

SI.NO	DATE	TITLE	PAGE NO	MARKS	SIGN
1.		Creation of HTML pages with frames, links, tables and other tags			
2.		Client Side Programming Form Validation			
3.		Payslip Generator Using Inheritance			
4.		Multithreading Implementation			
5.		Creating Simple Application to Access Data Base Using JDBC			
6.		Program for Manipulating database			
7.		Java Applet to display moving text			
8.		Java Applet To Show Animation Of Bouncing Ball			
9.		Creation of XML document			
10.		Creation of XML document styles with XSLT Style Sheet			

Ex. No. : 01 Creation of HTML pages with frames, links, tables and other tags

Date :

AIM:

To create a simple webpage using HTML that includes all tags.

ALGORITHM:

1. Write a HTML program in the notepad with the tags such as

A. FRAMES

With frames, you can display more than one HTML document in the same browser window. Each HTML document is called a frame, and each frame is independent of the others.

The Frameset Tag

The <frameset> tag defines how to divide the window into frames.

The Frame Tag

The <frame> tag defines what HTML document to put into each frame.

Example:

```
<frameset cols="25%, 75 %">  
<frame src="frame_a.htm">  
<frame src="frame_b.htm">  
</frameset>
```

Tags and their Description:

<frameset> Defines a set of frames

<frame> Defines a sub window (a frame)

B. LINKS

A hyperlink is a reference (an address) to a resource on the web.

Example:

```
<a href="http://www.w3schools.com/">Visit W3Schools!</a>
```

The href Attribute

The href attribute defines the link "address".

The target Attribute

The target attribute defines where the linked document will be opened.

Tag and its Description:

<a> Defines an anchor

C. TABLES

Tables are defined with the <table> tag. A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). The letters td stands for "table data," which is the content of a data cell.

Example:

```
<table border="1">
<tr>
<td>Row 1, cell 1</td>
<td>Row 1, cell 2</td>
</tr>
</table>
```

Tags and their Description:

<Table> Defines a table

<th> Defines a table header

<tr> Defines a table row

<td> Defines a table cell

2. Use appropriate tags to apply the background colors and desired styles as Required and save the file with .html extension.

3. Run the program in the Web Browser.

PROGRAM:

Home.html

```
<html>
<head>
<title>Home</title>
</head>
<frameset rows="25%,*">
<frame src="frame1.html">
<frameset cols="25%,*">
<frame src="frame2.html" name="f2">
<frame src="frame3.html" name="f3">
</frameset>
</html>
```

Frame1.html

```
<html>
<head><title>frame1</title>
</head>
<body bgcolor="blue">
<h1 style="color:green;font-size:15pt">
<marquee bgcolor="#cccccc" loop="-1" scrollamount="6" width="100%">
SAVEETHA COLLEGE OF ENGINEERING AND
TECHNOLOGY </marquee>
</h1>
</body>
</html>
```

Frame2.html

```
<html>
<head><title>frame2</title>
<style type="text/css">
h1
{
font-size:25pt;color:pink;
```

```
}
</style>
</head>
<body bgcolor="red">
<h1>click the link</h1>
<a href="intro.html" target=f3>Introduction</a><br>
<a href="dept.html" target=f3>Departments</a><br>
<a href="ad.html" target=f3>ADDRESS</a><br>
<a href="feed.html" target=f3>Feedback</a><br>
<a href="gall.html" target=f3>Gallery</a><br>
</body>
</html>
```

Frame3.html

```
<html>
<head><title>1st page</title>
<link rel="stylesheet" type="text/css" href="C:\Documents and
Settings\Administrator\Desktop\ab\css1.css"/>
</head>
<body bgcolor="tan">
<h2> <center>YOU ARE IN HOME PAGE</center></h2>
</body>
</html>
```

Intro.html

```
<html>
<head><title>intro</title>
</head>
<body bgcolor="black">
<font color=red>
<p>
Welcome to SAVEETHA College of Engineering and Technology -
Affiliated to Anna University<br>
<br>
```

“SAVEETHA College of Engineering and Technology resolves to mould a human task force useful to the society through transparent methods that lead to continuous improvement of the resources and state-of-the-art methodologies conforming to recognized standards.”

</p>

</body>
</html>

Ad.html

<html>
<head><title>ADDRESS</title>
</head>
<body bgcolor="black">
<p>

Name:SAVEETHA College of Engineering and Technology

Location:Vandavasi

Contact No:04183-221444

Website: www.google.co.in

</p>
</body>
</html>

Dept.html

<html>
<head><title>Departments</title>
</head>
<body>
<div align="center">
<table border=2>
<tr>
<th>Dept code</th>
<th>Dept name</th>
</tr>

```
<tr>
<td>01</td>
<td>CSE</td>
</tr>
<tr>
<td>02</td>
<td>ECE</td>
</tr>
<tr>
<td>03</td>
<td>EEE</td>
</tr>
<tr>
<td>04</td>
<td>IT</td>
</tr>
<tr>
<td>05</td>
<td>MECH</td>
</tr>
<tr>
<td>06</td>
<td>AERO</td>
</tr>
</table>
</div>
</body>
</html>
```

Feed.html

```
<html>
<head><title>feed</title>
</head>
<body bgcolor="black">
<p>
<font color=green>
```

To give your feedback mail to google_feedback@edu.in

</p>

</body>

</html>

Gall.html

<html>

<head><title>gall</title>

</head>

<body bgcolor="pink">

<p>

College Front View

</p>

</body>

</html>

OUTPUT:



RESULT:

Thus the HTML pages are created using frames, links, tables and other tags and have been successfully executed.

Ex. No. : 2

Client Side Programming Form Validation

Date :

AIM:

To write a Java script program for Form Validation including text field, radio buttons, check boxes, list box and other controls.

ALGORITHM:

Client Side Programming - Form validation

1. Create a form by including all the elements of java script.

Ex: `<input type="button" value="OK">`

`<input type="checkbox" name="c">`

`<input type="text">` etc.

2. Create a button in such a way that when the button is clicked the form has to be validated (i.e. it has to check whether all fields in the form are filled and the data entered are valid).

