EXPERIMENT 01:

AIM: Write a JDBC application to implement DDL and DML commands.

Execution steps:

File → New Project → java → java Application → Next→ Project Name → Finish

Goto → Services → Start DataBase → Right Click And Start Server.

Then GO to Source Packages → Add Library → select Java DB Driver.

And Right Click the Source File → click RUN FILE.

SOURCE CODE:

```
package jdbbcex;
import java.sql.*;
import java.util.Scanner;
public class Jdbc {
   public static void main(String[] args) {
      try {
         Class.forName("org.apache.derby.jdbc.ClientDriver");
         Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm474","navya","navya");
         Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
         Scanner sc = new Scanner(System.in);
         ResultSet rs;
         int ch, n, rno, s1, s2, s3, s4, s5, s6, tot;
         String q, reg, sname, g;
         while (true) {
            System.out.println("-----MENU-----");
            System.out.println("0.exit");
            System.out.println("1.create table");
            System.out.println("2.insert data");
```

```
System.out.println("3.adding new columns");
System.out.println("4.updating new columns");
System.out.println("5.Display data(before updating)");
System.out.println("6.Display data(after updating)");
System.out.println("7.deleting data");
System.out.println("8.dropping data");
System.out.println("Enter your choice:");
ch = sc.nextInt();
switch (ch) {
   case 0:
      con.close();
      System.exit(0);
   case 1:
      try {
         q = "create table student_marks1 (rno integer, rgd varchar(10), sname
varchar(20), s1 integer, s2 integer, s3 integer, s4 integer, s5 integer, s6 integer)";
         stmt.executeUpdate(q);
         System.out.println("Table created successfully");
      } catch (SQLException e) {
         System.out.println("Table already exists");
      }
      break;
   case 2:
      System.out.println("Enter number of students:");
      n = sc.nextInt();
      for (int i = 1; i <= n; i++) {
         System.out.println("Enter student " + i + " details:");
         System.out.println("Rno:");
         rno = sc.nextInt();
```

```
                System.out.println("Regdno:");
                reg = sc.next();
                System.out.println("Sname:");
                sname = sc.next();
                System.out.println("s1:");
                s1 = sc.nextInt();
                System.out.println("s2:");
                s2 = sc.nextInt();
                System.out.println("s3:");
                s3 = sc.nextInt();
                System.out.println("s4:");
                s4 = sc.nextInt();
                System.out.println("s5:");
                s5 = sc.nextInt();
                System.out.println("s6:");
                s6 = sc.nextInt();
                q = "insert into student_marks1 values(" + rno + ",'" + reg + "','" + sname +
"'," + s1 + "," + s2 + "," + s3 + "," + s4 + "," + s5 + "," + s6 + ")";
                stmt.executeUpdate(q);
            }
            System.out.println(n + " records are inserted");
            break;
        case 3:
            try {
                q = "alter table student_marks1 add column tot integer";
                stmt.executeUpdate(q);
                q = "alter table student_marks1 add column grade varchar(20)";
                stmt.executeUpdate(q);
                System.out.println("New columns are added");
```

```
                } catch (SQLException e) {
                    System.out.println("Already exists");
                }
                break;
            case 4:
                q = "select * from student_marks1";
                rs = stmt.executeQuery(q);
                while (rs.next()) {
                    tot = rs.getInt(4) + rs.getInt(5) + rs.getInt(6) + rs.getInt(7) + rs.getInt(8) +
rs.getInt(9);
                    if (tot >= 560) g = "A";
                    else if (tot >= 500) g = "B";
                    else if (tot >= 430) g = "C";
                    else if (tot >= 370) g = "D";
                    else if (tot >= 300) g = "E";
                    else g = "fail";
                    rs.updateInt(10, tot);
                    rs.updateString(11, g);
                    rs.updateRow();
                }
                System.out.println("New Columns are updated");
                break;
            case 5:
                q = "select * from student_marks1";
                rs = stmt.executeQuery(q);
                System.out.println("The student details are:");
                while (rs.next()) {
                    System.out.println("Rno:" + rs.getString(1));
                    System.out.println("Regdno:" + rs.getString(2));
```

```
            System.out.println("sname:" + rs.getString(3));
            System.out.println("s1:" + rs.getString(4));
            System.out.println("s2:" + rs.getString(5));
            System.out.println("s3:" + rs.getString(6));
            System.out.println("s4:" + rs.getString(7));
            System.out.println("s5:" + rs.getString(8));
            System.out.println("s6:" + rs.getString(9));
        }
        break;
    case 6:
        q = "select * from student_marks1";
        rs = stmt.executeQuery(q); // Re-fetch the ResultSet
        System.out.println("The student details are:");
        while (rs.next()) {
            System.out.println("/nRno:" + rs.getString(1));
            System.out.println("Regdno:" + rs.getString(2));
            System.out.println("sname:" + rs.getString(3));
            System.out.println("s1:" + rs.getString(4));
            System.out.println("s2:" + rs.getString(5));
            System.out.println("s3:" + rs.getString(6));
            System.out.println("s4:" + rs.getString(7));
            System.out.println("s5:" + rs.getString(8));
            System.out.println("s6:" + rs.getString(9));
            System.out.println("Total:" + rs.getString(10));
            System.out.println("Grade:" + rs.getString(11));
        }
        break;
    case 7:
        q = "delete from student_marks1";
```

```
                int x = stmt.executeUpdate(q);
                if (x != 0) System.out.println("Table data deleted");
                else System.out.println("No data to delete");
                break;
            case 8:
                try {
                    q = "drop table student_marks1";
                    stmt.executeUpdate(q);
                    System.out.println("Table deleted successfully");
                } catch (SQLException e) {
                    System.out.println("Table does not exist");
                }
                break;
            default:
                System.out.println("Invalid choice");
                break;
          }
        }
      } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
      }
    }
}
```

OUTPUT:

run: **-----MENU-----** 0. exit 1. create table 2. insert data 3. adding new columns 4. updating new columns 5. Display data(before updating)	Enter student 1 details: Rno: 100 Regdno: 474 Sname: navya s1:

6. Display data(after updating)
7. deleting data
8. dropping data

Enter your choice:

1

Table created successfully

-----MENU-----

0. exit
1. create table
2. insert data
3. adding new columns
4. updating new columns
5. Display data(before updating)
6. Display data(after updating)
7. deleting data
8. dropping data

Enter your choice:

2

Enter number of students:

1

6. Display data(after updating)
7. deleting data
8. dropping data

Enter your choice:

3

New columns are added

-----MENU-----

0. exit
1. create table
2. insert data
3. adding new columns
4. updating new columns
5. Display data(before updating)
6. Display data(after updating)
7. deleting data
8. dropping data

Enter your choice:

4

New Columns are updated

-----MENU-----

0. exit
1. create table
2. insert data
3. adding new columns

90
s2:
89
s3:
78
s4:
89
s5:
90
s6:
90
1 records are inserted

-----MENU-----

0. exit
1. create table
2. insert data
3. adding new columns
4. updating new columns
5. Display data(before updating)

6. Display data(after updating)
7. deleting data
8. dropping data

Enter your choice:

6

The student details are:
/nRno:100
Regdno:474
sname:navya
s1:90
s2:89
s3:78
s4:89
s5:90
s6:90
Total:526
Grade:B

-----MENU-----

0. exit
1. create table
2. insert data
3. adding new columns
4. updating new columns
5. Display data(before updating)
6. Display data(after updating)

4. updating new columns
5. Display data(before updating)
6. Display data(after updating)
7. deleting data
8. dropping data

Enter your choice:

5

The student details are:
Rno:99
Regdno:473
name: navya
s1:90
s2:89
s3:78
s4:89
s5:75
s6:90

-----MENU-----

0. exit
1. create table
2. insert data
3. adding new columns
4. updating new columns
5. Display data(before updating)
7. deleting data
8. dropping data

Enter your choice:

7

Table data deleted

-----MENU-----

0. exit
1. create table
2. insert data
3. adding new columns
4. updating new columns
5. Display data(before updating)
6. Display data(after updating)
7. deleting data
8. dropping data

Enter your choice:

8

Table deleted successfully

-----MENU-----

0.exit
Enter your choice:
0
BUILD SUCCESSFUL (total time: 7 minutes 53 seconds)

EXPERIMENT 02:

AIM: Write an application to demonstrate HTTP Servlets.

SOURCE CODE:

Execution steps:

FILE → NEW PROJECT → JAVA WEB – JAVA WEB APPLICATION → NEXT → SELECT APACHE TOMACT

VERSION → FINISH

Right CLICK PROJECT NAME → SELECT NEW → SERVLET CLASS → GIVE CLASS NAME AS IN ACTION ATTRIBUTE.

Code:

Get Method code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>GET METHOD</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
     <form action="LoginServlet>" method="GET">
       <center>
         <h1>This is servlet program</h1>
         <br> Username: <input type="text" name="username"/> <br/>
         <br>  Password: <input type="password" name="yourPassword"/> <br/>
        <input type="submit" value="NEXT" />
       </center>
      </form>
  </body>
</html>
```

LoginServlet.java:

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
   protected void doPost(HttpServletRequest request,
         HttpServletResponse response) throws ServletException, IOException {
      String username = request.getParameter("username");
      String password = request.getParameter("password");
      System.out.println("username: " + username);
      System.out.println("password: " + password);
       PrintWriter writer = response.getWriter();
      String htmlRespone = "<html>";
      htmlRespone += "<h2>Your username is: " + username + "<br/>";
      htmlRespone += "Your password is: " + password + "</h2>";
      htmlRespone += "</html>";
      writer.println(htmlRespone);
   }
}
```

Post method code:

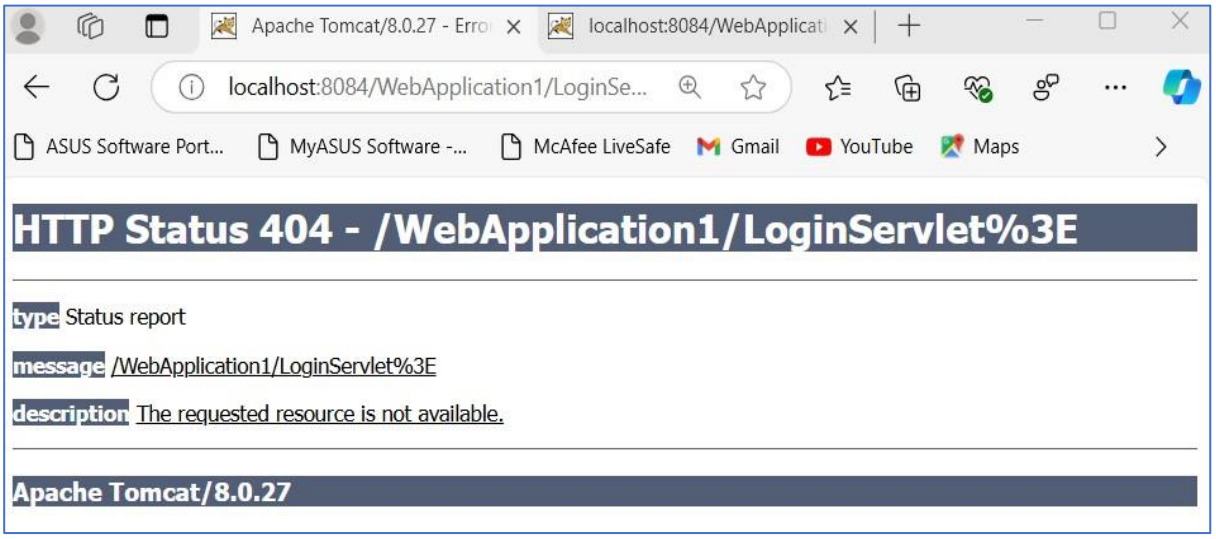
```
<html>
   <head>
      <title>POST METHOD</title>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
  <center>
    <form  method="post" action="LoginServlet">
  Username: <input type="text" name="username"/> <br/>
  Password: <input type="password" name="password"/> <br/>
  <input type="submit" value="Login" />
    </form>
  </center>
  </body>
</html>
```

