Course Name: B.SC. (H) COMPUTER SCIENCE
SEMESTER-V
Paper Title: INTERNET TECHNOLOGY PRACTICAL

Submitted by:

Kritika Dey
Examination Roll No: 18044570002
College RollNo: CSC/18/7

College Name:
Mata Sundri College for Women, University of Delhi
College Address: Mata Sundri Lane, New Delhi 110002

JAVA

1. Implement a Bank Account having Instance variables: Account Number, Balance and having methods:
float Deposit (float x)
float withdraw (float x)
int get account no ()
float get balance ()
tax deduction ()
Then implement class Bank having an array list of accounts of type BankAccount. Implement following methods:
AddAccount in Bank
Get Total balance in Bank
Get account number with max. and min. balance
Find an account given a bank account no.
Count no. of accounts having atleast specific balance

```
import java.util.ArrayList;



class bankaccount

{

  int accno;

  double balance;

   bankaccount(int ac,double ibal)

  {

    accno=ac;

    balance=ibal;

  }

  bankaccount(int acnum)

  {

    accno=acnum;

    balance=0;

  }

 double deposit(double amt)
```

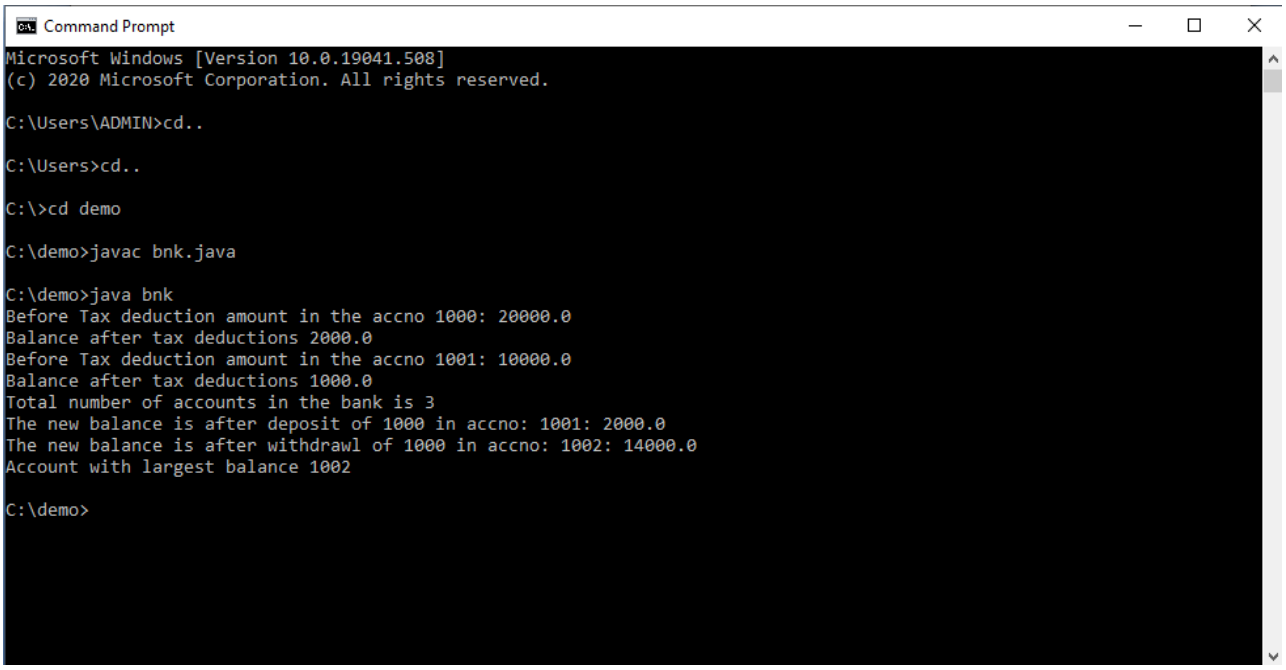
```
  {
     double new_balance=balance+amt;
     balance=new_balance;
     return new_balance;
  }
  double withdraw(double amt)
  {
    double new_balance=balance-amt;
    balance=new_balance;
    return new_balance;
  }
  int getaccount()
  {  return accno; }
  double getbalance()
  { return balance; }
  void tax_deduction()
  { double tax=balance*0.10;
    System.out.println("Balance after tax deductions "+tax);
    balance=tax;
  }
};

class bank
{
  ArrayList<bankaccount>account;
```

```
 bank()
{ account=new ArrayList<bankaccount>(); }

 void addaccount(bankaccount a)
 {  account.add(a);  }

 double get_total_balance()
 {   double total=0;
     for(bankaccount a:account)
      total+=a.getbalance();
    return total;
 }
 int count(double atleast)
{  int match=0;
   for(bankaccount a:account)
    match++;
   return match;
}
bankaccount find(int accno)
{
  for(bankaccount a:account)
   {
     if(a.getaccount()==accno)
      return a;
   }
```

```
   return null;
  }
  bankaccount getmax()
 {
   if(account.size()==0)
     return null;
   bankaccount large=account.get(0);
   for(int i=1;i<account.size();i++)
    {  bankaccount a=account.get(i);
       if(a.getbalance()>large.getbalance())
         large=a;
    }
  return large;
 }
};

public class bnk
 {
   public static void main(String args[])
    {
      bank ob=new bank();
      ob.addaccount(new bankaccount(1000,20000));
      ob.addaccount(new bankaccount(1001,10000));
      ob.addaccount(new bankaccount(1002,15000));
```

```
    int accountno=1000;
    bankaccount b=ob.find(accountno);

     if(b!=null)
     {
        System.out.println("Before Tax deduction amount in the accno "+b.getaccount()+":
"+b.getbalance());
        b.tax_deduction();
     }

    accountno=1001;
    bankaccount a=ob.find(accountno);
     if(a!=null)
     {
        System.out.println("Before Tax deduction amount in the accno "+a.getaccount()+":
"+a.getbalance());
        a.tax_deduction();
     }

    double threshold=15000;
    int c=ob.count(threshold);
    System.out.println("Total number of accounts in the bank is "+c);

    accountno=1001;
    bankaccount s=ob.find(accountno);
```

```
    System.out.println("The new balance is after deposit of 1000 in accno: "+s.getaccount()+":
"+s.deposit(1000));


    accountno=1002;

    bankaccount d=ob.find(accountno);

    System.out.println("The new balance is after withdrawl of 1000 in accno: "+d.getaccount()+":
"+d.withdraw(1000));


    bankaccount max=ob.getmax();

    System.out.println("Account with largest balance "+max.getaccount());


  }

};
```

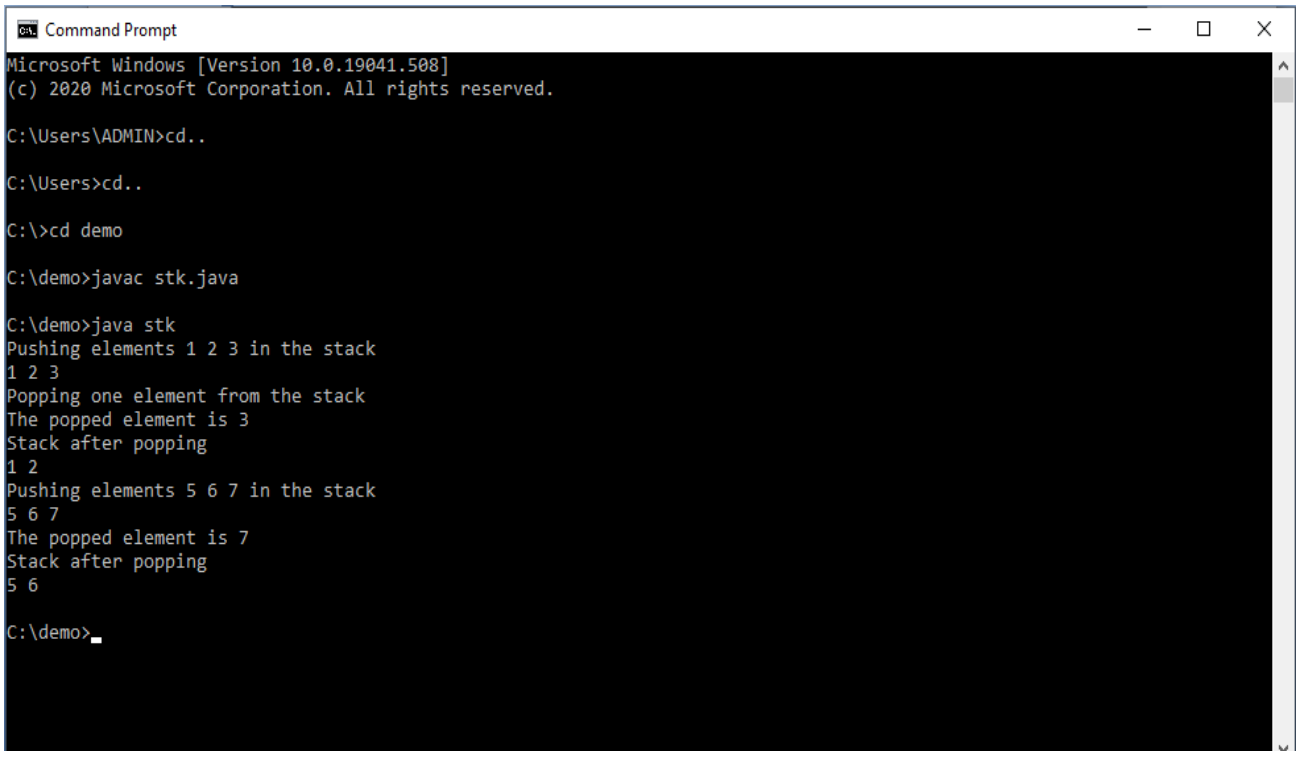


2. Implement an Abstract Class Stack with methods push, pop, display for two classes: StaticStack and DyanamicStack. StaticStack uses one dimensional integer array to store numbers and DyanamicStack uses an integer ArrayList to store.

```
import java.util.ArrayList;
abstract class stack
{

  int top;
   stack()
  {  top=-1;  }
   abstract  void push(int x);
   abstract  int pop();
   abstract  void display();
};
class staticstack extends stack
 {

   int ar[];
   staticstack()
   {  super();  }
   staticstack(int size)
   {
    super();
    ar=new int[size];
   }
  public void push(int item)
   {  if(super.top==ar.length-1)
       System.out.print("stack is full");
      else
       ar[++super.top]=item;
   }
  public int pop()
   {
       if(super.top<0)
       {  System.out.print("stack underflow");
          return 0;
       }
      else
       return ar[super.top--];

    }

  public void display()
   {   for(int i=0;i<=super.top;i++)
       System.out.print(ar[i]+" ");
       System.out.println();
   }
};
 class dynamicstack extends stack
```

```
{

  ArrayList<Integer>arr;
  dynamicstack()
   {  super();
      arr=new ArrayList<Integer>();
   }

  public void push(int item)
   {
       arr.add(++super.top,item);
   }
  public int pop()
   {  if(super.top<0)
       {  System.out.print("stack underflow");
          return 0;
       }
      else
       return arr.remove(super.top--);
   }

  public void display()
   {
        for(int i=0;i<arr.size();i++)
        System.out.print(arr.get(i)+" ");
        System.out.println();
   }
};


public class stk
 {
   public static void main(String args[])
   {
     staticstack a=new staticstack(3);
     System.out.println("Pushing elements 1 2 3 in the stack");
     a.push(1);
     a.push(2);
     a.push(3);
     a.display();
     System.out.println("Popping one element from the stack");
     int z=a.pop();
     System.out.println("The popped element is "+z);
     System.out.println("Stack after popping");
     a.display();
     System.out.println("Pushing elements 5 6 7 in the stack");
     dynamicstack b=new dynamicstack();
     b.push(5);
     b.push(6);
```

```
    b.push(7);
    b.display();
    int g=b.pop();
    System.out.println("The popped element is "+g);
    System.out.println("Stack after popping");
    b.display();


  }
};
```



]

JAVABEANS

1. Implement Student JavaBean using Serializability Interface

Stu.jsp

```
<%@ page contentType="text/html" %>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %> <html>

<head>

<title>User Info Entry Form</title>

</head>

<body bgcolor="white">

<jsp:useBean id="userInfo"

class="str.studentbean"> <jsp:setProperty name="userInfo" property="*" /> </jsp:useBean>

<form action="stud.jsp" method="post">

<input type="hidden" name="submitted" value="true"> <table>

<c:if

test="${param.submitted && !userInfo.userNameValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please enter your Name

</font></td></tr>

</c:if>

<tr>

<td>Name:</td>

<td>

<input type="text" name="userName"

value="<c:out value="${userInfo.userName}" />"> </td>

</tr>
```

```
<c:if test="${param.submitted && !userInfo.birthDateValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please enter a valid Birth Date

</font></td></tr>

</c:if>

<tr>

<td>Birth Date:</td>

<td>

<input type="text" name="birthDate"

value="<c:out value="${userInfo.birthDate}" />">

</td>

<td>(Use format yyyy-mm-dd)</td>

</tr>


<c:if test="${param.submitted && !userInfo.addressValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please enter a valid Address

</font></td></tr>

</c:if>

<tr>

<td>Address:</td>

<td>

<input type="text" name="address"

value="<c:out value="${userInfo.address}" />">

</td>
```

```
</tr>

<c:if test="${param.submitted && !userInfo.courseValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please enter a valid Course

</font></td></tr>

</c:if>

<tr>

<td>Course:</td>

<td>

<input type="text" name="course"

value="<c:out value="${userInfo.course}" />">

</td>

</tr>

<c:if test="${param.submitted && !userInfo.rollnoValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please enter a valid Rollno

</font></td></tr>

</c:if>

<tr>

<td>Roll Number:</td>

<td>

<input type="text" name="rollno"

value="<c:out value="${userInfo.rollno}" />">
```

```
</td>

</tr>


<c:if test="${param.submitted && !userInfo.genderValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please select a valid Gender

</font></td></tr>

</c:if>

<tr>

<td>Gender:</td>

<td>

<c:choose>

<c:when test="${userInfo.gender == 'f'}">

<input type="radio" name="gender" value="m" > Male<br>

<input type="radio" name="gender" value="f" checked> Female

</c:when>

<c:otherwise>

<input type="radio" name="gender" value="m" checked> Male<br>

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

</c:otherwise>

</c:choose>

</td>

</tr>


<tr>
```

```
<td colspan="3">
<input type="submit" value="Send Data">
</td>
</tr>
</table>
</form>
</body>
</html>
```

Studentbean.java

```
package str;

import java.io.Serializable;
public class studentbean implements Serializable {
// Validation constants
private static String DATE_FORMAT_PATTERN = "yyyy-MM-dd";
private static String[] GENDER_LIST = {"m", "f"};

 //properties
 private String userName;
 private String birthDate;
 private String emailAddr;
 private String gender;
 private String rollno;
 private String course;
```

```
 private String address;


 //getter method for username
 public String getUserName() {
 return userName;
 }
//setter method for username
 public void setUserName(String userName) {
 this.userName = userName;
 }
//validation for username
public boolean isUserNameValid()
 {
 boolean isValid = false;
 if (userName!=null )
 {
 isValid = true;
 }
 return isValid;
 }
//getter method for birthdate
 public String getBirthDate() {
 return (birthDate == null ? "" : birthDate);
 }
//setter method for birthdate
```

