INDERPRASTHA ENGINEERING COLLEGE

GHAZIABAD

Department of Information Technology Web Technology Lab (RIT-601) (2019-20)

Name : Natasha Sharma

Roll Number : 1703013043

Course : B.Tech. (I.T.)

Year : 3

Semester : 6

Section : A

INDEX

S.No	TITLE	Date	Pageno.	Remark
1	Write HTML/Java scripts to display your CV in navigator, your Institute website, Department website and tutorial website for specific subject.	24.01.20	4-7	
2	A) Create a web page using HTML having frames B) Create a web page using HTML having Iframe.	31.01.20	8-13 14-15	
3	Design HTML form for keeping student record and validate it using Java script	07.02.20	16-19	
4	Write programs using Java script for Web Page to display browsers information.	28.02.20	20-21	
5	Write a Java applet to display the Application Program screen i.e. calculator and other.	13.03.20	22-25	
6	Writing program in XML for creation of DTD, which specifies set of rules. Create a style sheet in CSS/ XSL & display the document in internet explorer.	13.03.20	26-27	
7	Program to illustrate JDBC connectivity. Install a database (Mysql or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number Write a java program to connect to that database and extract data from the tables	20.03.20	28-29	
8	Install TOMCAT web server and APACHE. Access the developed static web pages	20.03.20	30-32	

	for books web site, using these servers by putting the web pages developed.			
9	Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a servlet for doing the following. Create a Cookie and add these four user id's and passwords to this Cookie. 2. Read the user id and passwords entered in the Login form and authenticate with the values available in the cookies.	27.03.20	33-35	
10	Install a database (Mysql or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page.	27.03.20	36-41	
11	Write a JSP which insert the details of the 3 or 4 users who register with the web site by using registration form. Authenticate the user when he submits the login form using the user name and password from the database.	14.03.20	42-44	
12	Design and implement a simple shopping cart example with session tracking API.	14.04.20	45-47	

Practical No:1

Objective:Write HTML/Javascripts to display your CV in navigator and link your Institute website, Department website and Tutorial website for specific subject(HTML).

Theory:

HTML stands for *Hypertext Markup Language*, is the common *markup language* for creating or describing a web pages. It is not a programming language,it is a markup language.

A markup language is a language for communicating to a web browser how the contents of a page will be displayed. In HTML this extra information is communicated to the browser in the form of codes or "tags".
HTML is written in the form of "tags" that are surrounded by angle brackets like start tag <html> and end tag </html>.
An HTML file have an .htm or .html file extension as like index.html, about_us.htm, which can identify that the page is a web page or HTML Documents.

HTML Tags

Tags is the important part of Hypertext Markup language. HTML Tags give instruction a browser how to display the page content. In other words, HTML Tag is one kind of command, how and what type content will display in the browser window.

Source Code:

```
<!Doctype html>
<html>
<head><h1>
<center><u><b>RESUME</u></b></center></h1></head >
<style>
table, th, td {
 border: 1px solid black;
 border-collapse: collapse;
}
</style>
</head>
<body><p>
<img src="abc.jpg" height="190" width="180" align="right" hspace="120"></p>>
<p align="middle">
<h3><b>NAME:Natasha Sharma</b></h3>
<h3><b>ADDRESS:Patparganj,Delhi</b></h3>
<h3><b>EMAIL:natashasharma98.23@gmail.com</b></h3>
<h3><b>PHONE:8595549312</b></h3> </p>
<p><hr></p>
```

```
<p><h2>Career Objective:</h2> Seeking an engineering position that will incorporate my undergraduate engineering coursework as well as my experience as an administrator.</p>
<h2>Educational Qualification:</h2> <table style="width:100%">
<tr>
<th>Course/Degree</th>
<th>Institute/College</th>
<th>Board/University</th>
<th>CGPA</th>
</tr>
<tr> <td>10th</td>
<td>Bal Bhavan Public School</td>
<td>CBSE</td>
<td>9.5</td>
</tr>
 <tr>
 <td>12th</td>
<td>Bal Bhavan Public School</td>
<td>CBSE</td> <td>85%</td>
</tr>
<tr>
 <td>B.tech</td>
<td>Inderprastha Engineering College</td>
<td>AKTU</td> <td>76%</td>
 </tr>
 </table>
 <h2>Technical Skills:</h2>
 <ul>
  <li>Language Skills:</li>
  <ol>
  <li>C</li>
  <li>C++</li>
  </ol>
  <li><h2>Web Technology:</h2></li>
  <ol>
  <li>HTML</li>
  <li>CSS</li>
  </ol>
  <li><h2>Database:</h2></li>
  <ol>
  <li>DBMS</li>
  <li>Oracle</li>
  <li>MySql</li>
  </ol>
 </ul>
<h2>Academic Project:</h2>
 <ul>
 <li>Android with core java</li>
<li>Web development</li>
<li>Networking</li>
```

```
<li>Naayi Raah</li>
</ul>
<h2>Internship:</h2>
<li>Artificial Intelligence</li>
<h2>Personal Details:</h2>
<p>NAME: Natasha Sharma</br>
ADDRESS: Patparganj,Delhi</br>
EMAIL: natashasharma98.23@gmail.com</br>
PHONE: 8595549312</br></p>
<a href="http://ipeclive.ipec.org.in/">Institute Website</a><br> <a
href="https://www.w3schools.com/html/">Tutorial Website</a> <br>
<a href="https://www.ipec.org.in/academics/facultylist/it/"> Department website</a>
<p>
<h2><pre>Date:                                    Signature:</pre></h2>
</p>
</body>
</html>
```

OUTPUT

RESUME

>

NAME:Natasha Sharma

ADDRESS:Paparganj,Delhi

EMAIL:natashasharma98@gmail.com

PHONE:8800450940

Career Objective:

Seeking an engineering position that will incorporate my undergraduate engineering coursework as well as my experience as an administrator.

Educational Qualification:

Course/Degree	Institute/College	Board/University	CGPA
10th	Bal Bhavan Public School	CBSE	8.7
12th	Bal Bhavan Public School	CBSE	76.8%
B.tech	Inderprastha Engineering College	AKTU	75%

Technical Skills:

Language Skills:

1. C
2. C++

Web Technology:

1. HTML
2. CSS

Database:

1. DBMS
2. Oracle
3. MySql

Academic Project:

- Android with core java
- Web development
- Networking
- Naayi Raah

Internship:

- Artificial Intelligence

Personal Details:

NAME: Natasha Sharma
ADDRESS: Paparganj,Delhi
EMAIL: natashasharma98.23@gmail.com
PHONE: 8800450940

Institute Website
Tutorial Website
Department website

Date: Signature:

PRACTICAL NO:2 (A)

Objective: Write HTML code to create the following frames.

Theory:

A Framed document divides a browser window into multiple smaller window frames.
Each frame can contain different documents with frame we can define more than one web pages in the same browser window.
The <frame> tag defines one particular window within a <frameset>.
Each <frame> in a <frameset> can have different attributes, such as border, scrolling, ability to resize etc.

