**JSS ACADEMY OF TECHNICAL EDUCATION**

# WEB TECHNOLOGY LAB FILE (KCS 652)

|  |  |
| --- | --- |
| Name | Anunay Anand |
| Roll No. | 1809110037 |
| Batch-Section | A1-CS1 |

Department of Computer Science and Engineering JSS ACADEMY OF TECHNICAL EDUCATION



C-20/1, SECTOR-62, NOIDA

## VISION OF THE INSTITUTE

JSS Academy of Technical Education Noida aims to become an Institution of excellence in imparting quality Outcome Based Education that empowers the young generation with Knowledge, Skills,Research,Aptitude and Ethical values to solve Contemporary Challenging Problems.

## MISSION OF THE INSTITUTE



* Develop a platform for achieving globally acceptable level of intellectual acumenand technologicalcompetence.
* Create an inspiring ambience that raises the motivation level for conductingquality research.
* Provide an environment for acquiring ethical values and positiveattitude.

## VISION OF THE DEPARTMENT

To spark the imagination of the Computer Science Engineers with values, skills and creativity to solve the real world problems.

## MISSION OF THE DEPARTMENT

* To inculcate creative thinking and problem solving skills through effectiveteaching, learning andresearch.
* To empower professionals with core competency in the field of Computer Scienceand Engineering.
* To foster independent and life long learning with ethical and socialresponsibilities.

## PROGRAM EDUCATIONAL OUTCOMES (PEOs)

**PEO1**: To empower students with effective computational and problem solving skills.

**PEO2**: To enable students with core skills for employment and entrepreneurship.

**PEO3**: To imbibe students with ethical values and leadership qualities.

**PEO4:** To foster students with research oriented ability which helps them in analyzing and solving real life problems and motivate them for pursuing higher studies.

## PROGRAM OUTCOMES (POs)

### Engineering Graduates will be able to:

**PO1: Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2: Problem analysis**: Identify, formulate, review research literature, and analyze complexengineering problems reaching substantiatedconclusionsusingfirstprinciples of mathematics, natural sciences, and engineeringsciences.

**PO3: Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, andthecultural,societal, and environmentalconsiderations.

**PO4: Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide validconclusions.

**PO5: Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of thelimitations.

**PO6: The engineer and society**: Apply reasoning informedby the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineeringpractice.

**PO7: Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainabledevelopment.

**PO8: Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of theengineeringpractice.

**PO9: Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, andin multidisciplinary settings.

**PO10: Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clearinstructions.

**PO11: Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to



one’s own work, as a member and leader in a team, to manage projects and in multidisciplinaryenvironments.

**PO12: Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of

technologicalchange.

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

PSO1: An ability to apply foundation of Computer Science and Engineering, algorithmic principles and theory in designing and modeling computation based systems.

PSO2: The ability to demonstrate software development skills.

## COURSE OUTCOMES (COs)

**Bloom’s taxonomy**



**CO-PO-PSO MAPPING**

|  |  |  |
| --- | --- | --- |
| C310.1 | Design a static web page using HTML | K2, K3 |
| C310.2 | Develop Java programs for window/web-based applications. | K2, K3 |
| C310.3 | Design dynamic web pages using jabva script and XML | K3, K4 |
| C310.4 | Design dynamic web page using server side programming. | K3, K4 |
| C310.5 | Design server side applications using JDBC, ODBC and section-  tracking API. | K3, K4 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO5** | **PO 6** | **PO7** | **PO 8** | **PO9** | **PO1 0** | **PO1 1** | **PO1 2** | **PSO1** | **PSO2** |
| **C310.1** | **2** | **2** | **3** | **1** | **3** | **1** | **1** | **1** | **2** | **2** | **3** | **2** | **2** | **2** |
| **C310.2** | **2** | **2** | **2** | **1** | **3** | **1** | **1** | **1** | **2** | **2** | **3** | **2** | **2** | **2** |
| **C310.3** | **2** | **2** | **3** | **1** | **3** | **1** | **1** | **1** | **2** | **2** | **3** | **2** | **2** | **3** |
| **C310.4** | **2** | **2** | **3** | **1** | **3** | **1** | **1** | **1** | **2** | **2** | **3** | **2** | **2** | **3** |
| **C310.5** | **2** | **2** | **3** | **1** | **3** | **1** | **1** | **1** | **2** | **2** | **3** | **2** | **2** | **3** |