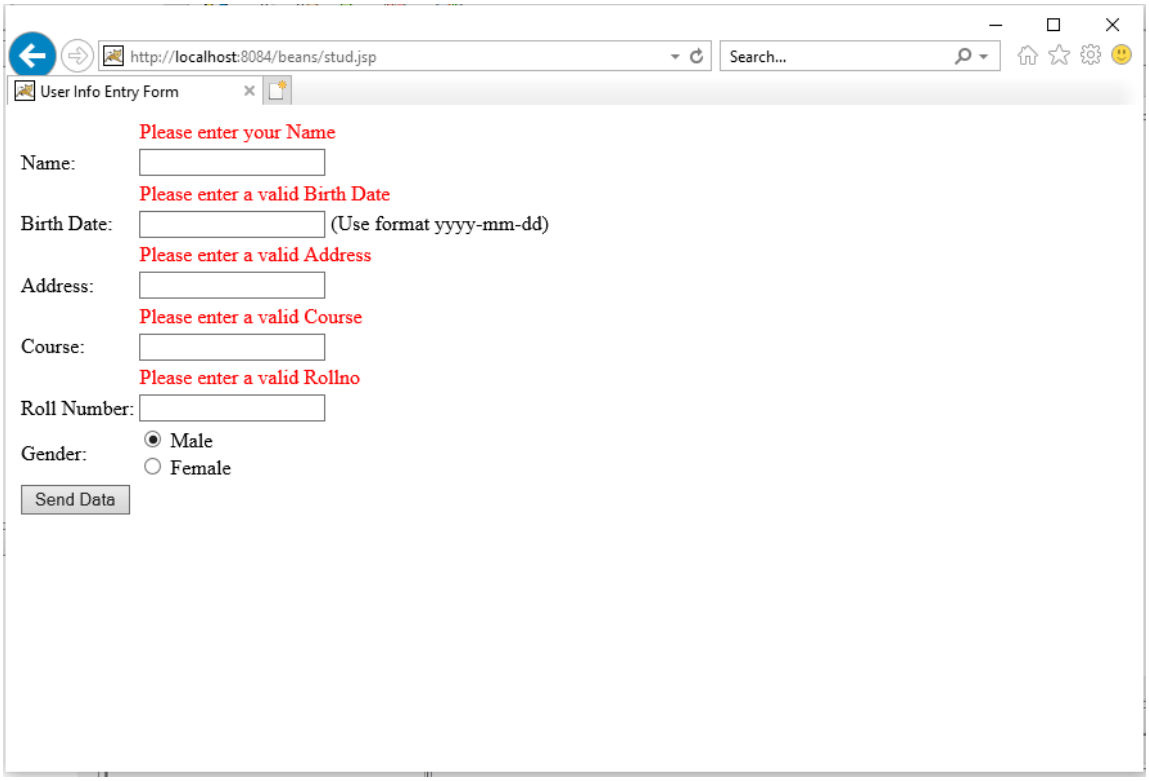
```
public void setBirthDate(String birthDate) {
this.birthDate = birthDate;
}
//validation for birthdate
public boolean isBirthDateValid( ) {
boolean isValid = false;
if (birthDate != null )
{
isValid = true;
}
return isValid;
}

//getter method for rollno
public String getRollno() {
 return rollno;
}
//setter method for rollno
public void setRollno(String rollno) {
   this.rollno = rollno;
 }
//validation for rollno
 public boolean isRollnoValid()
{
boolean isValid = false;
```

```
 if (rollno!=null )
 {
 isValid = true;
 }
 return isValid;
 }
//getter method for course
 public String getCourse() {
   return course;
    }
//setter method for course
 public void setCourse(String course) {
  this.course = course;
    }
//validation for course
public boolean isCourseValid()
 {
 boolean isValid = false;
 if (course!=null )
 {
 isValid = true;
 }
 return isValid;
 }
//getter method for address
```

```
public String getAddress() {
    return address;
   }
//setter method for address
 public void setAddress(String address) {
    this.address = address;
   }
//validation for address
 public boolean isAddressValid()
 {
 boolean isValid = false;
 if (address!=null )
 {
 isValid = true;
 }
 return isValid;
 }
 //getter method for emailaddr
 public String getEmailAddr() {
 return emailAddr;
 }
 //setter method for emailaddr
 public void setEmailAddr(String emailAddr) {
 this.emailAddr = emailAddr;
 }
```

```
//validation for emailaddr
public boolean isEmailAddrValid()
{
boolean isValid = false;
if (emailAddr.contains("@")&&emailAddr!=null ||emailAddr!="")   {
isValid = true;
}
return isValid;
}
 //getter method for gender
public String getGender() {
return gender;
}
//setter method for gender
public void setGender(String gender) {
this.gender = gender;
}
//validation for emailaddr
public boolean isGenderValid()
{
boolean isValid = false;
if (gender != null && (gender.equals("f")||gender.equals("m")))   {
isValid = true;
}
return isValid;
```

```
}

public boolean isValid() {
return isBirthDateValid() && isEmailAddrValid() &&
isGenderValid() && isUserNameValid();
}

}
```



2. Implement Employee JavaBean using Serializability Interface.

Empl.jsp

```
<%@ page contentType="text/html" %>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %> <html>

<head>

<title>Employee Info Entry Form</title>

</head>

<body bgcolor="white">

<jsp:useBean id="empInfo"

class="str.employeebean"> <jsp:setProperty name="empInfo" property="*" /> </jsp:useBean>

<form action="empl.jsp" method="post">

<input type="hidden" name="submitted" value="true"> <table>

<c:if

test="${param.submitted && !empInfo.empNameValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please enter your Name

</font></td></tr>

</c:if>

<tr>

<td>Name:</td>

<td>

<input type="text" name="empName"

value="<c:out value="${empInfo.empName}" />"> </td>

</tr>


<c:if test="${param.submitted && !empInfo.birthDateValid}"> <tr><td></td>
```

```
<td colspan="2"><font color="red">

Please enter a valid Birth Date

</font></td></tr>

</c:if>

<tr>

<td>Birth Date:</td>

<td>

<input type="text" name="birthDate"

value="<c:out value="${empInfo.birthDate}" />">

</td>

<td>(Use format yyyy-mm-dd)</td>

</tr>


<c:if test="${param.submitted && !empInfo.addressValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please enter a valid Address

</font></td></tr>

</c:if>

<tr>

<td>Address:</td>

<td>

<input type="text" name="address"

value="<c:out value="${empInfo.address}" />">

</td>

</tr>
```

```
<c:if test="${param.submitted && !empInfo.departmentValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please enter a valid Department

</font></td></tr>

</c:if>

<tr>

<td>Department:</td>

<td>

<input type="text" name="department"

value="<c:out value="${empInfo.department}" />">

</td>

</tr>


<c:if test="${param.submitted && !empInfo.empidValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please enter a valid EmpID

</font></td></tr>

</c:if>

<tr>

<td>Emp ID:</td>

<td>

<input type="text" name="empid"

value="<c:out value="${empInfo.empid}" />">

</td>
```

```
</tr>

<c:if test="${param.submitted && !empInfo.salaryValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please enter a valid Salary

</font></td></tr>

</c:if>

<tr>

<td>Salary:</td>

<td>

<input type="text" name="salary"

value="<c:out value="${empInfo.salary}" />">

</td>

</tr>

<c:if test="${param.submitted && !empInfo.genderValid}"> <tr><td></td>

<td colspan="2"><font color="red">

Please select a valid Gender

</font></td></tr>

</c:if>

<tr>

<td>Gender:</td>

<td>

<c:choose>

<c:when test="${empInfo.gender == 'f'}">
```

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

<input type="radio" name="gender" value="f" checked> Female

</c:when>

<c:otherwise>

<input type="radio" name="gender" value="m" checked> Male<br>

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

</c:otherwise>

</c:choose>

</td>

</tr>


<tr>

<td colspan="3">

<input type="submit" value="Send Data">

</td>

</tr>

</table>

</form>

</body>

</html>
```

Employeebean.java

```
package str;


import java.io.Serializable;
public class employeebean implements Serializable {
// Validation constants
private static String DATE_FORMAT_PATTERN = "yyyy-MM-dd";
private static String[] GENDER_LIST = {"m", "f"};


 //properties
 private String empName;
 private String birthDate;
 private String emailAddr;
 private String gender;
 private String empid;
 private String department;
 private String address;
 private String salary;


 //getter method for empname
 public String getEmpName() {
 return empName;
 }
//setter method for empname
 public void setEmpName(String empName) {
```

```
 this.empName = empName;
 }
//validation for empname
public boolean isEmpNameValid()
 {
 boolean isValid = false;
 if (empName!=null )
 {
 isValid = true;
 }
 return isValid;
 }
//getter method for birthdate
 public String getBirthDate() {
 return (birthDate == null ? "" : birthDate);
 }
//setter method for birthdate
 public void setBirthDate(String birthDate) {
 this.birthDate = birthDate;
 }
//validation for birthdate
 public boolean isBirthDateValid( ) {
 boolean isValid = false;
 if (birthDate != null )
 {
```

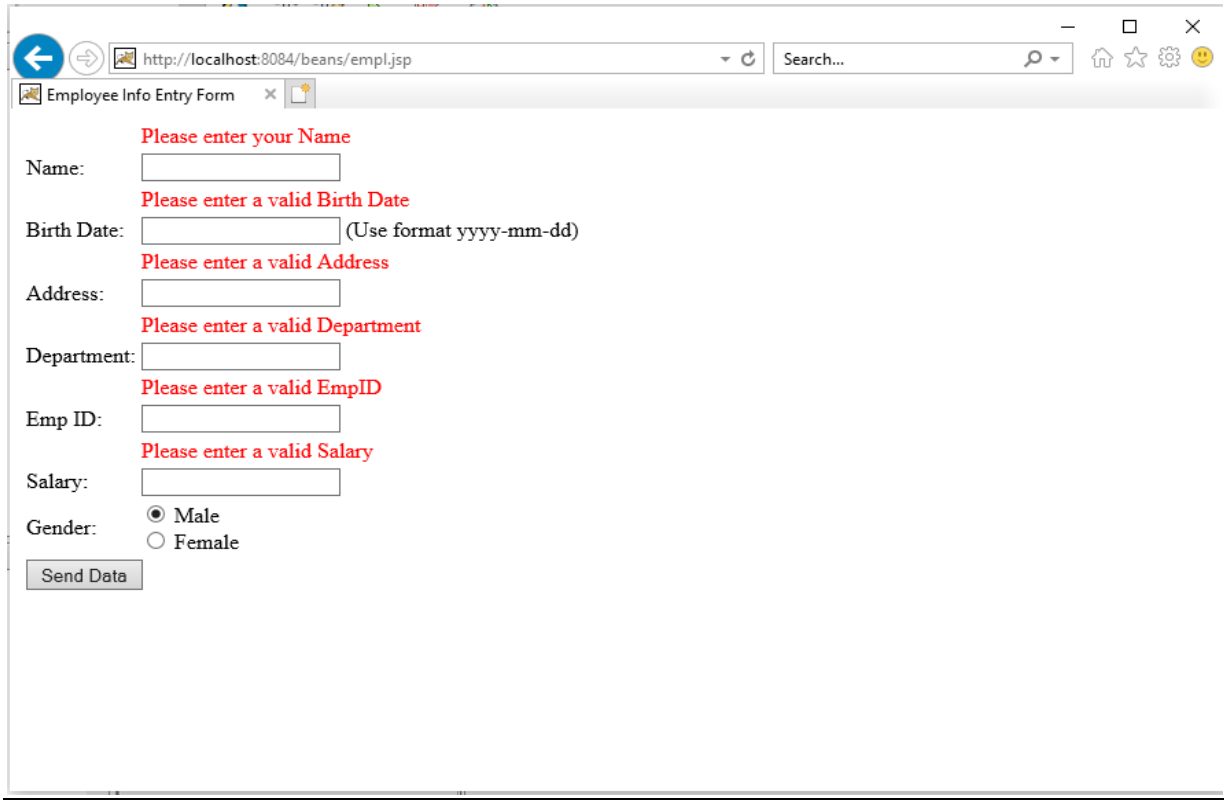
```
 isValid = true;
 }
 return isValid;
 }
//getter method for salary
public String getSalary() {
  return salary;
 }
//setter method for salary
public void setSalary(String salary) {
   this.salary = salary;
}
//validation for salary
 public boolean isSalaryValid()
 {
 boolean isValid = false;
 if (salary!=null )
 {
 isValid = true;
 }
 return isValid;
 }
 //getter method for empid
 public String getEmpid() {
  return empid;
```

```
 }
//setter method for rollno
 public void setEmpid(String empid) {
    this.empid = empid;
  }
//validation for rollno
  public boolean isEmpidValid()
 {
 boolean isValid = false;
 if (empid!=null )
 {
 isValid = true;
 }
 return isValid;
 }
//getter method for department
 public String getDepartment() {
   return department;
    }
//setter method for department
 public void setDepartment(String department) {
  this.department = department;
    }
//validation for department
public boolean isDepartmentValid()
```

```
{
 boolean isValid = false;
 if (department!=null )
 {
 isValid = true;
 }
 return isValid;
 }
//getter method for address
public String getAddress() {
    return address;
   }
//setter method for address
 public void setAddress(String address) {
    this.address = address;
   }
//validation for address
 public boolean isAddressValid()
 {
 boolean isValid = false;
 if (address!=null )
 {
 isValid = true;
 }
 return isValid;
```

```
}
//getter method for emailaddr
public String getEmailAddr() {
return emailAddr;
}
//setter method for emailaddr
public void setEmailAddr(String emailAddr) {
this.emailAddr = emailAddr;
}
//validation for emailaddr
public boolean isEmailAddrValid()
{
boolean isValid = false;
if (emailAddr.contains("@")&&emailAddr!=null ||emailAddr!="")   {
isValid = true;
}
return isValid;
}
 //getter method for gender
public String getGender() {
return gender;
}
//setter method for gender
public void setGender(String gender) {
this.gender = gender;
```

```
}
//validation for emailaddr
public boolean isGenderValid()
{
boolean isValid = false;
if (gender != null && (gender.equals("f")||gender.equals("m")))   {
isValid = true;
}
return isValid;
}

public boolean isValid() {
return isBirthDateValid() && isEmailAddrValid() &&
isGenderValid() && isEmpNameValid();
}



}
```



JDBC

1. Create Student and Results Database and perform the following using JDBC programs
a. Find total number of students
b. Print average marks for each subject input by user.
c. Find the name of student getting highest marks.
d. Find no of students getting first, second and third division.
e. Find subject wise toppers
f. Find the average marks
g. Find the student getting second highest marks.

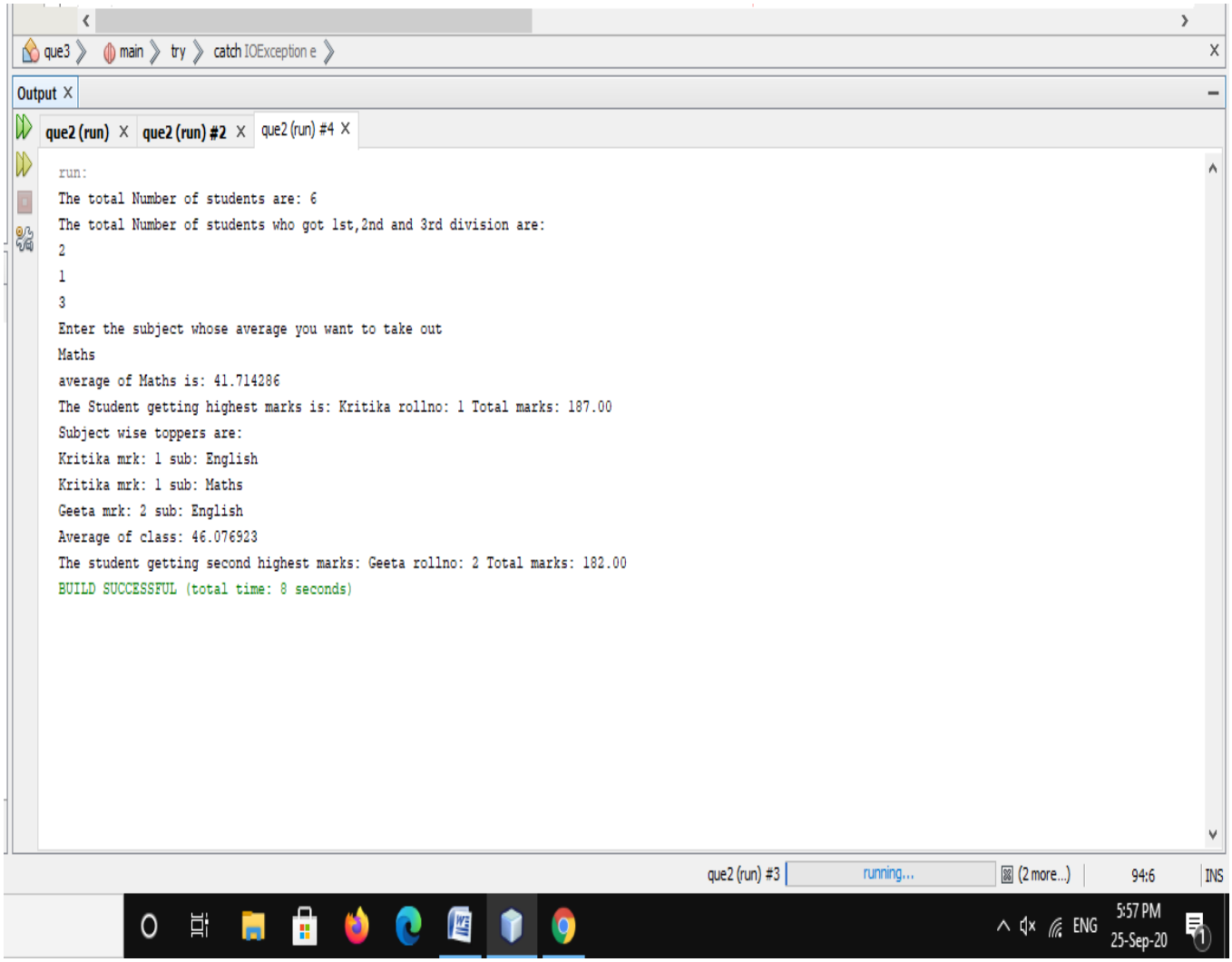
```
import java.io.*;

import java.sql.*;

public class que3 {

    public static void main(String[] args) {


      try{
```

```
Class.forName("com.mysql.cj.jdbc.Driver");
Connection conn;
conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/it_gals","root","root");
Statement stmt  = conn.createStatement();
String query = "SELECT count(*) FROM stu";