Source Code:

Main Page:

```
<html>
<frameset rows="20%,*">
<frame src="frame1.html" name="f1" noresize>
<frameset cols="20%,40%,40%">
<frame src="frame2.html" name="f2" noresize>
<frame src="frame3.html" name="f3" noresize>
<frame src="frame4.html" name="f4" noresize>
</frameset>
</frameset>
</html>
```

Frame1.html

```
<html>
<h2 align="middle">NATASHA SHARMA</h2>
<h2 align="middle">Welcome to our website</h2>
</html>
```

Frame2.html

```
<html>
<a href="obj.html" target="f3">Objective</a><br>
<br>
```

```
<a href="per.html" target="f3">Personal information</a><br>
<br>
<a href="exp.html" target="f3">Experience</a><br>
<br>
<a href="edu.html" target="f3">Educational information</a><br>
<br>
<a href="ach.html" target="f3">Achivements</a><br>
</html>
```

obj.html

```
<html>
<body>
<h3>OBJECTIVE</h3>
<p>A passionate under graduate of Btech(information Technology) having a dedication to learn new skills and to retail a position in futur that uses my skills and give me opportunity to perform.
</p>
</body>
</html>
```

per.html

```
<html>
<h3>PERSONAL INFORMATION</h3>
<p>NAME: NATASHA SHARMA</br>
ADDRESS: Patparganj,Delhi</br>
EMAIL: natashasharma98.23@gmail.com</br>
PHONE: 8595549312</br></p>
</html>
```

exp.html

```
<html>
<h3>EXPERIENCE</h3>
<p>I learned alot from my Btech .It helps me in learning different languages, data handling, algorithms etc. My experience in my projects is good as my teachers helped me alot in my projects.</p>
</html>
```

edu.html

```
<html>
<h3>EDUCATIONAL INFORMATION</h3>
<table style="width:100%" border="1">
<tr>
<th>Course/Degree</th>
<th>Institute/College</th>
<th>Board/University</th>
<th>CGPA</th>
```

```
</tr>
<tr> <td>10th</td>
<td>Bal Bhavan Public School</td>
<td>CBSE</td>
<td>9.5</td>
</tr>
 <tr>
 <td>12th</td>
<td>Bal Bhavan Public School</td>
<td>CBSE</td> <td>85%</td>
</tr>
<tr>
 <td>B.tech</td>
<td>Inderprastha Engineering College</td>
<td>AKTU</td> <td>76%</td>
 </tr>
 </table>
 </html>
```

ach.html

```
<html>
<h2>Academic Project:</h2>
 <ul>
 <li>Android with core java</li>
<li>Web development</li>
<li>Networking</li>
<li>Naayi Raah</li>
</ul>
```

Frame4.html

```
<html>
<img src="a.jpeg" height="100%" width="100%">
</html>
```

OUTPUT:

Natasha Sharma

Welcome to our website

Objective
Personal information
Experience
Educational information
Achivements

Natasha Sharma

Welcome to our website

Objective
Personal information
Experience
Educational information
Achivements

OBJECTIVE

A passionate under graduate of Btech(information Technology) having a dedication to learn new skills and to retail a position in futur that uses my skills and give me opportunity to perform.

Natasha Sharma

Welcome to our website

Objective Personal information Experience Educational information Achivements	**PERSONAL INFORMATION** NAME: Natasha Sharma ADDRESS: Gaziabad EMAIL: natashasharma98.23@gmail.com PHONE: 9876543573	

Natasha Sharma

Welcome to our website

Objective Personal information Experience Educational information Achivements	**EXPERIENCE** I am currently persuing bachelors degree in Information technology.	

Natasha Sharma

Welcome to our website

Objective

Personal information

Experience

Educational information

Achivements

EDUCATIONAL INFORMATION

Course/Degree	Institute/College	Board/University	CGPA
10th	Bal Bhavan Public School	CBSE	85%
12th	BBPS,Delhi	CBSE	78%
B.tech	Inderprastha Engineering College	AKTU	75%

Natasha Sharma

Welcome to our website

Objective

Personal information

Experience

Educational information

Achivements

Academic Project:

- Android with core java
- Web development
- Networking

PRACTICAL NO:2 (B)

Objective: Write HTML code to create the floating frame.

Theory:

The floating frame is used to create an inline framed region or window that acts similarly to any other embedded object insofar as text can be flowed around it.With in the<body> of an HTML element an inline frame is defined by the iframe element.To create an<iframe> tag the major attributes required is src,height and width.

Source Code:

```
<html>
<body>
<h2>Iframe - Target for a Link</h2>
<iframe height="200px" width="50%" src="demo_iframe.html"
name="iframe_a"></iframe>
<p><a href="https://www.ipec.org.in/wp/" target="iframe_a">Inderprastha Engineering
College,Ghazibad</a></p>
<p>When the target of a link matches the name of an iframe, the link will open in the
iframe.</p>
</body>
</html>
```

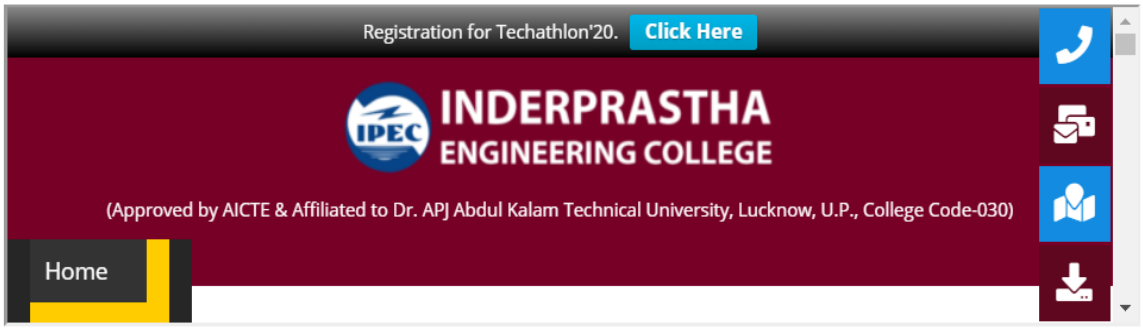
Output:

Iframe - Target for a Link

Inderprastha Engineering College,Ghazibad

When the target of a link matches the name of an iframe, the link will open in the iframe.

Iframe - Target for a Link



Inderprastha Engineering College,Ghazibad

When the target of a link matches the name of an iframe, the link will open in the iframe.

PRACTICAL NO: 3

Objective: Design HTML form for keeping student record and validate it using javascript.

Theory:

The <form> tag in HTML is used to create form for user input. There are many elements which are used within form tag. For example: <input>, <textarea>, <button>, <select>, <option>, <optgroup>, <fieldset>, <label>.