Output:

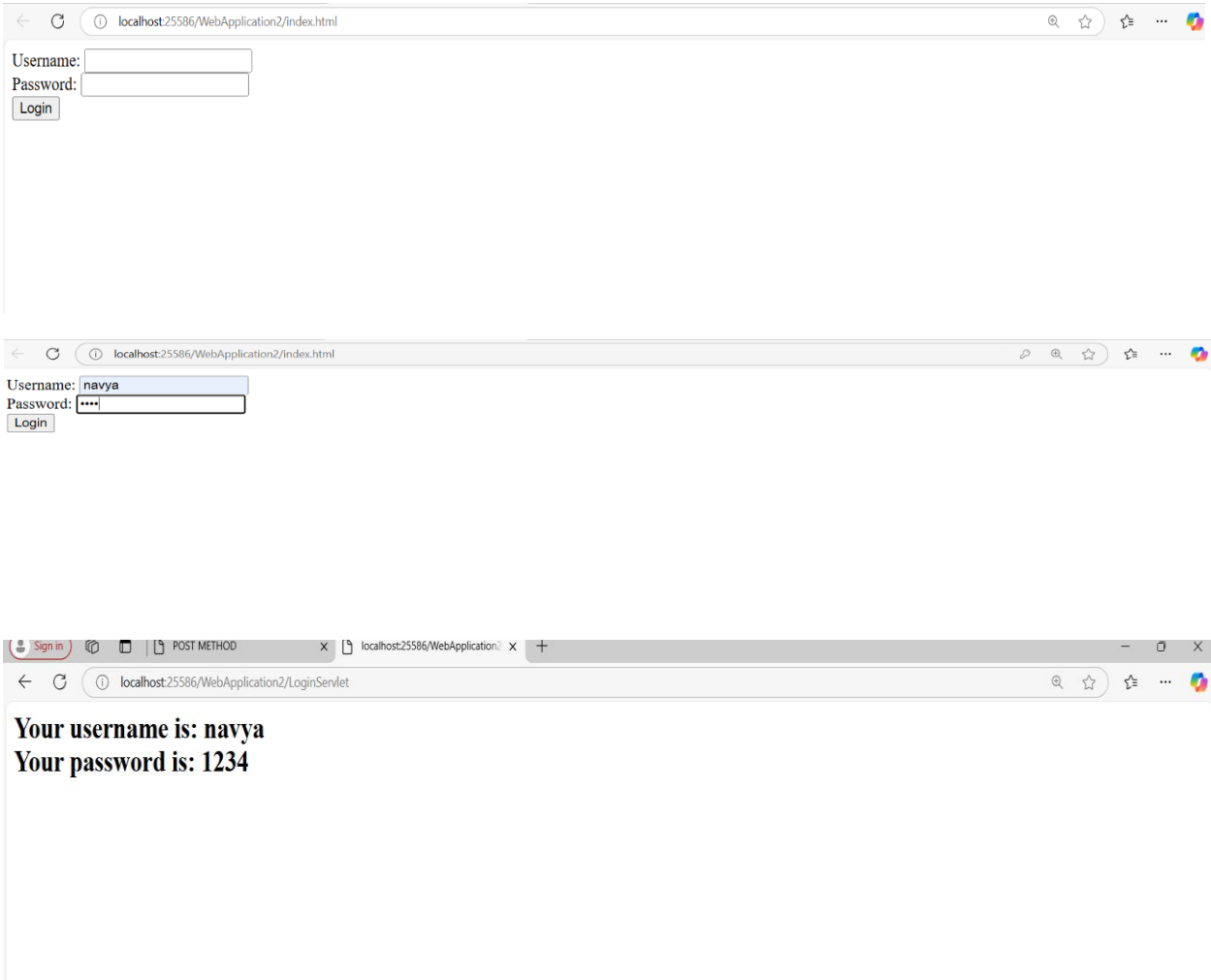
Get output:

Right click the index.html → click run file



POST OUTPUT:

Right click the post html file → click run file.



Experiment -03

AIM: Write an application to demonstrate Cookie & Sessions.

Source Code:

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Shopping</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>

    <div>
      <center>
      <form method="get" action="NewServlet">
        Username:<input type="text" name="username"/><br><br>
        Password:<input type="password" name="password"/><br><br>
        <input type="submit" name="submit"/>
      </form>
      </center>
    </div>
  </body>
</html>
```

NewServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
```

```
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class NewServlet extends HttpServlet {
   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
         throws ServletException, IOException, ClassNotFoundException, SQLException {
      response.setContentType("text/html;charset=UTF-8");
      Class.forName("org.apache.derby.jdbc.ClientDriver");
      Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm474","navya","navya
");
      Statement st=con.createStatement();
      String a;
      int r;
      ResultSet res;
      String name=request.getParameter("username");
      String pass=request.getParameter("password");
      res=st.executeQuery("select * from userdata where usename='"+name+"' and
password='"+pass+"'");
      ServletContext sc=request.getServletContext();
```

```
RequestDispatcher rd;
HttpSession ses=request.getSession(true);
Integer count=(Integer)ses.getAttribute("count");
if(count==null)
   count=0;
try (PrintWriter out = response.getWriter()) {
   out.println("<!DOCTYPE html>");
   out.println("<html>");
   out.println("<head>");
   out.println("<title>Servlet NewServlet2</title>");
   out.println("</head>");
   out.println("<body>");
   if(res.next())
   {
       rd=sc.getRequestDispatcher("/NewServlet1");
       rd.forward(request, response);
   }
   else
   {
      count=count+1;
      ses.setAttribute("count", count);
   if(count<3)
   {
      out.println("<h1>enter correct details</h1>");
      rd=sc.getRequestDispatcher("/index.html");
      rd.include(request, response);
   }
   else
   {
```

```
            out.println("<h1>No of Attempts execeded </h1>");
        }


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

}
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null, ex);
    }
}
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null, ex);
```

```
        }
    }
    @Override
    public String getServletInfo() {
        return "Short description";
    }
}
```

NewServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class NewServlet1 extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet NewServlet1</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<form method='get' action='brand'>");
            out.println("<h1>Choose your options:</h1>");
```

```
            out.println("<h4>Select watch brand you want</h4>");
            out.println("<input type='checkbox' name='n' value='noise'/>Noise<br>");
            out.println("<input type='checkbox' name='n' value='fire boltt'/>Fire Boltt<br>");
            out.println("<input type='checkbox' name='n' value='ptron'/>pTron<br>");
            out.println("<input type='checkbox' name='n' value='boat wave'/>boAT Wave<br>");
            out.println("<input type='submit' value='Next'/>");
            out.println("</body>");
            out.println("</html>");
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    public String getServletInfo() {
        return "Short description";
    }
}
```

brand.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
```

```
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class brand extends HttpServlet {


   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
          throws ServletException, IOException, ClassNotFoundException, SQLException {
      response.setContentType("text/html;charset=UTF-8");
      Class.forName("org.apache.derby.jdbc.ClientDriver");
      Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm474","navya","navya
");
      Statement st=con.createStatement();
      ResultSet res;
      String[] name=request.getParameterValues("n");
      String n,m,v;
      int amou,id;
      try (PrintWriter out = response.getWriter()) {
         out.println("<!DOCTYPE html>");
         out.println("<html>");
         out.println("<head>");
         out.println("<title>Servlet  brand</title>");
```

```
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Select the models in it.....</h1>");
        out.println("<form method='get' action='checkedout'>");
        for(int i=0;i<name.length;i++)
        {
           res=st.executeQuery("select * from cookie where name='"+name[i]+"'");
           while(res.next())
           {
              n=res.getString(1);
              m=res.getString(2);
              v=res.getString(3);
              amou=res.getInt(4);
              id=res.getInt(5);
              out.println("<input type='checkbox' name='brand' value='"+id+"'/>");
              out.println("Name: "+n);
              out.println("Model: "+m);
              out.println("version: "+v);
              out.println("Amount: "+amou);
              out.println("Id: "+id);
              out.println("<br>");
           }
        }
        out.println("<input type='submit' value='checkout'/>");
        out.println("</form></body>");
        out.println("</html>");
     }
  }
  @Override
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        try {
            processRequest(request, response);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(brand.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(brand.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        try {
            processRequest(request, response);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(brand.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(brand.class.getName()).log(Level.SEVERE, null, ex);
        }
    @Override
    public String getServletInfo() {
        return "Short description";
    }

}
```

checkedout.java

```
import java.io.IOException;
import java.io.PrintWriter;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


public class checkedout extends HttpServlet {
   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
          throws ServletException, IOException, ClassNotFoundException, SQLException {
      response.setContentType("text/html;charset=UTF-8");
      response.setContentType("text/html;charset=UTF-8");
      Class.forName("org.apache.derby.jdbc.ClientDriver");
      Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm474","navya","navya
");
      Statement st=con.createStatement();
      ResultSet res;
      Cookie c;

      String[] name=request.getParameterValues("brand");
      try (PrintWriter out = response.getWriter()) {
         out.println("<!DOCTYPE html>");
```

```
out.println("<html>");
out.println("<head>");
out.println("<title>Servlet brand</title>");
out.println("</head>");
out.println("<body>");
out.println("<form method='get' action='pay_money'>");
out.println("<h1>Payment hear...</h1>");
int id,amount=0;
int total=0;
for(int i=0;i<name.length;i++)
{
   res=st.executeQuery("select * from cookie where id="+name[i]+" ");
   while(res.next())
   {
      //total=total+res.getInt(1);
      id=res.getInt(5);
      amount=res.getInt(4);
      total=total+amount;
      out.println("product ID :"+id+"  Total Amount :"+amount);
      c=new   Cookie(String.valueOf(id),String.valueOf(amount));
      response.addCookie(c);
      out.println("<br>");
   }
}
out.println("<h2>Total Amount :"+total+"</h2>");
c=new Cookie("total",String.valueOf(total));
out.println("<input type='submit' value='pay'/>");
out.println("</form>");
response.addCookie(c);
```

```
            out.println("</body>");
            out.println("</html>");
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        try {
            processRequest(request, response);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(checkedout.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(checkedout.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        try {
            processRequest(request, response);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(checkedout.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(checkedout.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public String getServletInfo() {
        return "Short description";
```

```
    }
}
```

pay_money.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;


public class pay_money extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        HttpSession ses=request.getSession();
        ses.setMaxInactiveInterval(60);
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet pay_money</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<form method='get' action='payment_page'>");
            out.println("<h3>pay the amount...</h3><br>");
            out.println("Username :");
```

```
            out.println("<input type='text' name='username'/><br><br>");
            out.println("Id :");
            out.println("<input type='text'name='id'/><br><br>");
            out.println("<br>");
            out.println("<input type='submit' value='payment hear.....'/>");
            out.println("</form>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    public String getServletInfo() {
        return "Short description";
    }