ResultSet rs  = stmt.executeQuery(query);
rs.next();
//query1
int count=rs.getInt(1);
System.out.println("The total Number of students are: "+count);
//query2
query="SELECT COUNT(ROLLNO) AS TOTAL FROM DIVISION where TOTAL>=120\n" +
       "UNION\n" +
         "SELECT COUNT(ROLLNO) AS TOTAL FROM DIVISION where TOTAL>=80 AND\n" +
           "TOTAL<120\n" +
             "UNION\n" +
               "\n" +
         "SELECT COUNT(ROLLNO) AS TOTAL FROM DIVISION where TOTAL<80;";
      rs=stmt.executeQuery(query);
      System.out.println("The total Number of students who got 1st,2nd and 3rd division are: ");
      while(rs.next())
      {
          System.out.println(rs.getString(1));
```

```
 }
  rs.next();
//query3
  BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the subject whose average you want to take out");
String sub=br.readLine();
switch(sub)
{
   case "English":query="select avg(MARKS) from res where SUBJECT='English'"; break;
   case "Maths":query="select avg(MARKS) from res where SUBJECT='Maths'"; break;
   default:System.out.println("Wrong input");
}

 ResultSet r=stmt.executeQuery(query);
  r.next();
 String average=r.getString(1);
 System.out.println("average of "+sub+" is: "+ average);


 //query4
 query="select name, rollno, sum(marks) from stu, res where roll=rollno group by rollno order by sum(marks) desc limit 1;";
 rs=stmt.executeQuery(query);
 rs.next();
```

```
System.out.println("The Student getting highest marks is: "+rs.getString(1)+" rollno: "+rs.getString(2)+" Total marks: "+rs.getString(3));


//query5
query="select name,rollno,subject,marks as \"Highest\" from stu, res where\n" +
        "roll=rollno and marks in (select max(marks) from res group by subject); ";
rs=stmt.executeQuery(query);
System.out.println("Subject wise toppers are:");
while(rs.next())
{
System.out.println(rs.getString(1)+" mrk: "+rs.getString(2)+" sub: "+rs.getString(3));
}


//query 6
query="SELECT avg(marks) FROM res ";
rs=stmt.executeQuery(query);
rs.next();
String nd=rs.getString(1);
System.out.println("Average of class: "+nd);


//query7
query="select name, rollno, sum(marks) from stu, res where roll=rollno group\n" +
        "by rollno order by sum(marks) desc limit 1,1; ";
rs=stmt.executeQuery(query);
rs.next();
```

```
            System.out.println("The student getting second highest marks: "+rs.getString(1)+" rollno:
"+rs.getString(2)+" Total marks: "+rs.getString(3));




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

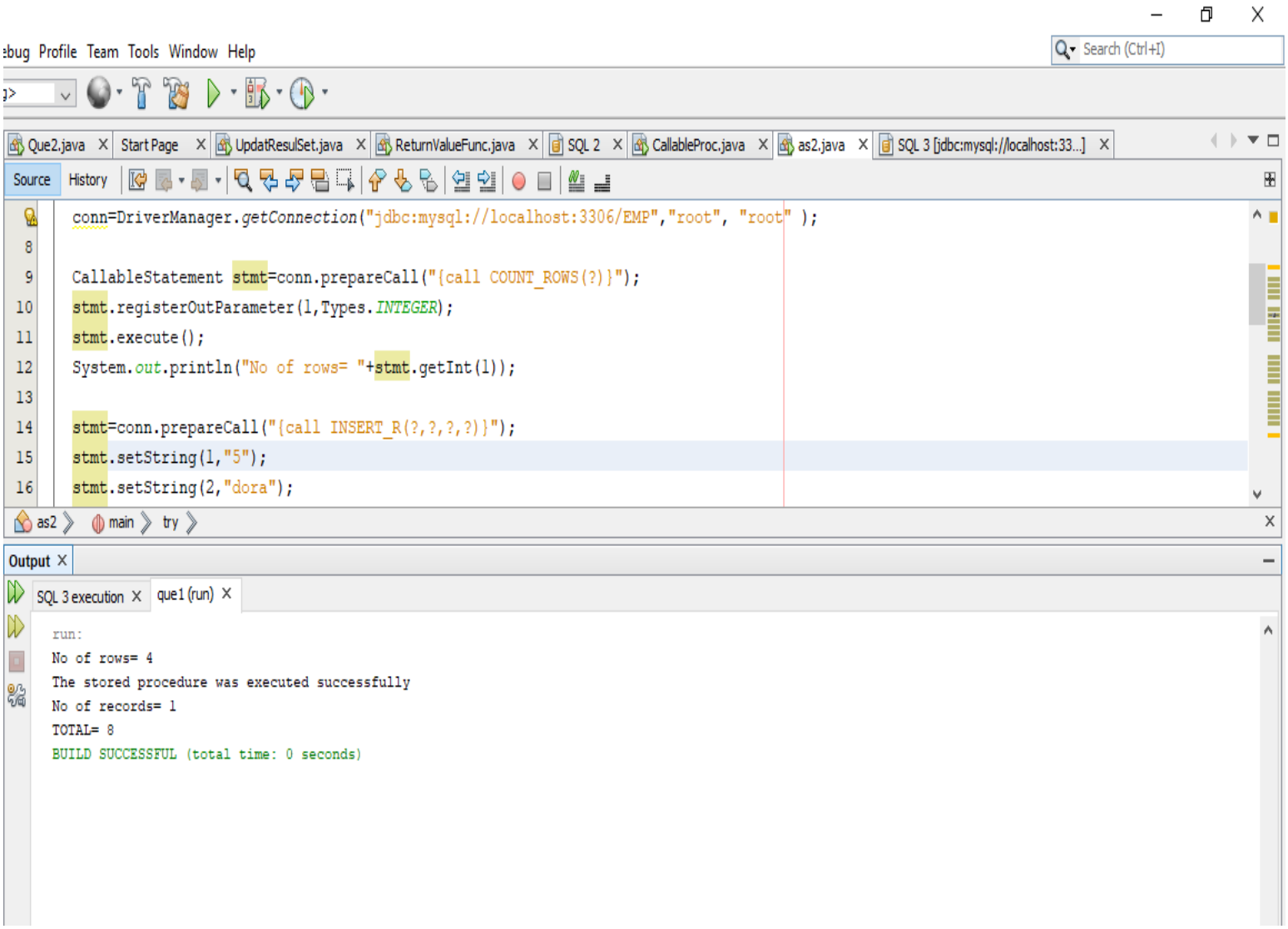


2. Create a procedure in MySQL to count the number of Rows in table 'Student'. Use CallableStatement to call this method from Java code.

```
import java.sql.*;

public class as2 {

   public static void main(String args[]){

try{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection

conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/EMP","root", "root" );
```

```
CallableStatement stmt=conn.prepareCall("{call COUNT_ROWS(?)}");
stmt.registerOutParameter(1,Types.INTEGER);
stmt.execute();
System.out.println("No of rows= "+stmt.getInt(1));

stmt=conn.prepareCall("{call INSERT_R(?,?,?,?)}");
stmt.setString(1,"5");
stmt.setString(2,"dora");
stmt.setString(3,"18");
stmt.setString(4,"F");
stmt.execute();
System.out.println("The stored procedure was executed successfully");

stmt=conn.prepareCall("{?=call Records(?)}");
stmt.setString(2,"50000");
stmt.registerOutParameter(1,Types.INTEGER);
stmt.execute();
System.out.println("No of records= "+stmt.getInt(1));

stmt=conn.prepareCall("{call addition(?,?,?)}");
stmt.setString(1,"5");
stmt.setString(2,"3");
stmt.registerOutParameter(3,Types.INTEGER);
stmt.execute();
System.out.println("TOTAL= "+stmt.getInt(3));
```

```
conn.close();

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

}

}
```



JavaScript

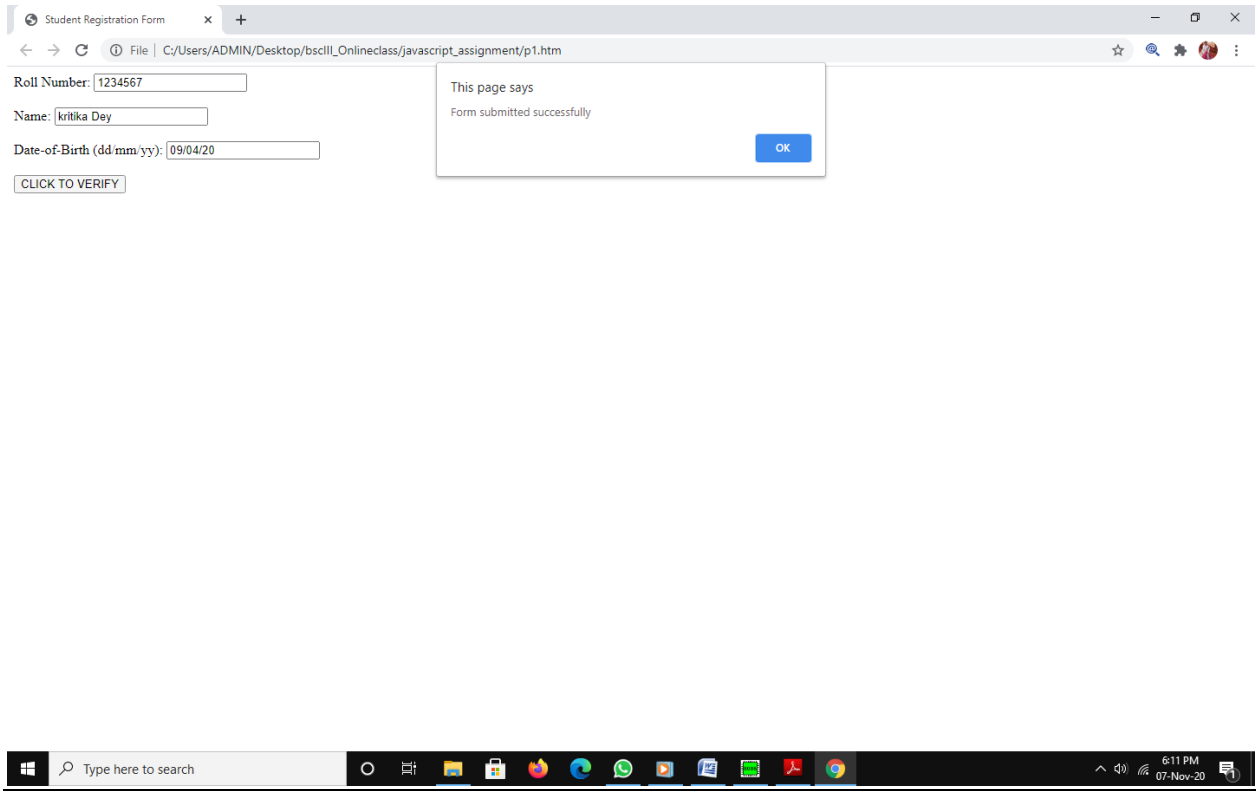
1. Create a student registration form. Create functions to perform the following checks:
a. Roll number is a 7 digit numeric value
b. Name should be an alphabetical value (String)
c. DOB entered in dd/mm/yy format and should be display in words (e.g. Saturday, January 01, 2000)
d. Check on non-empty fields

```
<html>

  <head>

    <title> Student Registration Form </title>

 <script>




        function check()

        {

                var flag=1;

       var rollno =myform.elements[0].value

       var uname=myform.elements[1].value

       var dob = myform.elements[2].value


          if(rollno =="" || rollno == null)

          {       alert("Your rollno field is empty");

                  myform.elements[0].focus();

         flag=0;

                  }

          else

          {       if(/[0-9]+/.test(rollno))

                  {           if(rollno.length!=7)
```

```
                    { flag=0;
        alert("Warning : Roll No. should contain 7 digits");
       }
          }
          else
   {   flag=0;
                    alert("Rollno should be a number");
   }
   }

   if(uname =="" ||uname== null)
   {   alert("Your name field is empty");
          document.myform.elements[1].focus();
flag=0;
          }
else
  {
          if(!/[a-zA-Z ]+/.test(uname))
          {
       flag=0;
                    alert("Your name is not valid !!!");
          }

   }
```

```
            if(dob =="" ||dob== null)
            {   alert("Your Date-of-Birth field is empty");
                    document.myform.elements[2].focus();
          flag =0;
                    }
            else
            {//regular expression
                    var frmat=/^(0?[1-9]|[12][0-9]|3[01])[\/](0?[1-9]|1[012])[\/]\d{2}$/;



                    //Check whether valid dd/MM/yy Date format
                    if(!frmat.test(dob))
              {  flag=0;
                            alert("Your date of birth is not valid ");
              }
            }

      if(flag==1)
      alert("Form submitted successfully");

         }
</script>
```

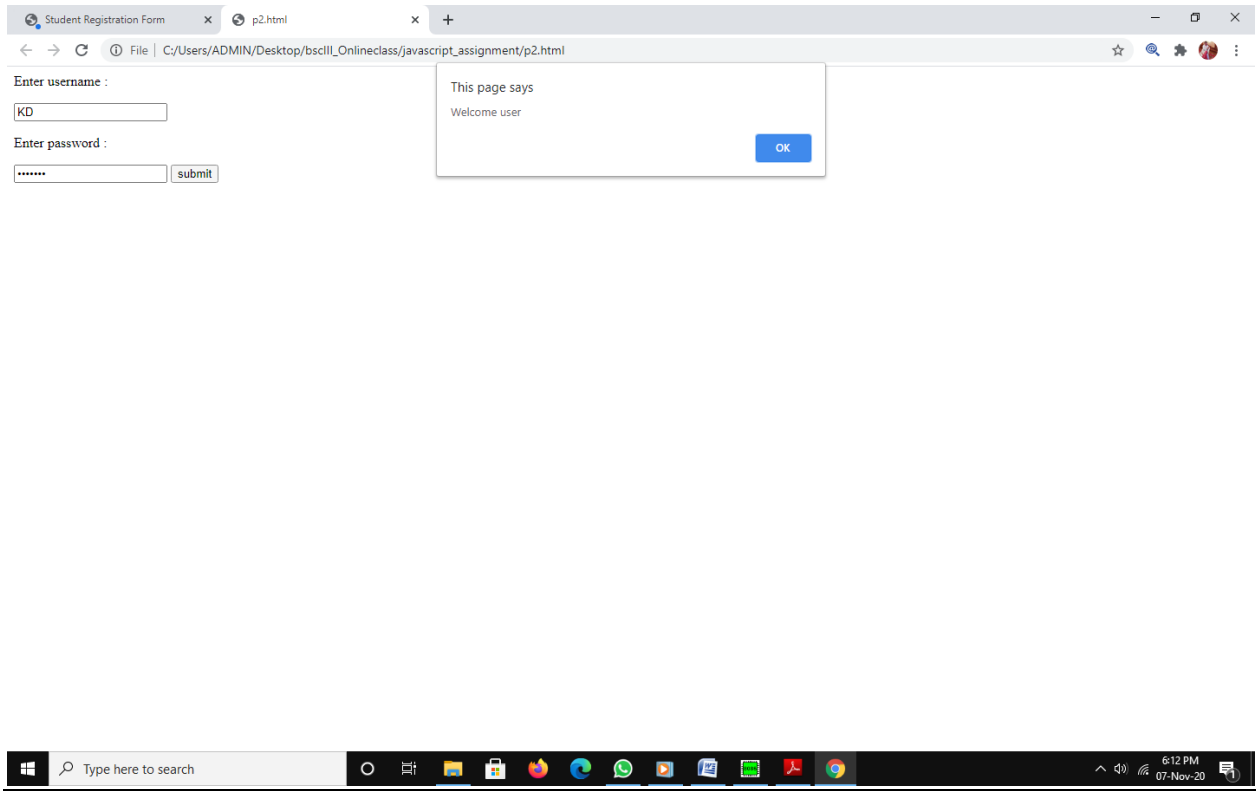
```
  </head>
  <body>
    <div>
      <form name=myform>
        Roll Number: <input type="text" name="rollno" value"" ><br/><br/>

        Name: <input type="text" name="uname" value="" ><br/><br/>

        Date-of-Birth (dd/mm/yy): <input type="text" name="dob" value=""><br/><br/>

        <button type="button" name="Submit" onclick = "check()"> CLICK TO VERIFY</button>
        </form>

    </div>
  </body>
</html>
```