Source Code:

Main page:

```
<html>
<body>
<form action="f1.html" target="_top" method="get""post"">
<center><fieldset>
<legend> STUDENT REGISTRATION FORM </legend>
USERNAME: <input type="text" name=" " id="name"/><br>
PASSWORD: <input type="password" name="pass" minlength="10" required/><br>
ADDRESS: <input type="text" placeholder="address" size="20%"/><br>
EMAIL: <input type="email" name="email" placeholder="email"/><br>
PHONE: <input type="number" maxlength="10" placeholder="phone" name="num"/><br>
DATE OF BIRTH: <input type="date" placeholder="dd/mm/yyyy" name="num"/><br>
GENDER: <input type="radio" value="male" name="gender" checked/>male <input
type="radio" value="female" name="gender" />female <br>
CATEGORY: <input type="checkbox" value="sc" name="category" checked/> SC <input
type="checkbox" value="st" name="category"/> ST <input type="checkbox" value="obc"
name="category"/>OBC  <input type="checkbox" value="general"
name="category"/>GENERAL<br>
SELECT YOUR COURSE: <select name="subject" multiple size=2>
<option value="web technology"> web technology </option>
<option value=" computer network"> computer network </option>
<option value="java"> java </option>
</select><br>
<input type="button" onclick="submit it " value="submit"/>
<input type="button" onclick="submit it " value="reset"/>
</center>
</fieldset>
</form>
</body>
</html>
```

F1.html:

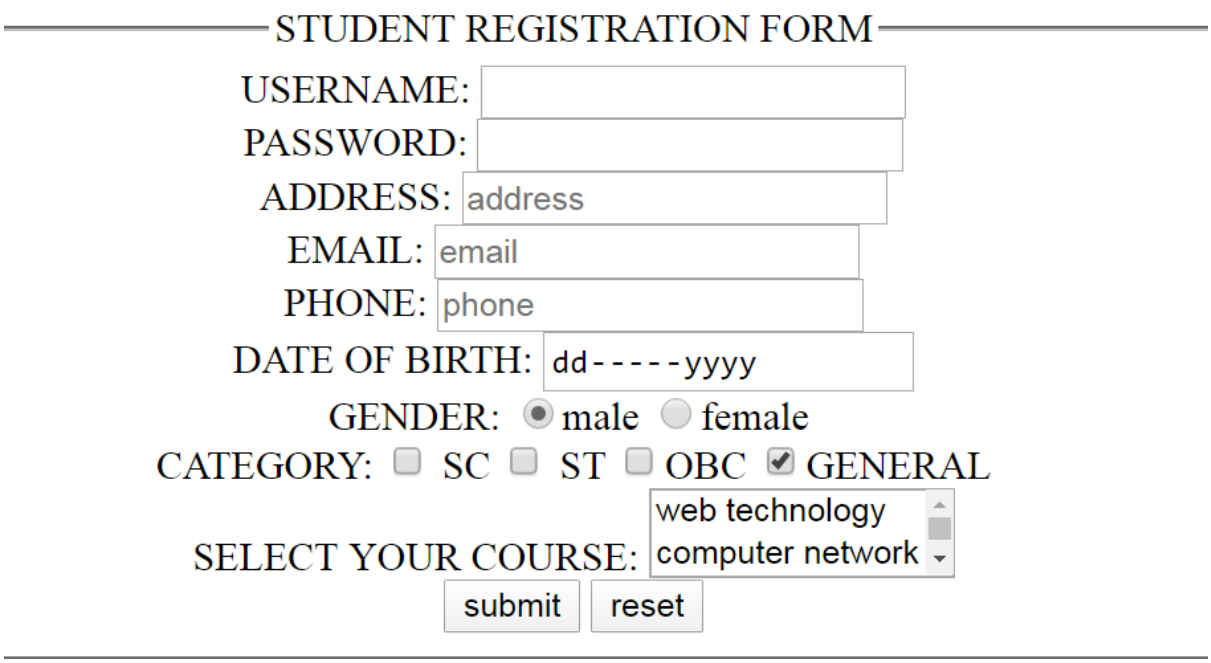
```
<html>
<body>
Sucessfully submitted the form.<br>
</body>
</html>
```

FORM VALIDATION:

```
<script>
function form()
{
   var username = document.forms["RegForm"]["Name"];
   var email = document.forms["RegForm"]["EMail"];
   var phone = document.forms["RegForm"]["Telephone"];
   var SelectYourCourse =  document.forms["RegForm"]["Subject"];
   var password = document.forms["RegForm"]["Password"];
   var address = document.forms["RegForm"]["Address"];
   var dateofbirth = document.forms["RegForm"]["DateOfBirth"];
   var gender = document.forms["RegForm"]["Gender"];
   var category = document.forms["RegForm"]["Category"];

   if (username.value == "")
   {
      window.alert("Please enter your name.");
      username.focus();
      return false;
   }

   if (address.value == "")
   {
      window.alert("Please enter your address.");
      address.focus();
      return false;
   }

   if (email.value == "")
   {
      window.alert("Please enter a valid e-mail address.");
      email.focus();
      return false;
   }

   if (phone.value == "")
   {
      window.alert("Please enter your telephone number.");
      phone.focus();
```

```
    return false;
  }

  if (password.value == "")
  {
    window.alert("Please enter your password");
    password.focus();
    return false;
  }

  if (dateofbirth.value ==””)
  {
    window.alert(“Please enter your date of birth.”);
    dateofbirth.focus();
    return false;
  }

  if (gender.selectedIndex <1)
  {
    alert(“Please enter your gender.”);
    gender.focus();
    return false;
  }

   if (category.selectedIndex <1)
   {
      alert(“please enter your category.”);
      category.focus();
      return false;
   }

  if (SelectYourCourse.selectedIndex < 1)
  {
    alert("Please enter your course.");
    SelectYourCourse.focus();
    return false;
  }

  return true;
}</script>
```

OUTPUT:


STUDENT REGISTRATION FORM
USERNAME:
PASSWORD:
ADDRESS: address
EMAIL: email
PHONE: phone
DATE OF BIRTH: dd-----yyyy
GENDER: male female
CATEGORY: SC ST OBC GENERAL
SELECT YOUR COURSE: web technology computer network
submit
reset

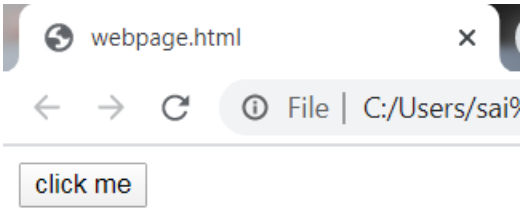
PRACTICAL NO : 4

Objective: Write a program using javascript for Web page to display browser.

Source Code:-

```
<html>
<head>
<script>
function show()
{
	document.write("Name "+navigator.appName+"<br>");
	document.write("Version "+navigator.appVersion+"<br>");
	document.write("CodeName "+navigator.appCodeName+"<br>");
	document.write("CookieEnabled "+navigator.cookieEnabled+"<br>");
	document.write("javaEnabled "+navigator.javaEnabled()+"<br>");
	document.write("MimeType "+navigator.mimeTypes+"<br>");
	document.write("platform "+navigator.platform+"<br>");
	document.write("Language "+navigator.language+"<br>");
	document.write("User agent "+navigator.userAgent+"<br>");
}
</script>
</head>

<form>
<input type="button" value="click me" onclick="show()">
</form>
</html>
```

OUTPUT:-



Name Netscape
Version 5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36
CodeName Mozilla
CookieEnabled true
javaEnabled false
MimeType [object MimeTypeArray]
platform Win32
Language en-US
User agent Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36