PROGRAM:

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
  <b><font face="Verdana"><font size=+1>Form Validation Example:
Feedback Form</font></font></b>
  <br><script LANGUAGE="JavaScript">
<!--
function ValidateForm(form){
  ErrorText= "";
  if (form.username.value == "") {ErrorText= "\nPlease enter your name"}

  if ((form.email.value == "" || form.email.value.indexOf('@', 0) == -1) ||
form.email.value.indexOf('.') == -1){
    ErrorText+=  "\nPlease enter a proper value for your email"}
    if (form.feedback.value == "") {ErrorText+= "\nPlease enter your
feedback"}
    if (ErrorText!= "") {
      alert("Error :" + ErrorText);
      return false;
    }
    if (ErrorText= "") { form.submit() }

  ErrorText= "";
  if ( ( form.gender[0].checked == false ) && ( form.gender[1].checked ==
false ) ) { alert ( "Please choose your Gender: Male or Female" ); return
false; }
  if (ErrorText= "") { form.submit() }

  ErrorText= "";
  if ( form.age.selectedIndex == 0 ) { alert ( "Please select your Age." );
return false; }
  if (ErrorText= "") { form.submit() }
```

```
alert ( "Thank u" )
}
```

```
-->
```

```
</script>
```

```
<br><form name="feedback" action="mailto:webgimmicks@email.com"
method=post>
```

```
<dl>
```

```
<dt>
```

```
<font face="Verdana">
```

```
<pre>
```

Please enter the following(To test form validation, skip entering one of the fields and click 'Submit' button):

```
</pre>
```

```
</font>
```

```
<font face="Verdana">Name: &nbsp;</font><input type="text" value=""
name="username" size=30></dt><br>
```

```
<br><dt> <font face="Verdana">E-mail: &nbsp;</font><input type="text"
value="" name="email" size=30></dt><br>
```

```
<br><dt><font face="Verdana">Your Gender:</font>
```

```
  <input type="radio" name="gender" value="Male">
```

Male

```
  <input type="radio" name="gender" value="Female">
```

Female </dt><dt> </dt>


```
  <font face="Verdana, Arial, Helvetica, sans-serif">Your Age</font>:
```

```
<select name="age"> <option value="">Please Select an Option:</option>
```

```
<option value="0-18 years">0-18 years</option> <option value="18-30
```

```
years">18-30 years</option> <option value="30-45 years">30-45
```

```
years</option> <option value="45-60 years">45-60 years</option>
```

```
<option value="60+ years">60+ years</option> </select>
```

```
<br>
```

```
<br><font face="Verdana">Feedback:</font>
```

```
<dt><textarea rows=7 cols=60 name="feedback"></textarea></dt>
```

```
</dl>
```

```
<input type="button" name="SubmitButton" value="Submit"
```

```
onClick="ValidateForm(this.form)">
```

```
<input type="reset" value="Reset">
```

```
</form>
```

```
<pre><font face="Verdana"><font size=+0>&nbsp; &nbsp;</font></font>
```

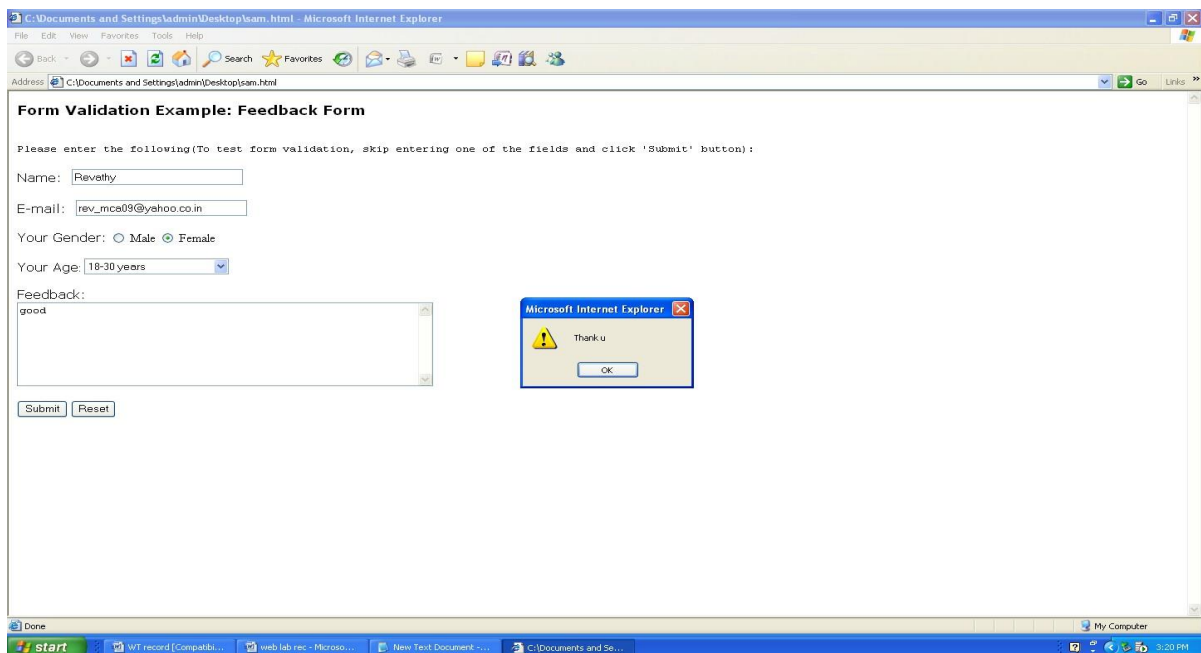
```
</pre>
```

```
</form>
```

```
</body>
```

```
</html>
```

OUTPUT:



RESULT:

Thus the Java script program for Form Validation including text field, radio buttons, check boxes, list box and other controls has been successfully written and output is verified.

Ex. No. : 2
Date:

Payslip Generator Using Inheritance

AIM:

To develop a java application to generate pay slip for different category of employees using the concept of inheritance.

ALGORITHM:

1. Start
2. Create the class Employee with name, Empid, address, mailid, mobileneno as data members.
3. Inherit the classes Programmer, Asstprofessor, Associateprofessor and Professor from employee class.
4. Add Basic Pay (BP) as the member of all the inherited classes.
5. Calculate DA as 97% of BP, HRA as 10% of BP, PF as 12% of BP, Staff club fund as 0.1% of BP.
6. Calculate gross salary and net salary.
7. Generate payslip for all categories of employees.
8. Create the objects for the inherited classes and invoke the necessary methods to display the Payslip
9. Stop

PROGRAM:

```
package java_project;
import java.util.*;
class employee
{
    int emp_id;
    long ph_no;
    String name, address, mailid;
    Scanner sc = new Scanner(System.in);
    void scdata()
    {