INDEX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Experiment No. | Experiment Name | Date of Conduction | Date of Submission | Faculty Signature |
| 1. | Write HTML/Java scripts to display  your CV in navigator | 27-4-2021 | 27-4-2021 |  |
| 2. | Write an HTML program to design an entry form of student  details | 4-5-2021 | 4-5-2021 |  |
| 3. | Write programs using Java script for Web  Page to display browsers information | 11-5-2021 | 11-5-2021 |  |
| 4. | Write a Java applet to display the Application Program screen i.e. calculator  and other. | 18-5-2021 | 18-5-2021 |  |
| 5. | Writing program in XML for creation of  DTD, which specifies set of rules | 25-5-2021 | 25-5-2021 |  |
| 6. | Program to illustrate JDBC connectivity and maintaining database by sending  queries. | 1-6-2021 | 1-6-2021 |  |
| 7. | Install APACHE TOMCAT web server . Access the static web pages for  login id, using this server | 8-6-2021 | 8-6-2021 |  |
| 8. | Create a Cookie and add four user id’s and  passwords to this Cookie. | 15-6-2021 | 15-6-2021 |  |
| 9. | Write a java program/servlet/JSP to connect database and extract data from the tables and display  them. | 22-6-2021 | 22-6-2021 |  |
| 10. | Write a JSP which insert the details of the 3 or 4 users who register with the web  site by using registration form. | 29-6-2021 | 29-6-2021 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11. | Design**D**a**e**n**p**d**artment of C**  implement a simple shopping cart  example with session tracking API. | 6**o**-**m**7**p**-**u**2**t**0**er**2**S**1**cience & Eng** | **i**6**n**-**e**7**e**-**ri**2**n**0**g**21 |  |
| 12. | MultiThreading in Java | 13-7-2021 | 13-7-2021 |  |
| 13. | Event Handling in java | 20-7-2021 | 13-7-2021 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**EXPERIMENT NO.-01**

Aim: Write HTML/Java scripts to display your CV in navigator.

Input:

<!DOCTYPE html>

<html>

<head><title>Bio-Data</title>

</head>

<body>

<h1>Anunay Anand </h1>

<h3> Block-I, Sector-22, Noida 201301<br>JSS Academy of Technical Education, Noida<br>Phone 989128931 <br> [anunayanand@gmail.com@gmail.com<](mailto:%20anunayanand@gmail.com@gmail.com%3c)/h3>