PRACTICAL NO : 5

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

Source Code :-

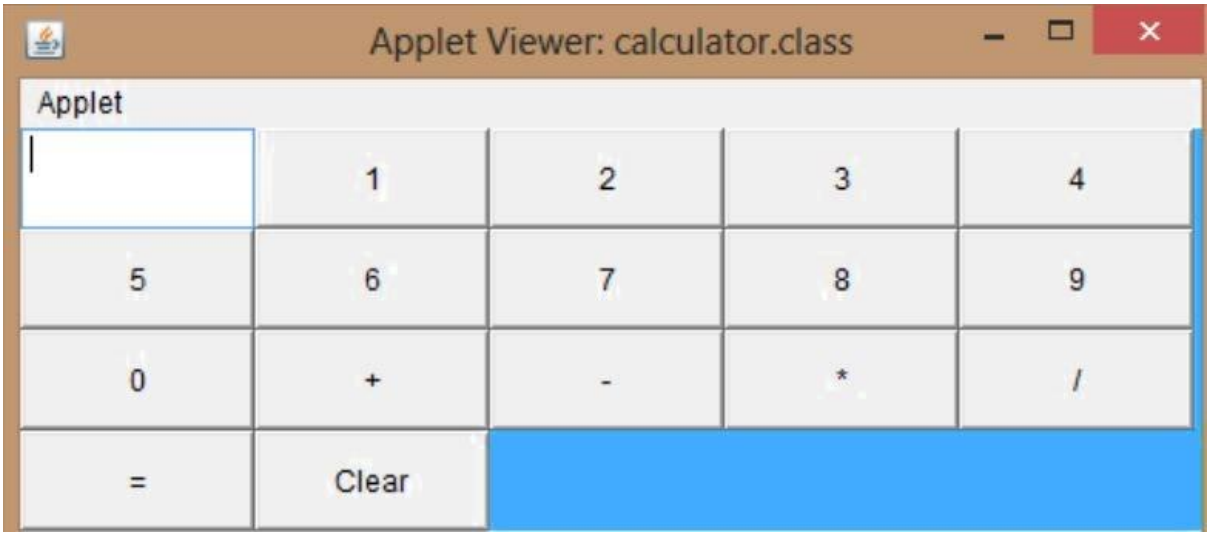
```
import java.applet.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.TextEvent;
import java.awt.event.TextListener;;
//<applet code="calculator" width=600 height=800>
//</applet>
public class calculator extends Applet implements ActionListener, TextListener
{
String s,s1,s2,s3,s4;
Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0;
Button add,sub,eq,cl,mul,div;
TextField t1;
int a,b,c;

public void init()
{
 t1=new TextField(10);
 b1=new Button("1");
 b2=new Button("2");
 b3=new Button("3");
 b4=new Button("4");
 b5=new Button("5");
 b6=new Button("6");
 b7=new Button("7");
 b8=new Button("8");
 b9=new Button("9");
 b0=new Button("0");
 add=new Button("+");
 sub=new Button("-");
 mul=new Button("*");
 div=new Button("/");
 eq=new Button("=");
 cl=new Button("Clear");
```

```
GridLayout gb=new GridLayout(4,5);
setLayout(gb);

add(t1);
add(b1);
add(b2);
add(b3);
add(b4);
add(b5);
add(b6);
add(b7);
add(b8);
add(b9);
add(b0);
add(add);
add(sub);
add(mul);
add(div);
add(eq);
add(cl);

b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
b6.addActionListener(this);
b7.addActionListener(this);
b8.addActionListener(this);
b9.addActionListener(this);
b0.addActionListener(this);
add.addActionListener(this);
sub.addActionListener(this);
mul.addActionListener(this);
div.addActionListener(this);
eq.addActionListener(this);
cl.addActionListener(this);
paint();
//t1.addTextListener(this);
}
public void paint()
{
```

```
 setBackground(Color.blue);
}

public void actionPerformed(ActionEvent e)
{
 s=e.getActionCommand();
 if(s.equals("0")||s.equals("1")||s.equals("2")||
s.equals("3")||s.equals("4")||s.equals("5")||s.equals("6")||s.equals("7")||s.equals("8")||
s.equals("9")||s.equals("0"))
 {
  s1=t1.getText()+s;
  t1.setText(s1);
 }
 if(s.equals("+"))
 {
   s2=t1.getText();
   t1.setText("");
   s3="+";
 }
 if(s.equals("-"))
 {
   s2=t1.getText();
   t1.setText("");
   s3="-";
 }
 if(s.equals("*"))
 {
   s2=t1.getText();
   t1.setText("");
   s3="*";
 }
 if(s.equals("*"))
 {
   s2=t1.getText();
   t1.setText("");
   s3="*";
 }
 if(s.equals("="))
 {
  s4=t1.getText();
  a=Integer.parseInt(s2);
  b=Integer.parseInt(s4);
  if(s3.equals("+"))
```

```
  c=a+b;
 if(s3.equals("-"))
  c=a-b;
 t1.setText(String.valueOf(c));
 }
 if(s.equals("Clear"))
 {
  t1.setText("");
 }
}
public void textValueChanged(TextEvent e)
{
}
}
```

Output:-



PRACTICAL NO : 6

Objective: Write a program in XML for creation of DTD, which specifies set of rules. Create a stylesheet in CSS/XSL and display the document in internet explorer.[Book information]

Source code:

book.DTD

```
<!ELEMENT books (heading,book*)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT book ((bookname|title),author+,publisher?,edition,price)>
<!ELEMENT bookname (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT edition (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

book.css

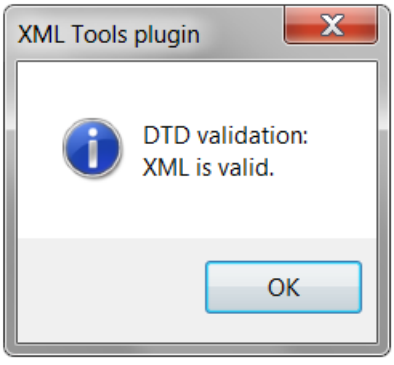
```
books {
	color: white;
	background-color:grey;
	width:100%;
}
heading {
	color: green;
	font-size: 40px;
	background-color: blue;
}
heading,title,author,publisher,edition,price{
	display: block;
}
title{
	font-size: 25px;
	font-weight: bold;
}
```

Book.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="book.css"?>
<!DOCTYPE books SYSTEM "book.dtd">
<books>
```

```
        <heading>xml with css</heading>
        <book>
                <title>Title: Web Technology</title>
                <author>Author: Gopal</author>
                <publisher>Publisher: PHI</publisher>
                <edition>Edition : 2nd</edition>
                <price>Price: Rs.200</price>
        </book>
        <book>
                <title>Title: J2REE</title>
                <author>Author: Santo</author>
                <publisher>Publisher: APL</publisher>
                <edition>Edition : 3rd</edition>
                <price>Price: Rs.500</price>
        </book>
</books>
```

OUTPUT:



PRACTICAL NO: 7

Objective:- Install a database (Mysql or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number Write a java program to connect to that database and extract data from the tables of "Web Technology Lab"

Source code:

Mysql Code:-

```
mysql> create database WebTechnologyLab;

Query OK, 1 row affected (0.00 sec)

mysql> use  WebTechnologyLab;
Database changed

mysql> create table WebTechnologyLab(name varchar(10), password varchar(10), email_id
varchar(20), phone int(10));

Query OK, 0 rows affected (0.01 sec)

mysql> insert into WebTechnologyLab Values('ram','ram','ram@9000gmail.com',98765678);

Query OK, 1 row affected (0.00 sec)

mysql> insert into WebTechnologyLab
values('rohan','rohan','rohan@9000gmail.com',854675678);

Query OK, 1 row affected (0.00 sec)

mysql> insert into WebTechnologyLab
values('abhi','abhi','abhi@9000gmail.com',896775678);

Query OK, 1 row affected (0.00 sec)

mysql> select * from WebTechnologyLab;
```