        System.out.println("Enter the name:");
        name = sc.nextLine();
        System.out.println("Enter the mail id:");
        mailid = sc.nextLine();
        System.out.println("Enter Address:");
        address = sc.nextLine();
        System.out.println("Enter employee id ");
        emp_id = sc.nextInt();
        System.out.println("Enter Phone Number:");
        ph_no = sc.nextLong();
    }
    void display()
    {
        System.out.println("Employee Name: "+name);
        System.out.println("Employee id : "+emp_id);
        System.out.println("Mail id : "+mailid);
        System.out.println("Address: "+address);
        System.out.println("Phone Number: "+ph_no);
    }
}
class Programmer extends employee
{
    double salary,bp,da,hra,pf,club,net,gross;
    void scprogrammer()
    {
        System.out.println("Enter basic pay");
        bp = sc.nextDouble();
    }
    void calculateprog()
    {
        da=(0.97*bp);
```

```

hra=(0.10*bp);
pf=(0.12*bp);
club=(0.1*bp);
gross=(bp+da+hra);
net=(gross-pf-club);
System.out.println("PAY SLIP FOR PROGRAMMER");
System.out.println("Basic Pay:" +bp);
System.out.println("DA: "+da);
System.out.println("HRA: "+hra);
System.out.println("PF: "+pf);
System.out.println("CLUB: "+club);
System.out.println("GROSS PAY: "+gross);
System.out.println("NET PAY: "+net);
}
}
class Asstprofessor extends employee
{
double salary,bp,da,hra,pf,club,net,gross;
void scasst()
{
System.out.println("Enter basic pay");
bp = sc.nextDouble();
}
void calculateasst()
{
da=(0.97*bp);
hra=(0.10*bp);
pf=(0.12*bp);
club=(0.1*bp);
gross=(bp+da+hra);
net=(gross-pf-club);
System.out.println("PAY SLIP FOR ASSISTANT PROFESSOR");
System.out.println("Basic Pay: "+bp);
System.out.println("DA: "+da);
System.out.println("HRA: "+hra);
System.out.println("PF: "+pf);
System.out.println("CLUB: "+club);
System.out.println("GROSS PAY: "+gross);
System.out.println("NET PAY: "+net);
}
}
class Associateprofessor extends employee
{
double salary,bp,da,hra,pf,club,net,gross;

```

```

void scassociate()
{
    System.out.println("Enter basic pay");
    bp = sc.nextDouble();
}
void calculateassociate()
{
    da=(0.97*bp);
    hra=(0.10*bp);
    pf=(0.12*bp);
    club=(0.1*bp);
    gross=(bp+da+hra);
    net=(gross-pf-club);
    System.out.println("PAY SLIP FOR ASSOCIATE PROFESSOR");
    System.out.println("Basic Pay: "+bp);
    System.out.println("DA: "+da);
    System.out.println("HRA: "+hra);
    System.out.println("PF: "+pf);
    System.out.println("CLUB: "+club);
    System.out.println("GROSS PAY: "+gross);
    System.out.println("NET PAY: "+net);
}
}
class Professor extends employee
{
    double salary,bp,da,hra,pf,club,net,gross;
    void scprofessor()
    {
        System.out.println("Enter basic pay");
        bp = sc.nextDouble();
    }
    void calculateprofessor()
    {
        da=(0.97*bp);
        hra=(0.10*bp);
        pf=(0.12*bp);
        club=(0.1*bp);
        gross=(bp+da+hra);
        net=(gross-pf-club);
        System.out.println("PAY SLIP FOR PROFESSOR");
        System.out.println("Basic Pay: "+bp);
        System.out.println("DA: "+da);
        System.out.println("HRA: "+hra);
        System.out.println("PF: "+pf);
    }
}

```

```

System.out.println("CLUB: "+club);
System.out.println("GROSS PAY: "+gross);
System.out.println("NET PAY: "+net);
}
}

```

```

public class EX_4 {
    public static void main(String[] args) {
        int choice,cont;
        Scanner c = new Scanner(System.in);
        do
        {
            System.out.println("PAYROLL");
            System.out.println(" 1.PROGRAMMER \n 2.ASSISTANT
PROFESSOR \n 3.ASSOCIATE PROFESSOR \n 4.PROFESSOR ");
            System.out.println("Enter your choice:");
            choice=c.nextInt();
            if(choice==1)
            {
                Programmer p=new Programmer();
                p.sdata();
                p.scprogrammer();
                p.display();
                p.calculateprog();
            }
            else if(choice==2)
            {
                Asstprofessor asst =new Asstprofessor();
                asst.sdata();
                asst.scasst();
                asst.display();
                asst.calculateasst();
            }
            else if(choice==3)
            {
                Associateprofessor asso=new Associateprofessor();
                asso.sdata();
                asso.scassociate();
                asso.display();
                asso.calculateassociate();
            }
            else if(choice==4)
            {
                Professor prof=new Professor();

```

```
        prof.scddata();
        prof.scprofessor();
        prof.display();
        prof.calculateprofessor();
    }
else {
    System.out.println("Invalid choice!");
}
    System.out.print("Please enter 0 to quit and 1 to continue: ");
    cont=c.nextInt();
    }    while(cont==1);
}
}
```

OUTPUT:

```
PAYROLL
1.PROGRAMMER
2.ASSISTANT PROFESSOR
3.ASSOCIATE PROFESSOR
4.PROFESSOR
Enter your choice:
1
Enter the name:
Sowjanya
Enter the mail id:
sowjanya@gmail.com
Enter Address:
ABC Street
Enter employee id
12345
Enter Phone Number:
9176563333
Enter basic pay
150000
Employee Name: Sowjanya
Employee id : 12345
Mail id : sowjanya@gmail.com
Address: ABC Street
Phone Number: 9176563333
PAY SLIP FOR PROGRAMMER
Basic Pay:150000.0
DA: 145500.0
HRA: 15000.0
PF: 18000.0
CLUB: 15000.0
GROSS PAY: 310500.0
NET PAY: 277500.0
```

RESULT:

Thus the Java application to generate pay slip for different category of employees was implemented using inheritance and the program was executed successfully.

Ex. No:4

Multithreading Implementation

Date:

AIM:

To write a java program that implements a multi-threaded application.

ALGORITHM:

1. Start
2. Create a class Songs which implements first thread that plays songs.
3. run() method implements the code to be executed when thread gets executed.
4. Create a class Songqueue which implements second thread keeps on adding songs simultaneously to be played next.
5. The Multithreading is performed, and the task switched between multiple threads.
7. The sleep () method makes the thread to suspend for the specified time.
8. Stop.

PROGRAM:

```
package java_project;

class Songs implements Runnable{
    public void run() {
        for(int i=0;i<=5;i++) {
            System.out.println("Playing song:" + (i+1));
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
};

class Songqueue extends Thread{
    public void run() {
        for(int i=1;i<=6;i++) {
            System.out.println("Adding song:" + i + " to the queue");
            try {
                Thread.sleep(3000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
};

public class Multithreading_2 {
    public static void main(String[] args) {
        Songs s = new Songs();
    }
}
```