<hr/>

<br>

<h2><b><u>Education</u></b></h2>

<table border=5px>

<tr>

<th>S.No.</th>

<th>Institute</th>

<th>Class</th>

<th>Year of passing</th>

<th>Percentage or Grade </th>

</tr>

<tr>

<td>1</td>

<td>Dav School </td>

<td>10th</td>

<td>2015</td>

<td> A2 </td>

</tr>

<tr>

<td>2</td>

<td>Dav School </td>

<td>12th</td>

<td>2017</td>

<td>75</td>

</tr>

<tr>

<td>3</td>

<td>JSSATEN</td>

<td>B.Tech</td>

<td>2022</td>

<td>-</td>

</tr>

</table>

<hr>

<h2><b><u>Skills</u></b></h2>

<ol type="1">

<li>Web Design with HTML & CSS</li>

<li>C</li>

<li>C++</li>

<li>Python</li>

</ol>

<hr>

<h2><b><u>Experience</u></b></h2>

<ol type="1">

<li>Sparks Foundation Web Development Intern </li>

<li>Fresher</li>

<li>Did various internships</li>

</ol>

<hr>

<h2><b><u>Extracurricolars</u></b></h2>

<ol type="1">

<li>Singing </li>

<li>Football </li>

<li>Poems and Literature </li>

</ol>

<hr>

<h2><b><u>Interests</u></b></h2>

<ol type="1">

<li>Football </li>

<li>Singing</li>

<li>Programming</li>

<li>Stand Up Comedy </li>

<li>Technology</li>

<li>Movies</li>

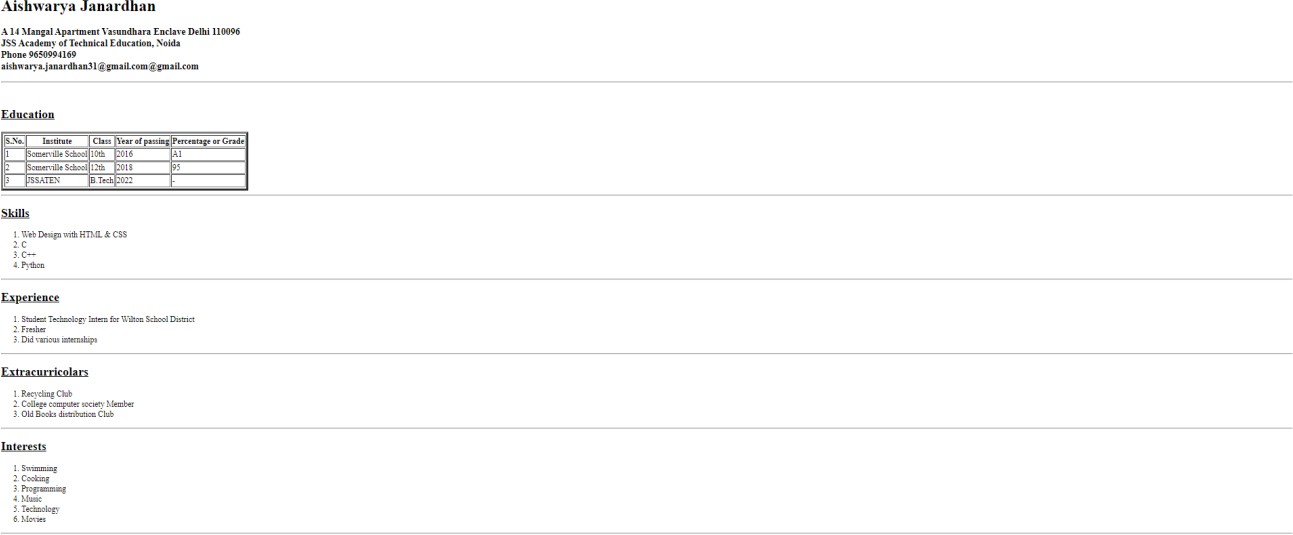
</ol>

<hr>

</body>

</html>

Output:



## EXPERIMENT NO.-02

Aim: Write an HTML program to design an entry form of student details.

Source Code:

<!DOCTYPE html>

<html>

<head>

<title>registartion form</title>

<style> table, th, td {

border: 1px solid black; border-collapse: collapse;

}

th, td { padding: 5px;

text-align: center;

</style>

</head>

<body style="background-color:#ffc7e3;">

<h1 style="text-align: center">REGISTRATION FORM</h1>

<form>

<strong>First name:</strong><br>

<input type="text" name="firstname">

<br>

<strong>Last name:</strong><br>

<input type="text" name="lastname">

<br>

<strong>Sex:</strong><br>

<input type="radio" name="male">Male<br>

<input type="radio" name="female">Female

<hr>

<strong>Username:</strong><br>

<input type="text" name="username">

<br>

<strong>Password:</strong><br>

<input type="Password" name="password">

<br>

<hr>

<table style="width:100%">

<caption style="font-size: 20px;"><strong>Educational Qualifications</strong></caption>

<tr>

<th>Course</th>

<th>Marks Obtained(%)or CGPA</th>

<th>Institute Name</th>

</tr>

<tr>

<td>10th</td>

<td>10</td>

<td>Somerville School </td>

</tr>

<tr>

<td>12th</td>

<td>94.6%</td>

<td>DAV Public School, Sahibabad</td>

</tr>

<tr>

<td>Graduation</td>

<td>Persuing</td>

<td>JSSATEN</td>

</tr>

</table>

<hr>

<strong>Address:</strong><br>

<textarea name="address" rows="10" cols="80"></textarea>

<br>

<hr>

<strong>Elective Subject:</strong><br>

<input type="radio" name="subject" value="Computer">Computer<br>

<input type="radio" name="subject" value="Arts">Arts<br>

<input type="radio" name="subject" value="Maths">Maths<br>

<hr>

<strong>Comment:</strong>

<textarea name="comment" rows="10" cols="40"></textarea>

<hr>

<button type="button" onclick="alert('Your form is submitted')" style="margin-left: 600px">Submit</button>

<input type="reset" name="reset">

<br>

<hr>

</form>

</body>

</html>

Output:



## Experiment-03

Aim: Write programs using Java script for Web Page to display browsers information.