Name	Password	Email-Id	Phone
Ram	ram	ram@9000gmail.com	987656784
rohan	rohan	rohan@9000gmail.com	854675678
Abhi	abhi	abhi@9000gmail.com	896775678

3 rows in set (0.00 sec)

Java Code:-

```
import java.sql.*;

class MysqlCon{
```

```
public static void main(String args[]){
try{
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/WebTechnologyLab","root","tiger");
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from WebTechnologyLab");
while(rs.next())
System.out.println(rs.getString(1)+"  "+rs.getString(2)+"  "+rs.getString(3)+"  "+rs.getInt(4));
con.close();
}
catch(Exception e){ System.out.println(e);
}
}
}
```

Output:-

```
C:\Users\hp>javac MysqlCon.java

C:\Users\hp>java  MysqlCon.java
ram       ram       ram@9000gmail.com          987656784
rohan     rohan     rohan@9000@gmail.com       896775678
abhi      abhi      abhi@9000gmail.com          854675678

C:\Users\hp>
```

PRACTICAL NO: 8

OBJECTIVE : Install TOMCAT web server and APACHE. Access the above developed static web pages for books web site, using these servers by putting the web pages developed.

THEORY

A *web application* (or webapp), unlike standalone application, runs over the Internet. Examples of webapps are google, amazon, ebay, facebook and twitter.

A webapp is typically a *3-tier* (or *multi-tier*) *client-server database application* run over the Internet as illustrated in the diagram below. It comprises five components:

1. **HTTP Server**: E.g., Apache HTTP Server, Apache Tomcat Server, Microsoft Internet Information Server (IIS), nginx, Google Web Server (GWS), and others.
2. **HTTP Client (or Web Browser)**: E.g., Internet Explorer (MSIE), FireFox, Chrome, Safari, and others.
3. **Database**: E.g., Open-source MySQL, Apache Derby, mSQL, SQLite, PostgreSQL, OpenOffice's Base; Commercial Oracle, IBM DB2, SAP SyBase, MS SQL Server, MS Access; and others.
4. **Client-Side Programs**: could be written in HTML Form, JavaScript, VBScript, Flash, and others.
5. **Server-Side Programs**: could be written in Java Servlet/JSP, ASP, PHP, Perl, Python, CGI, and others.

SOURCE CODE

Steps to install TOMCAT Web Server

Step1: Create a directory to keep all your works.

Step2: Download and Install TOMCAT

For Windows:

Goto http://tomcat.apache.org

Tomcat's Directories

- **bin**: contains the binaries; and startup script (startup.bat for Windows and startup.sh for Unixes and Mac OS), shutdown script (shutdown.bat for Windows and shutdown.sh for Unix and Mac OS), and other binaries and scripts.
- **conf**: contains the system-wide configuration files, such as server.xml, web.xml, and context.xml.
- **webapps**: contains the webapps to be deployed. You can also place the WAR (Webapp Archive) file for deployment here.
- **lib**: contains the Tomcat's system-wide JAR files, accessible by all webapps. You could also place external JAR file (such as MySQL JDBC Driver) here.
- **logs**: contains Tomcat's log files. You may need to check for error messages here.

- **work**: Tomcat's working directory used by JSP, for JSP-to-Servlet conversion.

Step3: Create an Environment variable JAVA_HOME
Step4: Cofigure TOMCAT Server
The Tomcat configuration files are located in the "conf" sub-directory of your Tomcat installed directory, e.g. "c:\myWebProject\tomcat\conf" (for Windows). There are 4 configuration files.

server.xml
web.xml
context.xml
Step5: Start TOMCAT Server
Step6: Develop and Deploy Web Application.
Write a Welcome Page

```
<html>
 <head><title>My Home Page</title></head>
 <body>
  <h1>My Name is so and so. This is my HOME.</h1>
 </body>
</html>
```

Write HelloWorld.java Servlet

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

public class HelloServlet extends HttpServlet {
  @Override
  public void doGet(HttpServletRequest request, HttpServletResponse response)
      throws IOException, ServletException {

    // Set the response MIME type of the response message
    response.setContentType("text/html");
    // Allocate a output writer to write the response message into the network socket
    PrintWriter out = response.getWriter();

    // Write the response message, in an HTML page
    try {
      out.println("<html>");
      out.println("<head><title>Hello, World</title></head>");
      out.println("<body>");
      out.println("<h1>Hello, world!</h1>");  // says Hello
      // Echo client's request information
      out.println("<p>Request URI: " + request.getRequestURI() + "</p>");
      out.println("<p>Protocol: " + request.getProtocol() + "</p>");
      out.println("<p>PathInfo: " + request.getPathInfo() + "</p>");
```

```
      out.println("<p>Remote Address: " + request.getRemoteAddr() + "</p>");
      // Generate a random number upon each request
      out.println("<p>A Random Number: <strong>" + Math.random() +
"</strong></p>");
      out.println("</body></html>");
    } finally {
      out.close();  // Always close the output writer
    }
  }
}
```

Configure Servlet's Request URL in "webapps\hello\WEB-INF\web.xml"

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app version="3.0"
 xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">

  <!-- To save as "hello\WEB-INF\web.xml" -->

  <servlet>
    <servlet-name>HelloWorld</servlet-name>
    <servlet-class>HelloServlet</servlet-class>
  </servlet>

  <!-- Note: All <servlet> elements MUST be grouped together and
       placed IN FRONT of the <servlet-mapping> elements -->

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

OUTPUT

PRACTICAL NO: 9

OBJECTIVE: Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a servlet for doing the following. Create a Cookie and add these four user id's and passwords to this Cookie. 2. Read the user id and passwords entered in the Login form and authenticate with the values available in the cookies.

THEORY

Servlet:

Servlets are the Java programs that runs on the Java-enabled web server or application server. They are used to handle the request obtained from the web server, process the request, produce the response, then send response back to the web server. Properties of **Servlets** : **Servlets** work on the server-side. A **cookie** is a small piece of information that is persisted between the multiple client requests. A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.

SOURCE CODE

Index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<h1>Simple Cookie example</h1>

<form action="CookieController" method="post">
      Username: <input type="text" name="uname"> <br>
  Password: <input type="password" name="pass"> <br>
<input type="submit" value="Login">
</form>
</body>
</html>
```

Servlet CookieController to create cookie

```
package com.candidjava;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class CookieController
 */
@WebServlet("/CookieController")
public class CookieController extends HttpServlet {

  protected void doPost(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    String un = request.getParameter("uname");
    String pw = request.getParameter("pass");

    Cookie ck = new Cookie("mycookie", un);
    response.addCookie(ck);

    response.sendRedirect("home.jsp");

  }
}
```

Display list of Cookie in browser

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1>Retrieving Cookie from browser</h1>
<br>
<%
```

```
  Cookie[] cks=request.getCookies();
  for(Cookie ck:cks)
  {
    String cn=ck.getName();
    String cv=ck.getValue();