2. Implement a Static Password Protection.

```
<html>

<body>

<form name="form1">

<p>Enter username :</p>

<input name="nm" type='text'><br>

<p>Enter password  :</p>

<input name="pw" type='password'>

<input type="submit" onclick="myFunction()" value="submit">

</form>


<script>

function myFunction() {
```

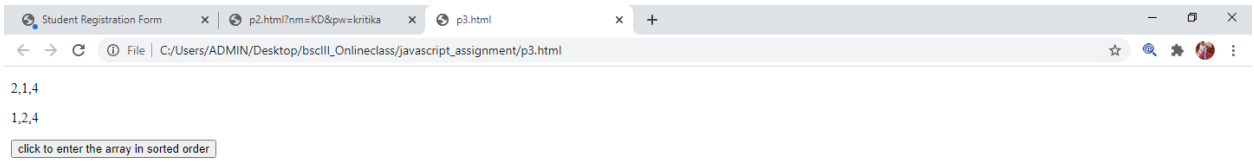
```
 if(document.form1.pw.value=="kritika" && document.form1.nm.value=="KD" )
   alert("Welcome user");
else
 alert("Incorrect username or password");

}
</script>

</body>
</html>
```



3. Write a java script to sort an array using bubble sort. Take the number of elements and array from user.

```
<!DOCTYPE html>

<html>

<body>



<p id ="dm"></p>

<p id ="abs"></p>



<script>

var n=prompt("Enter the number of elements in the array :");

var  ar=[];



for(i=0;i<n;i++)

ar.push(prompt("Enter the "+i+"th element  in the array :"));

document.getElementById("dm").innerHTML=ar;



function stack()

{

var i,j;

var min;



for(j=0;j<n;j++)

{

if(ar[j]>ar[j+1])

{
```

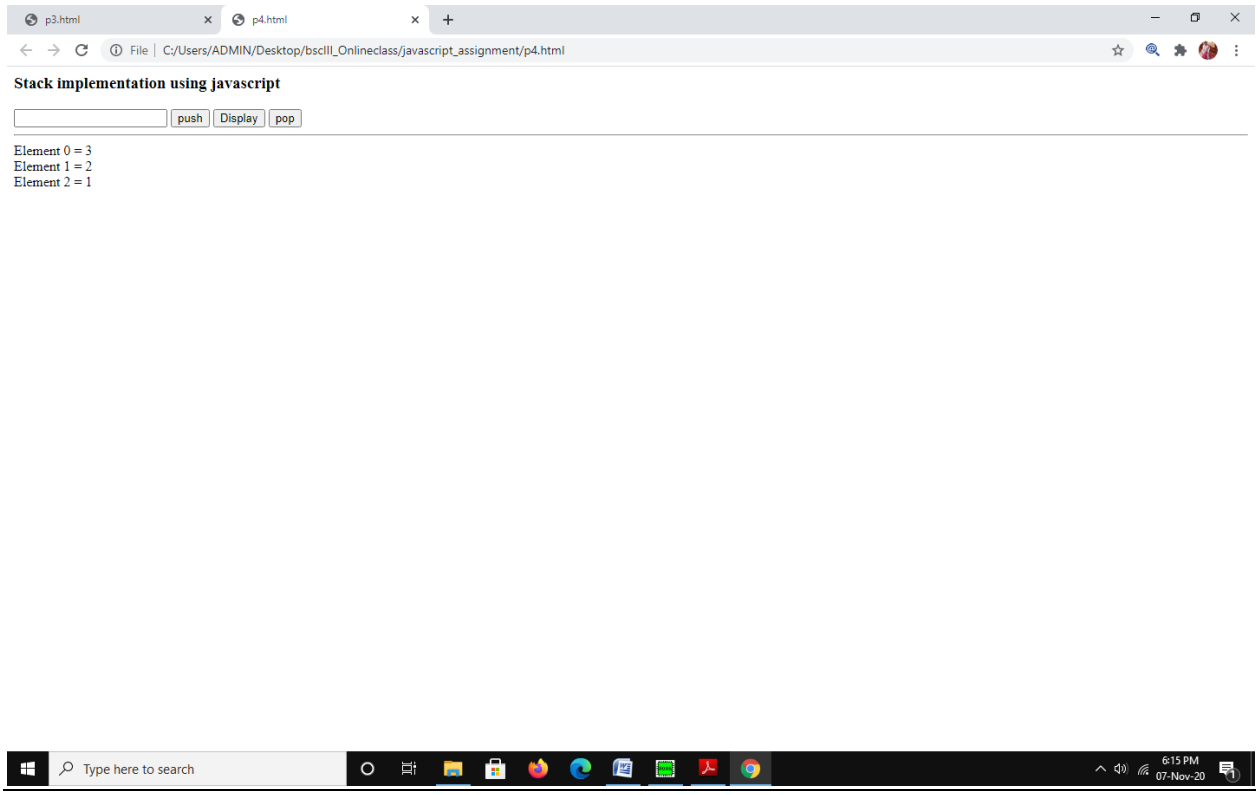
```
min=ar[j];
 ar[j]=ar[j+1];
ar[j+1]=min;
}

}
document.getElementById("abs").innerHTML=ar;
}

</script>
<button type="button" onClick="stack()">click to enter the array in sorted order</button>
</body>
</html>
```



4. Write a JavaScript to implement stack methods (push and pop).

```
<script>

var x = 0;

var array = Array();


function push_element_to_array()

{

 array[x] = document.getElementById("text1").value;

 alert("Element: " + array[x] + " Added at index " + x);

 x++;

 document.getElementById("text1").value = "";

}


function display_array()

{

  var e = "<hr/>";


  for (var y=0; y<array.length; y++)

  {

    e += "Element " + y + " = " + array[y] + "<br/>";

  }

  document.getElementById("Result").innerHTML = e;

}

function ppop_array()

{
```

```
  array.pop();
x--;
 display_array();
}
</script>



<html>
<head>
</head>
<body >
<h3>Stack implementation using javascript</h3>
<input type="text" id="text1"></input>
<input type="button" id="button1" value="push" onclick="push_element_to_array();"></input>
<input type="button" id="button2" value="Display" onclick="display_array();"></input>
<input type="button" id="button3" value="pop" onclick="ppop_array();"></input>
<div id="Result"></div>
</body>
</html>
```



5. Write a JavaScript
a. to change the color of text using setTimeOut()
b. to move an image across screen using setInterval()

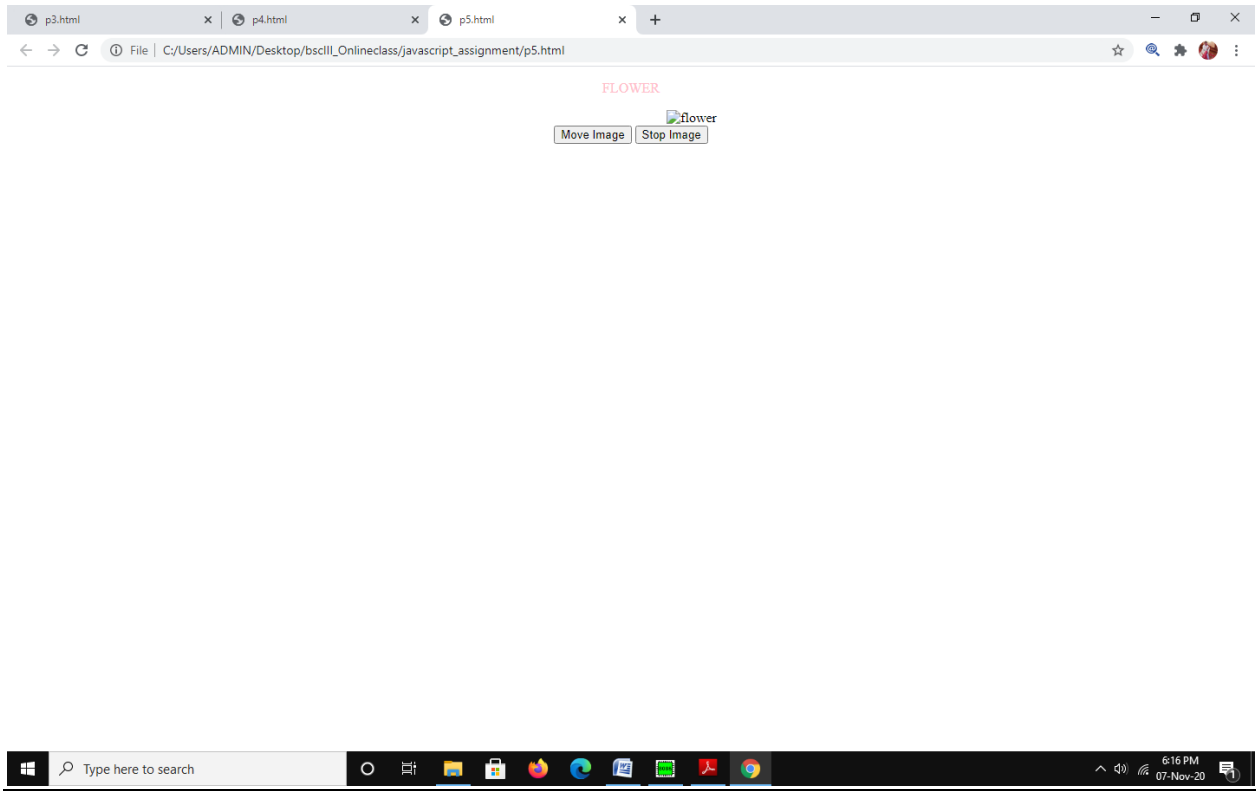
```
<!DOCTYPE html>

<html>


<body onLoad="f()"><center>


                    <p id="text">FLOWER</p>


                    <img src="images/flower.jpg" alt="flower" id="im" height="350"
width="350"><br>

                    <button onclick="g()" id="b1">Move Image</button>

                    <button onclick="h()" id="b2">Stop Image</button>
```

```
</center>

<script>
                    var l=0;var w;
                    function f()
                    {       //Display text in different colour after 3 seconds (3000 milliseconds)
        setTimeout(()=>{document.getElementById("text").style.color="pink";},3000);

                    }
                    function a()
                    {
                            var img = document.getElementById("im");
                            img.style.position="relative";
                            img.style.left=(l)+"px"; //HORIZONTAL MOVEMENT
                            l+=10;
                    }
                    function g()
                    {
                            w=setInterval("a()",500); //after every 0.5 second or 500 milliseconds
the position changes
                    }
                    function h()
                    {
                            clearInterval(w);
                    }
```

</script>

</body>

</html>



6. Implement the question no. 1 of hands on exercises of chapter 10 (page 190).

<!DOCTYPE html>

<html>

<head>

<script>

function validate(frm) {

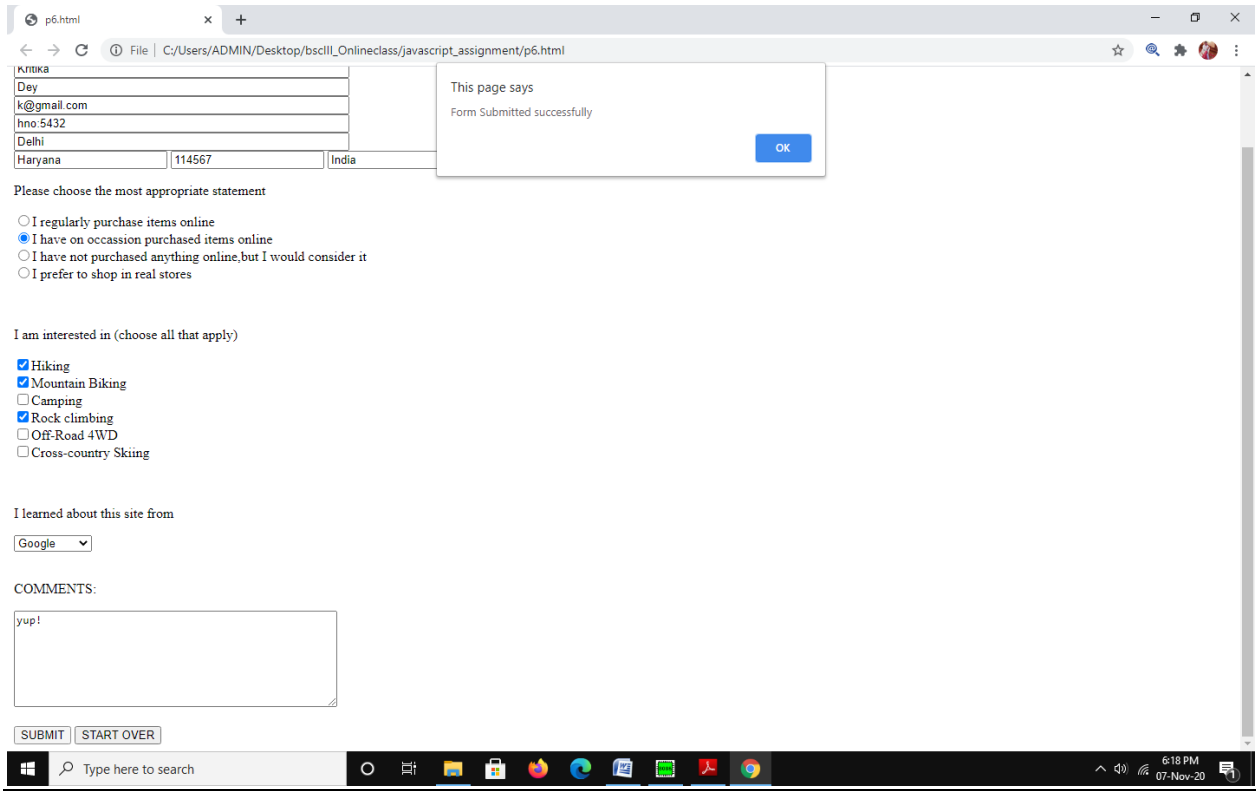
var elem=document.getElementById('frm').elements;

for(var i=0;i<elem.length;i++)

```
    if(document.frm.elements[i].value=="")
     {
     alert("This feild  cannot be empty");
      document.frm.elements[i].focus();
      break;
     }
else
{ alert("Form Submitted successfully");
  break;
}
}
</script>
</head>
<body>

<h1><u>INFONET SERVICES</u></h1>

<form name="frm" id="frm" >
<input type="text" class="button" id="fname" placeholder="Enter firstname" size="50px"
title="alphabet characters only" pattern="[A-Za-z]{1,}"/> <br/>

<input type="text" class="button" id="lname" placeholder="Enter lastname" size="50px" title="alphabet
characters only" pattern="[A-Za-z]{1,}"/> <br/>

<input type="text" class="button" id="email" placeholder="Enter email id" size="50px" title="must
contain @" pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,3}$"/><br/>
```

```
<input type="alphanumeric" class="button" id="address" placeholder="Enter address"
size="50px"/><br/>


<input type="text" class="button" id="city" placeholder="Enter city" size="50px"  pattern="[A-Za-
z]{1,}"/><br/>


<input type="text" class="button" id="state" placeholder="Enter state"   pattern="[A-Za-z]{1,}"/>


<input type="numeric" class="button" id="postalcode" placeholder="Enter postal code"
pattern=".{6,}"/>


<input type="text" class="button" id="country" placeholder="Enter country name" pattern="[A-Za-
z]{1,}"/>


<p>Please choose the most appropriate statement</p>

<input type="radio" id="1" name="opt1" >I regularly purchase items online<br>

<input type="radio" id="2" name="opt1">I have on occassion purchased items online<br>

<input type="radio" id="3" name="opt1">I have not purchased anything online,but I would consider it
<br>

<input type="radio" id="4" name="opt1">I prefer to shop in real stores<br><br><br>


<p>I am interested in (choose all that apply)</p>

<input type="checkbox" id="c1" name="c1" >Hiking<br>

<input type="checkbox" id="c2" name="c2" >Mountain Biking<br>

<input type="checkbox" id="c3" name="c3" >Camping<br>

<input type="checkbox" id="c4" name="c4" >Rock climbing<br>

<input type="checkbox" id="c5" name="c5" >Off-Road 4WD<br>
```

```
<input type="checkbox" id="c6" name="c6" >Cross-country Skiing<br><br><br>


<p>I learned about this site from</p>

<select id="subjects" class="button" name="subjects">

<option value="printad">Printads</option>

<option value="google">Google</option>

<option value="noticebard">NoticeBard</option>

<option value="gmail">Gmail</option>

<option value="fb">Facebook</option>

 </select><br><br>


<p>COMMENTS:</p><textarea rows="7" cols="50" class="button" name="comment">

Please type any comment here...</textarea>


 <br/> <br/><input type="submit" onClick="validate(frm)" value="SUBMIT">

 <input type="reset"  value="START OVER">

</form>




</body>

</html>
```