```
Thread t1 = new Thread(s);
t1.start();

Songqueue q = new Songqueue();
q.start();
t1.setName("Songs thread");
q.setName("Song adding to queue");
if(t1.isAlive())
    System.out.println(t1.getName() + " is alive");
try {
    t1.join();
} catch (InterruptedException e) {
    e.printStackTrace();
}
try {
    q.join();
} catch (InterruptedException e) {
    e.printStackTrace();
}

System.out.println("Both works are done!");

}

}
```

OUTPUT:

```
Playing song:1  
Adding song:1 to the queue  
Songs thread is alive  
Adding song:2 to the queue  
Adding song:3 to the queue  
Adding song:4 to the queue  
Playing song:2  
Adding song:5 to the queue  
Adding song:6 to the queue  
Playing song:3  
Playing song:4  
Playing song:5  
Playing song:6
```

RESULT:

Thus, the Java program for multi-threaded application has been implemented and executed successfully.

Ex. No. : 05 Creating Simple Application to Access Data Base Using JDBC

Date:

AIM:

To develop and retrieve Online Applications for displaying student's details using JDBC.

ALGORITHM:

- 1. Import the packages:** Requires that you include the packages containing the JDBC classes needed for database programming. Most often, using *import java.sql.** will suffice.
- 2. Register the JDBC driver:** Requires that you initialize a driver so you can open a communication channel with the database.
- 3. Open a connection:** Requires using the *DriverManager.getConnection()* method to create a Connection object, which represents a physical connection with the database.
- 4. Execute a query:** Requires using an object of type Statement for building and submitting an SQL statement to the database.
- 5. Extract data from result set:** Requires that you use the appropriate *ResultSet.getXXX()* method to retrieve the data from the result set.
- 6. Clean up the environment:** Requires explicitly closing all database resources versus relying on the JVM's garbage collection.

PROGRAM:

```
import java.sql.*;

public class FirstExample {

    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver"; static final
    String DB_URL = "jdbc:mysql://localhost/EMP";

    // Database credentials
    static final String USER = "username"; static final String PASS = "password";

    public static void main(String[] args) { Connection conn = null;
    Statement stmt = null;
    try{

        //STEP 2: Register JDBC driver Class.forName("com.mysql.jdbc.Driver");

        //STEP 3: Open a connection System.out.println("Connecting to
        database...");

        conn = DriverManager.getConnection(DB_URL,USER,PASS);

        //STEP 4: Execute a query System.out.println("Creating statement...");
        stmt = conn.createStatement();

        String sql;
        sql = "SELECT id, first, last, age FROM Employees";
        ResultSet rs = stmt.executeQuery(sql);

        //STEP 5: Extract data from result set while(rs.next()){

            //Retrieve by column name
            int id = rs.getInt("id"); int age = rs.getInt("age");
            String first = rs.getString("first");
            String last = rs.getString("last");

            //Display values System.out.print("ID: " + id);
            System.out.print(", Age: " + age);
            System.out.print(", First: " + first); System.out.println(", Last: " + last);
        }
    }
```

```
//STEP 6:

Clean-up environment rs.close();
stmt.close();
conn.close();
}
catch(SQLException se){
    //Handle errors for JDBC se.printStackTrace();
}
catch(Exception e){
    //Handle errors for Class.forName e.printStackTrace();
}
finally{
    //finally block used to close resources
    try{
        if(stmt!=null)
            stmt.close();
    }
    catch(SQLException se2){
        }// nothing we can do
    try{
        if(conn!=null)
            conn.close();
    }
    catch(SQLException se)
    {
        se.printStackTrace();
    }//end finally try
}//end try
System.out.println("Goodbye!");
```

```
}//end main
```

```
}//end FirstExample
```

OUTPUT:

Connecting to database...

Creating statement...

ID: 100, Age: 18, First: Zara, Last: Ali

ID: 101, Age: 25, First: Mahnaz, Last: Fatma

RESULT:

Thus a 3-tier application for displaying student's mark list using java server pages and MS-Access database has been developed.

Ex. No. : 06

Program for Manipulating database

Date:

AIM:

To develop a Online application with data access for displaying student's mark list using java servlet and MS-Access database.

ALGORITHM:

Step 1: Start the program.

Step 2: Create a MS-Access database table named student for storing student's marks.

Step 3: Create an html form for entering the register number of the student.

Step 4: Open control panel and make the database connectivity settings.

Step 5: Create a servlet for processing the student's marks from the database.

Step 6: Deploy the servlet and the html page in the Eclipse IDE to create the application package.

Step 7: Run the application in the IDE.

Step 8: The student's mark list for the corresponding register number will be displayed.

Step 9: Stop the program.

PROGRAM:

mark.html

```
<html>
<body bgcolor=#779966 text=black>
<center><font color=blue><h2>LOGIN FORM</h2></font></center>
<form method =get action="http://localhost:8084/marklist/Student">
<table align=center border=0 bgcolor=#777799>
<tr><td>REG NUMBER</td><td>:</td><td><input type=text name=login
value="">
</td>
</table><br><br><center>
<input type=submit value=SUBMIT>
<input type=Reset value=RESET></center>
</form>
</body>
</html>
```

Student.java

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

public class Student extends HttpServlet
{
    int m1,m2,m3,m4,m5,m6;
    public void doGet(HttpServletRequest r,HttpServletResponse rs)throws
    IOException,ServletException
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection con=DriverManager.getConnection("jdbc:odbc:student");
            Statement st=con.createStatement();
            ResultSet res;
            String query="select *from student";
            res=st.executeQuery(query);
            rs.setContentType("text/html");
            PrintWriter out=rs.getWriter();
            out.println("<html><body>");
            int reg1=Integer.parseInt(r.getParameter("login"));
            int valid=0;
            while(res.next())
            {
                if(res.getInt(1)==reg1)
                {
                    valid=1;
                    m1=res.getInt(2);
                    m2=res.getInt(3);
                    m3=res.getInt(4);
                    m4=res.getInt(5);
                    m5=res.getInt(6);
                    m6=res.getInt(7);
                    break;
                }
            }
        }
    }
}
```