    %>
    Cookie name : <b><%=cn %> </b><br>
    Cookie Value : <b><%=cv %> </b><br>
    <%
  }
%>
</body>
</html>
```

OUTPUT:-

PRACTICAL NO: 10

OBJECTIVE: Install a database (Mysql or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page.

THEORY:

Software required to be installed in our system are Oracle, java and Tomcat server.

To start with interfacing Java Servlet Program with JDBC Connection:

1. Proper JDBC Environment should set-up along with database creation.
2. To do so, download the mysql-connector.jar file from the internet,
3. As it is downloaded, move the jar file to the apache-tomcat server folder,
4. Place the file in **lib** folder present in the apache-tomcat directory.
5. **To start with the basic concept of interfacing:**

Step 1: Creation of Database and Table in MySQL

As soon as jar file is placed in the folder, create a database and table in MySQL,

Step 2: Implementation of required Web-pages

Create a form in HTML file, where take all the inputs required to insert data into the database. Specify the servlet name in it, with the POST method as security is important aspects in database connectivity.

Step 3: Creation of Java Servlet program with JDBC Connection

To create a JDBC Connection steps are

1. Import all the packages
2. Register the JDBC Driver

3. Open a connection
4. Execute the query, and retrieve the result
5. Clean up the JDBC Environment

Create a separate class to create a connection of database, as it is a lame process to writing the same code snippet in all the program. Create a .java file which returns a Connection object.

Step 4: To use this class method, create an object in Java Servlet program

Below program shows Servlet Class which create a connection and insert the data in the **demo** table,

Step 5: Get the data from the HTML file

To get the data from the HTML file, the request object is used which calls getParameter() Method to fetch the data from the channel. After successful insertion, the writer object is created to display a success message.

After insertion operation from Servlet, data will be reflected in MySQL Database

SOURCE CODE:

SQL CREATE Statement

Create table personalvalues(name varchar2(20), password varchar2(10), age number, address varchar2(2030), email varchar2(30), phone number(10));

Registration.html

```
<html>
<head>
<title>Registration page</title>
</head>
<body bgcolor="#00FFFf">
<form METHOD="POST" ACTION="register">
<CENTER>
<table>
<center>
<tr> <td> Username </td>
<td><input type="text" name="usr"> </td> </tr>
<tr><td> Password </td>
<td><input type="password" name="pwd"> </td> </tr>
<tr><td>Age</td>
<td><input type="text" name="age"> </td> </tr>
<tr> <td>Address</td>
<td> <input type="text" name="add"> </td> </tr>
```

```
<tr> <td>email</td>
<td> <input type="text" name="mail"> </td> </tr>
<tr> <td>Phone</td>
<td> <input type="text" name="phone"> </td> </tr>
<tr> <td colspan=2 align=center> <input type="submit" value="submit"> </td>
</tr>
</center>
</table>
</form>
</body>
```

Login.html

```
<html>
<head>
<title>Registration page</title>
</head>
<body bgcolor=pink> <center> <table>
<form METHOD="POST" ACTION="authent">
<tr> <td> Username </td>
<td><input type="text" name="usr"></td> </tr>
<tr> <td> Password </td>
<td> <input type="password" name="pwd"> </td> </tr>
<tr> <td align=center colspan="2"><input type="submit" value="submit"></td>
</tr>
</table> </center>
</form>
</body>
</html>
```

Ini.java:

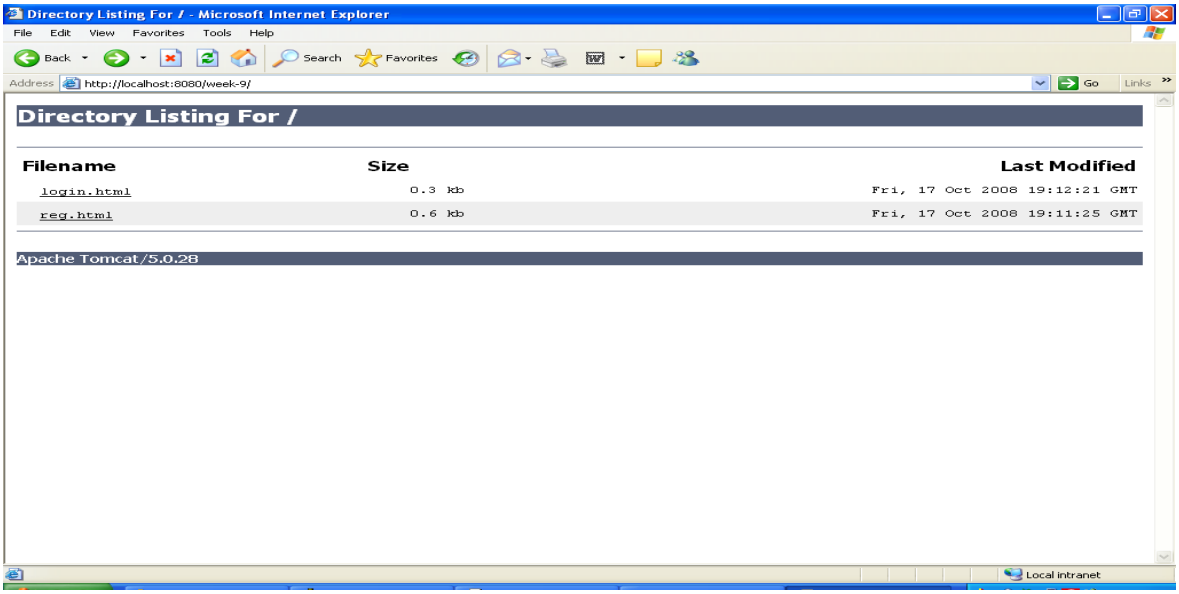
```
import javax.servlet.*;
import java.sql.*;
import java.io.*;
public class Ini extends GenericServlet
{
private String user1,pwd1,email1;