}
```

payment_page.java

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
import static java.lang.System.out;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;


public class payment_page extends HttpServlet {
   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
          throws ServletException, IOException, ClassNotFoundException, SQLException {
      response.setContentType("text/html;charset=UTF-8");
      Class.forName("org.apache.derby.jdbc.ClientDriver");
      HttpSession ses=request.getSession(false);
      Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm430","venu","venu
");
      Statement
st=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE)
;
```

```
ResultSet res;
ServletContext sc=request.getServletContext();
RequestDispatcher rd;
Cookie[] c=request.getCookies();
int amount=0;
int total=0;
String name=request.getParameter("username");
res=st.executeQuery("select * from userdata where usename='"+name+"'");
if(ses==null){
   System.out.println("session expired");
}
if(res.next())
{
   amount=res.getInt(3);
}
for(Cookie cookie:c)
{
   if(cookie.getName().equals("total"))
   {
      total=Integer.parseInt(cookie.getValue());
   }
}
if(total<=amount)
   {
      amount=amount-total;
      res.updateInt("amount",amount);
      res.updateRow();
      rd=sc.getRequestDispatcher("/NewServlet2");
      rd.include(request, response);
```

```
        }
      else
        {
          rd=sc.getRequestDispatcher("/NewServlet3");
          rd.include(request, response);
        }

      try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet payment_page</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("</body>");
        out.println("</html>");
      }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
      try {
         processRequest(request, response);
       } catch (ClassNotFoundException ex) {
         Logger.getLogger(payment_page.class.getName()).log(Level.SEVERE, null, ex);
      } catch (SQLException ex) {
         Logger.getLogger(payment_page.class.getName()).log(Level.SEVERE, null, ex);
      }
    }
```

```
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        try {
            processRequest(request, response);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(payment_page.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(payment_page.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public String getServletInfo() {
        return "Short description";
    }
}
```

NewServlet2.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(urlPatterns = {"/NewServlet2"})
public class NewServlet2 extends HttpServlet {
```

```
    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet NewServlet2</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>payment complete</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    public String getServletInfo() {
        return "Short description";
```

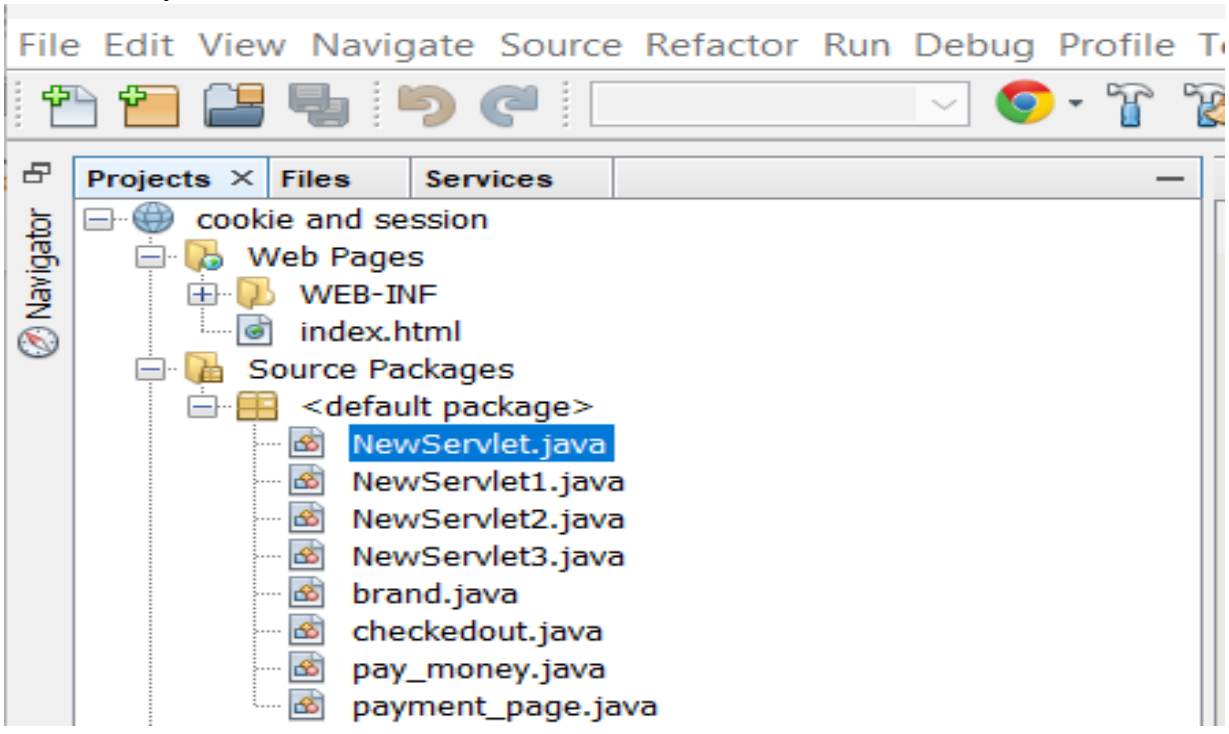
```
    }
}
```

NewServlet.java

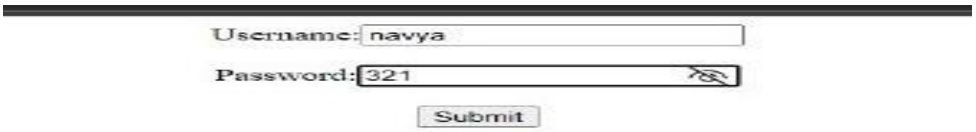
```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class NewServlet3 extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet NewServlet3</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>payment insufficent</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
```

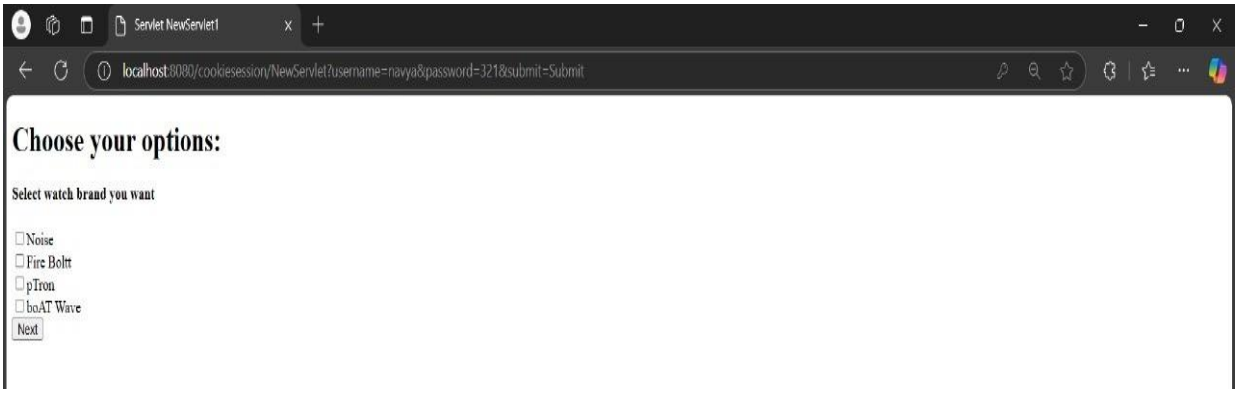
```
        throws ServletException, IOException {
    processRequest(request, response);
  }
  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
  }
  @Override
  public String getServletInfo() {
    return "Short description";
  }
}
```

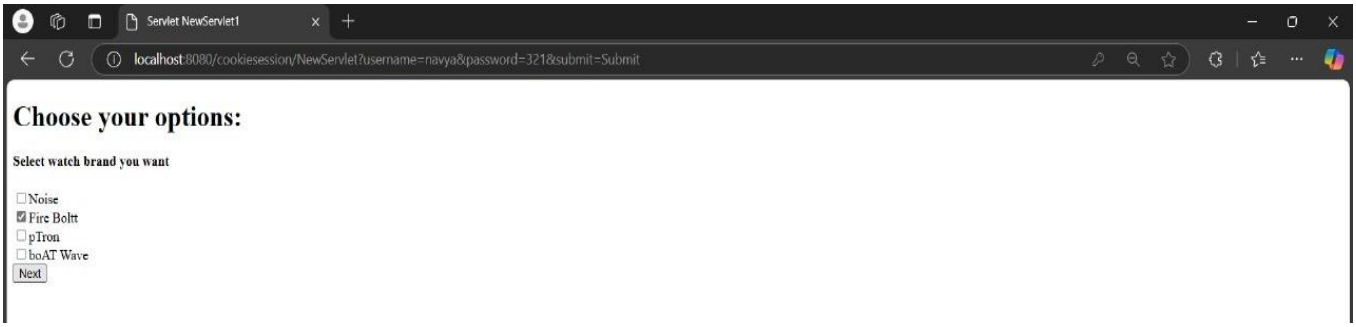
Output:

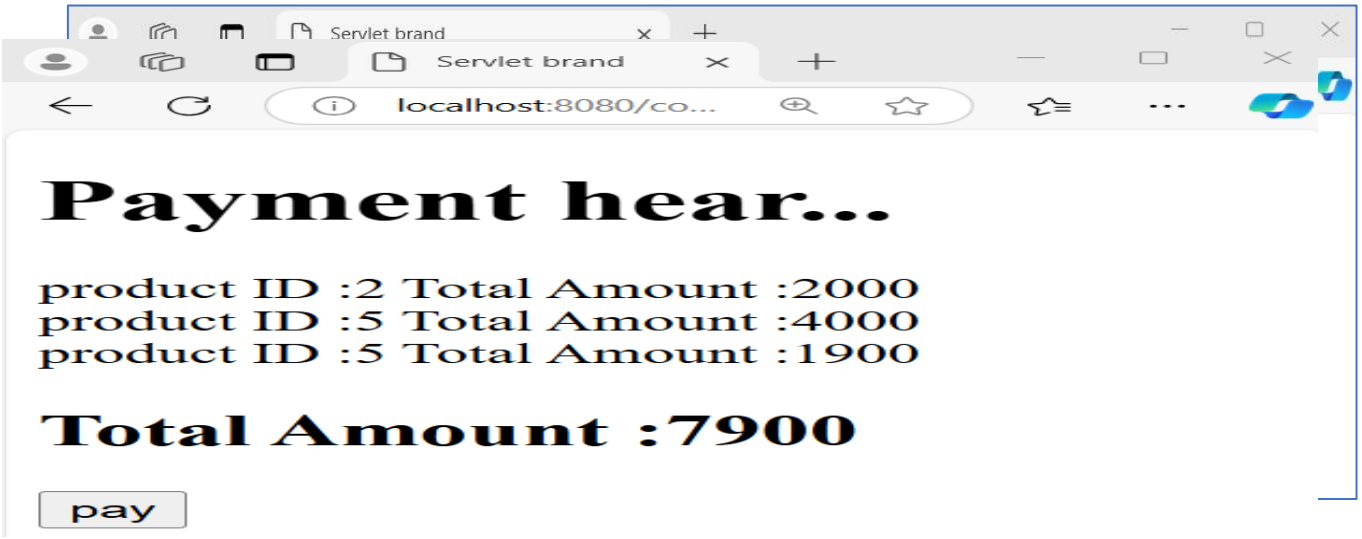


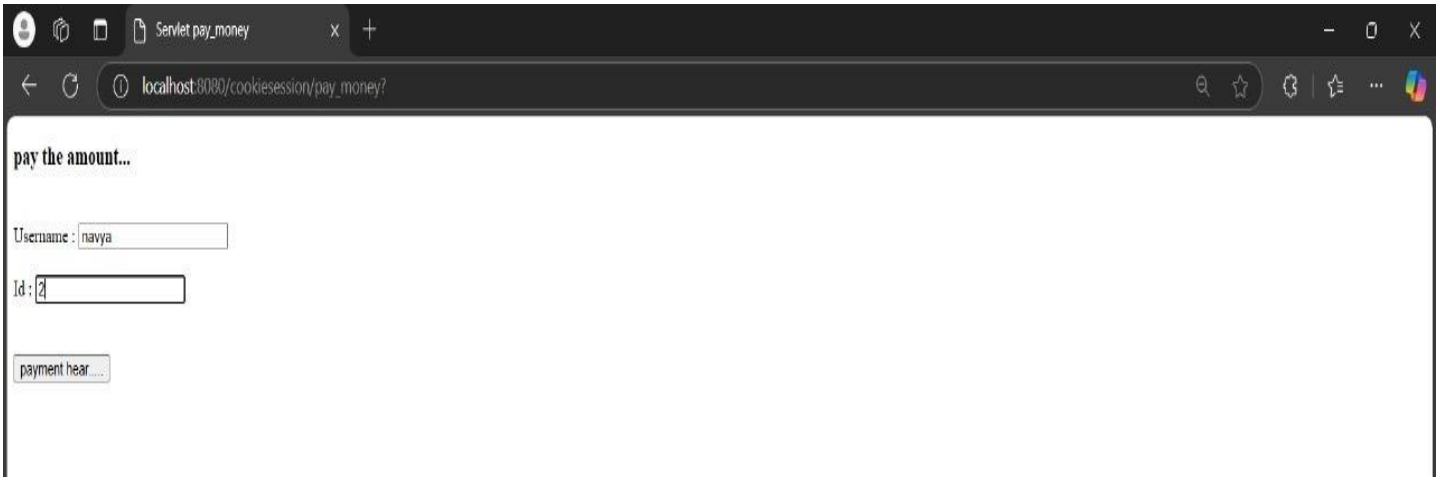
Right click the index.html file click run file:








Servlet brand
Servlet brand
localhost:8080/co...
Payment hear...
product ID :2 Total Amount :2000
product ID :5 Total Amount :4000
product ID :5 Total Amount :1900
Total Amount :7900
pay


Servlet pay_money
localhost:8080/cookiesession/pay_money?
pay the amount...
Username : navya
Id : 2
payment hear....

EXPERIMENT 04:

AIM: Write an application to integrate JSP & Servlets.

Source code:

insert.jsp