Source Code:

<html>

<head>

<title>

Browser Information

</title>

</head>

<body>

<h1>Browser Information</h1>

<hr>

<p> The <b>navigator</b> object contains the following information about the browser you areusing. </p>

<ul>

<script LANGUAGE="JavaScript" type="text/javascript"> document.write("<li><b>Code Name:</b> " +navigator.appCodeName); document.write("<li><b>App Name:</b> " + navigator.appName); document.write("<li><b>AppVersion:</b> " + navigator.appVersion); document.write("<li><b>User Agent:</b> " + navigator.userAgent); document.write("<li><b>Language:</b> " + navigator.language); document.write("<li><b>Platform:</b> " + navigator.platform);

</script>

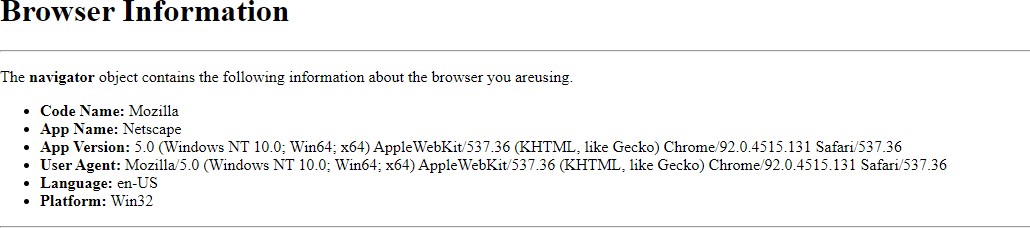
</ul>

<hr>

</body>

</html>

Output:



## Experiment-04

Aim: Write a Java applet to display the Application Program screen i.e. calculator and other. Source:

import java.awt.\*; import java.awt.event.\*; import java.applet.\*;

/\* <applet code="Calculator" width="700" height="200">

</applet>\*/

public class Calculator extends Applet implements ActionListener { String msg = "";

TextField t1, t2, t3; Button b1, b2, b3, b4; Label l1, l2, l3; public void init() {

l1 = new Label("First Number"); add(l1);

t1 = new TextField(15); add(t1);

l2 = new Label("Second Number"); add(l2);

t2 = new TextField(15); add(t2);

l3 = new Label("Result"); add(l3);

t3 = new TextField(15); add(t3);

b1 = new Button("ADD"); add(b1); b1.addActionListener(this);

b2 = new Button("SUB"); add(b2); b2.addActionListener(this); b3 = new Button("MULT"); add(b3); b3.addActionListener(this); b4 = new Button("DIV"); add(b4); b4.addActionListener(this);

}

public void actionPerformed(ActionEvent e) { if (e.getSource() == b1) {

int x = Integer.parseInt(t1.getText()); int y = Integer.parseInt(t2.getText()); int sum = x + y;

t3.setText(" " + sum);

}

if (e.getSource() == b2) {

int x = Integer.parseInt(t1.getText()); int y = Integer.parseInt(t2.getText()); int sub = x - y;

t3.setText(" " + sub);

}

if (e.getSource() == b3) {

int x = Integer.parseInt(t1.getText()); int y = Integer.parseInt(t2.getText()); int mul = x \* y;

t3.setText(" " + mul);

}

if (e.getSource() == b4) {

int x = Integer.parseInt(t1.getText()); int y = Integer.parseInt(t2.getText()); int div = x / y;

t3.setText(" " + div);

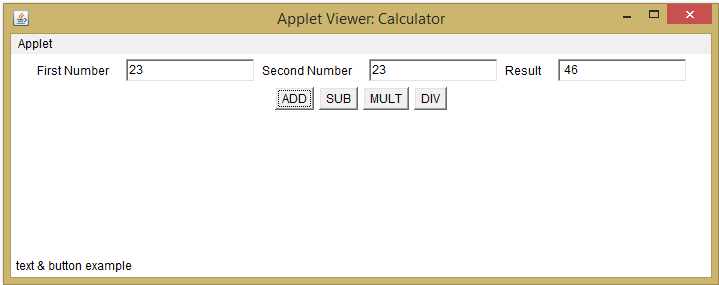
}

showStatus(" text & button example"); repaint();

}

}

Output:



## Experiment-05

Aim: Writing program in XML for creation of DTD, which specifies set of rules.