public void service(ServletRequest req,ServletResponse res) throws
ServletException,IOException
{
user1=req.getParameter("user");
```

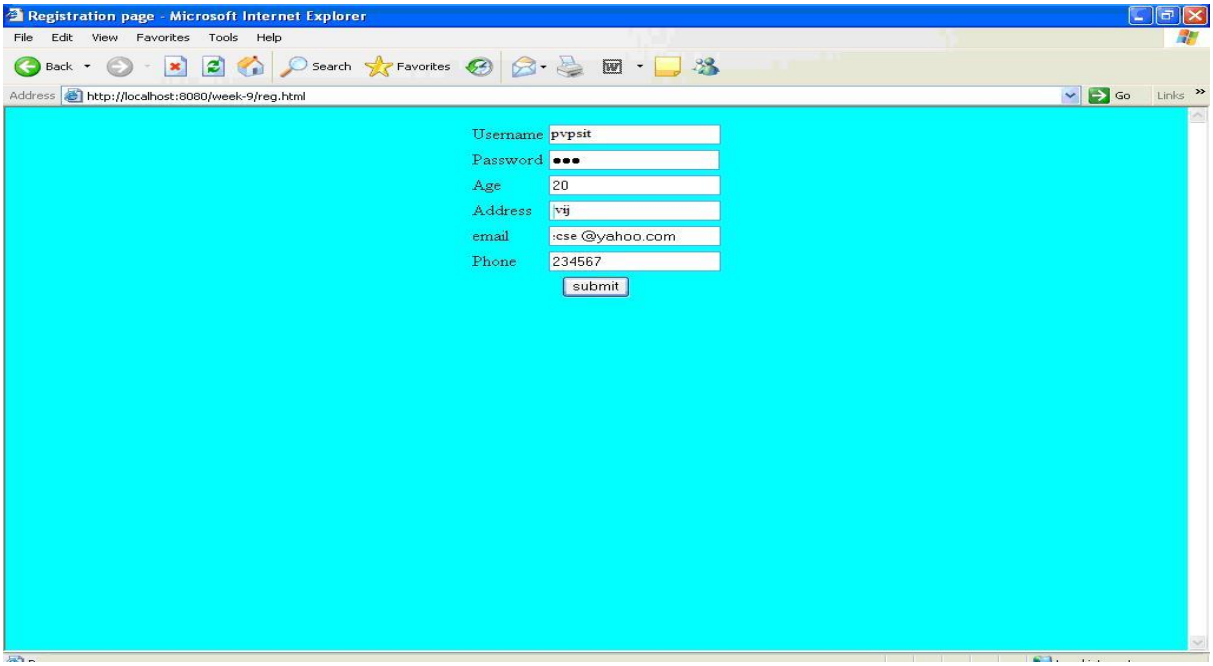
```
pwd1=req.getParameter("pwd");
email1=req.getParameter("email");
res.setContentType("text/html");
PrintWriter out=res.getWriter();
try
  {
   Class.forName("oracle.jdbc.driver.OracleDriver");
   Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@195.100.101.158:1521:ccla
b","scott","tiger");
  PreparedStatement st=con.prepareStatement("insert into
personalvalues(?,?,?,?,?,?)");
 st.setString(1,user1);
st.setString(2,pwd1);
st.setString(3,"25");
st.setString(4,"hyd");
st.setString(5,email1);
st.setString(6,"21234");
st.executeUpdate();
con.close();

 }
catch(SQLException s)
{  out.println("not found "+s);
}
catch(ClassNotFoundException c)
{
  out.println("not found "+c);
}
} }
```

web.xml:

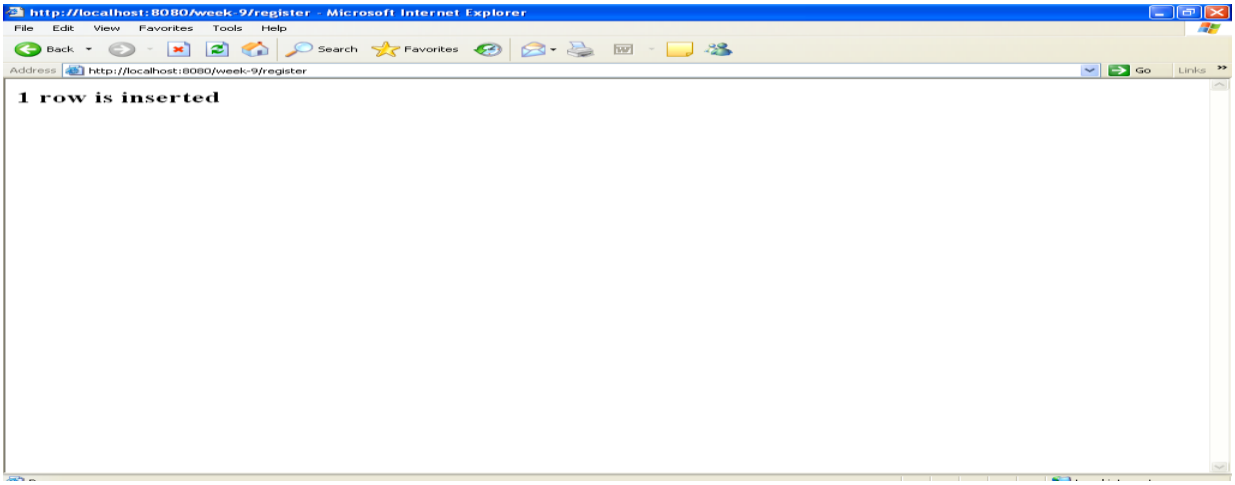
```
<web-app>
<servlet>
<servlet-name>init1</servlet-name>
<servlet-class>Ini</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>init1</servlet-name>
<url-pattern>/regis</url-pattern>
</servlet-mapping>
</web-app>
```

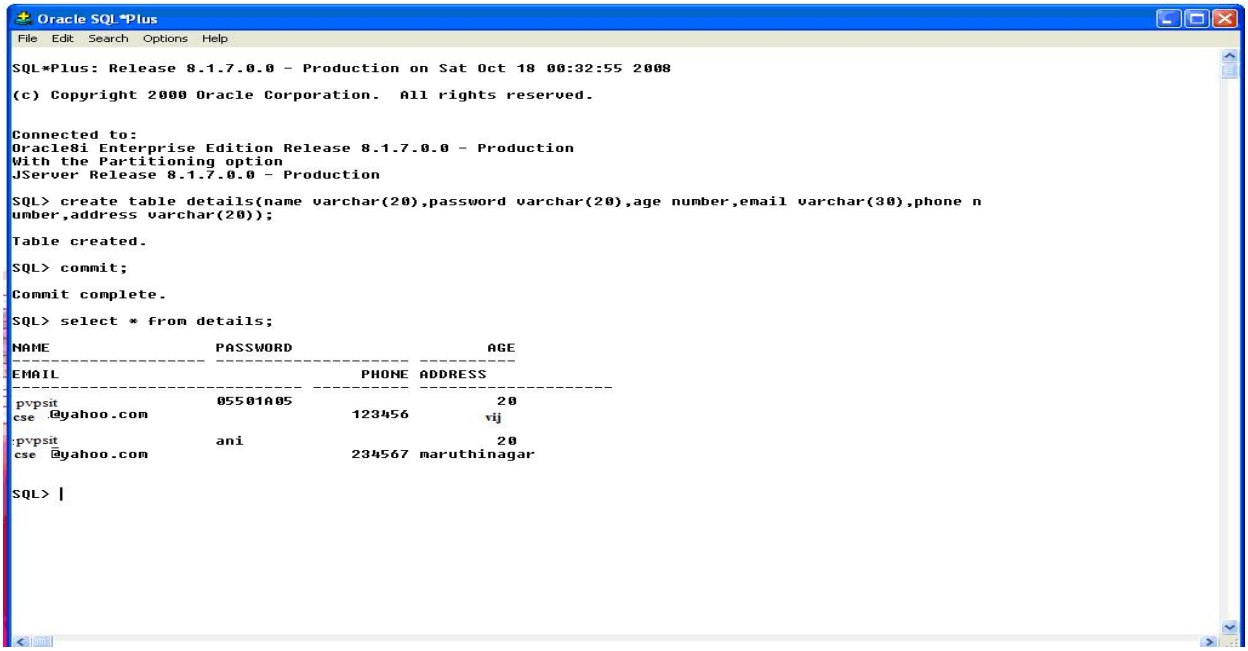
OUTPUT:-


Directory Listing For / - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/week-9/
Go Links
Directory Listing For /
Filename Size Last Modified
login.html 0.3 kb Fri, 17 Oct 2008 19:12:21 GMT
reg.html 0.6 kb Fri, 17 Oct 2008 19:11:25 GMT
Apache Tomcat/5.0.28
Local intranet



After inserting 1 users record





PRACTICAL NO: 11

OBJECTIVE: Write a JSP which insert the details of the 3 or 4 users who register with the web site by using registration form. Authenticate the user when he submits the login form using the user name and password from the database.

THEORY

Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. This tutorial will teach you how to use Java Server Pages to develop your web applications in simple and easy steps.

SOURCE CODE

UserPass.html