```
<%--
    Document   : index
    Created on : 25 Oct, 2024, 9:57:32 PM
    Author     : y22acm430
--%>
<%@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 method="post" action="Login">
            Username :<input type="text" name="uname"/><br><br>
            Password :<input type="password" name="pass"/><br><br>
            <input type="submit" value="Login"/>
        </form>
        <br>
        <form action="register.jsp">
            <input type="submit" value="Register"/>
        </form>
    </body>
</html>
```

Login,java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class Login extends HttpServlet {
   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
           throws ServletException, IOException {
       response.setContentType("text/html;charset=UTF-8");

       String name = request.getParameter("uname");
       String pa = request.getParameter("pass");

       try (Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm430", "venu", "venu");
            PreparedStatement pst = con.prepareStatement("SELECT * FROM login WHERE
username = ? AND password = ?")) {

           pst.setString(1, name);
           pst.setString(2, pa); // Replace with hashed password comparison in production

           try (ResultSet res = pst.executeQuery()) {
               try (PrintWriter out = response.getWriter()) {
```

```
                out.println("<!DOCTYPE html>");
                out.println("<html>");
                out.println("<head><title>User Details</title>");
                out.println("<style>");
                out.println("table { width: 50%; border-collapse: collapse; }");
                out.println("th, td { border: 1px solid black; padding: 10px; text-align: left; }");
                out.println("th { background-color: #f2f2f2; }");
                out.println("</style>");
                out.println("</head>");
                out.println("<body>");
                out.println("<h1>User Details</h1>");

                if (res.next()) {
                    out.println("<table>");
                    out.println("<tr><th>Name</th><td>" + res.getString("name") +
"</td></tr>");
                    out.println("<tr><th>Username</th><td>" + res.getString("username") +
"</td></tr>");
                    out.println("<tr><th>Password</th><td>Can't be displayed</td></tr>");
                    out.println("<tr><th>Registration Date</th><td>" + res.getString("regd") +
"</td></tr>");
                    out.println("<tr><th>Gender</th><td>" + res.getString("gender") +
"</td></tr>");
                    out.println("<tr><th>Branch</th><td>" + res.getString("branch") +
"</td></tr>");
                    out.println("</table>");

                } else {
                    out.println("<p>Invalid credentials. Please try again.</p>");
                    RequestDispatcher rd =
request.getServletContext().getRequestDispatcher("/NewServlet");
                    rd.include(request, response);
                }
```

```
                out.println("</body>");
                out.println("</html>");
            }
        }
    } catch (SQLException ex) {
        throw new ServletException("Database error", ex);
    }
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "Short description";
}
}
```

NewServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
```

```
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class NewServlet extends HttpServlet
{
   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
           throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
      try (PrintWriter out = response.getWriter()) {
         out.println("<!DOCTYPE html>");
         out.println("<html>");
         out.println("<head>");
         out.println("<title>Servlet NewServlet</title>");
         out.println("</head>");
         out.println("<body>");
         out.println("<form method='post' action='register.jsp'>");
         out.println("<h1>You are the new user so Register Now</h1>");
         out.println("<input type='submit' value='Register here'/>");
         out.println("</form></body>");
         out.println("</html>");
      }
   }
   @Override
   protected void doGet(HttpServletRequest request, HttpServletResponse response)
           throws ServletException, IOException {
      processRequest(request, response);
   }
   @Override
   protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    public String getServletInfo() {
        return "Short description";
    }
}
```

register.jsp

```
<%--
    Document   : register
    Created on : 17 Oct, 2024, 3:42:38 PM
    Author     : y22acs430
--%>
<%@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>
        <style>
            table, th, td
            {
                border:1px solid black;
            }
        </style>
    </head>

    <body>
        <h2>Register Form</h2>
```

```
<table style="width: 30%">
    <form method="post" action="details.jsp">
      <tr>
        <th>NAME</th>
        <td><input type="text" name="name" size="30%"/></td>
      </tr>
      <tr>
       <th>USERNAME</th>
       <td><input type="text" name="username" size="30%"/></td>
      </tr>
      <tr>
        <th>PASSWORD</th>
        <td><input type="password" name="password" size="30%"/></td>
      </tr>
      <tr>
        <th>REGD</th>
        <td><input type="text" name="regd" size="30%"/></td>
      </tr>
      <tr >
        <th>GENDER</th>
        <td size="30%"><input type="radio" name="gender" value="male"/>Male
          <input type="radio" name="gender" value="female"/>Female
          <input type="radio" name="gender" value="others"/>Others
        </td>
      </tr>
      <tr>
        <th>BRANCH</th>
        <td size="30%">
          <select id="branch" name="branch">
            <option>Select Branch</option>
```

```
                    <option value="CSE">CSE</option>
                    <option value="AIML">ECE</option>
                    <option value="CBDS">EEE</option>
                    <option value="IT">IT</option>
                  </select>
                </td>
              </tr>
               <input type="submit" value="submit"/>
          </form>
   </table>
   </body>
</html>
```

bean.java --> create a normal java file

```
package t;
public class bean
{
   private String name,username,pass,regd,gender,branch;
   public String getName()
   {
      return name;
   }
   public void setName(String name)
   {
      this.name=name;
   }
   public String getUname()
   {
      return username;
   }
```

```
public void setUsername(String username)
{
   this.username=username;
}
public String getPass()
{
   return pass;
}
public void setPass(String password)
{
   this.pass=password;
}
public String getRegd()
{
   return regd;
}
public void setRegd(String regd)
{
   this.regd=regd;
}
public String getGender()
{
   return gender;
}
public void setGender(String gender)
{
   this.gender=gender;
}
public String getBranch()
{
```

```
        return branch;
    }
    public void setBranch(String branch)
    {
        this.branch=branch;
    }
}
```

details.jsp

```
<%--
    Document   : details
    Created on : 25 Oct, 2024, 8:02:08 AM
    Author     : y22acs430
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="t.bean"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Connection"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
     <body>
    <%
```

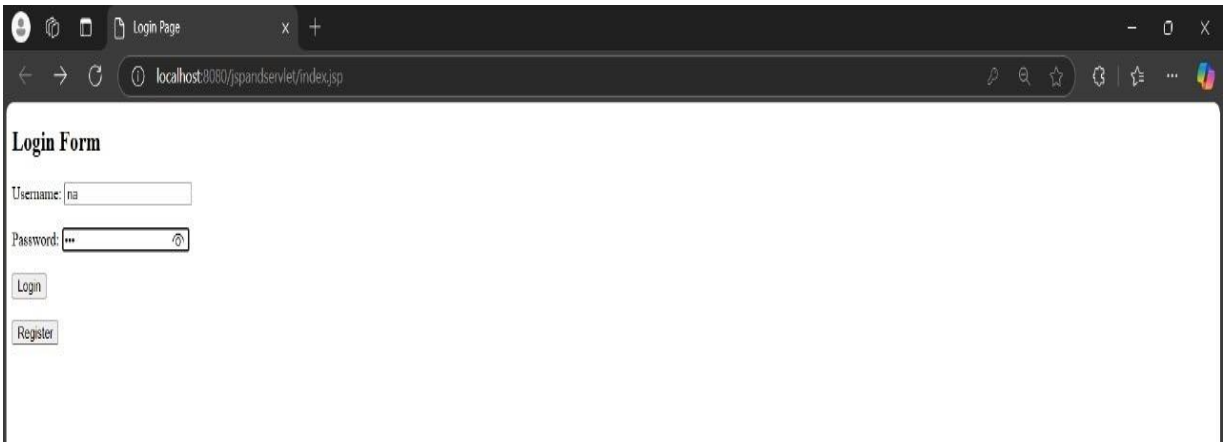
```
Class.forName("org.apache.derby.jdbc.ClientDriver");
Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm430","venu","venu"
);
Statement st=con.createStatement();
String name=request.getParameter("name");
String uname=request.getParameter("username");
String pass=request.getParameter("password");
String regd=request.getParameter("regd");
String gen=request.getParameter("gender");
String branch=request.getParameter("branch");
PreparedStatement pst;
bean jb=new bean();
jb.setName(name);
jb.setUsername(uname);
jb.setPass(pass);
jb.setRegd(regd);
jb.setGender(gen);
jb.setBranch(branch);
pst=con.prepareStatement("insert into login values(?,?,?,?,?,?)");
pst.setString(1,name);
pst.setString(2,uname);
pst.setString(3,pass);
pst.setString(4,regd);
pst.setString(5,gen);
pst.setString(6,branch);
int res =pst.executeUpdate();
if(res==1)
{
   out.println("<!DOCTYPE html>");
   out.println("<html>");
```

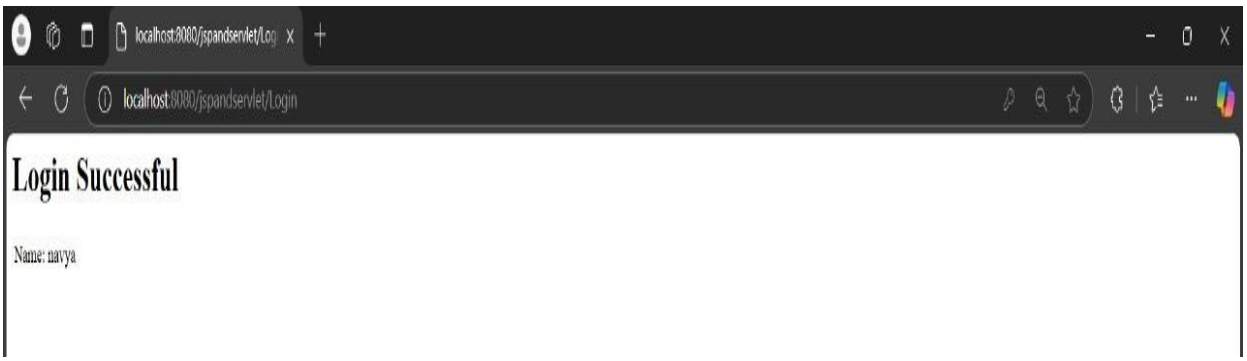
```
            out.println("<h1>Registation Completed Successfully</h1>");
            out.println("<style><table,th{ border:1px solid black; }>");
            out.println("<td{border:1px solid black; width:30%}></style>");
            out.println("</head>");
            out.println("<body>");
            out.println("<table>");
            out.println("<tr><th>NAME</th>"+"<td>"+jb.getName()+"</td></tr>");
            out.println("<tr><th>USERNAME:</th>"+"<td>"+jb.getUname()+"</td></tr>");
            out.println("<tr><th>PASSWORD</th>"+"<td>can't be displayed</td></tr>");
            out.println("<tr><th>REGD</th>"+"<td>"+jb.getRegd()+"</td></tr>");
            out.println("<tr><th>GENDER</th>"+"<td>"+jb.getGender()+"</td></tr>");
            out.println("<tr><th>BRANCH</th>"+"<td>"+jb.getBranch()+"</td></tr>");
            out.println("</table>");
            out.println("</body>");
            out.println("</html>");
            out.println("<form action='index.jsp'>");
            out.println("<input type='submit' value='login'>");
        }
        else
        {
            out.println("<h1>Registation fails</h1>");
        }