Source Code:

<?xml version="1.0"?>

<!DOCTYPE student[

<!ELEMENT student (name,address,std,marks)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT address (#PCDATA)>

<!ELEMENT std (#PCDATA)>

<!ELEMENT marks (#PCDATA)>]>

<student>

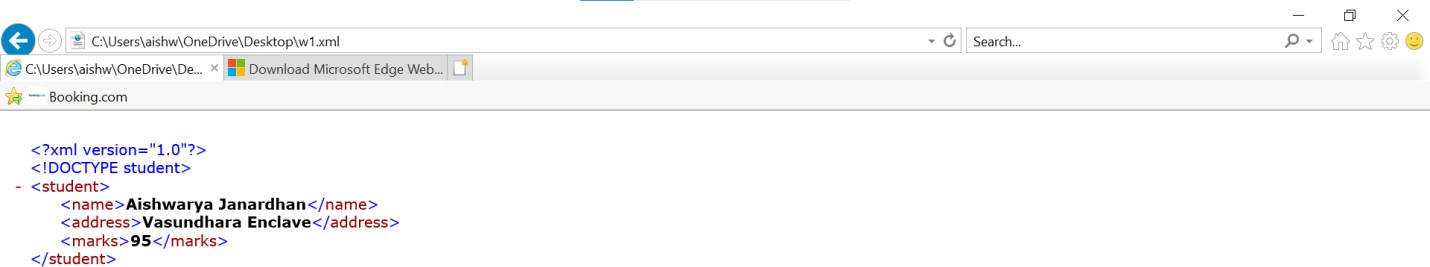
<name>Riddhi Gupta</name>

<address>Ghaziabad</address>

<marks>95</marks>

</student>

Output:



## Experiment-06

Aim: Program to illustrate JDBC connectivity and maintaining database by sending queries.

Source Code:

Import java.sql.\*; Import java.util.\*; class Main

{

public static void main(String a[])

{

//Creating the connection

String url = "jdbc:oracle:thin:@localhost:1521:xe"; String user = "system";

String pass = "12345";

//Entering the data

Scanner k = new Scanner(System.in); System.out.println("enter name"); String name = k.next(); System.out.println("enter roll no"); int roll = k.nextInt(); System.out.println("enter class"); String cls = k.next();

String sql = "insert into student1 values('"+name+"',"+roll+",'"+cls+"')"; Connection con=null;

try

{

DriverManager.registerDriver(new oracle.jdbc.OracleDriver());

//Reference to connection interface

con = DriverManager.getConnection(url,user,pass);

Statement st = con.createStatement(); int m = st.executeUpdate(sql);

if (m == 1)

System.out.println("inserted successfully : "+sql); else

System.out.println("insertion failed"); con.close();

}

catch(Exception ex)

{

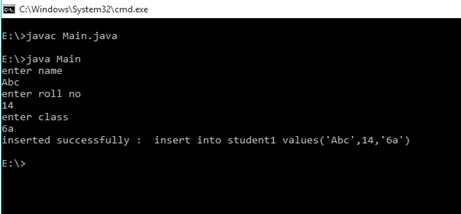
System.err.println(ex);

}

}

}

Output :



## Experiment-07

Aim: Install APACHE TOMCAT web server . Access the static web pages for login id, using this server.

Source Code:

Client-side HTML Login Validation Form:

File Name: SignIn.html

<body>

<h2 align="center">Sign in</h2>

<form method="get" action="http://localhost:8888/india/Login.jsp">

<h3>

Enter ID <input type="text" name="t1"><br>

Enter Password <input type="password" name="t2"><br>

<input type="submit" value="Sign in">

</h3>

</body>

Following is the server-side JSP file that includes a Scriptlet to understand what and how to code Scriptlet.

File Name: Login.jsp

<body>

<h2 align="center">Log in Validation </h2>

<%

String str1=request.getParameter("t1"); String str2=request.getParameter("t2");

if(str1.equalsIgnoreCase("snrao") && str2.equals("java"))

{

out.println("<h3>Your Login is Successful</h3>");

}

else

{

out.println("<h3>Sorry, your Login is Failed</h3>");

}

%>

</body>

Output:

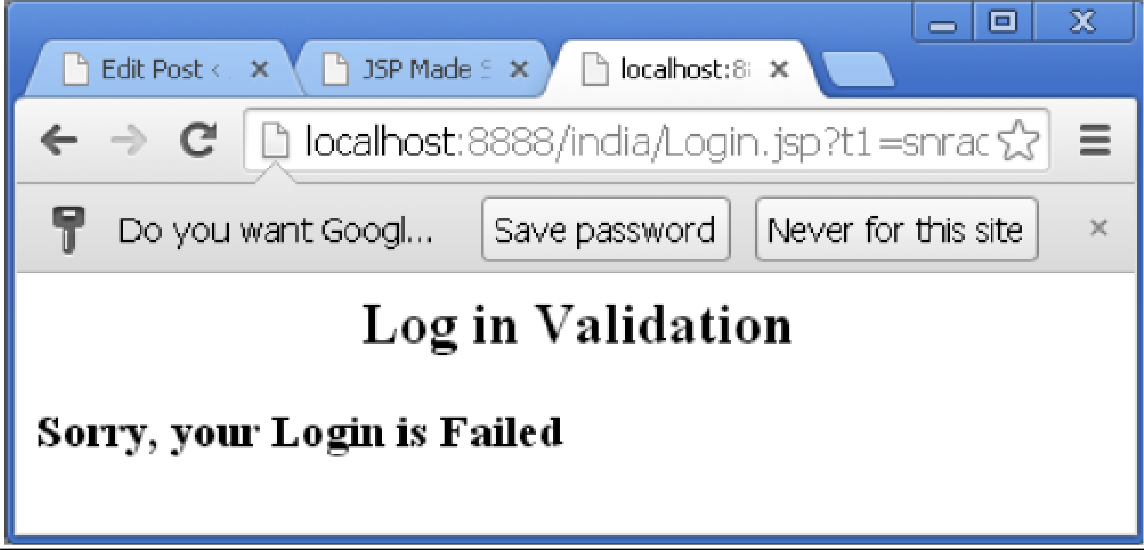
Screenshot of HTML file SignIn.html when values are entered.



Screenshot of response when correct values are entered.



Screenshot of response when wrong values are entered.



## Experiment-08

Aim: Create a Cookie and add four user id’s and passwords to this Cookie.

Source Code:

### Index.html

<form action="servlet1" method="post">

Name:<input type="text" name="userName"/><br/><input type="submit" value="go"/>

<input type="submit" value="go"/>

</form>

**FirstServlet.java** import java.io.\*; import javax.servlet.\*;

import javax.servlet.http.\*;

public class FirstServlet extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response){ try{

response.setContentType("text/html"); PrintWriter out = response.getWriter(); String n=request.getParameter("userName"); out.print("Welcome "+n);

Cookie ck=new Cookie("uname",n);//creating cookie object response.addCookie(ck);//adding cookie in the response

//creating submit button out.print("<form action='servlet2'>");

out.print("<input type='submit' value='go'>"); out.print("</form>");

out.close();

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

}

}

### SecondServlet.java

import java.io.\*;

import javax.servlet.\*; import javax.servlet.http.\*;

public class SecondServlet extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response){ try{

response.setContentType("text/html"); PrintWriter out = response.getWriter(); Cookie ck[]=request.getCookies(); out.print("Hello "+ck[0].getValue()); out.close();

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

}

<servlet>

<servlet-name>s1</servlet-name>

<servlet-**class**>FirstServlet</servlet-**class**>

</servlet>

<servlet-mapping>

<servlet-name>s1</servlet-name>

<url-pattern>/servlet1</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>s2</servlet-name>

<servlet-**class**>SecondServlet</servlet-**class**>

</servlet>

<servlet-mapping>

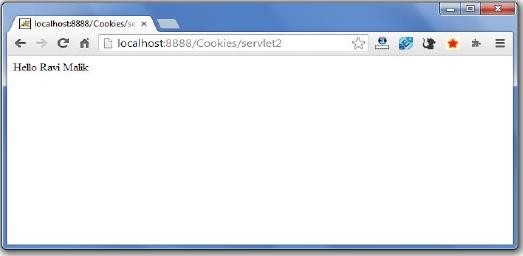
<servlet-name>s2</servlet-name>

<url-pattern>/servlet2</url-pattern>

</servlet-mapping>

</web-app>

Output:



## Experiment-09

Aim: Write a java program/servlet/JSP to connect database and extract data from the tables and display them.