```
<body>
<form method="get" action="http://localhost:8888/india/Validation.jsp">
<h3>
  Enter User Name <input type="text" name="t1"> <br>
  Enter Password   <input type="password" name="t2"> <br>
  <input type="submit" value="Please Validate">
  <input type="reset" value="Clear Please">
</h3>
</body>
```

Validation.jsp

```
<body>
<h2 align="center"> Validating User Name and Password </h2>

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

 if(str1.equalsIgnoreCase("snrao") && str2.equals("java"))
 {
  out.println("<h3>Thankyou, you are VALID</h3>");
 }
 else
 {
  out.println("<h3>Sorry, you are INVALID</h3>");
 }
%>

</body>
```

Validation.java

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

public class Validation extends HttpServlet
{
 public void service(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException
 {
  res.setContentType("text/html");
  PrintWriter pw = res.getWriter( );
```

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

   if(str1.equalsIgnoreCase("snrao") && str2.equals("java"))
   {
    pw.println("<h3>Thankyou, you are VALID</h3>");
   }
   else
   {
    pw.println("<h3>Sorry, you are INVALID</h3>");
   }
   pw.close( );
  }
}
```

OUTPUT

PRACTICAL NO: 12

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

THEORY

Session Tracking is a way to maintain state (data) of an user. It is also known as **session** management in**servlet**. Http protocol is a stateless so we need to maintain state using **session tracking** techniques. Each time user requests to the server, server treats the request as the new request.

SOURCE CODE

ShoppingCart.html

```
<h3>Cookie Example through Shopping Cart</h3>
<body>
<form method="get" action="http://localhost:8888/india/SC">

 Enter Item Name <input type="text" name="item"><br>
 Enter Item Quantity <input type="text" name="qty"><br>
 <input type="submit" value="Add Cookie" name="add">
 <input type="submit" value="List Cookies" name="list">

</form>
</body>
```

web.xml entry for ShoppingCart servlet

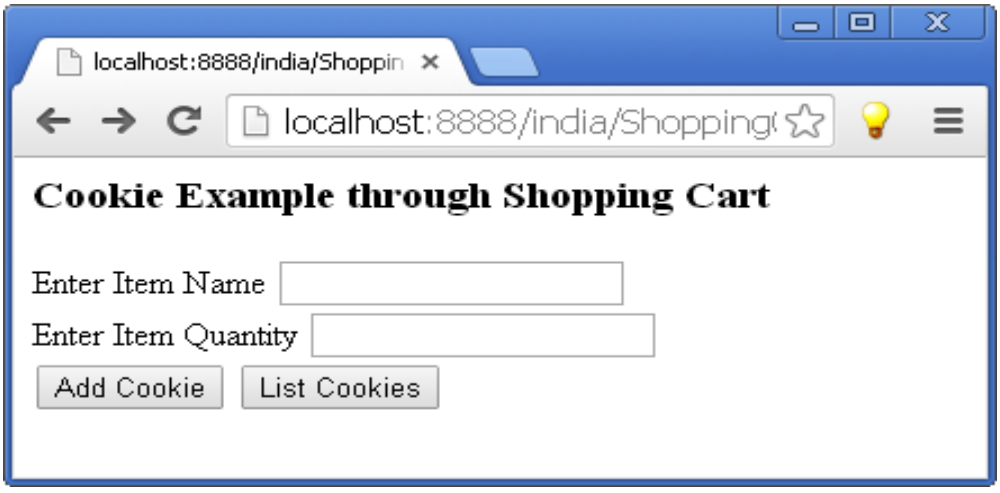
```
<servlet>
 <servlet-name>snrao1</servlet-name>
 <servlet-class>ShoppingCart</servlet-class>
</servlet>

<servlet-mapping>
 <servlet-name>snrao1</servlet-name>
 <url-pattern>/SC</url-pattern>
</servlet-mapping>
```

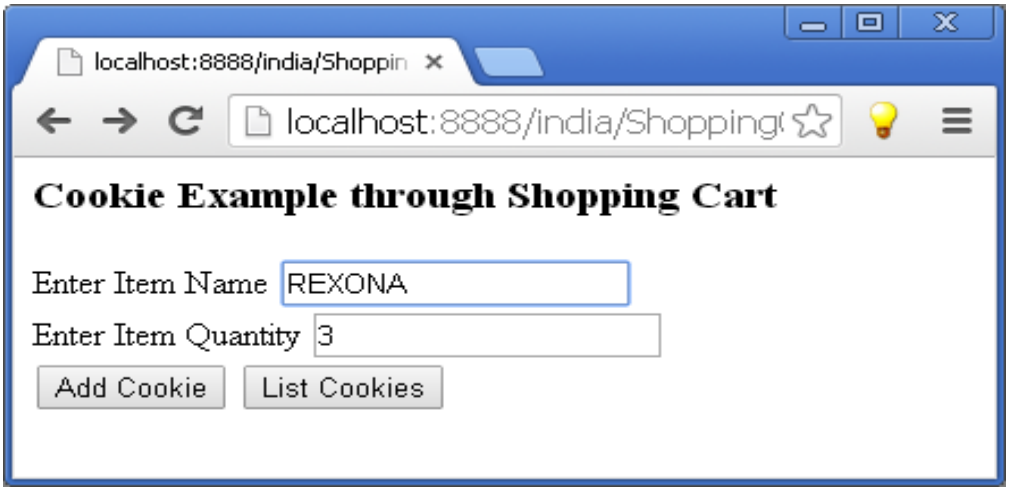
ShoppingCart.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class ShoppingCart extends HttpServlet
{
 public void service(HttpServletRequest req,HttpServletResponse res) throws
ServletException, IOException
 {
  String str1 = req.getParameter("item");      // item name
  String str2 = req.getParameter("qty");       // item quantity
  String str3 = req.getParameter("add");       // submit button by name add
  String str4 = req.getParameter("list");      // submit button by name list

  res.setContentType("text/html");
  PrintWriter out = res.getWriter();

  if(str3 != null)
  {
```

```
    Cookie c1 = new Cookie(str1, str2);
    res.addCookie(c1);
    res.sendRedirect("ShoppingCart.html");
   }
   else if(str4 != null)
   {
    Cookie clientCookies[] = req.getCookies();
    for( int i = 0; i < clientCookies.length; i++)
    {
     out.print("<B>" + clientCookies[i].getName() + " : " + clientCookies[i].getValue() +
"</B><BR>");
    }
   }
   out.close( ) ;
  }
}
```

OUTPUT:-




localhost:8888/india/Shoppin
localhost:8888/india/Shopping
Cookie Example through Shopping Cart
Enter Item Name REXONA
Enter Item Quantity 3
Add Cookie
List Cookies


localhost:8888/india/SC?item
localhost:8888/india/SC?item=
JSESSIONID : 68CC7353C7606CA75C5D634204C425A3
LUX : 10
SANTOOR : 5
REXONA : 3