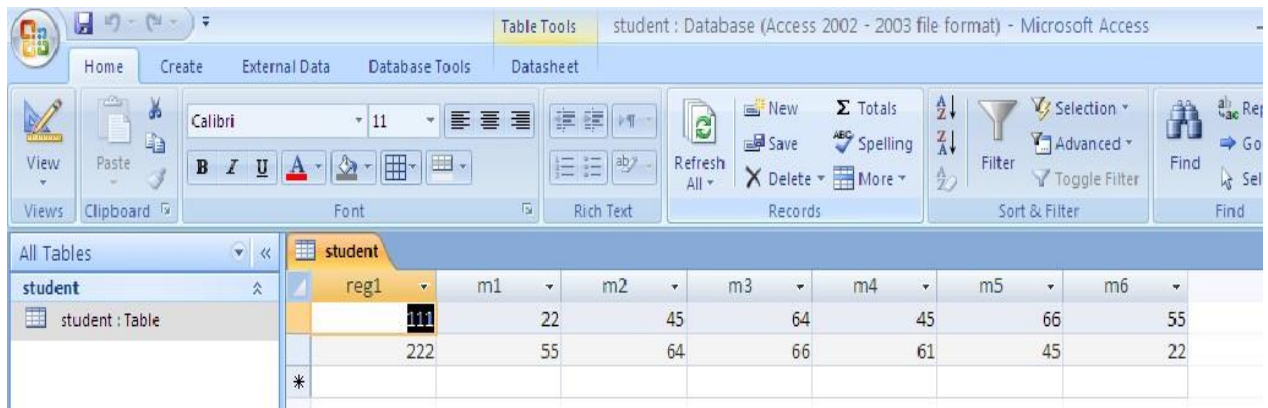
```

    }
    }
    if(valid==0)
    {
        out.println("not valid RegNo");
    }
    else
    {
        out.println("<html><body bgcolor=#ccbbaa");
        out.println("<p>&nbsp;</p>");
        out.println("<center><font color=blue><h2>GOOGLE
        </h2></font></center>");
        out.println("<p align=center>");
        out.println("MARK1<input type=text name=mark1 value="+m1+">");
        out.println("MARK2<input type=text name=mark2 value="+m2+">");
        out.println("<p>");
        out.println("<p align=center>");
        out.println("MARK3<input type=text name=mark3 value="+m3+">");
        out.println("MARK4<input type=text name=mark4 value="+m4+">");
        out.println("<p>");
        out.println("<p align=center>");
        out.println("MARK5<input type=text name=mark5 value="+m5+">");
        out.println("MARK6<input type=text name=mark6 value="+m6+">");
        int tot1;
        double avg1;
        tot1=m1+m2+m3+m4+m5+m6;
        avg1=((tot1)/6.0);
        out.println("<hr>");
        out.println("<p>&nbsp;</p>");
        out.println("<p>&nbsp;</p>");
        out.println("<p align=center>");
        out.println("TOTAL<input type=text name=total value="+tot1+">");
        out.println("AVERAGE<input type=text name=avg value="+avg1+">");
        out.println("<p>");
        out.println("</body></html>");
    }
}

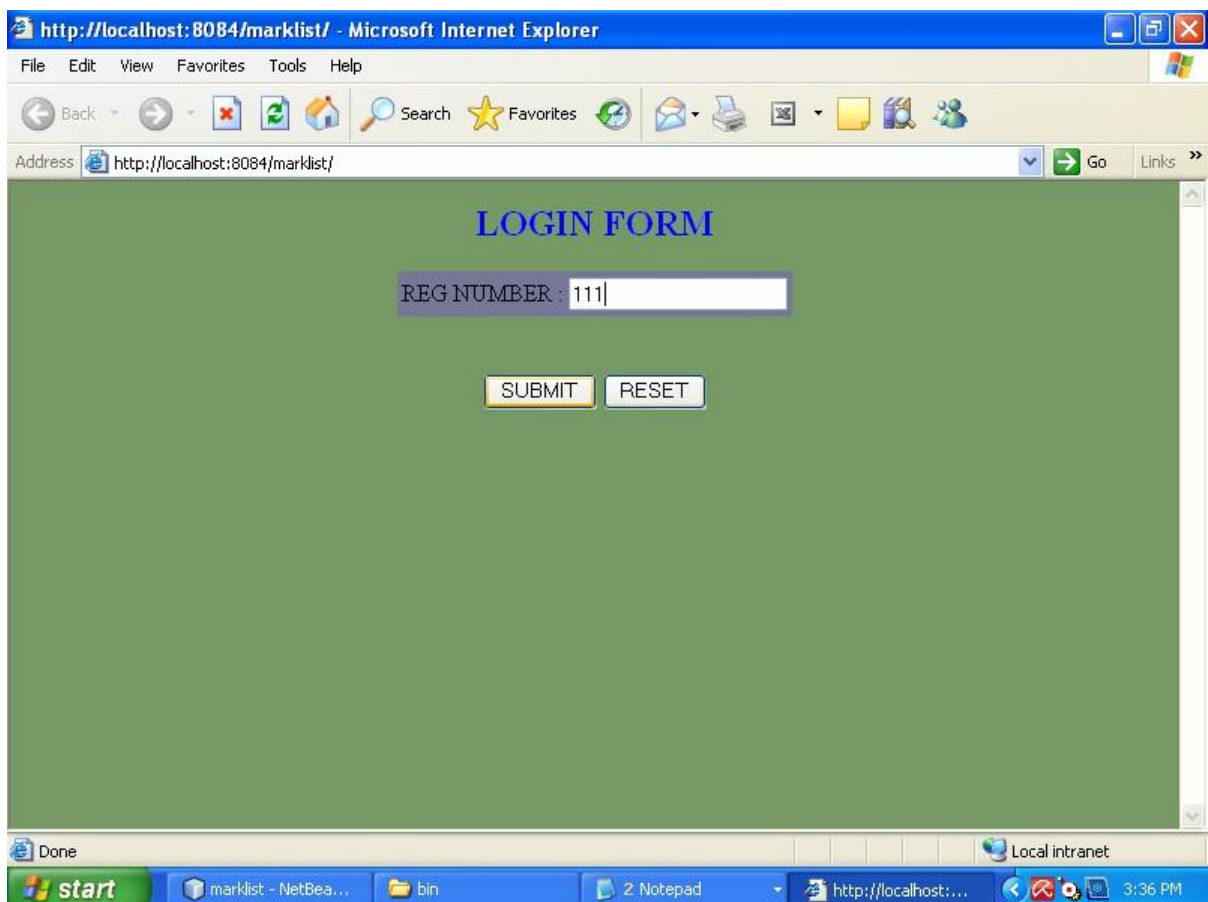
```

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

OUTPUT:



reg1	m1	m2	m3	m4	m5	m6
111	22	45	64	45	66	55
222	55	64	66	61	45	22



http://localhost:8084/marklist/ - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Reload Print Mail Links