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

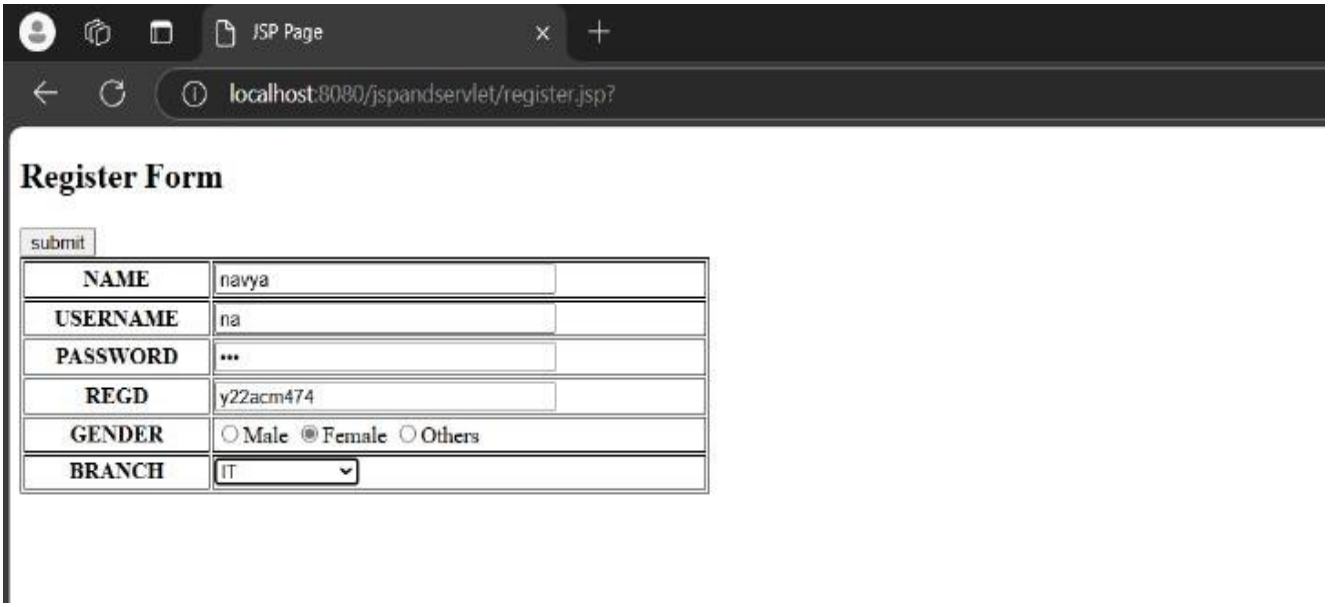
Execution steps:

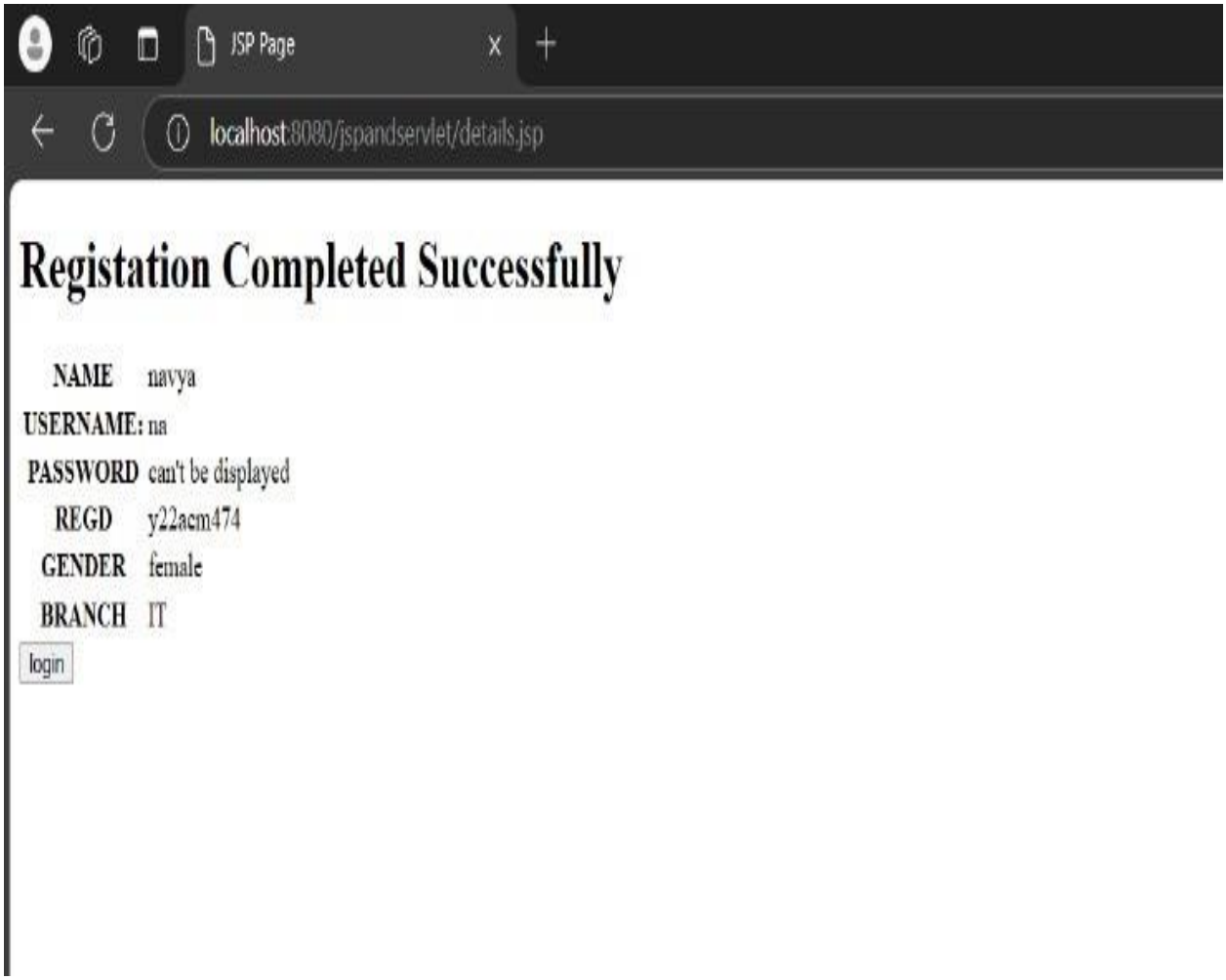
File → new project → java web → java web application →next → next → finish. Create dataset and create a login table.

OUTPUT:


Login Page
localhost:8080/jspandservlet/index.jsp
Login Form
Username: na
Password:
Login
Register


localhost:8080/jspandservlet/Login
Login Successful
Name: navya


JSP Page
localhost:8080/jspandservlet/register.jsp?
Register Form
submit
NAME navya
USERNAME na
PASSWORD
REGD y22acm474
GENDER Male Female Others
BRANCH IT


JSP Page
localhost:8080/jspandservlet/details.jsp
Registation Completed Successfully
NAME navya
USERNAME: na
PASSWORD can't be displayed
REGD y22acm474
GENDER female
BRANCH IT
login

EXPERIMENT 05:

AIM: Write an application to demonstrate Standard and Custom Tags in JSP.

Source code:

index.html

```
<%--
    Document   : index.jsp
    Created on : 26 Oct, 2024, 2:24:18 PM
    Author     : y22acm474
--%>

<%@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="newjsp1.jsp">
            <h1>Enter Register number :</h1>
            <input type="text" name="regno" placeholder="REGNO(Y22XXXXXX)">
            <input type="submit" value="submit">
        </form>

    </body>
</html>
```

newjsp1.jsp

```
<%--
    Document   : newjsp1
```

```
    Created on : 22 Oct, 2024, 2:27:02 PM
    Author     : Y22ACM430
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
<%@ taglib uri="/WEB-INF/tlds/newtag_library.tld" prefix="ct" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
         <sql:setDataSource var='dt' driver="org.apache.derby.jdbc.ClientDriver"
            url="jdbc:derby://localhost:1527/y22acm474" user="navya" password="navya"/>
        <sql:query dataSource="${dt}" var='rs'>
            select * from lab5 where regno=?
            <sql:param value="${param.regno}" />
        </sql:query>

        <p style="color:blue">< ****Welcome ${param.regno}****></p>
        <table border="2">
        <th colspan="2">-*-*-Exam Results-*-*-</th>
        <c:forEach items="${rs.rows}" var="r">
            <tr><td>Regd No</td><td><c:out value="${r.regno}" /></td></tr>
            <tr><td>Name</td><td><c:out value="${r.name}" /></td></tr>
            <tr><td>Sub 1 :</td><td><c:out value="${r.s1}" /></td></tr>
            <tr><td>Sub 2 :</td><td><c:out value="${r.s2}" /></td></tr>
```

```
        <tr><td>Sub 3 :</td><td><c:out value="${r.s3}" /></td></tr>
        <tr><td>Sub 4 :</td><td><c:out value="${r.s4}" /></td></tr>
        <tr><td>Sub 5 :</td><td><c:out value="${r.s5}" /></td></tr>
        <tr><td>Sub 6 :</td><td><c:out value="${r.s6}" /></td></tr>
        <ct:NewTagHandler1 regno="${r.regno}"
                           s1="${r.s1}"
                           s2="${r.s2}"
                           s3="${r.s3}"
                           s4="${r.s4}"
                           s5="${r.s5}"
                           s6="${r.s6}" />
      </c:forEach>
      </table><br>
      <form action="index.jsp">
        <input type='submit' value="Login page">
      </form>
   </body>
</html>
```

newtag_lib --> project --> right click --> new --> other--> search as tag --> Tag Library Discriptor select -->finish.

```
<?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>newtag_library</short-name>
  <uri>/WEB-INF/tlds/newtag_library</uri>
  <!-- A validator verifies that the tags are used correctly at JSP
       translation time. Validator entries look like this:
    <validator>
```

```
        <validator-class>com.mycompany.TagLibValidator</validator-class>
        <init-param>
          <param-name>parameter</param-name>
          <param-value>value</param-value>
        </init-param>
     </validator>
  -->
  <!-- A tag library can register Servlet Context event listeners in
       case it needs to react to such events. Listener entries look
       like this:
    <listener>
      <listener-class>com.mycompany.TagLibListener</listener-class>
    </listener>
  -->
 <tag>
  <name>NewTagHandler1</name>
  <tag-class>aaa.NewTagHandler1</tag-class>
  <body-content>scriptless</body-content>
  <attribute>
   <name>regno</name>
   <rtexprvalue>true</rtexprvalue>
   <type>java.lang.String</type>
  </attribute>
  <attribute>
   <name>name</name>
   <rtexprvalue>true</rtexprvalue>
   <type>java.lang.String</type>
  </attribute>
  <attribute>
   <name>s1</name>
```

```
    <rtexprvalue>true</rtexprvalue>
    <type>int</type>
   </attribute>
   <attribute>
    <name>s2</name>
    <rtexprvalue>true</rtexprvalue>
    <type>int</type>
   </attribute>
   <attribute>
    <name>s3</name>
    <rtexprvalue>true</rtexprvalue>
    <type>int</type>
   </attribute>
   <attribute>
    <name>s4</name>
    <rtexprvalue>true</rtexprvalue>
    <type>int</type>
   </attribute>
   <attribute>
    <name>s5</name>
    <rtexprvalue>true</rtexprvalue>
    <type>int</type>
   </attribute>
   <attribute>
    <name>s6</name>
    <rtexprvalue>true</rtexprvalue>
    <type>int</type>
   </attribute>
  </tag>
  <tag>
```

```
    <name>NewTagHandler1</name>
    <tag-class>aaa.NewTagHandler1</tag-class>
    <body-content>scriptless</body-content>
  </tag>
</taglib>
```