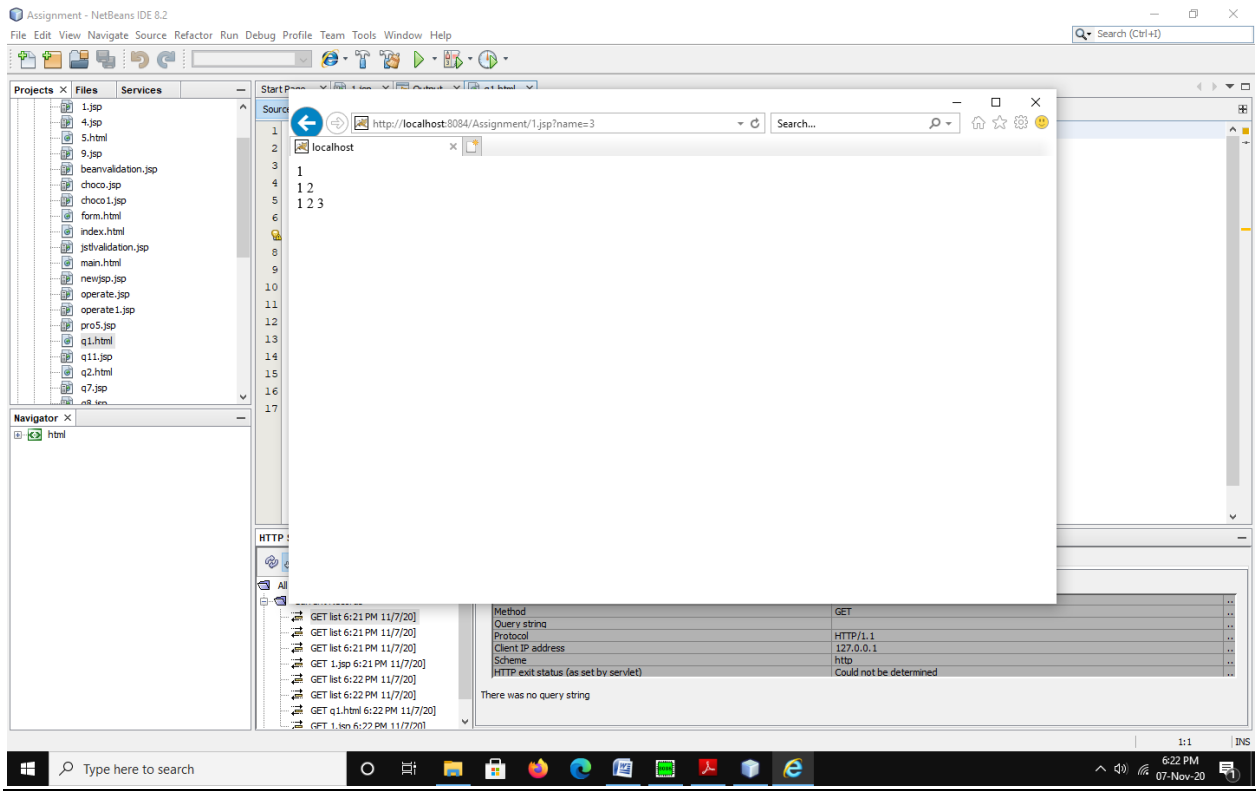


JSP

1. Display the pattern:
1
1 2
1 2 3
Take 'n' in a textbox from user. Display this pattern using
□ Scriptlets
□ <c:forEach> loop

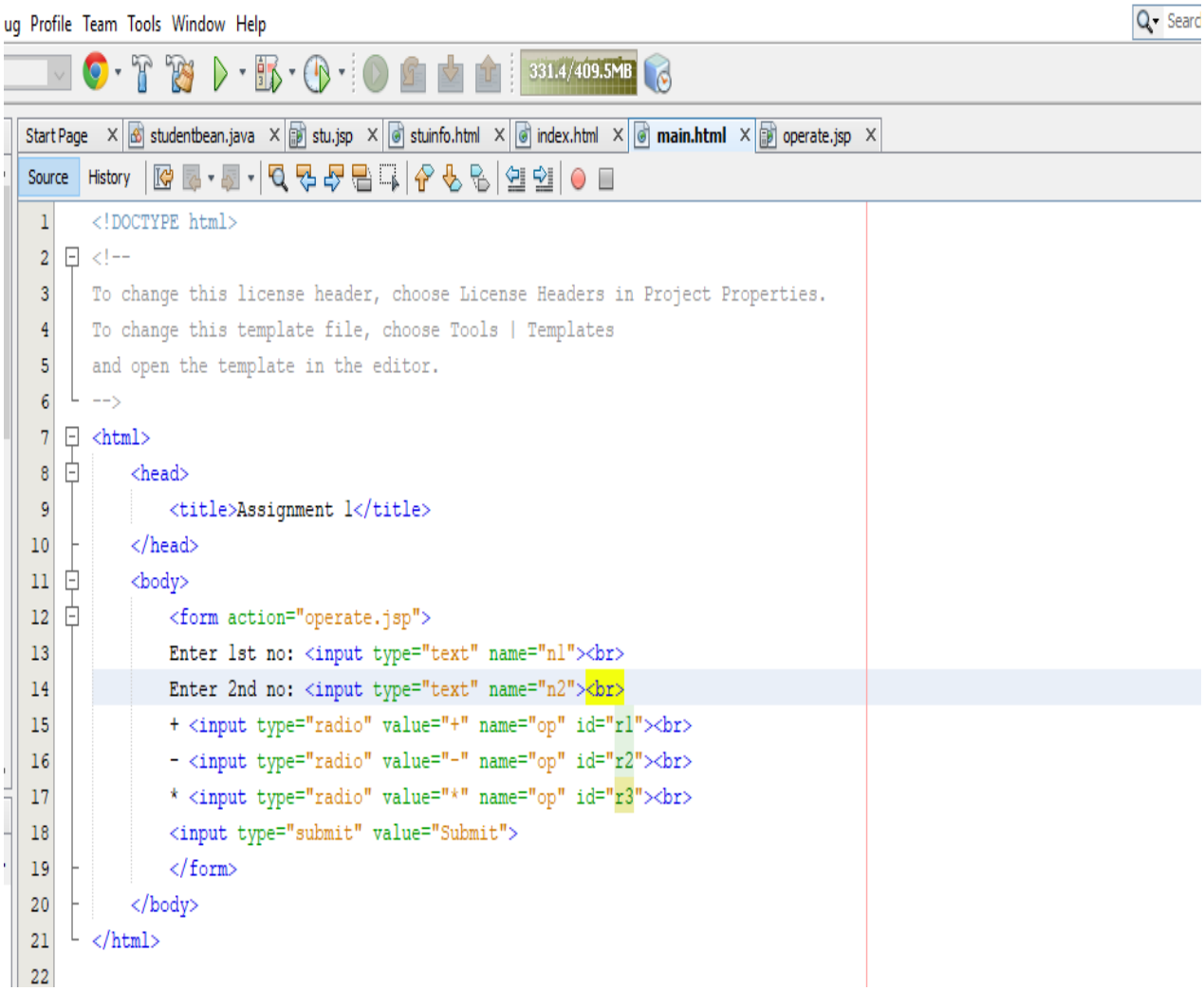
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>


  <body>

        <% int n=Integer.parseInt((request.getParameter("name")));

                int i,j;
```

```
                for(i=1;i<=n;i++)
                {
                        for(j=1;j<=i;j++)
            {out.print(j);
            out.print(" ");
            }
                        out.println("<br>");
                }
          %>


  </body>
</html>
```



2. Make two files as follows:
a. main.html: shows 2 text boxes and 3 radio buttons with values "addition", "subtraction" and "multiplication"
b. operate.jsp: depending on what the user selects perform the corresponding function (Give two implementations: using request.getParameter() and using expression language)

Main.html



Operate.jsp//using get paramenter

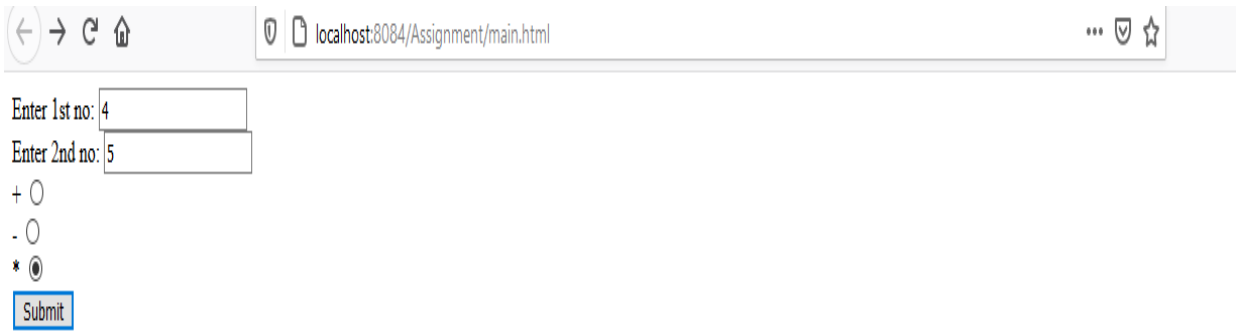
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <body>
      <%!   String a; String b; int x; int y; %>
<%
String operator = request.getParameter("op");

out.println("The result is");
if(operator.equals("+"))
{a=request.getParameter("n1");
 b=request.getParameter("n2");
 x=Integer.parseInt(a);
 y=Integer.parseInt(b);
 out.println(x+y);
}
else if(operator.equals("-"))
{
 a=request.getParameter("n1");
 b=request.getParameter("n2");
 x=Integer.parseInt(a);
 y=Integer.parseInt(b);
```

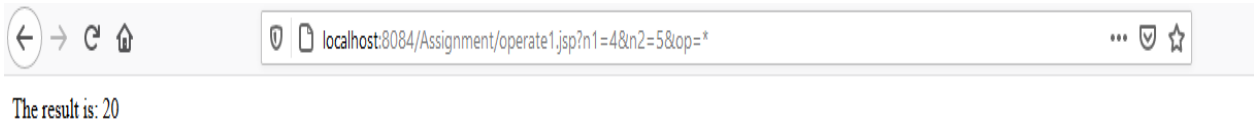
```
 out.println(x-y);
}
else if(operator.equals("*"))
{
 a=request.getParameter("n1");
 b=request.getParameter("n2");
 x=Integer.parseInt(a);
 y=Integer.parseInt(b);
 out.println(x*y);
}
else
out.println ("No operator selected");

%>
   </body>
</html>
```

The result is -1

Operator1.jsp //using expression language

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
   <body>
       The result is:

       <%  String operator=request.getParameter("op");
          if(operator.equals("+")){%>
      ${param.n1+param.n2}
      <% }else if(operator.equals("-")){%>
         ${param.n1-param.n2}
         <% } else if(operator.equals("*")) {%>
         ${param.n1*param.n2}
        <% } else
         out.println("No operator selected");%>


   </body>
</html>
```





3. Validate User input entered in a form. The input must include Name, DOB, Email ID, Lucky Number, Favorite food etc. (Refer Chapter 8)

v1.jsp

```
<%@ page contentType="text/html" %>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<html>

<head>

<title>User Info Entry Form</title>

</head>

<body bgcolor="white">

<jsp:useBean id="userInfo"
```

```
class="com.UserInfoBean">
<jsp:setProperty name="userInfo" property="*" />
</jsp:useBean>
<form action="v1.jsp" method="post">
<input type="hidden" name="submitted" value="true">
<table>
<c:if test="${param.submitted && userInfo.userNameValid == false}">
<tr><td></td>
<td colspan="2"><font color="red">
Please enter your Name
</font></td></tr>
</c:if>
<tr>
<td>Name:</td>
<td>
<input type="text" name="userName"
value="<c:out value="${userInfo.userName}"/>">
</td>
</tr>
<c:if test="${param.submitted && !userInfo.birthDateValid}">
<tr><td></td>
<td colspan="2"><font color="red">
Please enter a valid Birth Date
</font></td></tr>
</c:if>
```

```
<tr>
<td>Birth Date:</td>
<td>
<input type="text" name="birthDate"
value="<c:out value="${userInfo.birthDate}" />">
</td>
<td>(Use format yyyy-mm-dd)</td>
</tr>
<c:if test="${param.submitted && !userInfo.genderValid}">
<tr><td></td>
<td colspan="2"><font color="red">
Please select a valid Gender
</font></td></tr>
</c:if>

<tr>
<td>Gender:</td>
<td>
<c:choose>
<c:when test="${userInfo.gender == 'f'}">
<input type="radio" name="gender" value="m">
Male<br>
<input type="radio" name="gender" value="f" checked>
Female
</c:when>
```

```
<c:when test="${userInfo.gender == 'm'}">
<input type="radio" name="gender" value="m" checked>
Male<br>
<input type="radio" name="gender" value="f">
Female
</c:when>
<c:otherwise>
<input type="radio" name="gender" value="m">
Male<br>
<input type="radio" name="gender" value="f">
Female
</c:otherwise>
</c:choose>
</td>
</tr>
<c:if test="${param.submitted && !userInfo.foodValid}">
<tr><td></td>
<td colspan="2"><font color="red">
Please select only valid Favorite Foods
</font></td></tr>
</c:if>
<tr>
<td>Favorite Foods:</td>
<td>
<input type="checkbox" name="food" value="z"
```

```
${userInfo.pizzaSelected ? 'checked' : ''}>Pizza<br>
<input type="checkbox" name="food" value="p"
${userInfo.pastaSelected ? 'checked' : ''}>Pasta<br>
<input type="checkbox" name="food" value="c"
${userInfo.chineseSelected ? 'checked' : ''}>Chinese
</td>
</tr>
<tr>
<td colspan="3"> <input type="submit" value="Send Data"> </td>
</tr>
</table>
</form>
</body>
</html>
```

UserInfoBean.java

```
package com;

import java.io.Serializable;
public class UserInfoBean implements Serializable {
//properties
private String userName;
private String birthDate;
private String emailAddr;
private String gender;
```

```
private String luckyNumber;

private String[] food;

boolean birthDateValid = false;

boolean genderValid = false;

boolean userNameValid = false;

boolean foodValid = false;

boolean emailAddrValid = false;

boolean valid = false;

boolean luckyNumberValid = false;

boolean pizzaSelected=false;

boolean pastaSelected=false;

boolean chineseSelected=false;

public boolean isPizzaSelected() {

return pizzaSelected;

}

public void setPizzaSelected(boolean pizzaSelected) {

this.pizzaSelected = pizzaSelected;

}

public boolean isPastaSelected() {

return pastaSelected;

}

public void setPastaSelected(boolean pastaSelected) {

this.pastaSelected = pastaSelected;