Address http://localhost:8084/marklist/ Go

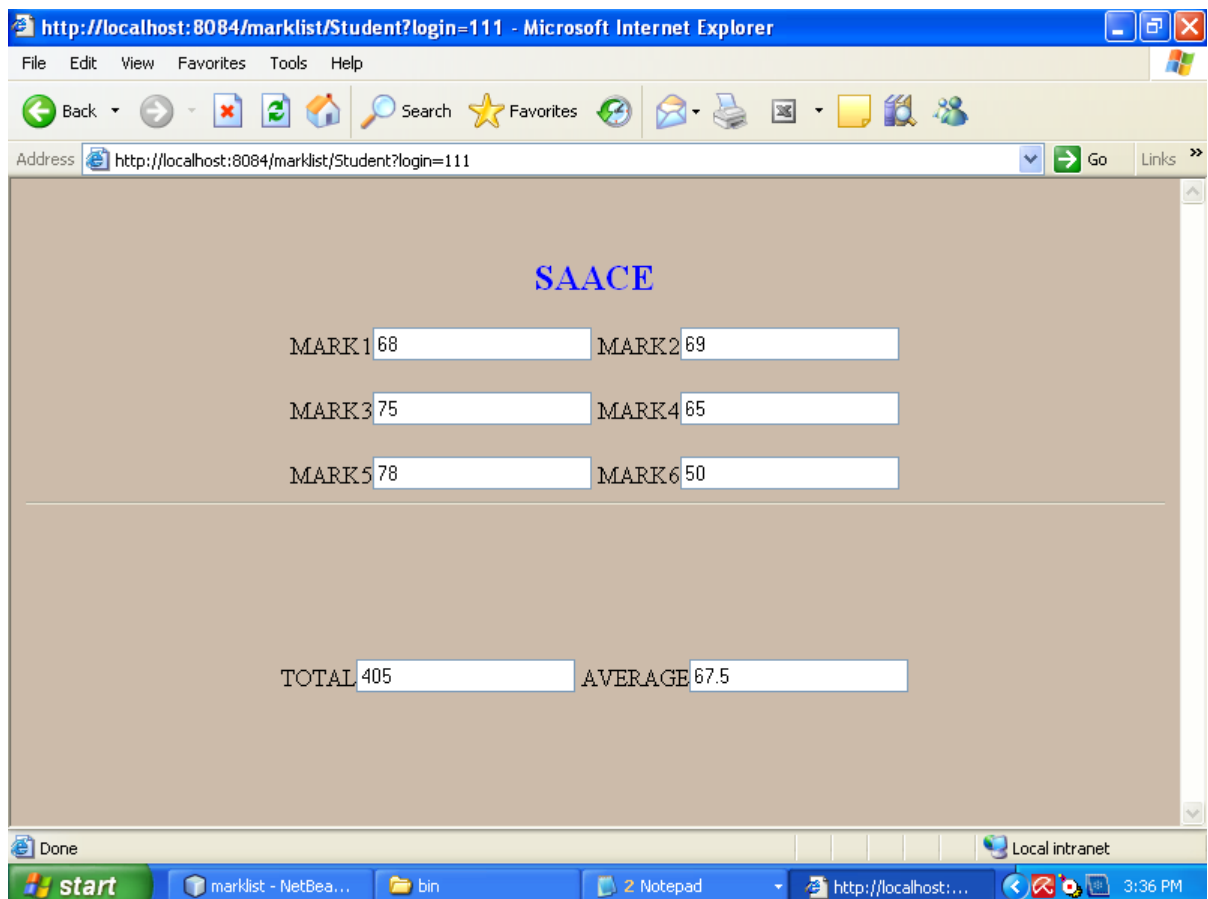
LOGIN FORM

REG NUMBER : 111

SUBMIT RESET

Done Local intranet

start marklist - NetBea... bin 2 Notepad http://localhost:... 3:36 PM



RESULT:

Thus a 3-tier application for displaying student's mark list using java servlet and MS-Access database has been successfully developed.

EX. NO :7 JAVA APPLET TO DISPLAY MOVING TEXT

DATE:

AIM:

To develop a java applet program to display moving text

ALGORITHM:

1. Start
2. Import Applet Libraries
3. Create a class MovingContent
4. Extend the class from Applet
5. Initialize the flag
6. Assign thread value to display the time duration of the text
7. Stop

PROGRAM:

```
package Applet;
import java.awt.*;
import java.applet.*;
public class MovingContent extends Applet implements Runnable {
    // enter message
    String msg = "Welcome to Includehelp.";
    Thread t = null;
    // initialize here.
    int state;
    boolean stopFlag;
    // Set colors and initialize text..
    public void init() {
        setBackground(Color.cyan);
        setForeground(Color.red);
    }

    // Start the text....
    public void start() {
        t = new Thread(this);
        stopFlag = false;
        t.start();
    }
    // Entry point which runs the text.
    public void run() {
        char ch;
        // Display text repeated times.
        for (;;)
        {
```

```
try {
    repaint();
    Thread.sleep(250);
    ch = msg.charAt(0);
    msg = msg.substring(1, msg.length());
    msg += ch;
    if (stopFlag)
        break;
} catch (InterruptedException e) {
    System.out.println(e);
}
}
}
// Pause the text.
public void stop() {
    stopFlag = true;
    t = null;
}
// Display the text.
public void paint(Graphics g) {
    g.drawString(msg, 50, 30);
}
}
```

OUTPUT:



Welcome To Includehelp

Applet started.

RESULT:

Thus a java applet program to display a moving text was successfully developed.

EX. NO :8 Java Applet To Show Animation Of Bouncing Ball

DATE:

AIM:

To develop a java applet program to show the animation of a bouncing ball.

ALGORITHM:

1. Start
2. Import the applet packages
3. Create a class BouncingBall and extend it from Applet
4. Initialize the coordinates of x and y
5. Assign the value to flag and start the thread
6. Assign value to the coordinates so that if a ball hits the edge it returns back
7. Stop.

PROGRAM:

```
package Applet;
import java.applet.*;
import java.awt.*;
public class BouncingBall extends Applet implements Runnable {
    // x,y coordinates and radius of the circle.
    int x = 150, y = 50, r = 20;
    int dx = 11, dy = 7;

    // create thread.
    Thread t;
    boolean stopFlag;

    // Function to start thread.
    public void start() {
        t = new Thread(this);
        stopFlag = false;
        t.start();
    }

    // Draw circle from its present position.
    public void paint(Graphics g) {
        g.setColor(Color.red);
        g.fillOval(x - r, y - r, r * 2, r * 2);
    }

    // function to move the image.
    public void run() {
        while (true) {
```

```

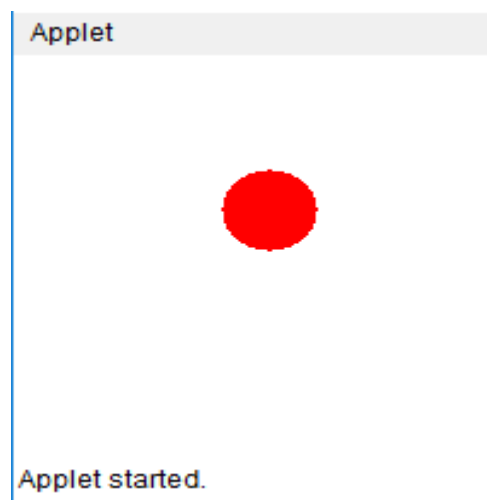
        if (stopFlag)
            break;
        // Bounce if we've hit an edge.
        if ((x - r + dx < 0) || (x + r + dx > bounds().width)) dx = -dx;
        if ((y - r + dy < 0) || (y + r + dy > bounds().height)) dy = -dy;
        // Move the circle.
        x += dx;
        y += dy;

        try {
            Thread.sleep(100);
        } catch (Exception e) {
            System.out.println(e);
        };
        // print circle again n again.
        repaint();
    }
}

// function to stop printing.
public void stop() {
    stopFlag = true;
    t = null;
}
}

```

OUTPUT:



RESULT:

Thus a java applet program to show the animation of the bouncing ball has been developed.

Ex. No. : 09

Creation of XML document

Date :

AIM:

To write a XML program for creating a cd catalog.

ALGORITHM:

1. Create a .xml file with the following header,

<?xml version="1.0"?>

2. Include all the description of the CDs inside the <CATALOG> tag.

3. For each CD create a separate tag within the parent tag

<CATALOG>which includes

item title

artist

country

company

price and

year.

4. Call the CSS file in which the format has been specified.

5. Run the XML program.

PROGRAM:

cd_catalog_css.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
  <CD>
    <TITLE>Greatest Hits</TITLE>
    <ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>RCA</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1982</YEAR>
  </CD>
  <CD>
    <TITLE>Still got the blues</TITLE>
    <ARTIST>Gary Moore</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>Virgin records</COMPANY>
    <PRICE>10.20</PRICE>
```

<YEAR>1990</YEAR>

</CD>

<CD>

<TITLE>Eros</TITLE>

<ARTIST>Eros Ramazzotti</ARTIST>

<COUNTRY>EU</COUNTRY>

<COMPANY>BMG</COMPANY>

<PRICE>9.90</PRICE>

<YEAR>1997</YEAR>

</CD>

<CD>

<TITLE>One night only</TITLE>

<ARTIST>Bee Gees</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>Polydor</COMPANY>

<PRICE>10.90</PRICE>

<YEAR>1998</YEAR>

</CD>

<CD>

<TITLE>Sylvias Mother</TITLE>

<ARTIST>Dr.Hook</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>CBS</COMPANY>

<PRICE>8.10</PRICE>

<YEAR>1973</YEAR>

</CD>

<CD>

<TITLE>Maggie May</TITLE>

<ARTIST>Rod Stewart</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>Pickwick</COMPANY>

<PRICE>8.50</PRICE>

<YEAR>1990</YEAR>

</CD>

<CD>

<TITLE>Romanza</TITLE>
<ARTIST>Andrea Bocelli</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>Polydor</COMPANY>
<PRICE>10.80</PRICE>
<YEAR>1996</YEAR>
</CD>

<CD>
<TITLE>When a man loves a woman</TITLE>
<ARTIST>Percy Sledge</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Atlantic</COMPANY>
<PRICE>8.70</PRICE>
<YEAR>1987</YEAR>
</CD>

<CD>
<TITLE>Black angel</TITLE>
<ARTIST>Savage Rose</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>Mega</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1995</YEAR>
</CD>

<CD>
<TITLE>1999 Grammy Nominees</TITLE>
<ARTIST>Many</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Grammy</COMPANY>
<PRICE>10.20</PRICE>
<YEAR>1999</YEAR>
</CD>

<CD>
<TITLE>For the good times</TITLE>
<ARTIST>Kenny Rogers</ARTIST>
<COUNTRY>UK</COUNTRY>

<COMPANY>Mucik Master</COMPANY>

<PRICE>8.70</PRICE>

<YEAR>1995</YEAR>

</CD>

<CD>

<TITLE>Big Willie style</TITLE>

<ARTIST>Will Smith</ARTIST>

<COUNTRY>USA</COUNTRY>

<COMPANY>Columbia</COMPANY>

<PRICE>9.90</PRICE>

<YEAR>1997</YEAR>

</CD>

<CD>

<TITLE>Tupelo Honey</TITLE>

<ARTIST>Van Morrison</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>Polydor</COMPANY>

<PRICE>8.20</PRICE>

<YEAR>1971</YEAR>

</CD>

<CD>

<TITLE>Soulsville</TITLE>

<ARTIST>Jorn Hoel</ARTIST>

<COUNTRY>Norway</COUNTRY>

<COMPANY>WEA</COMPANY>

<PRICE>7.90</PRICE>

<YEAR>1996</YEAR>

</CD>

<CD>

<TITLE>The very best of</TITLE>

<ARTIST>Cat Stevens</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>Island</COMPANY>

<PRICE>8.90</PRICE>

<YEAR>1990</YEAR>

</CD>

<CD>

<TITLE>Stop</TITLE>

<ARTIST>Sam Brown</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>A and M</COMPANY>

<PRICE>8.90</PRICE>

<YEAR>1988</YEAR>

</CD>

<CD>

<TITLE>Bridge of Spies</TITLE>

<ARTIST>T` Pau</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>Siren</COMPANY>

<PRICE>7.90</PRICE>

<YEAR>1987</YEAR>

</CD>

<CD>

<TITLE>Private Dancer</TITLE>

<ARTIST>Tina Turner</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>Capitol</COMPANY>

<PRICE>8.90</PRICE>

<YEAR>1983</YEAR>

</CD>

<CD>

<TITLE>Midt om natten</TITLE>

<ARTIST>Kim Larsen</ARTIST>

<COUNTRY>EU</COUNTRY>

<COMPANY>Medley</COMPANY>

<PRICE>7.80</PRICE>

<YEAR>1983</YEAR>

</CD>

<CD>

<TITLE>Pavarotti Gala Concert</TITLE>

<ARTIST>Luciano Pavarotti</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>DECCA</COMPANY>

<PRICE>9.90</PRICE>

<YEAR>1991</YEAR>

</CD>

<CD>

<TITLE>The dock of the bay</TITLE>

<ARTIST>Otis Redding</ARTIST>

<COUNTRY>USA</COUNTRY>

<COMPANY>Atlantic</COMPANY>

<PRICE>7.90</PRICE>

<YEAR>1987</YEAR>

</CD>

<CD>

<TITLE>Picture book</TITLE>

<ARTIST>Simply Red</ARTIST>

<COUNTRY>EU</COUNTRY>

<COMPANY>Elektra</COMPANY>

<PRICE>7.20</PRICE>

<YEAR>1985</YEAR>

</CD>

<CD>

<TITLE>Red</TITLE>

<ARTIST>The Communards</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>London</COMPANY>

<PRICE>7.80</PRICE>

<YEAR>1987</YEAR>

</CD>

<CD>

<TITLE>Unchain my heart</TITLE>

<ARTIST>Joe Cocker</ARTIST>

<COUNTRY>USA</COUNTRY>

<COMPANY>EMI</COMPANY>

<PRICE>8.20</PRICE>

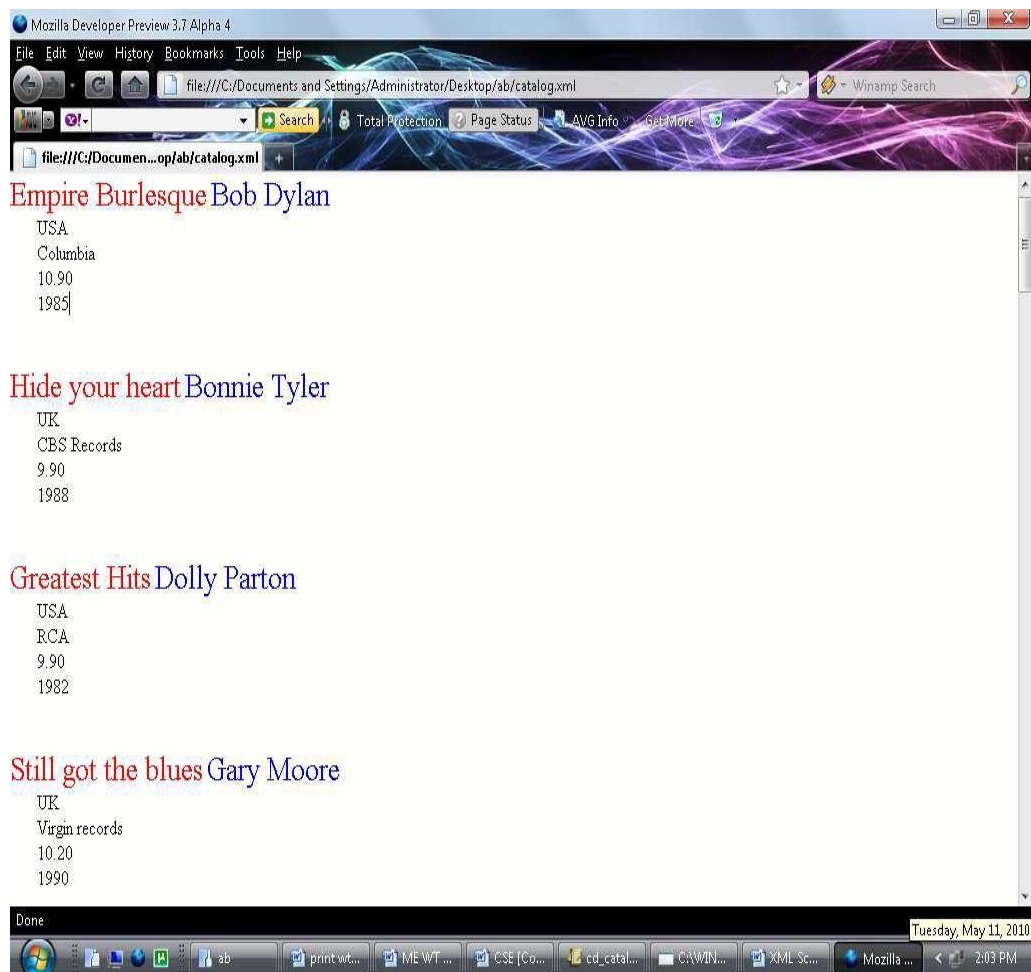
<YEAR>1987</YEAR>