Source Code:

Retrieve.jsp

%@page import="java.sql.DriverManager"%

%@page import="java.sql.ResultSet"%

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.Connection"%>

<% String id = request.getParameter("userid"); String driver = "com.mysql.jdbc.Driver";

String connectionUrl = "jdbc:mysql://localhost:3306/"; String database = "test";

String userid = "root"; String password = ""; try { Class.forName(driver);

}

catch (ClassNotFoundException e)

{

e.printStackTrace();

}

Connection connection = null; Statement statement = null; ResultSet resultSet = null; %>

<!DOCTYPE html>

<html>

<body>

<h1>Retrieve data from database in jsp</h1>

<table border="1">

<tr>

<td>first name</td>

<td>last name</td>

<td>City name</td>

<td>Email</td>

</tr>

<% try{ connection = DriverManager.getConnection(connectionUrl+database, userid, password); statement=connection.createStatement();

String sql ="select \* from users"; resultSet =statement.executeQuery(sql); while(resultSet.next())

{

%>

<tr>

<td><%=resultSet.getString("first\_name") %></td>

<td><%=resultSet.getString("last\_name") %></td>

<td><%=resultSet.getString("city\_name") %></td>

<td><%=resultSet.getString("email") %></td>

</tr>

<%

}

connection.close();

} catch (Exception e) {

e.printStackTrace();

}

%>

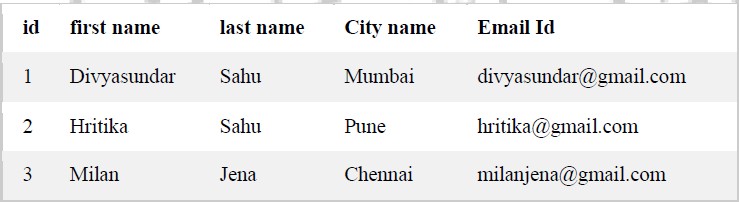
</table>

</body>

</html>

Output:

After retrieve the data from the data base the table look like this.



## Experiment-10

Aim: Write a JSP which insert the details of the 3 or 4 users who register with the web site by using registration form.

Source Code:

### Register\_1.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" ["http://www.w3.org/TR](http://www.w3.org/TR/html4/loose.dtd)/[html4/loose.dtd">](http://www.w3.org/TR/html4/loose.dtd)

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Guru Registration Form</title>

</head>

<body>

<h1>Guru Register Form</h1>

<form action="guru\_register" method="post">

<table style="with: 50%">

<tr>

<td>First Name</td>

<td><input type="text" name="first\_name" /></td>

</tr>

<tr>

<td>Last Name</td>

<td><input type="text" name="last\_name" /></td>

</tr>

<tr>

<td>UserName</td>

<td><input type="text" name="username" /></td>

</tr>

<tr>

<td>Password</td>

<td><input type="password" name="password" /></td>

</tr>

<tr>

<td>Address</td>

<td><input type="text" name="address" /></td>

</tr>

<tr>

<td>Contact No</td>

<td><input type="text" name="contact" /></td>

</tr></table>

<input type="submit" value="Submit" /></form>

</body>

</html>

### Guru\_register.java

package demotest;

import java.io.IOException;

import javax.servlet.RequestDispatcher; import javax.servlet.ServletException; import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class guru\_register

\*/

public class guru\_register extends HttpServlet { private static final long serialVersionUID = 1L;

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

// TODO Auto-generated method stub

String first\_name = request.getParameter("first\_name"); String last\_name = request.getParameter("last\_name"); String username = request.getParameter("username"); String password = request.getParameter("password"); String address = request.getParameter("address"); String contact = request.getParameter("contact");

if(first\_name.isEmpty() || last\_name.isEmpty() || username.isEmpty() || password.isEmpty() || address.isEmpty() || contact.isEmpty())

{

RequestDispatcher req = request.getRequestDispatcher("register\_1.jsp"); req.include(request, response);

}

else

{ RequestDispatcher req =request.getRequestDispatcher("register\_2.jsp");

req.forward(request, response);

}

}

}

### Register\_2.jsp

<html>

<head>

<title>Success Page</title>

</head>

<body>

<h5> --DataFlair-- </h5>

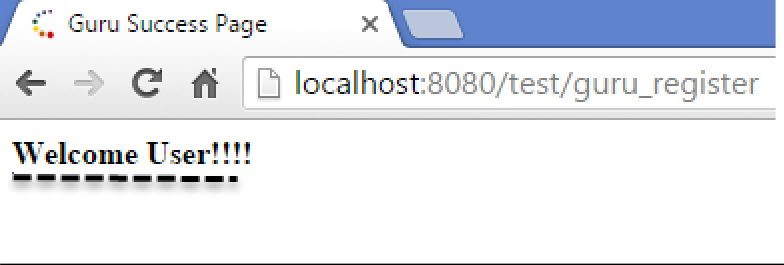
<a><b>Welcome User..</b></a>

</body>

</html>

Output:





## EXPERIMENT NO.-11

Aim**:** Design and implement a simple shopping cart example with session tracking API.

