print("</html> ");
    }
}
```

```

        out.print("<tr>");
        out.print("<td>");
        out.print(rs1.getString(1)+ "</td> <td>" +
rs1.getString(2) + "</td> <td>" + rs1.getString(3));
        out.print("</td>");
        out.print("</tr>");
    }
    out.print("</table>");
    out.print("</html>");
}
}

```

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

Thus the program is executed and output is obtained.

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

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

**PROGRAM:**

```

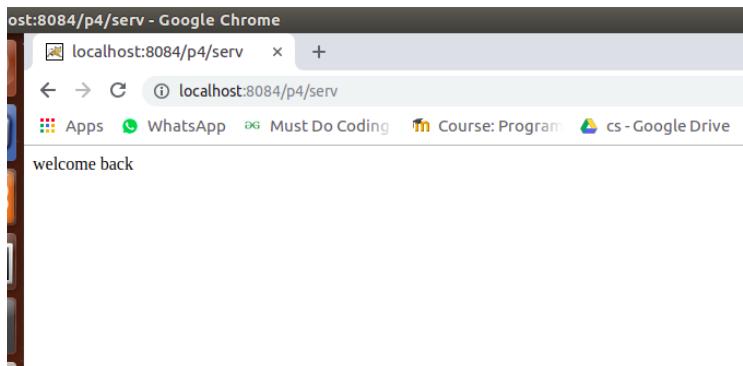
server.java
package com.serv4;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;

```

```
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class serv extends HttpServlet {
    private int c=0;
    private HttpSession session=null;
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
        c++;
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        if(session==null){
            out.println("Welcome guest");
            session=request.getSession();
        }
        else
        {
            out.println("welcome back");
        }
    }
desc="HttpServlet methods. Click on the + sign on the left to
edit the code.">
    @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";
    } // </editor-fold>
}
```

index.html

```
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  </head>
  <body>
    <a href="/p4/serv">kk</a>
  </body>
</html>
```

**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:**

```
server.java
import java.io.*;
import java.util.Random;
```

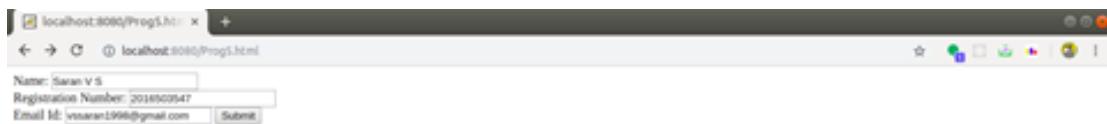
```
import javax.servlet.*;
import javax.servlet.http.*;
public class Prog5 extends HttpServlet {
protected void doGet(HttpServletRequest req,
HttpServletResponse res) throws IOException {
res.setContentType("text/html");
PrintWriter out = res.getWriter();
Cookie name = null, reg_no = null, email = null;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

```
else {
    out.println("Email: " + req.getParameter("email"));
    if (email.getValue().equals("Unknown")) {
        email.setValue(req.getParameter("email"));
        res.addCookie(email);
    }
}
}
}
else {
    if (req.getParameter("name").length() == 0)
        name = new Cookie("Name", "Unknown");
    else
        name = new Cookie("Name", req.getParameter("name"));
    out.println("Name: " + name.getValue());
    if (req.getParameter("reg_no").length() == 0)
        reg_no = new Cookie("Reg_No", "Unknown");
    else
        reg_no = new Cookie("Reg_No", req.getParameter("reg_no"));
    out.println("Registration Number: " + reg_no.getValue());
    if (req.getParameter("email").length() == 0)
        email = new Cookie("Email", "Unknown");
    else
        email = new Cookie("Email", req.getParameter("email"));
    out.println("Email: " + email.getValue());
    name.setMaxAge(60 * 60 * 24);
    reg_no.setMaxAge(60 * 60 * 24);
    email.setMaxAge(60 * 60 * 24);
    res.addCookie(name);
    res.addCookie(reg_no);
    res.addCookie(email);
}
out.close();
} catch (Exception e) {
    System.out.println(e);
}
}
}
index.html
<html>
    <body>
        <form action = "Prog5" method = "GET">
            Name: <input type = "text" name = "name">
```

```

<br />
Registration Number: <input type="text"
name="reg_no">
<br />
Email Id: <input type = "text" name = "email" />
<input type = "submit" value = "Submit" />
</form>
</body>
</html>

```

**OUTPUT:****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:**

```

server.java
package com.serv6;
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 serv6 extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html");
}

```

```
PrintWriter out = response.getWriter();
String title = "HTTP Header Request Example";
String docType =
    "<!doctype html public \\"-//w3c//dtd html 4.0\\" +
"transitional//en\\">\n";

out.println(docType +
    "<html>\n" +
    "<head><title>" + title + "</title></head>\n");
final String clientBrowser =
getClientBrowser(request);
if(clientBrowser.equals("IE"))
    out.println("<body bgcolor= 'red' "
text='yellow'>");
else
    out.println("<body bgcolor = 'yellow' "
text='red'>");
out.println("<body bgcolor = \"#f0f0f0\">\n" +
"<h1 align = \"center\">" + title + "</h1>\n" +
"<table width = \"100%\" border = \"1\" align =
= \"center\">\n" +
"<tr bgcolor = \"#949494\">\n" +
"<th>Header Name</th><th>Header Value(s)</th>\n" +
"</tr>\n"
);

Enumeration headerNames = request.getHeaderNames();

while(headerNames.hasMoreElements()) {
    String paramName =
(String)headerNames.nextElement();
    out.print("<tr><td>" + paramName + "</td>\n");
    String paramValue = request.getHeader(paramName);
    out.println("<td> " + paramValue + "</td></tr>\n");
}
out.println("</table>\n</body></html>");
}

// <editor-fold defaultstate="collapsed"
desc="HttpServlet methods. Click on the + sign on the left to
edit the code.">
```

```
@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";
}// </editor-fold>
public String getClientBrowser(HttpServletRequest request) {
    final String browserDetails =
request.getHeader("User-Agent");
    final String user = browserDetails.toLowerCase();

    String browser = "";

    //=====Browser=====
    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=(userAgent.substring(userAgent.indexOf("MSIE")).spl
it(" ")[0]).replace("/", "-");
        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;
    }

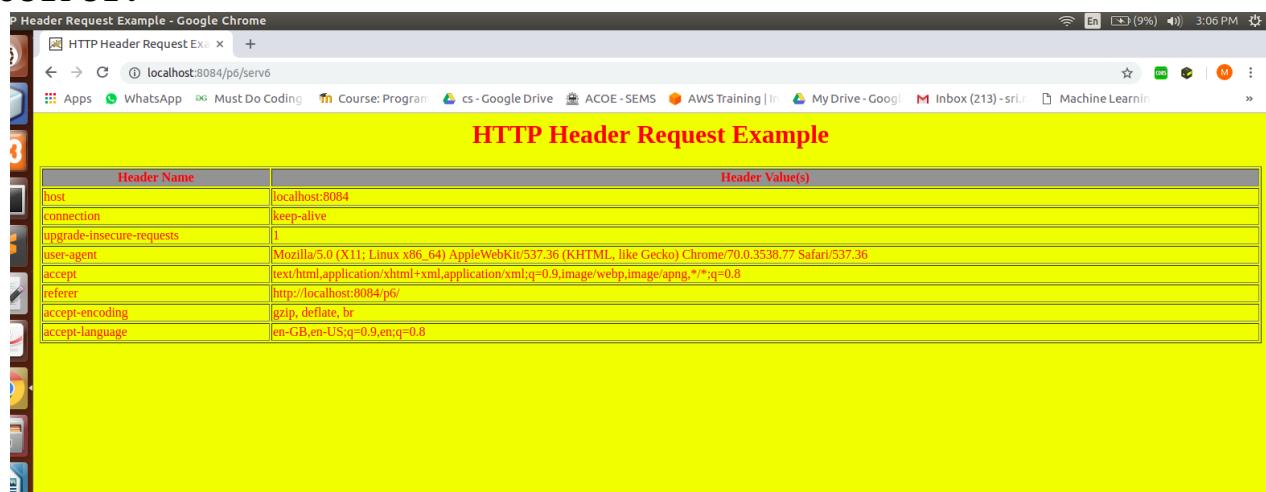
    return browser;
}
}

```

```

index.html
<html>
    <head>
        <title>Servlet 6</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    </head>
    <body>
        <a href="/p6/serv6">hello</a>
    </body>
</html>

```

**OUTPUT:**


Header Name	Header Value(s)
host	localhost:8084
connection	keep-alive
upgrade-insecure-requests	1
user-agent	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.77 Safari/537.36
accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
referer	http://localhost:8084/p6/
accept-encoding	gzip, deflate, br
accept-language	en-GB,en-US;q=0.9,en;q=0.8

**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:**

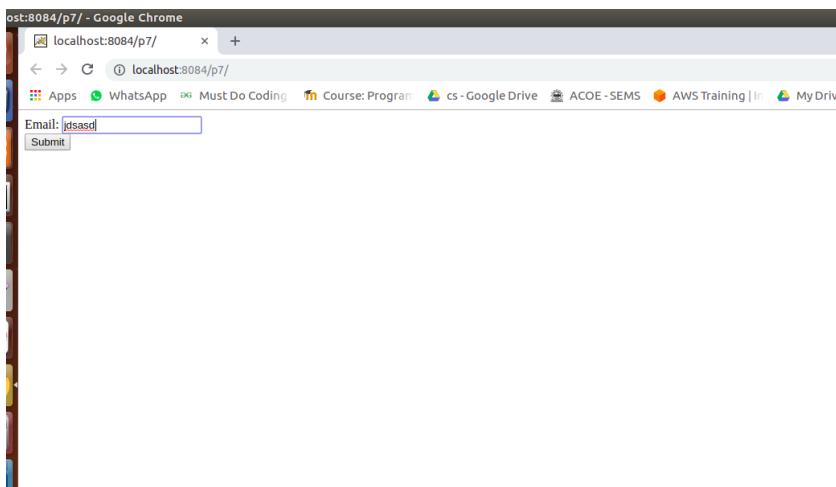
```
server.java
package com.serv7;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class serv7 extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
    try{
        PrintWriter out = response.getWriter();

        response.setContentType("text/html");
        String email = request.getParameter("email");
        if(email.contains("@")){
            out.println("<html>");
            out.println("<head>");
            out.println("<title>" + "Success" + "</title>");
            out.println("</head>");
            out.println(" <body>Success</body> ");
            out.println("</html>");

        }
        else{
            response.sendError(404,"BadRequest");
        }
    }
```

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

desc="HttpServlet methods. Click on the + sign on the left to
edit the code.">
@Override
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
processRequest(request, response);
}
index.html
<html>
<body>
    <form action ="serv7" method = "POST">
        Email: <input type = "text" name = "email">
        <br />
        <input type = "submit" value = "Submit" />
    </form>
</body>
</html>
```

**OUTPUT:****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.

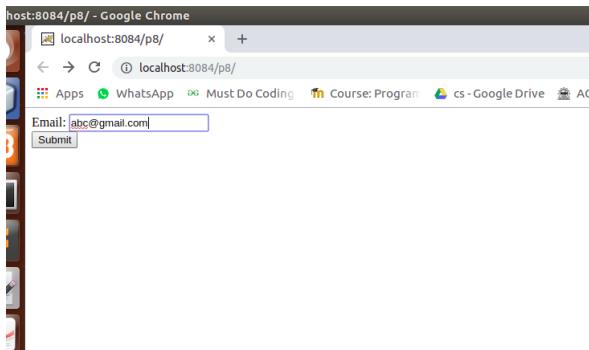
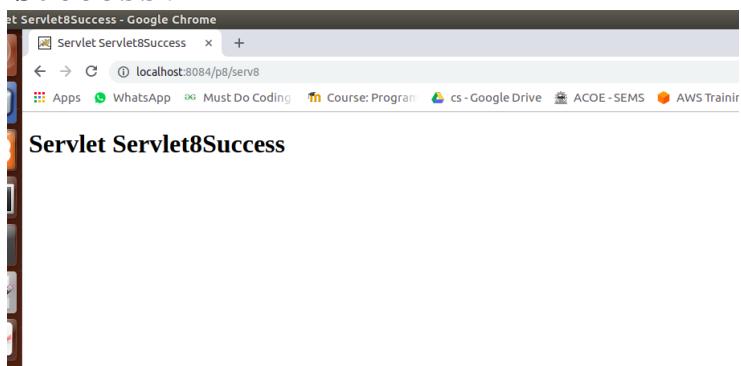
**PROGRAM:**

server.java

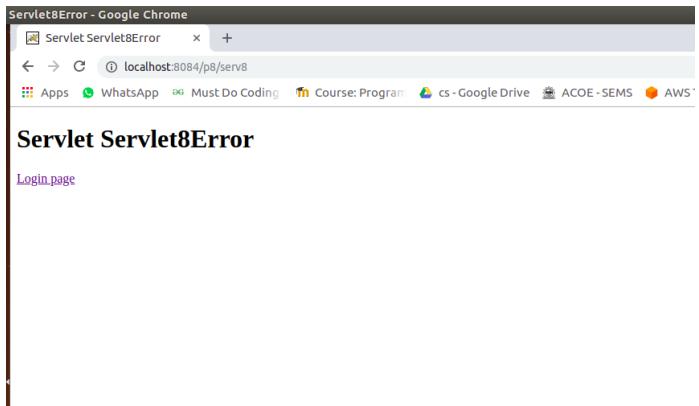
package com.serv8;

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class serv8 extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
    try{
        response.setContentType("text/html");
        String email = request.getParameter("email");
        if(email.contains("@")){
            RequestDispatcher rd =
request.getRequestDispatcher("Servlet8Success");
            rd.include(request,response);
        }
        else{
            RequestDispatcher rd =
request.getRequestDispatcher("Servlet8Error");
            rd.include(request,response);
        }
    }
    catch(Exception e){
        e.printStackTrace();
    }
}
```

```
desc="HttpServlet methods. Click on the + sign on the left to
edit the code.">
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletRequest response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}
index.html
<html>
    <body>
        <form action ="serv8" method = "POST">
            Email: <input type = "text" name = "email">
            <br />
            <input type = "submit" value = "Submit" />
        </form>
    </body>
</html>
```

**OUTPUT:****Success :**

**Failure:**



**RESULT:**

Thus the program is executed and output is obtained.

# Programming with AJAX, Jquery, JSON

**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:**

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

**RESULT:**

Thus the program is executed and output is obtained.

**PROGRAM 2****AIM:**

To create an XML document with atleast three instances of the car element defined in the DTD of Program 1. Process the

document by using the DTD of Program 1, and produce a display of raw XML document.

**PROGRAM:**

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

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

    <car>
        <make>German</make>
```

```

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

```

**OUTPUT:**


The screenshot shows a web browser window with the following details:

- Address bar: view-source:file:///home/saran/Desktop/Web Lab/Lab 7/prog2.xml
- Toolbar icons: back, forward, search, etc.
- Content area: The XML code from the previous block is displayed, with line numbers 1 through 47 on the left. The XML is color-coded: blue for tags like <model>, <year>, <color>, <engine>, <number\_of\_cylinders>, <fuel\_system>, <transmission\_type>, <accessories>, <radio>, <air\_conditioning>, <power\_windows>, <power\_steering>, and <power\_brakes>; red for attributes like radio = "yes", air\_conditioning = "no", power\_windows = "yes", power\_steering = "no", and power\_brakes = "yes"; and purple for values like Volkswagen, 1957, Red, 4, carburated, auto, 6, yes, no, yes, and yes.

**RESULT:**

Thus the program is executed and output is obtained.

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

To modify the example application of Program 1 to allow the user to select a make and model of used cars. The make must be in a menu. When a make is chosen, a meny of models must be displayed. This menu is produced by hardwired data in the original document. When a model is chosen, an AJAX request must be made to get a list of the years and colors of the chosen make and model that are available. Make up a server-resident script to produce the data from an example array or hash.

**PROGRAM:**

```
<html>
    <head>
        <title> Program 3 </title>
        <style>
            #display{
                visibility: hidden;
            }
        </style>
        <script>
            function getData(){
                var xml_load1 = new XMLHttpRequest();
                xml_load1.onreadystatechange =function(){
                    if(this.readyState==4 &&
this.status==200) {
                        getMakeElement(this);
                    }
                };
                xml_load1.open("GET","index.xml",true);
                xml_load1.send();
            }
            function getMakeElement(xml){
                var xmlDoc = xml.responseXML;
                var content = "";
                var x =
xmlDoc.getElementsByTagName( "make");
                for(i = 0;i < x.length;i++){
                    var name = "button"+i.toString();
                    var id1 = "value"+i.toString();

```

```

        content += "<br> <input type =
'text'" + "value=' " + x[i].childNodes[0].nodeValue + "' id="
+ id1+ " /><button id='"+name+" onclick =
displayUtil('"+x[i].childNodes[0].nodeValue+ "')> Click to
display value </button><br>";
    }
    document.getElementById("list").innerHTML
= content;
}

function displayUtil(name){
document.getElementById("list").style.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+="

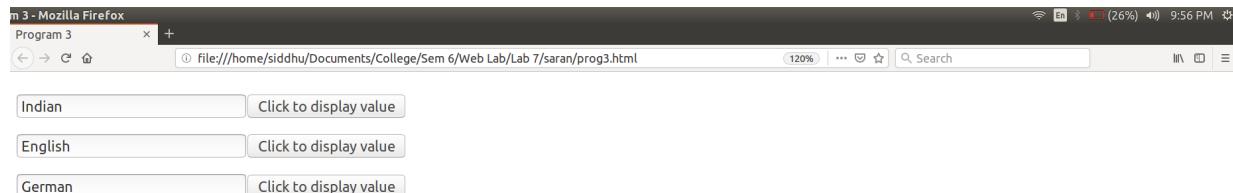
```

```

        }
    }
    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>

```

**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:**

prog.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)>
```

prog.xml:

```
<?xml version="1.0"?>
<!DOCTYPE catalog SYSTEM "prog4.dtd">
<catalog>
  <car>
```

```
<year>1998</year>
<make>Indian</make>
<model>BMW</model>
<color>Blue</color>
<engine>
    <number_of_cylinders>6</number_of_cylinders>
    <fuel_system>fuel_injected</fuel_system>
</engine>
<transmission_type>auto</transmission_type>
<number_of_doors>4</number_of_doors>
    <accessories radio = "yes" air_conditioning = "no"
power_windows = "yes" power_steering = "no" power_brakes =
"yes"></accessories>
</car>

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

<car>
    <make>German</make>
    <model>Volkswagen</model>
    <year>1957</year>
    <color>Red</color>
    <engine>
        <number_of_cylinders>4</number_of_cylinders>
        <fuel_system>carburated</fuel_system>
    </engine>
    <number_of_doors>6</number_of_doors>
    <transmission_type>auto</transmission_type>
```

```

        <accessories radio = "yes" air_conditioning = "no"
power_windows = "yes" power_steering = "no" power_brakes =
"yes"></accessories>
    </car>
<customer>
    <id>1</id>
    <name>Anu</name>
    <address>
        <city>Chennai</city>
        <state>TN</state>
        <zip>600073</zip>
    </address>
</customer>

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

prog.html:

```

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

```

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

    function dispXML(xml) {
        var xmlDoc = xml.responseXML;
        var x = xmlDoc.getElementsByTagName("customer");
        var content = "";
        alert(x.length);
        for (var i = 0; i < x.length; i++) {
            console.log(i);
            content += "Name: " +
x[i].getElementsByTagName("name")[0].childNodes[0].nodeValue
+ " <br>";
            content += "Address: <br>

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

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

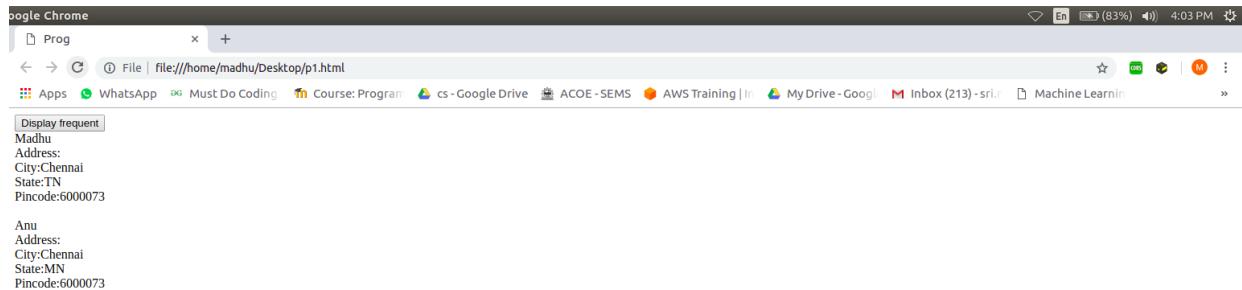
```

```

document.getElementById( "demo" ).innerHTML =
content

}

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

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

Thus the program is executed and output is obtained.

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

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

**PROGRAM:**

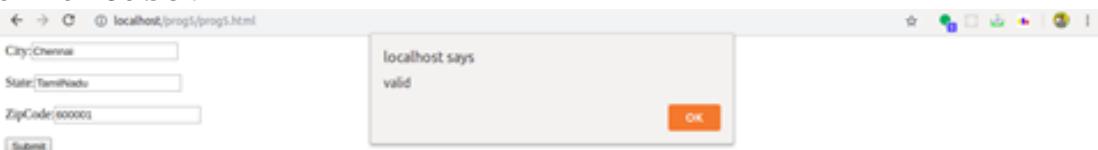
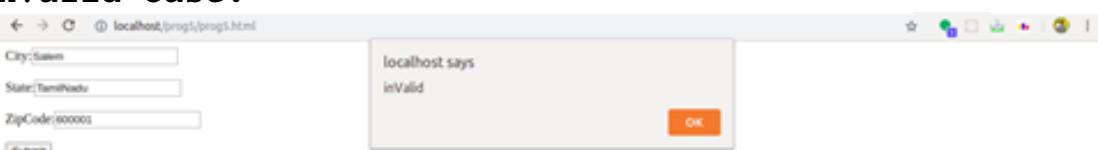
prog.html:

```
<html>
    <head>
        <title>Program 5</title>
        <meta charset="utf-8">
        <script>
            function myfunc(){
                var x =
document.getElementById("city").value;
                var y =
document.getElementById("state").value;
                var z = document.getElementById("zipcode")
.value;
                console.log(x);
                var req = x + " " + y + " " + z;
                var xml_load1 = new XMLHttpRequest();
                xml_load1.onreadystatechange = function(){
                    if(this.readyState == 4 && this.status ==
200){
                        alert(this.responseText);
                    }
                };
                xml_load1.open("GET", "prog5.php?
val="+req,true);
                xml_load1.send();
            }
        </script>
    </head>
    <body>
        <form>
            City:<input id="city" type="text" value="" /><br><br>
            State:<input id="state" type="text" value="" /><br><br>
            ZipCode:<input id="zipcode" type="text" value="" /><br><br>
            <input type="button" value="Submit"
onclick=myfunc() />
        </form>
    </body>
</html>
```

**prog.php:**

```
<?php
$address[ ] = "Chennai TamilNadu 600001";
$address[ ] = "Salem TamilNadu 600002";
$address[ ] = "Madurai TamilNadu 600003";

$val = $_REQUEST["val"];
$f1 = 0;
for($i=0;$i<sizeof($address);$i++){
    if($address[$i] == $val){
        $f1 = 1;
        break;
    }
}
if($f1==0){
    echo 'inValid';
}
else{
    echo 'valid';
}
?>
```

**OUTPUT:****Valid Case:****Invalid Case:****RESULT:**

Thus the program is executed and output is obtained.

# XML, DTD, Parsers, XSLT, XPATH and SAX

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

To design an XML document that stores information about patients in a hospital. The information must include their name, age, social security number, room number, primary insurance policy and secondary insurance policy.

**PROGRAM:**

```
<hospital>
    <patient>
        <name firstname="madhu" middlename="bala"
lastname="B" />
        <sex>Female</sex>
        <room-number>1</room-number>
        <age>21</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>9999999999</phone>
            <address>Chennai</address>
        </secondary-insurance-company>
    </patient>
    <patient>
        <name firstname="A" middlename="B" lastname="C" />
        <sex>Female</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>2222222222</phone>
            <address>Mumbai</address>
        </primary-insurance-company>
```

```
<secondary-insurance-company>
    <id>22</id>
    <group-id>51</group-id>
    <phone>8888888888</phone>
    <address>Delhi</address>
</secondary-insurance-company>
</patient>
<patient>
    <name firstname="X" middlename="Y" lastname="Z" />
    <sex>Female</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>3333333333</phone>
        <address>Mumbai</address>
    </primary-insurance-company>
    <secondary-insurance-company>
        <id>22</id>
        <group-id>51</group-id>
        <phone>7777777777</phone>
        <address>Delhi</address>
    </secondary-insurance-company>
</patient>
<patient>
    <name firstname="H" middlename="I" lastname="J" />
    <sex>Female</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>4444444444</phone>
        <address>Mumbai</address>
    </primary-insurance-company>
    <secondary-insurance-company>
        <id>22</id>
        <group-id>51</group-id>
        <phone>6666666666</phone>
```

```

        <address>Delhi</address>
    </secondary-insurance-company>
</patient>
</hospital>

```

**OUTPUT:**

```

<Google Chrome>
  pt.xml
  + File //file:///home/madhu/Desktop/xml/p1.xml
  Apps WhatsApp Must Do Coding Course: Program cs - Google Drive ACOE - SEMS AWS Training | In My Drive - Google Inbox (213) - sri Machine Learnin
  <hospital>
    <patient>
      <name firstname="madhu" middlename="bala" lastname="B"/>
      <sex>Female</sex>
      <room-number>1</room-number>
      <age>21</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>9999999999</phone>
        <address>Chennai</address>
      </secondary-insurance-company>
    </patient>
    <patient>
      <name firstname="A" middlename="B" lastname="C"/>
      <sex>Female</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>2222222222</phone>
        <address>Mumbai</address>
      </primary-insurance-company>
      <secondary-insurance-company>
        <id>22</id>
        <group-id>31</group-id>
        <phone>8888888888</phone>
        <address>Delhi</address>
      </secondary-insurance-company>
    </patient>
    <patient>
      <name firstname="X" middlename="Y" lastname="Z"/>
      <sex>Female</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>3333333333</phone>
        <address>Mumbai</address>
      </primary-insurance-company>
    </patient>
  </hospital>

```

**RESULT:**

Thus the program is executed and output is obtained.

**PROGRAM 2****AIM:**

To write a DTD for the above mentioned XML document.

**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)>
```

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

To create a CSS style sheet for the XML document and use it to display the document.

**PROGRAM:**

```
hospital
{
    color: black;
    background-color: white;
    height: 20px;
    font-size: 40px;
}

patient
{
    color: blue;
    font-size: 35px;
}

name,sex,room-number,age,social-security-number,primary-
insurance-company,secondary--insurance-company
{
    color: red;
    font-size: 30px;
    display: block;
    background-color:grey;
}
```

```
id,group-id,phone,address
{
    color:yellow;
    font-size:28px;
}
```

**OUTPUT:**

madhubala  
Female  
1  
21  
1234  
11 21 1111111111 Chennai  
21 41 9999999999 Chennai  
ABC  
Female  
2  
20  
1234  
12 31 2222222222 Mumbai  
22 51 8888888888 Delhi  
XYZ  
Female  
2  
20  
1234  
12 31 3333333333 Mumbai  
22 51 7777777777 Delhi

**RESULT:**

Thus the program is executed and output is obtained.

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

To create an XSLT style sheet for one patient element of XML 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>
    <head>
```

```

<title>patient's details</title>
</head>
<body bgcolor="#888888">
    <table border = "1">
        <tr>
            <th>FName</th>
            <th>MName</th>
            <th>LName</th>
            <th>age</th>
            <th>social-security-number</th>
            <th>primary-insurance-company:ID</th>
            <th>primary-insurance-company:Group</th>
            <th>primary-insurance-company:Phone</th>
            <th>primary-insurance-company:Address</th>
            <th>secondary-insurance-policy:ID</th>
            <th>secondary-insurance-policy:Group</th>
            <th>secondary-insurance-policy:Phone</th>
            <th>secondary-insurance-policy:Address</th>
        </tr>
        <xsl:for-each
select="hospital/patient[name='madhubala']">
            <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="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-policy/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	age	social-security-number	primary-insurance-company:ID	primary-insurance-company:Group	primary-insurance-company:Phone	primary-insurance-company:Address	secondary-insurance-policy:ID	secondary-insurance-policy:Group	secondary-insurance-policy:Phone	secondary-insurance-policy:Address
madhu	bala	B	21	1234	11	21	1111111111	Chennai	41		9999999999	Chennai

**RESULT:**

Thus the program is executed and output is obtained.

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

To modify the XSLT style sheet to display the overall details of all the patients.

**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>
    <head>
        <title>patient's details</title>
    </head>
    <body>
```

```

<table border = "1">
    <tr>
        <th>FName</th>
        <th>MName</th>
        <th>LName</th>
        <th>age</th>
        <th>social-security-number</th>
        <th>primary-insurance-company:ID</th>
        <th>primary-insurance-company:Group</th>
        <th>primary-insurance-company:Phone</th>
        <th>primary-insurance-company:Address</th>
        <th>secondary-insurance-policy:ID</th>
        <th>secondary-insurance-policy:Group</th>
        <th>secondary-insurance-policy:Phone</th>
        <th>secondary-insurance-policy: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="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-policy/id" /></td>
            <td><xsl:value-of select="secondary-
insurance-company/group-id" /></td>
            <td><xsl:value-of select="secondary-
insurance-company/phone" /></td>
        </tr>
    </xsl:for-each>
</table>

```

```

        <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	age	social-security-number	primary-insurance-company:ID	primary-insurance-company:Group	primary-insurance-company:Phone	primary-insurance-company:Address	secondary-insurance-policy:ID	secondary-insurance-policy:Group	secondary-insurance-policy:Phone	secondary-insurance-policy:Address
madhu	bala	B	21	1234	11	21	1111111111	Chennai	41	9999999999	Chennai	
A	B	C	20	1234	12	31	2222222222	Mumbai	51	8888888888	Delhi	
X	Y	Z	20	1234	12	31	3333333333	Mumbai	51	7777777777	Delhi	
H	I	J	20	1234	12	31	4444444444	Mumbai	51	6666666666	Delhi	

**RESULT:**

Thus the program is executed and output is obtained.