```
</CD>
</CATALOG>
```

cd_catalog.css

```
<style>
CATALOG
{
background-color: #ffffff;
width: 100%;
}
CD
{
display: block;
margin-bottom: 30pt;
margin-left: 0;
}
TITLE
{
color: #FF0000;
font-size: 20pt;
}
ARTIST
{
color: #0000FF;
font-size: 20pt;
}
COUNTRY,PRICE,YEAR,COMPANY
{
display: block;
color: #000000;
margin-left: 20pt;
}
</style>
```

OUTPUT:



RESULT:

Thus the XML program for creating cd catalog has been successfully completed and output is verified.

Ex. No. : 10 Creation of XML document styles with XSLT Style Sheet

Date :

AIM:

To write a XML program for creating a cd catalog and style it with XSLT Style Sheet.

ALGORITHM:

1. Create a .xml file with the following header,

```
<?xml version="1.0"?>
```
2. Include all the description of the CDs inside the <CATALOG> tag.
3. For each CD create a separate tag within the parent tag
4. Create an XSL Style Sheet ("**cdcatalog.xsl**") with a transformation **template**
5. Add an XSL Style Sheet reference to your XML document ("**cdcatalog.xml**")
6. Run the XML program.

cdcatalog.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
  <cd>
    <title>Greatest Hits</title>
    <artist>Dolly Parton</artist>
    <country>USA</country>
    <company>RCA</company>
    <price>9.90</price>
    <year>1982</year>
  </cd>
  <cd>
    <title>Still got the blues</title>
    <artist>Gary Moore</artist>
    <country>UK</country>
    <company>Virgin records</company>
    <price>10.20</price>
    <year>1990</year>
  </cd>
  <cd>
    <title>Eros</title>
    <artist>Eros Ramazzotti</artist>
    <country>EU</country>
    <company>BMG</company>
    <price>9.90</price>
    <year>1997</year>
  </cd>
</catalog>
```

cdcatalog.xsl

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th>Title</th>
      <th>Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="title"/></td>
        <xsl:choose>
          <xsl:when test="price > 10">
            <td bgcolor="#ff00ff">
              <xsl:value-of select="artist"/>
            </td>
          </xsl:when>
          <xsl:otherwise>
            <td><xsl:value-of select="artist"/></td>
          </xsl:otherwise>
        </xsl:choose>
      </tr>
    </xsl:for-each>
  </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Output:

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees

RESULT:

Thus the XML program for creating cd catalog styled with XSLT sheet has been successfully completed and output is verified.