}

public boolean isChineseSelected() {
```

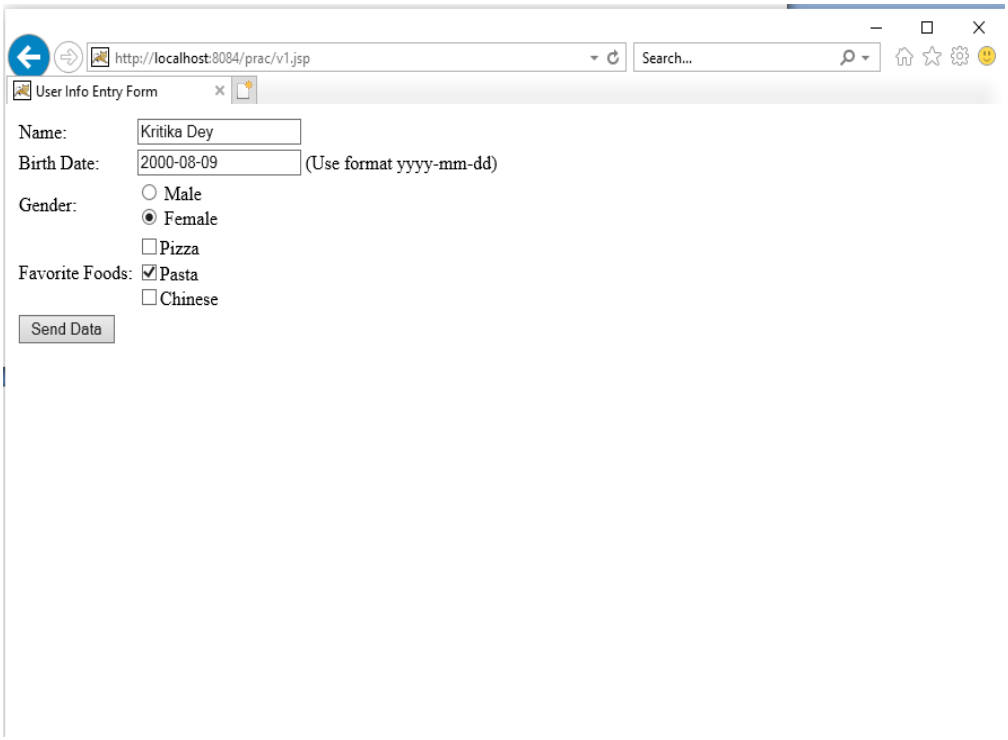
```
return chineseSelected;
}
public void setChineseSelected(boolean chineseSelected) {
this.chineseSelected = chineseSelected;
}
public String getUserName() {
return (userName == null ? "" : userName);
}
public void setUserName(String userName) {
this.userName = userName;
}
public String getBirthDate() {
return (birthDate == null ? "" : birthDate);
}
public void setBirthDate(String birthDate) {
this.birthDate = birthDate;
}
public String getEmailAddr() {
return emailAddr;
}
public void setEmailAddr(String emailAddr) {
this.emailAddr = emailAddr;
}
public String getGender() {
```

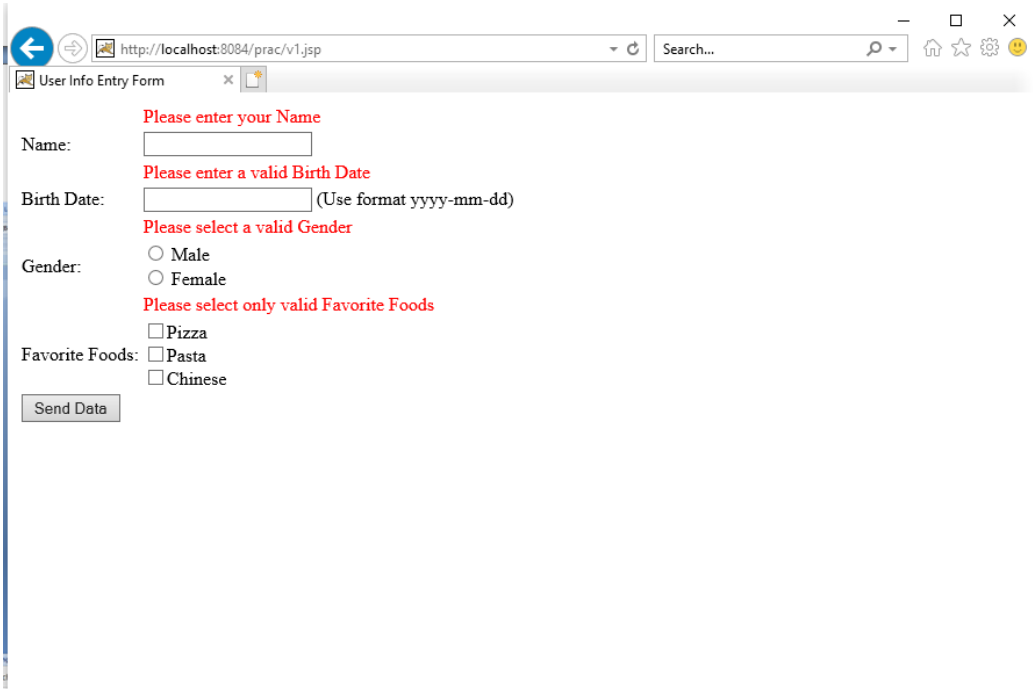
```
return gender;
}
public void setGender(String gender) {
this.gender = gender;
}
public String getLuckyNumber() {
return luckyNumber;
}
public void setLuckyNumber(String luckyNumber) {
this.luckyNumber = luckyNumber;
}
public String[] getFood() {
return food;

}
public void setFood(String[] food) {
this.food = food;
}
public boolean isBirthDateValid() {
if (birthDate != null )
{
birthDateValid = true;
}
return birthDateValid;
}
```

```
public boolean isGenderValid() {
if(gender!=null)
{
if((gender.equals("f")||gender.equals("m")))
{
genderValid = true;
}
}
return genderValid;
}
public boolean isUserNameValid() {
if (userName!=null)
{
userNameValid = true;
}
return userNameValid;
}
public boolean isFoodValid() {
if (food == null) {
foodValid = false;
return foodValid;
}
for(int i=0;i<getFood().length;i++)
{
if (food[i].equals("c"))
```

```
{
chineseSelected=true;


}
else if (food[i].equals("p"))
{
pastaSelected=true;
}
else //if (food[i].equals("z"))
{
pizzaSelected=true;
}
}
foodValid = pastaSelected || pizzaSelected || chineseSelected;
return foodValid;
}
public boolean isEmailAddrValid() {
if (emailAddr.contains("@")&&emailAddr!=null ||emailAddr!="")
{
emailAddrValid = true;
}
return emailAddrValid;
}
public boolean isLuckyNumberValid() {
if (Integer.parseInt(luckyNumber) >=1 && Integer.parseInt(luckyNumber) <=100)
```

```
{
luckyNumberValid = true;
}
return luckyNumberValid;
}
public boolean isValid() {
valid = isBirthDateValid() && isEmailAddrValid() &&
isFoodValid() && isLuckyNumberValid() &&
isGenderValid() && isUserNameValid();
return valid;
}

}
```


http://localhost:8084/prac/v1.jsp
Search...
User Info Entry Form
Name:
Kritika Dey
Birth Date:
2000-08-09
(Use format yyyy-mm-dd)
Gender:
Male
Female
Favorite Foods:
Pizza
Pasta
Chinese
Send Data


http://localhost:8084/prac/v1.jsp
Search...
User Info Entry Form
Please enter your Name
Name:
Please enter a valid Birth Date
Birth Date:
(Use format yyyy-mm-dd)
Please select a valid Gender
Gender:
Male
Female
Please select only valid Favorite Foods
Favorite Foods:
Pizza
Pasta
Chinese
Send Data

v2.jsp

```
<%--
Validating User Input Using JSTL Actions
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
<title>User Info Entry Form</title>
</head>
<body bgcolor="white">
<form action="v2.jsp" method="post">
<input type="hidden" name="submitted" value="true">
<table>
<c:if test="${param.submitted && empty param.userName}">
<tr><td></td>
<td colspan="2"><font color="red">
Please enter your Name
</font></td></tr>
</c:if>
<tr>
<td>Name:</td>
<td>
<input type="text" name="userName"
```

```
value="<c:out value="${param.userName}" />">
</td>
</tr>
<c:if test="${param.submitted && empty param.birthDate}">
<tr><td></td>
<td colspan="2"><font color="red">
Please enter your Birth Date
</font></td></tr>
</c:if>
<tr>
<td>Birth Date:</td>
<td>
<input type="text" name="birthDate"
value="<c:out value="${param.birthDate}" />">
</td>
<td>(Use format yyyy-mm-dd)</td>
</tr>

<c:if test="${param.submitted && empty param.emailAddr}">
<tr><td></td>
<td colspan="2"><font color="red">
Please enter your Email Address
</font></td></tr>
</c:if>
<tr>
```

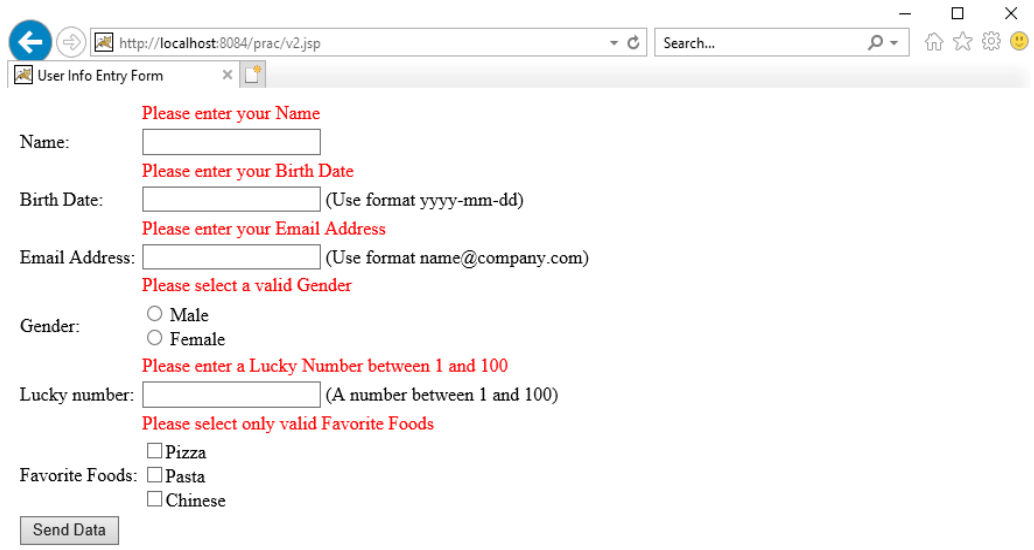
```
<td>Email Address:</td>
<td>
<input type="text" name="emailAddr"
value="<c:out value="${param.emailAddr}" />">
</td>
<td>(Use format name@company.com)</td>
</tr>
<c:if test="${param.submitted &&
param.gender != 'm' && param.gender != 'f'}">
<tr><td></td>
<td colspan="2"><font color="red">
Please select a valid Gender
</font></td></tr>
</c:if>
<tr>
<td>Gender:</td>
<td>
<c:choose>
<c:when test="${param.gender == 'f'}">
<input type="radio" name="gender" value="m">
Male<br>
<input type="radio" name="gender" value="f" checked>
Female
</c:when>
<c:when test="${param.gender == 'm'}">
```

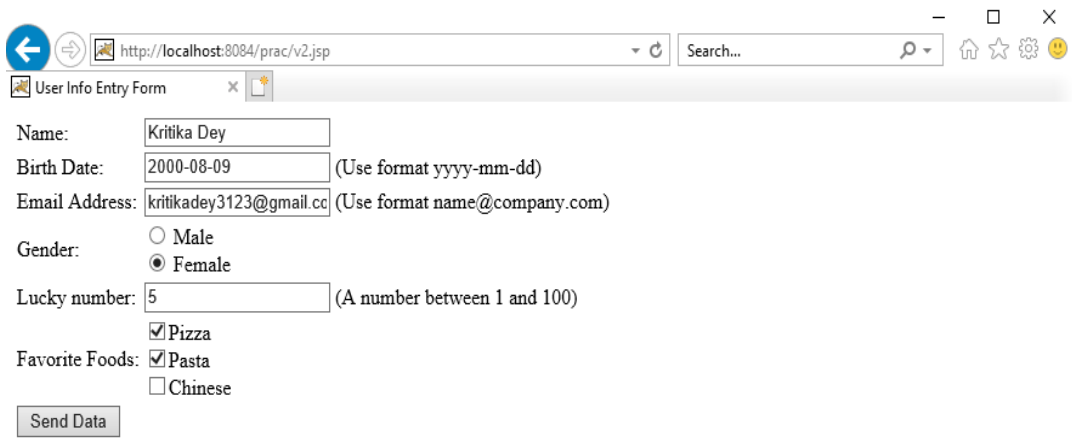
```
<input type="radio" name="gender" value="m" checked>
Male<br>
<input type="radio" name="gender" value="f">
Female
</c:when>
<c:otherwise>
<input type="radio" name="gender" value="m">
Male<br>
<input type="radio" name="gender" value="f">
Female


</c:otherwise>
</c:choose>
</td>
</tr>
<c:if test="${param.submitted &&
(param.luckyNumber < 1 || param.luckyNumber > 100)}">
<tr><td></td>
<td colspan="2"><font color="red">
Please enter a Lucky Number between 1 and 100
</font></td></tr>
</c:if>
<tr>
<td>Lucky number:</td>
<td>
```

```
<input type="text" name="luckyNumber"
value="<c:out value="${param.luckyNumber}" />">
</td>
<td>(A number between 1 and 100)</td>
</tr>
<c:forEach items="${paramValues.food}" var="current">
<c:choose>
<c:when test="${current == 'z'}">
<c:set var="pizzaSelected" value="true" />
<c:set var="foodValid" value="true" />
</c:when>
<c:when test="${current == 'p'}">
<c:set var="pastaSelected" value="true" />
<c:set var="foodValid" value="true" />
</c:when>
<c:when test="${current == 'c'}">
<c:set var="chineseSelected" value="true" />
<c:set var="foodValid" value="true" />
</c:when>
<c:otherwise>
<c:set var="foodValid" value="false" />
</c:otherwise>
</c:choose>
</c:forEach>
<c:if test="${param.submitted && !foodValid}">
```

```
<tr><td></td>
<td colspan="2"><font color="red">
Please select only valid Favorite Foods

</font></td></tr>
</c:if>
<tr>
<td>Favorite Foods:</td>
<td>
<input type="checkbox" name="food" value="z"
${pizzaSelected ? 'checked' : ''}>Pizza<br>
<input type="checkbox" name="food" value="p"
${pastaSelected ? 'checked' : ''}>Pasta<br>
<input type="checkbox" name="food" value="c"
${chineseSelected ? 'checked' : ''}>Chinese
</td>
</tr>
<tr>
<td colspan="3">
<input type="submit" value="Send Data">
</td>
</tr>
</table>
</form>
</body>
```

</html>


http://localhost:8084/prac/v2.jsp
Search...
User Info Entry Form
Please enter your Name
Name:
Please enter your Birth Date
Birth Date:
(Use format yyyy-mm-dd)
Please enter your Email Address
Email Address:
(Use format name@company.com)
Please select a valid Gender
Gender:
Male
Female
Please enter a Lucky Number between 1 and 100
Lucky number:
(A number between 1 and 100)
Please select only valid Favorite Foods
Favorite Foods:
Pizza
Pasta
Chinese
Send Data


http://localhost:8084/prac/v2.jsp
Search...
User Info Entry Form
Name:
Kritika Dey
Birth Date:
2000-08-09
(Use format yyyy-mm-dd)
Email Address:
kritikadey3123@gmail.cc
(Use format name@company.com)
Gender:
Male
Female
Lucky number:
5
(A number between 1 and 100)
Favorite Foods:
Pizza
Pasta
Chinese
Send Data

4. Display Good Morning <uname>, Good Afternoon <uname> or Good Evening <uname> based on the current time of the day.

```
package den;



import java.beans.*;

import java.io.Serializable;

import javax.servlet.jsp.tagext.*;

import javax.servlet.jsp.*;

import java.io.*;

import java.util.*;


public class q4 extends SimpleTagSupport  {


  String name1;

        public void setName1(String name1)

        {

    this.name1 = name1;

  }


  public String getName1() {

    return name1;

  }


        int hr, min, sec;

        public void doTag() throws JspException, IOException
```

```
   {
JspWriter out = getJspContext().getOut();
      out.print("Time : ");
      hr = Calendar.getInstance().get(Calendar.HOUR_OF_DAY);
      min = Calendar.getInstance().get(Calendar.MINUTE);
      sec = Calendar.getInstance().get(Calendar.SECOND);
      out.print(hr+"hr-");
      out.print(min+"min-");
      out.print(sec+"sec<br>");

                  if(hr>=5 && hr<= 11 )
      {
         out.println(name1+"Good Morning ");
      }
      else if(hr>=12 && hr<= 4)
      {
         out.println(name1+"Good Afternoon ");
      }
      else
      {
         out.println(name1+" Good Evening ");
      }
   }
}
```

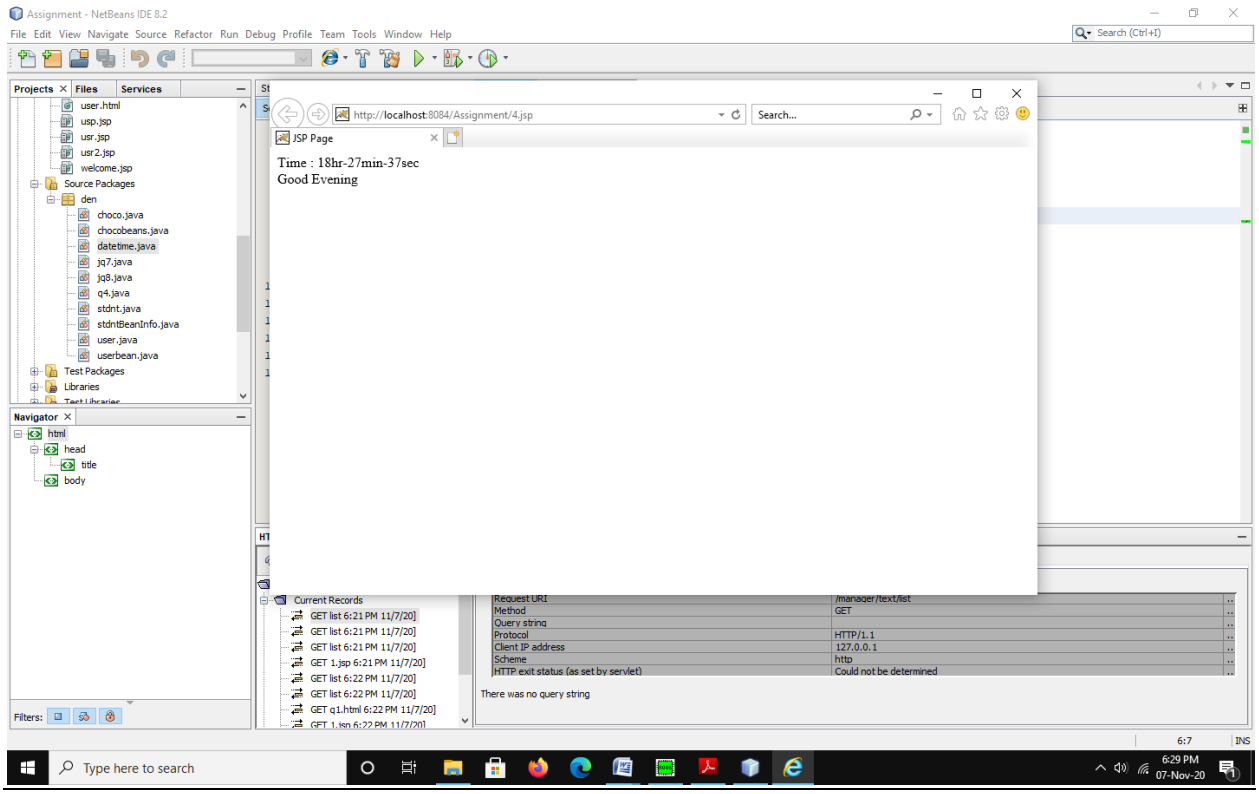
jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@taglib uri="/WEB-INF/tlds/gret.tld" prefix="k" %>



<html>
  <head>
    <title>JSP Page</title>
  </head>
  <body>
    <k:today name1="${param.name1}">
    </k:today>
  </body>
</html>
```

Tld file

```
<?xml version="1.0" encoding="UTF-8"?>

<taglib version="2.1" 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-
jsptaglibrary_2_1.xsd">
 <tlib-version>1.0</tlib-version>
 <short-name>gret</short-name>
 <uri>gret</uri>
 <tag>
    <name>today</name>
    <tag-class>den.q4</tag-class>
```

```
    <body-content>scriptless</body-content>
    <attribute>
    <name>name1</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    </attribute>
  </tag>

</taglib>
```

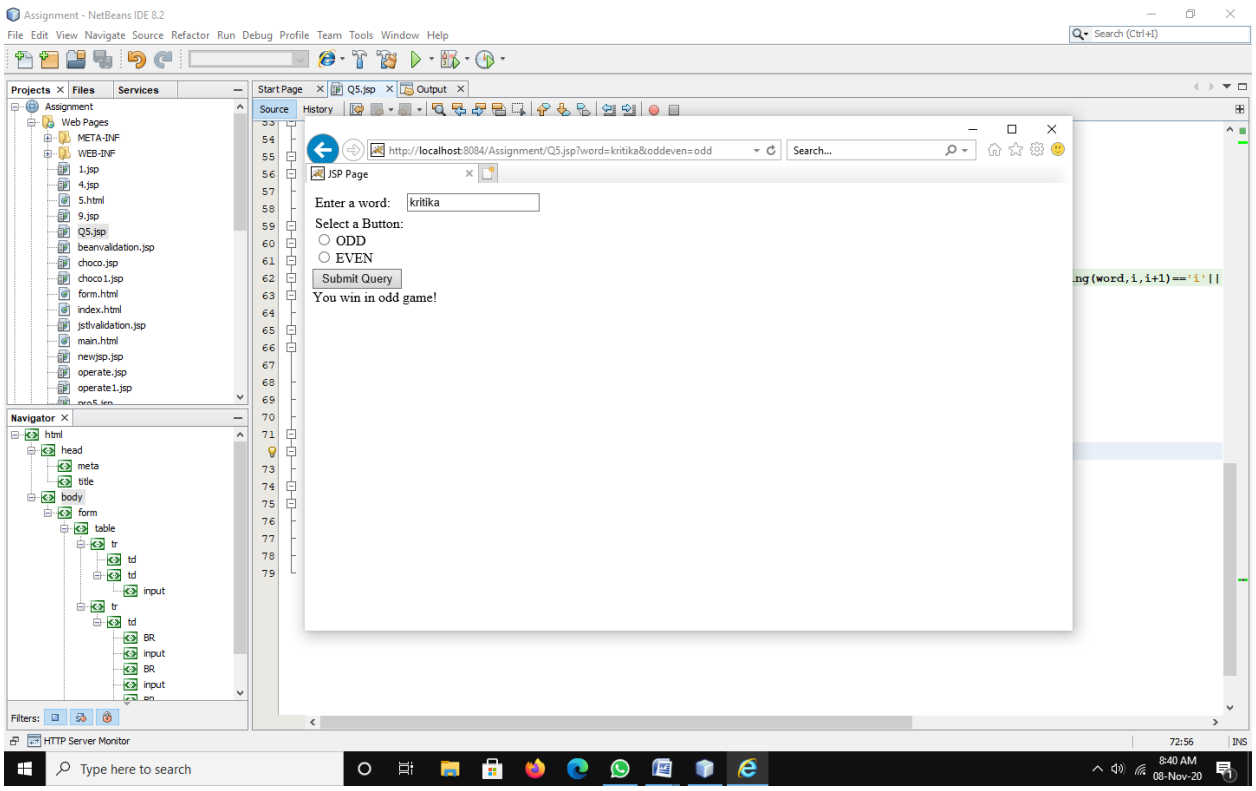


5. Let the user enter a word a in a textbox and let her/him select one of even or odd radio buttons. If she/he selects odd, check the odd positions in the word entered, if they all contain vowels, then display the message 'You win', else display 'You lose'. Similarly, if the user selects even, check for vowels in all even positions in the word entered. Use jstl's 'fn' library.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

  <head>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>JSP Page</title>

  </head>

  <body>



    <form action="Q5.jsp" method="get">



    <table>

      <tr>

        <td>Enter a word:</td><td><input type="text" name="word" value = "${param.word!=null ?
param.word: ''}"></td>

      </tr>

      <tr>

        <td>

          Select a Button:<BR/>

          <input type="radio" name="oddeven" value="odd"/> ODD <BR/>

          <input type="radio" name="oddeven" value="even"/> EVEN <BR/>

        </td>



    </table>
```

```
                        <input type="submit"/><br/>
    </form>

    <%@taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
    <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

    <c:set var="winflag" value="t"/>
    <c:set var="word" value="${param.word}"/>

    <c:if test = "${param.oddeven == 'odd'}">
       <c:forEach var="i" begin="0" end="${param.word.length()-2}" step="2">
          <c:choose>
             <c:when test="${fn:substring(word,i,i+1)=='a' || fn:substring(word,i,i+1)=='e'||
fn:substring(word,i,i+1)=='i'|| fn:substring(word,i,i+1)=='o'|| fn:substring(word,i,i+1)=='u'}">
                <c:set var = "winflag" value = 't'/>
             </c:when>
             <c:otherwise>
                <c:set var = "winflag" value = 'f'/>
                <c:set var = "i" value = "${param.word.length()}"/>
             </c:otherwise>
          </c:choose>

       </c:forEach>
       <c:if test="${winflag == 't'}">
          <c:out value="You win in odd game! "/>
       </c:if>
```

```
        <c:if test="${winflag == 'f'}">
            <c:out value="You lose in odd game! "/>
        </c:if>
    </c:if>
    <c:if test = "${param.oddeven == 'even'}">
        <c:forEach var="i" begin="1" end="${param.word.length()-1}" step="2">
            <c:choose>
                <c:when test="${fn:substring(word,i,i+1)=='a' || fn:substring(word,i,i+1)=='e'||
fn:substring(word,i,i+1)=='i'|| fn:substring(word,i,i+1)=='o'|| fn:substring(word,i,i+1)=='u'}">
                    <c:set var = "winflag" value = 't'/>
                </c:when>
                <c:otherwise>
                    <c:set var = "winflag" value = 'f'/>
                    <c:set var = "i" value = "${param.word.length()}"/>
                </c:otherwise>
            </c:choose>
        </c:forEach>
        <c:if test="${winflag == 't'}">
            <c:out value="You win in even game! "/>
        </c:if>
        <c:if test="${winflag == 'f'}">
            <c:out value="You lose in even game! "/>
        </c:if>
    </c:if>
  </body>
</html>
```



6. Create your custom library which contains two tags: <hello>, <choco>.
Usage of the tags:
□ <hello name=”Ajay”>: Output should be Hello Ajay. It contains a mandatory attribute ‘name’ which can accept Dynamic value.

□ <choco texture=”Chewy”>: Output should be FiveStar, BarOne.

<choco texture=”Crunchy”>: Output should be Munch. KitKat.
That means the mandatory attribute must accept a value, and based on the attributes value, it should give output. You must use a bean ChocoBean for this purpose.

q6_tag.tlds

```
<?xml version="1.0" encoding="UTF-8"?>

<taglib version="2.1" 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-jsptaglibrary_2_1.xsd">

 <tlib-version>1.0</tlib-version>

 <jsp-version>2.0</jsp-version>

 <short-name>q6_tag</short-name>
```

```
<uri>/WEB-INF/tlds/q6_tag</uri>


<tag>
   <name>hello</name>
   <tag-class>den.choco</tag-class>
   <body-content>scriptless</body-content>
   <attribute>
      <name>name</name>
      <required>true</required>
      <type>String</type>
      <rtexprvalue>true</rtexprvalue>
   </attribute>
</tag>
<tag>
   <name>choco</name>
   <tag-class>den.chocobeans</tag-class>
   <body-content>scriptless</body-content>
   <attribute>
      <name>texture</name>
      <required>true</required>
      <type>String</type>
      <rtexprvalue>true</rtexprvalue>
   </attribute>
</tag>
```

```
</taglib>
```

Choco.java

```
package den;


import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
import java.io.*;


public class choco extends SimpleTagSupport
{
   String name;


   public void setName(String name) {
      this.name = name;
   }
   public void doTag()throws IOException,JspException
   {
      JspWriter out=getJspContext().getOut();
      if(name!=null)
         out.println("Hello! "+name+"<br>");
   }


}
```

Chocobeans.java

```
package den;
```

```
import java.io.*;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;



public class chocobeans extends SimpleTagSupport implements Serializable  {

 String texture;

public void setTexture(String texture) {
       this.texture = texture;
}
   public void doTag()throws IOException,JspException{
JspWriter out = getJspContext().getOut();
if (texture != null) {

if (texture.equalsIgnoreCase("Chrunchy"))
out.println("Munch, Kitkat!");
if (texture.equalsIgnoreCase("Chewy"))
out.println("Fivestar,Barone!");

     }
if (texture == null)
out.println("Attribute value not set!<br/>");
```

```
  }
}
```

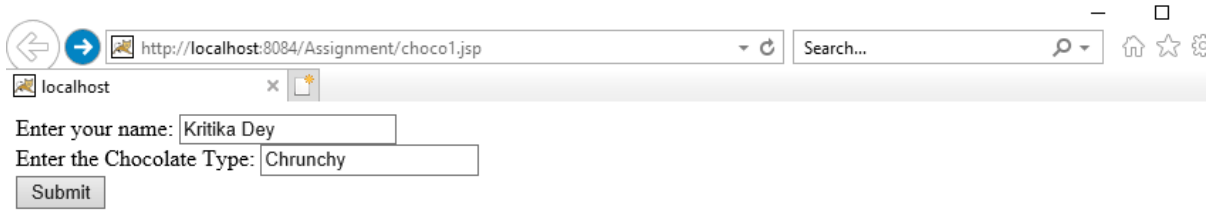
Choco1.jsp

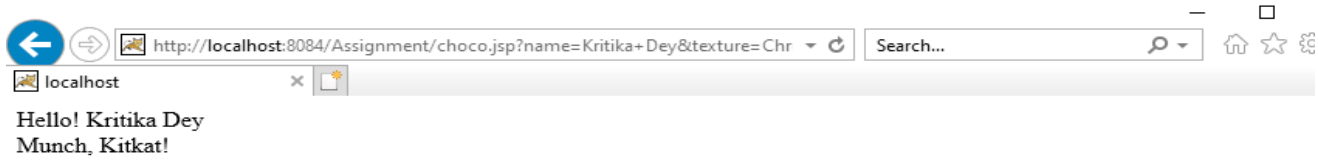
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <body>
      <form action="choco.jsp" >
         Enter your name: <input type="text" name="name"><br>
         Enter the Chocolate Type: <input type="text" name="texture"><br>
         <input type="submit" value="Submit">
      </form>

   </body>
</html>
```

Choco.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="coco" uri="/WEB-INF/tlds/q6_tag.tld" %>

<html>
```

```
<coco:hello name="${param.name}"> </coco:hello>

<coco:choco texture="${param.texture}"> </coco:choco>


<br><br>




</html>
```





7. Create a custom tag “substring” with 3 mandatory attributes “input”, “start”, “end” which will do substring operation on given input

Tlds file:

```
<?xml version="1.0" encoding="UTF-8"?>

<taglib version="2.1" 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-jsptaglibrary_2_1.xsd">

 <tlib-version>1.0</tlib-version>

 <jsp-version>2.0</jsp-version>

 <short-name>q7_tag</short-name>

 <uri>/WEB-INF/tlds/q7_tag</uri>

 <tag>

  <name>substring</name>

  <tag-class>den.jq7</tag-class>

  <body-content>scriptless</body-content>

  <attribute>

   <name>input</name>

   <required>true</required>

  </attribute>

  <attribute>

   <name>start</name>

   <required>true</required>

  </attribute>

  <attribute>

   <name>end</name>
```

```
    <required>true</required>
   </attribute>
  </tag>
</taglib>
```

Java file:

```
package den;

import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
import java.io.*;

public class jq7 extends SimpleTagSupport {

   String input;
   int start,end;

   public String getInput() {
      return input;
   }

   public void setInput(String input) {
      this.input = input;
   }
```

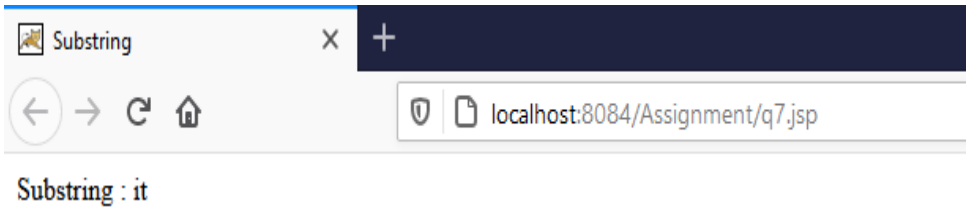
```
public int getStart() {
   return start;
}

public void setStart(int start) {
   this.start = start;
}

public int getEnd() {
   return end;
}

public void setEnd(int end) {
   this.end = end;
}



public void doTag() throws IOException,JspException
{
   if(input!=null)
   {
   JspWriter out=getJspContext().getOut();
   StringBuffer br=new StringBuffer(input);
   String str=br.substring(start,end);
```

```
        out.println("Substring : "+str);
        }

    }

}
```

Jsp file:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="sub" uri="/WEB-INF/tlds/q7_tag.tld"%>
<!DOCTYPE html>
<html>
    <head>
        <title>Substring</title>
    </head>

        <sub:substring input="Kritika" start="2" end="4" />

</html>
```

Output:



8. Create a custom tag “reverse” with a mandatory attribute “input” to reverse a string.
Tlds file:

```
<?xml version="1.0" encoding="UTF-8"?>

<taglib version="2.1" 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-
jsptaglibrary_2_1.xsd">

 <tlib-version>1.0</tlib-version>

 <jsp-version>2.0</jsp-version>

 <short-name>q8_tag</short-name>

 <uri>/WEB-INF/tlds/q8_tag</uri>

  <tag>

    <name>reverse</name>

    <tag-class>den.jq8</tag-class>

    <body-content>scriptless</body-content>

   <attribute>

    <name>input</name>

    <required>true</required>

   </attribute>
```

```
 </tag>
</taglib>
```

Java file:

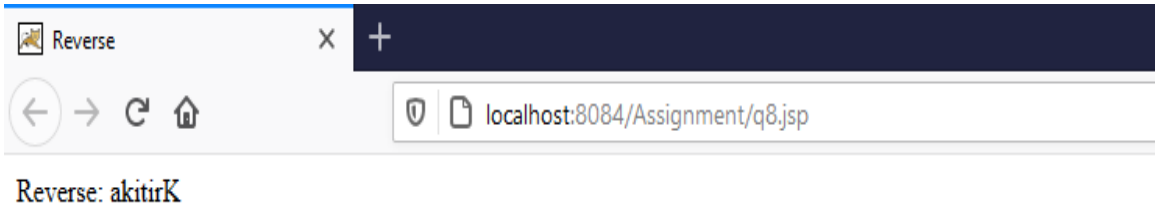
```
package den;

import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
import java.io.*;


public class jq8 extends SimpleTagSupport {
  String input;

    public String getInput() {
        return input;
    }

    public void setInput(String input) {
        this.input = input;
```

```
  }
 public void doTag() throws IOException,JspException
 {
    if(input!=null)
    { JspWriter out=getJspContext().getOut();
    StringBuffer br=new StringBuffer(input);
    StringBuffer rev=br.reverse();
    out.println("Reverse: "+rev);
    }

 }
}
```

Jsp file:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix= "res" uri= "/WEB-INF/tlds/q8_tag.tld"%>
<!DOCTYPE html>
<html>
  <head>
     <title>Reverse</title>
  </head>
      <res:reverse input="Kritika"/>
</html>
```

Output:



9. Create a custom tag "today" that displays today's date and time

```
package den;


import java.beans.*;

import java.io.Serializable;

import javax.servlet.jsp.tagext.*;

import javax.servlet.jsp.*;

import java.io.*;

import java.util.*;

public class datetime  extends SimpleTagSupport {

 public void doTag() throws JspException, IOException

  {

     JspWriter out = getJspContext().getOut();

     try

     {

        out.print("Date : ");

        out.print(Calendar.getInstance().get(Calendar.DAY_OF_MONTH)+"/");
```

```
        out.print(Calendar.getInstance().get(Calendar.MONTH)+"/");
        out.print(Calendar.getInstance().get(Calendar.YEAR)+"<br/>");
        out.print("Time : ");
        out.print(Calendar.getInstance().get(Calendar.HOUR_OF_DAY)+"hr-");
        out.print(Calendar.getInstance().get(Calendar.MINUTE)+"min-");
        out.print(Calendar.getInstance().get(Calendar.SECOND)+"sec");
      }
      catch(Exception e)
      {
        System.out.println(e);
      }
   }

}
```

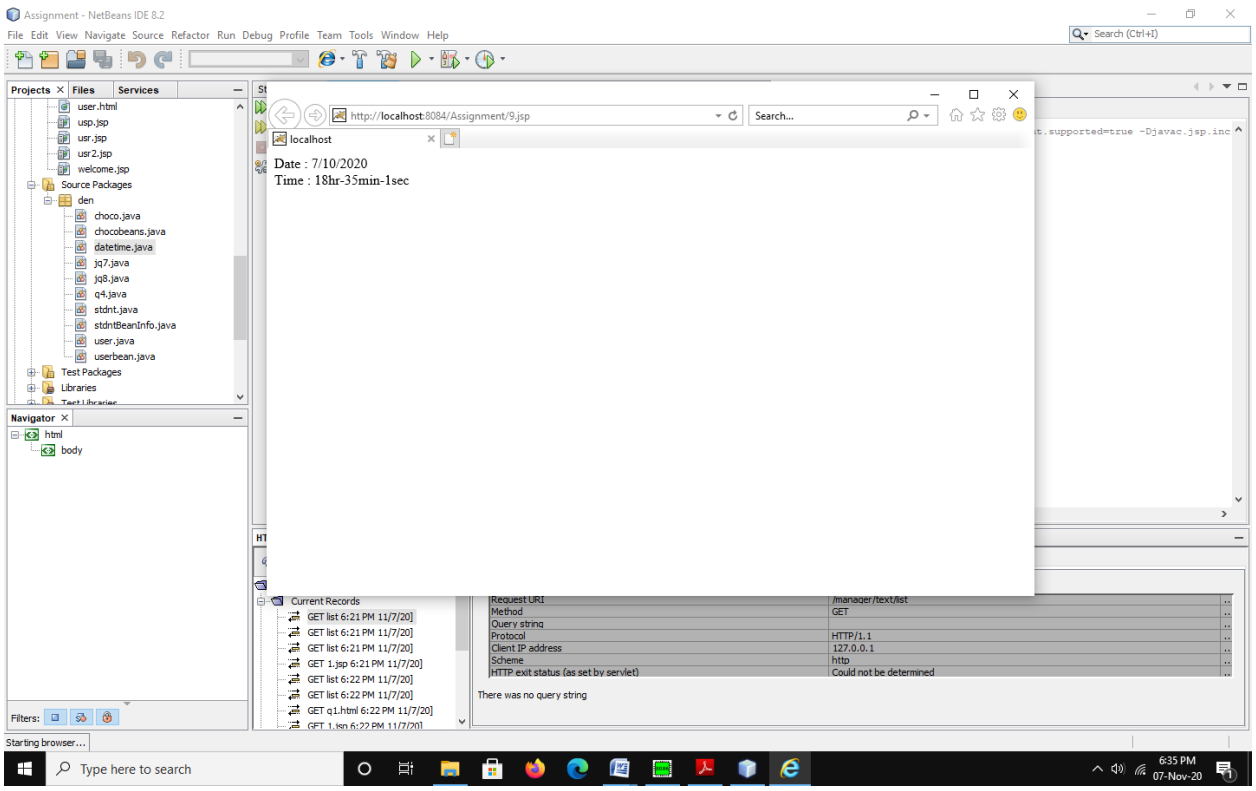
Tlds

```
<?xml version="1.0" encoding="UTF-8"?>

<taglib version="2.1" 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-
jsptaglibrary_2_1.xsd">

  <tlib-version>1.0</tlib-version>

  <short-name>date</short-name>

  <uri>/WEB-INF/tlds/date</uri>

 <tag>

      <name> xyz </name>

      <tag-class>den.datetime </tag-class>
```

```
    <body-content>scriptless</body-content>
  </tag>

</taglib>
```

Jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="/WEB-INF/tlds/date" prefix="x" %>
<!DOCTYPE html>
<html>

  <body>
    <x:xyz/>
  </body>
</html>
```



10. Ask a user's name and age on a HTML form. Then display Hello <uname> on a JSP. On the same page ask the product the user would like to buy. Then redirect to another JSP which would display: Hello <uname>, You have ordered <product>. (Use Session Scope Variable using setTag)

Java File

```
package den;



public class user implements java.io.Serializable {



    String name;

    String age;

    String product;
```

```
  public user()
  {
     name=age=product=null;

  }
   public void setName(String name)
  { this.name=name; }

  public String getName()
  { return name; }

  public void setAge(String age)
  { this.age=age; }

  public String getAge()
  { return age;}

  public void setProduct(String product)
  { this.product= product; }

  public String getProduct()
  { return product;}

}
```

Jsp file1

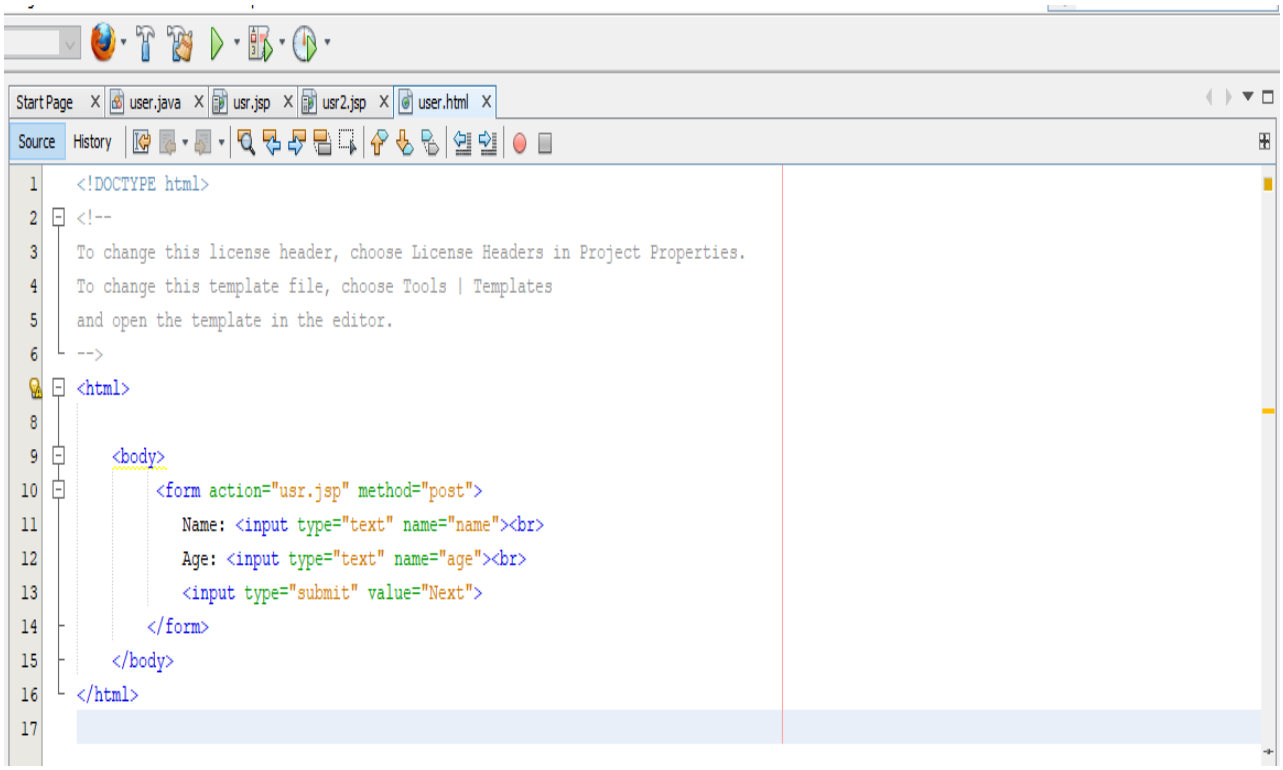
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>

  <body>



    <form action="usr2.jsp" method="post">
       <%
       String name=request.getParameter("name");
       out.println("Hello! "+name);
       session.setAttribute("nm",name); %><br>
       Which product you want?<br>
    Product: <input type="text" name="product"><br>
    <input type="submit" value="Next" >


    </form>



  </body>
</html>
```

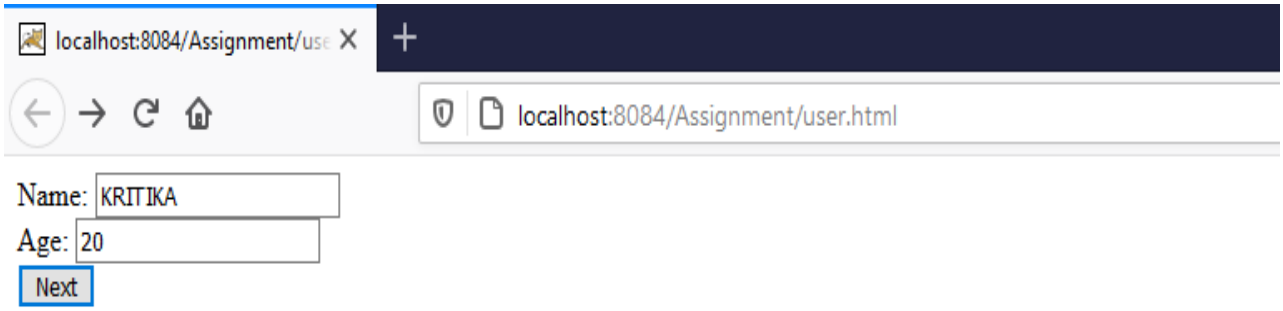
Jsp file2

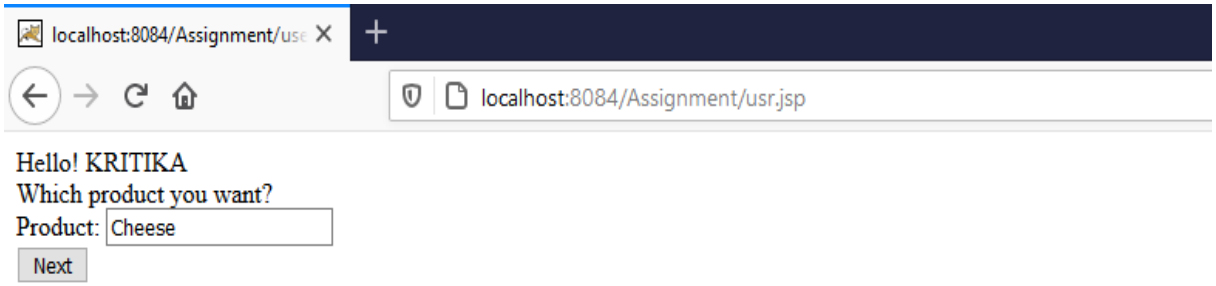
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

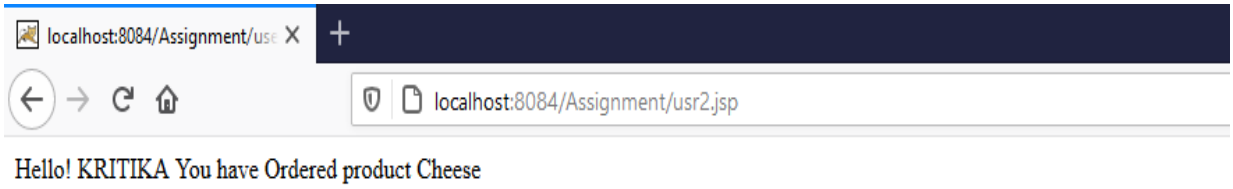
```
<!DOCTYPE html>

<html>

  <body>

     <%

        String a=(String)session.getAttribute("nm");

     out.println("Hello! "+a);

     String pro=request.getParameter("product");

       out.println(" You have Ordered product "+pro);

     %>


  </body>

</html>
```

Html file




localhost:8084/Assignment/use
localhost:8084/Assignment/user.html
Name: KRITIKA
Age: 20
Next


localhost:8084/Assignment/use
localhost:8084/Assignment/usr.jsp
Hello! KRITIKA
Which product you want?
Product: Cheese
Next


localhost:8084/Assignment/use
localhost:8084/Assignment/usr2.jsp
Hello! KRITIKA You have Ordered product Cheese