TagNewHandler1 -----> project name --->right click --> new --> other--> search as tag --->select TagNewHandler--> next --> classname= TagNewHandler1 and give package aa--> next

-->give tld file location and tag name --> finish.

```
package aaa;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.tagext.JspFragment;
import javax.servlet.jsp.tagext.SimpleTagSupport;
public class NewTagHandler1 extends SimpleTagSupport {
    private String regno;
    private int s1;
    private int s2;
    private int s3;
    private int s4;
```

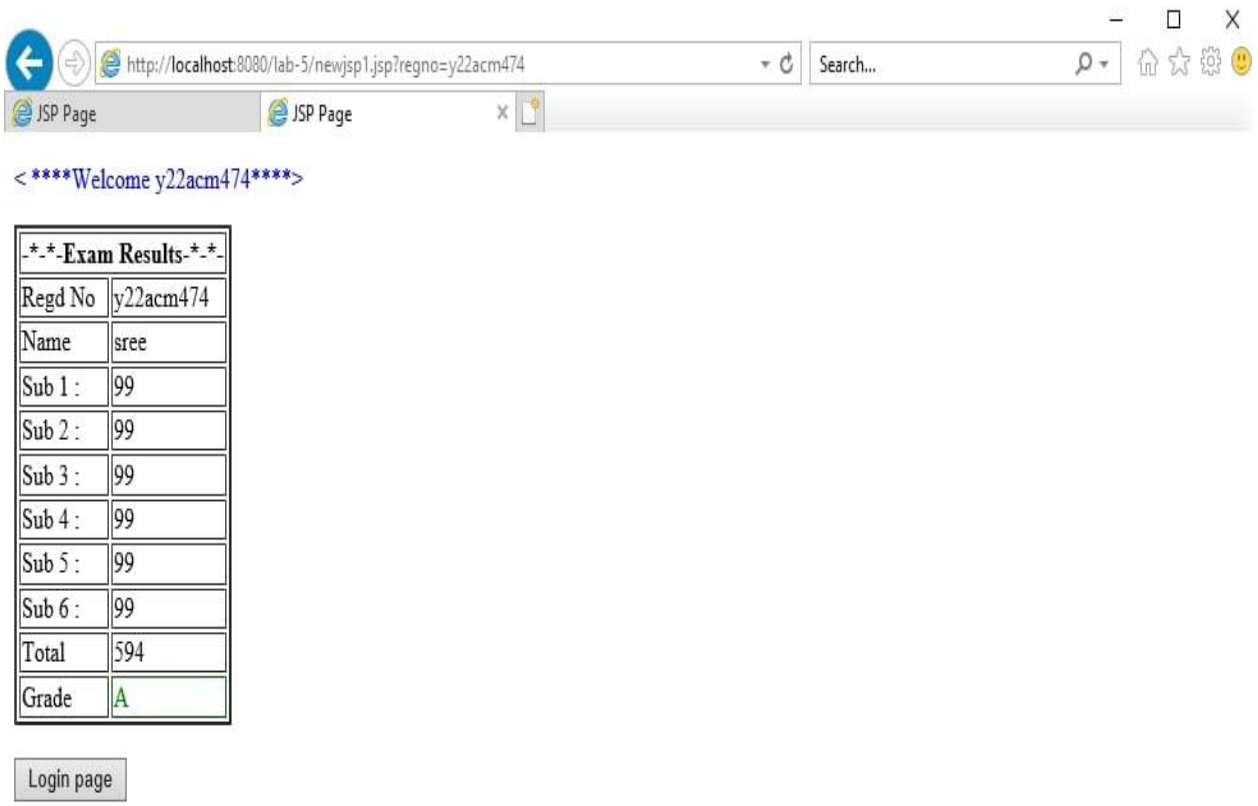
```
    private int s5;
    private int s6;

    @Override
    public void doTag() throws JspException, IOException
    {
        JspWriter out = getJspContext().getOut();
        int total = s1 + s2 + s3 + s4 + s5 + s6;
        String grade;
        String color;
        if (total >= 550)
        {
            grade = "A";
            color = "green";
        }
        else if (total >= 450)
        {
            grade = "B";
            color = "blue";
        }
        else if (total >= 350)
        {
            grade = "C";
            color = "orange";
        }
        else
        {
            grade = "F";
            color = "red";
        }
```

```
        out.write("<tr><td>Total</td><td>" + total + "</td></tr>");
        out.write("<tr><td>Grade</td><td style='color:" + color + "'>" + grade + "</td></tr>");
        try
        {
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm474","navya","navya"
);
            Statement st=con.createStatement();
            ResultSet rs;
            rs=st.executeQuery("select * from lab5 where regno='"+regno+"'");
            if(rs.next())
            {
                st.execute("update lab5 set total="+total);
                st.execute("update lab5 set grade="+grade);
            }
            JspFragment f = getJspBody();
            if (f!= null)
            {
                f.invoke(out);
            }

        }
        catch (java.io.IOException ex)
        {
            throw new JspException("Error in NewTagHandler tag", ex);
        }
        catch (ClassNotFoundException | SQLException ex)
        {
            Logger.getLogger(NewTagHandler1.class.getName()).log(Level.SEVERE,  null,  ex);
        }
```

```
}
public void setRegno(String regno)
{
   this.regno = regno;
}

public void setS1(int s1)
{
   this.s1 = s1;
}

public void setS2(int s2)
{
   this.s2 = s2;
}

public void setS3(int s3)
{
   this.s3 = s3;
}

public void setS4(int s4)
{
   this.s4 = s4;
}

public void setS5(int s5)
{
   this.s5 = s5;
}
```

```
    public void setS6(int s6)
    {
        this.s6=s6;
    }
}
```

Output:



EXPERIMENT 6

AIM: Write an application to demonstrate Java Server Faces (JSF) Validators, Event handlers and converters.

SOURCE CODE:

index.HTML:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
     xmlns:f="http://xmlns.jcp.org/jsf/core">
   <h:head>
      <title>Registration Page</title>
      <style>
         .cen {
            text-align: center;
         }
      </style>
   </h:head>
   <h:body>
      <center>
         <h:form>
            <h1>PERSONAL INFORMATION</h1>
            <h:panelGrid columns="2">
               <h:outputLabel value="First Name:" for="fname" />
               <h:inputText id="fname" value="#{mb.fname}" required="true">
                  <f:validateRequired />
               </h:inputText>
```

```
<h:outputLabel value="Last Name:" for="lname" />
<h:inputText id="lname" value="#{mb.lname}" required="true">
  <f:validateRequired />
</h:inputText>

<h:outputLabel value="Father Name:" for="father" />
<h:inputText id="father" value="#{mb.father}" required="true">
  <f:validateRequired />
</h:inputText>

<h:outputLabel value="Email Address:" for="email" />
<h:inputText id="email" value="#{mb.email}" required="true">
  <f:validator validatorId="newval" />
</h:inputText>

<h:outputLabel value="Register Number:" for="regdno" />
<h:inputText id="regdno" value="#{mb.regdno}" required="true">
  <f:validator validatorId="regd" />
</h:inputText>

<h:outputLabel value="Password:" for="password" />
<h:inputSecret id="password" value="#{mb.password}" required="true" />

<h:outputLabel value="Confirm Password:" for="confirmpass" />
<h:inputSecret id="confirmpass" value="#{mb.confirmpass}" required="true" />

<h:outputLabel value="Gender:" for="gender" />
<h:selectOneRadio id="gender" value="#{mb.gender}">
  <f:selectItem itemLabel="Male" itemValue="Male" />
```

```
            <f:selectItem itemLabel="Female" itemValue="Female" />
          </h:selectOneRadio>
          <h:outputLabel value="Language:" for="lang" />
          <h:selectManyCheckbox id="lang" value="#{mb.lang}">
            <f:selectItem itemLabel="English" itemValue="English" />
            <f:selectItem itemLabel="Hindi" itemValue="Hindi" />
            <f:selectItem itemLabel="Telugu" itemValue="Telugu" />
          </h:selectManyCheckbox>
          <h:outputLabel value="Phone Number:" for="phone" />
          <h:inputText id="phone" value="#{mb.phone}">
            <f:converter converterId="phn" />
          </h:inputText>
        </h:panelGrid>
        <h:commandButton action="preview.xhtml" value="Submit" />
      </h:form>
    </center>
  </h:body>
</html>
```

Managedbean.java

```
import java.util.Date;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
@ManagedBean(name = "mb")
@RequestScoped
public class managedbean {
  String
regdno,fname,lname,father,email,password,confirmpass,gender,lang[],phone,state,dis,man;
```

```
Date dob;

 public Date getDob() {
    return dob;
 }

 public void setDob(Date dob) {
    this.dob = dob;
 }
 public String getRegdno() {
    return regdno;
 }
 public void setRegdno(String regdno) {
    this.regdno = regdno;
 }

 public String getFname() {
    return fname;
 }

 public void setFname(String fname) {
    this.fname = fname;
 }

 public String getLname() {
    return lname;
 }

 public void setLname(String lname) {
```

```
    this.lname = lname;
  }

  public String getFather() {
    return father;
  }

  public void setFather(String father) {
    this.father = father;
  }

  public String getEmail() {
    return email;
  }

  public void setEmail(String email) {
    this.email = email;
  }

  public String getPassword() {
    return password;
  }

  public void setPassword(String password) {
    this.password = password;
  }

  public String getConfirmpass() {
    return confirmpass;
```

```
    }

    public void setConfirmpass(String confirmpass) {
        this.confirmpass = confirmpass;
    }

    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    public String[] getLang() {
        return lang;
    }

    public void setLang(String[] lang) {
        this.lang = lang;
    }

    public String getPhone() {
        return phone;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }
```

```
    public String getMan() {
        return man;
    }

    public void setMan(String man) {
        this.man = man;
    }
}
```

Regdno.java

```
import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.validator.FacesValidator;
import javax.faces.validator.Validator;
import javax.faces.validator.ValidatorException;

@FacesValidator("regd")
public class Regdno implements Validator {
    @Override
    public void validate(FacesContext context, UIComponent component, Object value)
throws ValidatorException {
        String regdno = (String) value;

        // Example validation logic (ensure it's numeric and has a specific length)
        if (regdno == null || !regdno.matches("\\d{9}|.*")) {
            FacesMessage msg = new FacesMessage("Invalid Register Number");
```

```
            msg.setSeverity(FacesMessage.SEVERITY_ERROR);
            throw new ValidatorException(msg);
        }
    }
}
```

EmailCheck.java

```
import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.validator.FacesValidator;
import javax.faces.validator.Validator;
import javax.faces.validator.ValidatorException;

@FacesValidator("newval")
public class EmailCheck implements Validator {

    @Override
    public void validate(FacesContext fc, UIComponent uic, Object value) throws
ValidatorException {
        if (value == null || value.toString().trim().isEmpty()) {
            // You might want to handle required fields separately.
            FacesMessage msg = new FacesMessage("Email is required");
            throw new ValidatorException(msg);
        }

        String email = value.toString();
        // Regex for validating email addresses
        String emailRegex = "^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}$";
```

```
        if (!email.matches(emailRegex)) {
            FacesMessage msg = new FacesMessage("Please enter a valid email address (e.g., user@example.com)");
            throw new ValidatorException(msg);
        }
    }
}
```

PhnConverter.java

```
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import javax.faces.convert.FacesConverter;
import java.util.logging.Level;
import java.util.logging.Logger;

@FacesConverter("phn")
public class PhnConverter implements Converter {

    private static final String COUNTRY_CODE = "+91";
    private static final Logger LOGGER = Logger.getLogger(PhnConverter.class.getName());

    @Override
    public Object getAsObject(FacesContext fc, UIComponent uic, String value) {
        if (value == null || value.isEmpty()) {
            return null;
        }
        if (!isValidPhoneNumber(value)) {
            LOGGER.log(Level.WARNING, "Invalid phone number: {0}", value);
```

```
            return null;
        }
        return COUNTRY_CODE + value.trim();
    }

    @Override
    public String getAsString(FacesContext fc, UIComponent uic, Object object) {
        if (object == null) {
            return "";
        }

        String phoneNumber = object.toString();
        return phoneNumber.replace(COUNTRY_CODE, "").trim();
    }
    private boolean isValidPhoneNumber(String phoneNumber) {

        return phoneNumber.matches("\\d{10}");
    }
}
```

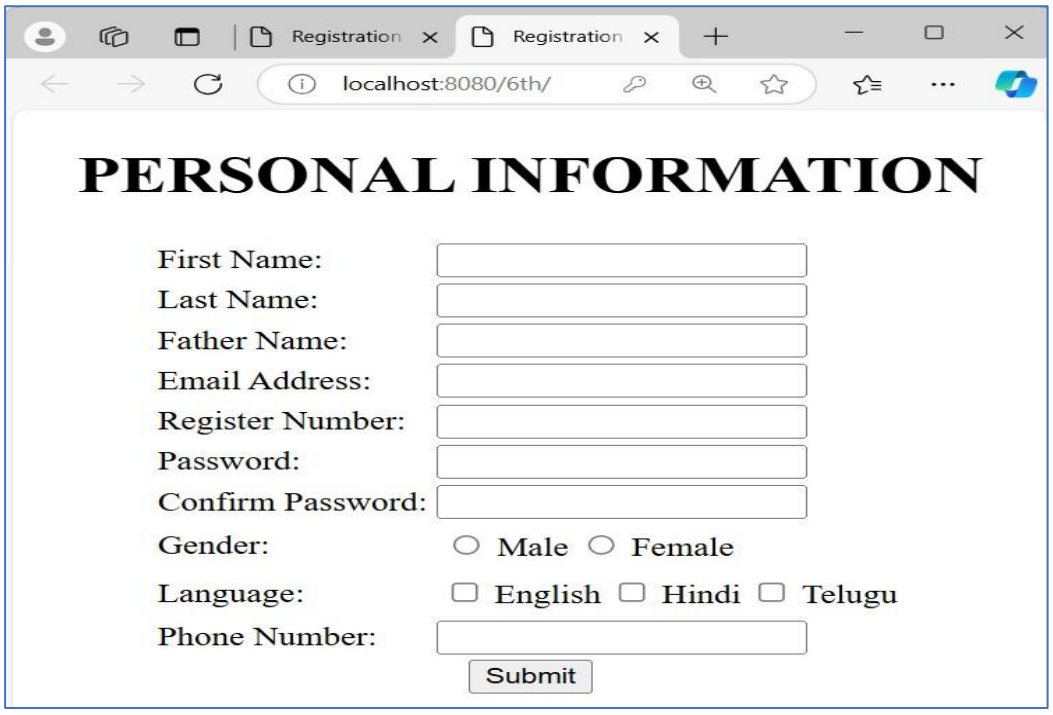
preview.html

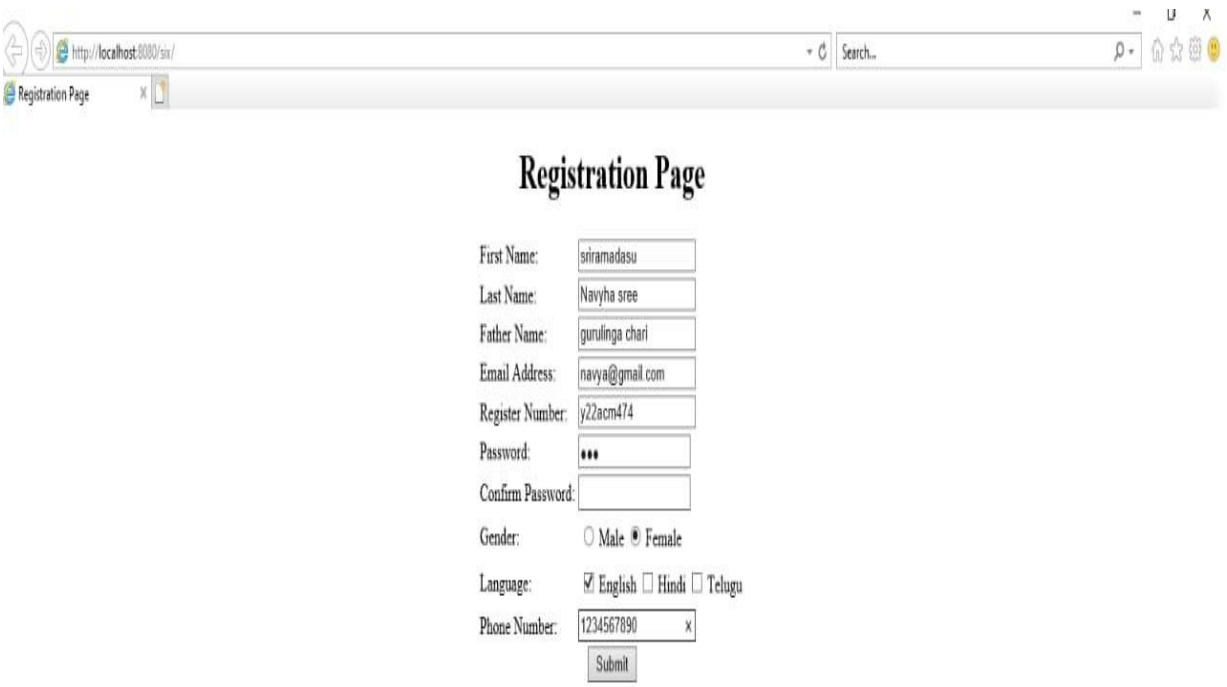
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:f="http://xmlns.jcp.org/jsf/core">
<head>
    <title>User Information</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
```

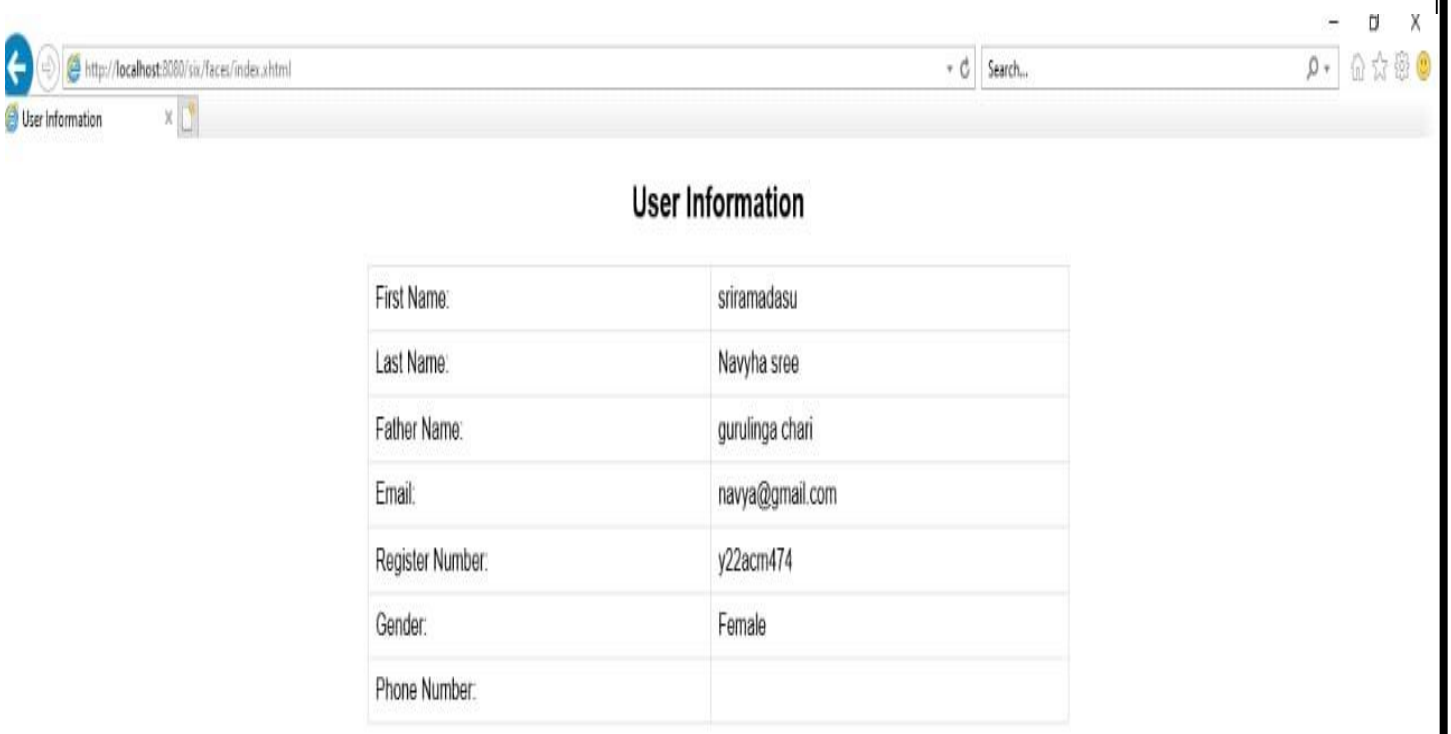
```
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
        }
        h2 {
            text-align: center;
        }
        .info-table {
            margin: 0 auto;
            border-collapse: collapse;
            width: 50%;
        }
        .info-table th, .info-table td {
            border: 1px solid #ddd;
            padding: 8px;
            text-align: left;
        }
        .info-table th {
            background-color: #f2f2f2;
        }
    </style>
</head>
<body>
    <h2>User Information</h2>
    <div>
        <h:panelGrid columns="2" class="info-table">
            <h:outputLabel value="First Name:" />
            <h:outputText value="#{mb.fname}" />
```

```
        <h:outputLabel value="Last Name:" />
        <h:outputText value="#{mb.lname}" />
        <h:outputLabel value="Father Name:" />
        <h:outputText value="#{mb.father}" />
        <h:outputLabel value="Email:" />
        <h:outputText value="#{mb.email}" />
        <h:outputLabel value="Register Number:" />
        <h:outputText value="#{mb.regdno}" />
        <h:outputLabel value="Gender:" />
        <h:outputText value="#{mb.gender}" />
        <h:outputLabel value="Phone Number:" />
        <h:outputText value="#{phn.phone}" />
      </h:panelGrid>
    </div>
</body>
</html>
```

Output:




http://localhost:8080/six/
Search...
Registration Page
Registration Page
First Name: sriramadasu
Last Name: Navyha sree
Father Name: gurulinga chari
Email Address: navya@gmail.com
Register Number: y22acm474
Password: •••
Confirm Password:
Gender: Male Female
Language: English Hindi Telugu
Phone Number: 1234567890
Submit


http://localhost:8060/six/faces/index.xhtml
Search...
User Information
User Information
First Name: sriramadasu
Last Name: Navyha sree
Father Name: gurulinga chari
Email: navya@gmail.com
Register Number: y22acm474
Gender: Female
Phone Number:

EXPERIMENT: 07

AIM: Write an application to demonstrate web service.

SOURCE CODE:

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Arithmetic Operations Application</title>
</head>
<body>
   <form action="http://localhost:8080/7th/webresources/generic" method="get">
      <div align="center">
         <h1>
            Enter a Number: <input type="text" name="a"/><br/><br/>
            Enter b Number:<input type="text" name="b"/><br/><br/><br/>
            <div style="color: red;" align="center">
               <input type="submit" value="Calculate"/>
            </div>
         </h1>
      </div>
   </form>
</body>
</html>
```

GenericResource.java

```
package pkg;
import javax.ws.rs.*;
import javax.ws.rs.core.*;
@Path("generic")
public class GenericResource {

    @Context
    private UriInfo context;

    public GenericResource() {
    }
    @GET
    @Produces(MediaType.TEXT_HTML)
    public String getHtml() {
        try {
            // Parse input numbers
            int a = Integer.parseInt(context.getQueryParameters().getFirst("a"));
            int b = Integer.parseInt(context.getQueryParameters().getFirst("b"));

            // Perform calculations
            int addition = a + b;
            int subtraction = a - b;
            int multiplication = a * b;
            double division = b != 0 ? (double) a / b : Double.NaN;
            int modulo = b != 0 ? a % b : Integer.MIN_VALUE;
            long factorialA = factorial(a);
            long factorialB = factorial(b);
            boolean isPrimeA = isPrime(a);
```

```
        boolean isPrimeB = isPrime(b);
        boolean isPerfectA = isPerfect(a);
        boolean isPerfectB = isPerfect(b);
        String multiplicationTableA = multiplicationTable(a);
        String multiplicationTableB = multiplicationTable(b);
        int sumEven = sumOfEvenNumbers(a, b);
        int sumOdd = sumOfOddNumbers(a, b);

        // Build the result HTML
        StringBuilder result = new StringBuilder("<div align='center'><h1>Results:</h1>");

        // Basic arithmetic results
        result.append("<h2>Basic  Operations</h2>");
        result.append("<table border='1' style='border-collapse:
collapse;'><tr><th>Operation</th><th>Result</th></tr>");
        result.append("<tr><td>Addition</td><td>").append(addition).append("</td></tr>");

result.append("<tr><td>Subtraction</td><td>").append(subtraction).append("</td></tr>");

result.append("<tr><td>Multiplication</td><td>").append(multiplication).append("</td></t
r>");
        result.append("<tr><td>Division</td><td>").append(Double.isNaN(division) ?
"undefined" : division).append("</td></tr>");
        result.append("<tr><td>Modulo</td><td>").append(modulo == Integer.MIN_VALUE ?
"undefined" : modulo).append("</td></tr>");
        result.append("<tr><td>Sum of Even
Numbers</td><td>").append(sumEven).append("</td></tr>");
        result.append("<tr><td>Sum of Odd
Numbers</td><td>").append(sumOdd).append("</td></tr>");
        result.append("</table>");

        // Factorial results
```

```
        result.append("<h2>Factorial</h2>");
        result.append("<table border='1' style='border-collapse:
collapse;'><tr><th>Number</th><th>Factorial</th></tr>");

result.append("<tr><td>").append(a).append("</td><td>").append(factorialA).append("</td
></tr>");

result.append("<tr><td>").append(b).append("</td><td>").append(factorialB).append("</td
></tr>");
        result.append("</table>");


        // Prime results
        result.append("<h2>Prime  Check</h2>");
        result.append("<table border='1' style='border-collapse:
collapse;'><tr><th>Number</th><th>Is Prime?</th></tr>");
        result.append("<tr><td>").append(a).append("</td><td>").append(isPrimeA ? "Yes" :
"No").append("</td></tr>");
        result.append("<tr><td>").append(b).append("</td><td>").append(isPrimeB ? "Yes" :
"No").append("</td></tr>");
        result.append("</table>");


        // Perfect number results
        result.append("<h2>Perfect Number Check</h2>");
        result.append("<table border='1' style='border-collapse:
collapse;'><tr><th>Number</th><th>Is  Perfect?</th></tr>");
        result.append("<tr><td>").append(a).append("</td><td>").append(isPerfectA  ?  "Yes"
: "No").append("</td></tr>");
        result.append("<tr><td>").append(b).append("</td><td>").append(isPerfectB  ?  "Yes"
: "No").append("</td></tr>");
        result.append("</table>");


        // Multiplication tables
        result.append("<h2>Multiplication  Tables</h2>");
```

```
            result.append("<h3>Table for
").append(a).append(":</h3><p>").append(multiplicationTableA).append("</p>");
            result.append("<h3>Table for
").append(b).append(":</h3><p>").append(multiplicationTableB).append("</p>");


            result.append("</div>");


            return result.toString();
        } catch (NumberFormatException e) {
            return "<div align='center'><h1>Error: Please enter valid integers.</h1></div>";
        }
    }

    private long factorial(int number) {
        if (number < 0) return -1; // Error for negative numbers
        long result = 1;
        for (int i = 1; i <= number; i++) {
            result *= i;
        }
        return result;
    }

    private boolean isPrime(int number) {
        if (number <= 1) return false;
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) return false;
        }
        return true;
    }
```

```
    private int sumOfEvenNumbers(int a, int b) {
        int sum = 0;
        for (int i = Math.min(a, b); i <= Math.max(a, b); i++) {
            if (i % 2 == 0) {
                sum += i;
            }
        }
        return sum;
    }

    private int sumOfOddNumbers(int a, int b) {
        int sum = 0;
        for (int i = Math.min(a, b); i <= Math.max(a, b); i++) {
            if (i % 2 != 0) {
                sum += i;
            }
        }
        return sum;
    }

    private String multiplicationTable(int number) {
        StringBuilder table = new StringBuilder();
        for (int i = 1; i <= 10; i++) {
            table.append(number).append(" x ").append(i).append(" = ").append(number *
i).append("<br/>");
        }
        return table.toString();
    }
```

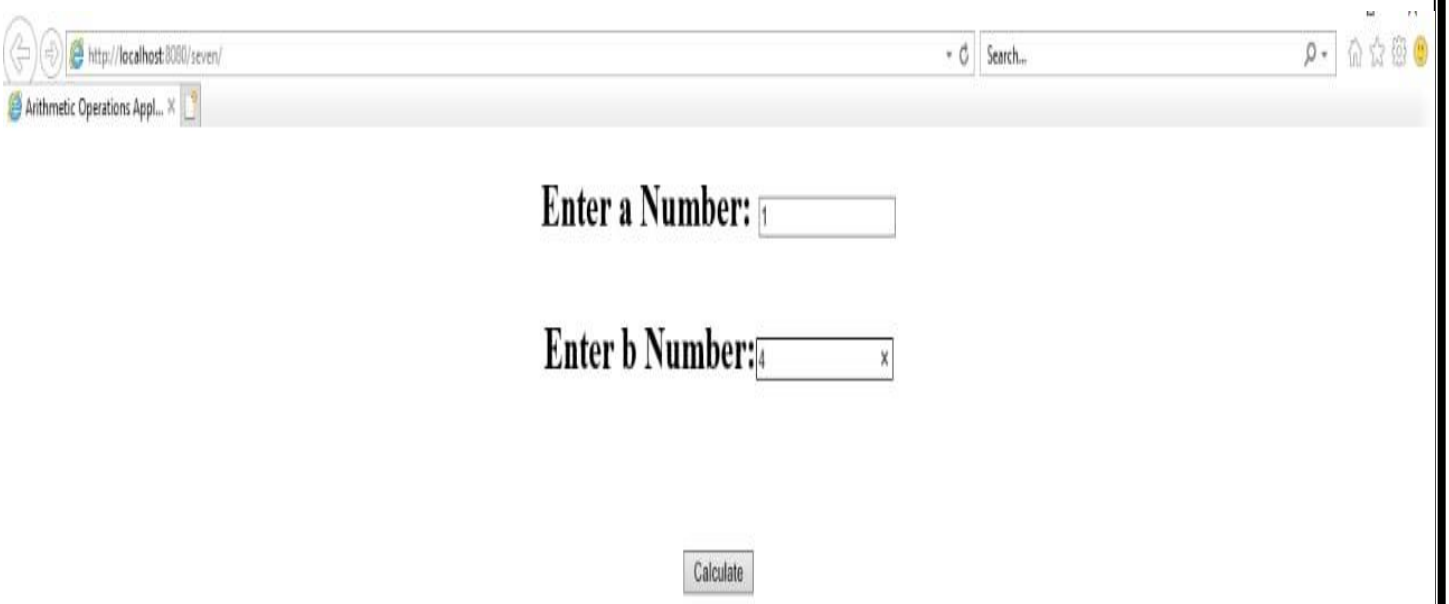
```
    private boolean isPerfect(int number) {
        if (number < 1) return false;
        int sum = 0;
        for (int i = 1; i < number; i++) {
            if (number % i == 0) {
                sum += i;
            }
        }
        return sum == number;
    }

    @PUT
    @Consumes(MediaType.TEXT_HTML)
    public void putHtml(String content) {
        // Not implemented
    }
}
```

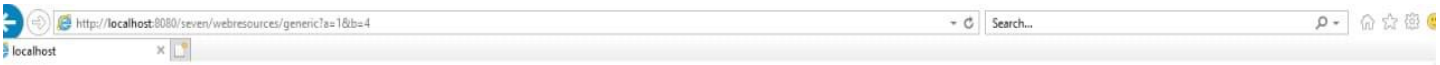
ApplicationConfig.java

```
package pkg;
import java.util.Set;
import javax.ws.rs.core.Application;
@javax.ws.rs.ApplicationPath("webresources")
public class ApplicationConfig extends Application {
    @Override
    public Set<Class<?>> getClasses() {
        Set<Class<?>> resources = new java.util.HashSet<>();
        addRestResourceClasses(resources);
        return resources;
    }
```

```
    private void addRestResourceClasses(Set<Class<?>> resources) {
        resources.add(pkg.GenericResource.class);
    }
}
```

Output:





Results:

Basic Operations

Operation	Result
Addition	5
Subtraction	-3
Multiplication	4
Division	0.25
Modulo	1
Sum of Even Numbers	6
Sum of Odd Numbers	4

Factorial

Number	Factorial
1	1.0
4	24.0

Prime Check

Number	Is Prime?
1	No
4	No

Perfect Number Check

Number	Is Perfect?
1	No
4	No

Multiplication Tables

Table for 1:

1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10

Table for 4:

4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40