Source Code:

**INPUT:**

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

import javax.servlet.http.\*;

public class ShoppingCartViewerSession extends HttpServlet { public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException { res.setContentType("text/html");

PrintWriter out = res.getWriter();

// Get the current session object, create one if necessary. HttpSession session = req.getSession(true);

// Cart items are maintained in the session object. String[] items = (String[])session.getValue("cart.items");

System.out.println("<HTML><HEAD><TITLE>SessionTracker</TITLE></HEAD>"); System.out.println("<BODY><H1>Session Tracking Demo</H1>");

// Print the current cart items.

System.out.println("You currently have the following items in your cart:<BR>"); if (items == null) {

System.out.println("<B>None</B>");

}

else { System.out.println("<UL>");

for (int i = 0; i <items.length; i++) { System.out.println("<LI>" + items[i]);

}

System.out.println("</UL>");

}

// Ask if they want to add more items or check out.

System.out.println("<FORM ACTION=\"/servlet/ShoppingCart\" METHOD=POST>"); System.out.println("Would you like to<BR>");

System.out.println("<INPUT TYPE=submit VALUE=\" Add More Items \">"); System.out.println("<INPUT TYPE=submit VALUE=\" Check Out \">"); System.out.println("</FORM>");

// Offer a help page. Encode it as necessary. System.out.println("For help, click <A HREF=\"" + res.encodeUrl("/servlet/Help?topic=ShoppingCartViewer") + "\">here</A>");

System.out.println("</BODY></HTML>");

}

}

Output:

SessionTracker

Session Tracking Demo

currently have the following items in your cart: APPLE

BANANA MANGO

Would you like to Add More Items Check Out

For help here

## EXPERIMENT NO.-12

Aim: Write a program of MutiThreading in Java.

Source Code:

class MultithreadingDemo extends Thread

{

public void run()

{

try

{

// Displaying the thread that is running

System.out.println ("Thread " + Thread.currentThread().getId() + " is running");

}

catch (Exception e)

{

// Throwing an exception System.out.println ("Exception is caught");

}

}

}

// Main Class

public class Multithread

{

public static void main(String[] args)

{

int n = 8; // Number of threads for (int i=0; i<8; i++)

{

MultithreadingDemo object = new MultithreadingDemo(); object.start();

}

}

}

Output :

Thread 8 is running Thread 9 is running Thread 10 is running Thread 11 is running Thread 12 is running Thread 13 is running Thread 14 is running Thread 15 is running

Thread creation by implementing the Runnable Interface

We create a new class which implements java.lang.Runnable interface and override run() method. Then we instantiate a Thread object and call start() method on this object.

// Java code for thread creation by implementing

// the Runnable Interface

class MultithreadingDemo implements Runnable

{

public void run()

{

try

{

// Displaying the thread that is running

System.out.println ("Thread " + Thread.currentThread().getId() + " is running");

}

catch (Exception e)

{

// Throwing an exception System.out.println ("Exception is caught");

}

}

}

// Main Class class Multithread

{

public static void main(String[] args)

{

int n = 8; // Number of threads for (int i=0; i<8; i++)

{

Thread object = new Thread(new MultithreadingDemo()); object.start();

}

}

}

Output :

Thread 8 is running Thread 9 is running Thread 10 is running Thread 11 is running Thread 12 is running Thread 13 is running Thread 14 is running Thread 15 is running

### EXPERIMENT NO.-13

Aim: Write a program on Event Handling in Java Source code

import java.awt.\*; import java.awt.event.\*;

class AEvent extends Frame implements ActionListener{ TextField tf;

AEvent(){

//create components tf=new TextField();

tf.setBounds(60,50,170,20); Button b=new Button("click me"); b.setBounds(100,120,80,30);

//register listener b.addActionListener(this);//passing current instance

//add components and set size, layout and visibility add(b);add(tf);

setSize(300,300); setLayout(null); setVisible(true);

}

public void actionPerformed(ActionEvent e){ tf.setText("Welcome");

}

public static void main(String args[]){ new AEvent();

}

}

public void setBounds(int xaxis, int yaxis, int width, int height); have been used in the above example that sets the position of the component it may be button, textfield etc

Output:

