EXPERIMENT 01:

Aim: Write a JDBC application to implement DDL and DML commands

Execution Steps:

Open **NetBeans** and follow the below steps:

1. **Create a Project:**
   - Go to **File → New Project → Java → Java Application**. Enter a **Project Name** and click **Finish**.
2. **Add Java File:**
   - Right-click **Source Packages → New → Java File**. Name the file JDBC and paste the required code.
3. **Set Up Database Connection:**
   - Install **Apache Derby** in your Java folder. In **Libraries**, add **Java DB** and ensure the paths are correct in **Properties**.
4. **Create Database:**
   - In **Services → Databases**, start the **Java DB server**.
   - Create a new database by specifying the username, password, and database name.
   - Connect to the database by right-clicking it and selecting **Connect**.
5. **Run the Project:**
   - Right-click the project → **Clean and Build**.
   - Click **Run** to view the output.
6. **View Table Data:**
   - In **Services → Databases**, expand your database → **Tables**.
   - Right-click your table → **View Data**.

Source Code:

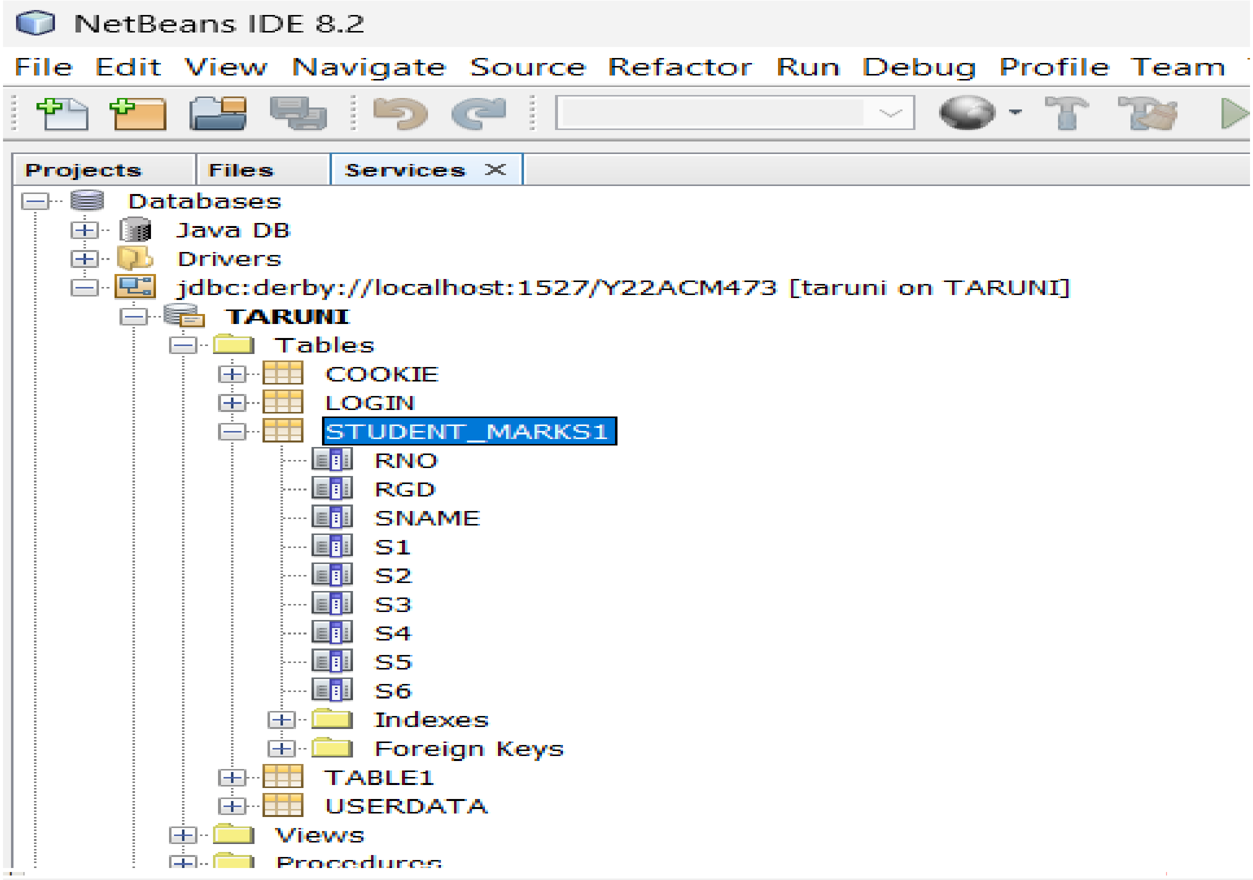
```
import java.sql.*;
import java.util.Scanner;
public class JDBCEX {
    public static void main(String[] args) {
        try {
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/Y22ACM473",
"taruni","taruni");
            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:

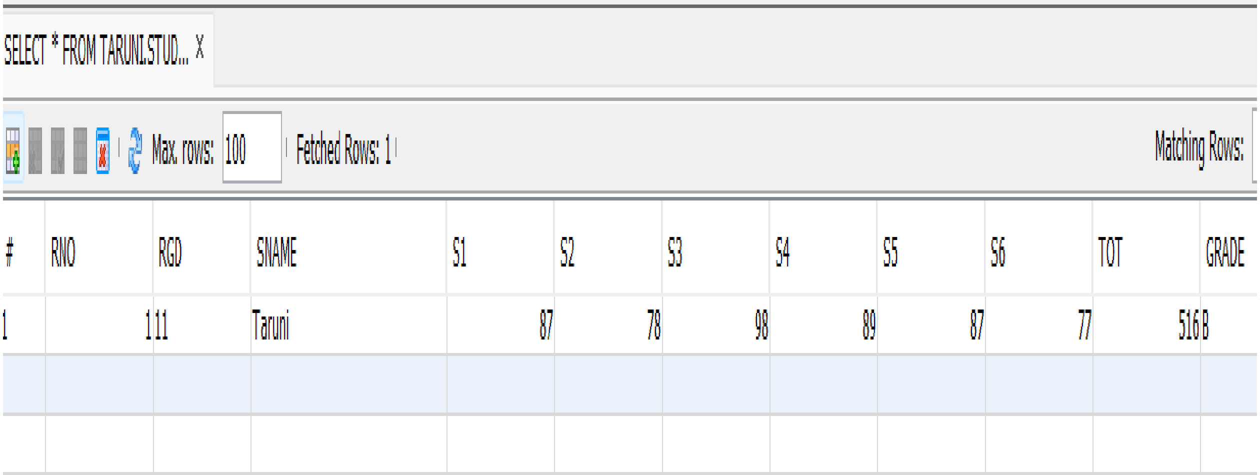
```
run:
-----MENU-----
0.exit
1.create table
2.insert data
3.adding new columns
4.updating new columns
5.Display data(before updating)
6.Display data(after updating)
7.deleting data
8.dropping data
Enter your choice:
1
Table created successfully

-----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:
2
Enter student 1 details:
Rno:
1
Regdno:
11
Sname:
Taruni
s1:
67
s2:
65
s3:
76
s4:
78
s5:
88
s6:
80
Enter student 2 details:
Rno:
2
Regdno:
22
Sname:
nandini
s1:
77
s2:
88
s3:
89
s4:
78
s5:
87
s6:
77
2 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:
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
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:1
Regdno:11
sname:Taruni
s1:67
s2:65
s3:76
s4:78
s5:88
s6:80
Rno:2
Regdno:22
sname:nandini
s1:77
s2:88
s3:89
s4:78
s5:87
s6:77

-----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:1
Regdno:11
sname:Taruni
s1:67
s2:65
s3:76
s4:78
s5:88
s6:80
Total:454
Grade:C
/nRno:2
Regdno:22
sname:nandini
s1:77
s2:88
s3:89
s4:78
s5:87
s6:77
Total:496
Grade:C
```

```
-----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)s
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
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:
0
BUILD SUCCESSFUL (total time: 3
minutes 31 seconds)
```


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team
Projects Files Services ×

```
Databases
  Java DB
  Drivers
  jdbc:derby://localhost:1527/Y22ACM473 [taruni on TARUNI]
    TARUNI
      Tables
        COOKIE
        LOGIN
        STUDENT_MARKS1
          RNO
          RGD
          SNAME
          S1
          S2
          S3
          S4
          S5
          S6
          Indexes
          Foreign Keys
        TABLE1
        USERDATA
      Views
      Procedures
```



#	RNO	RGD	SNAME	S1	S2	S3	S4	S5	S6	TOT	GRADE
1	111		Taruni	87	78	98	89	87	77	516	B

EXPERIMENT 02:

AIM: Write an application to demonstrate HTTP Servlets.

Execution steps:

1. **Create the Project:**
   - **File→NewProject→JavaWeb→Web Application →Next**

     Select **GlassFish Server → Finish**.
2. **Add a Servlet Class:**
   - Right-click on the project → **New →Servlet Class**.

     Name the servlet class to match the **action** attribute (e.g., LoginServlet).
3. **Set Up HTML Forms:**
   - Create separate HTML files for GET and POST methods, as shown below.

Source 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>
```

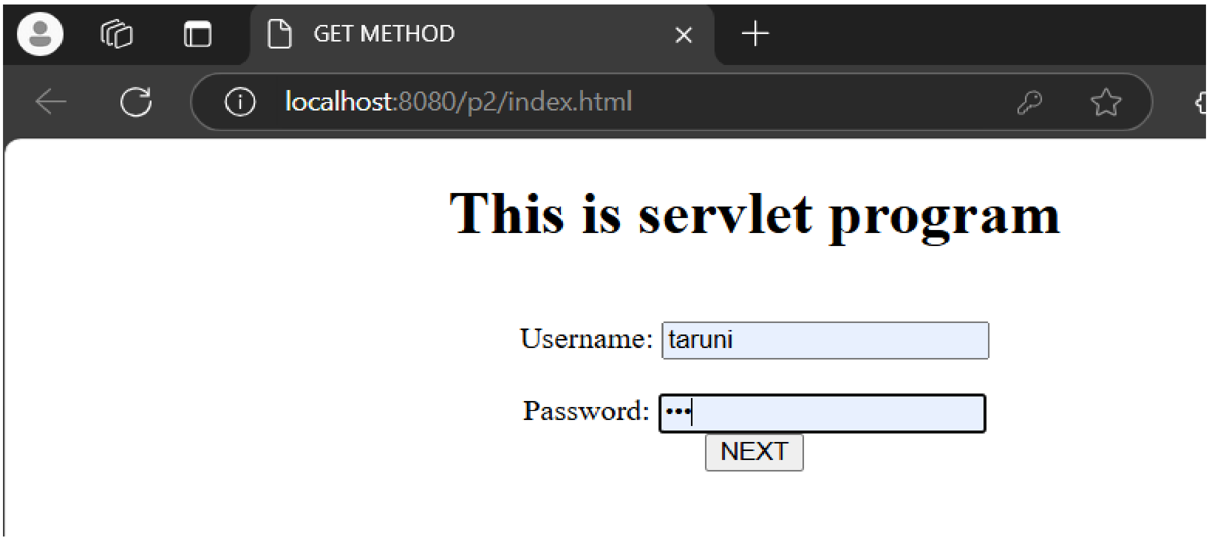
LoginServlet.java:

```
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>
```

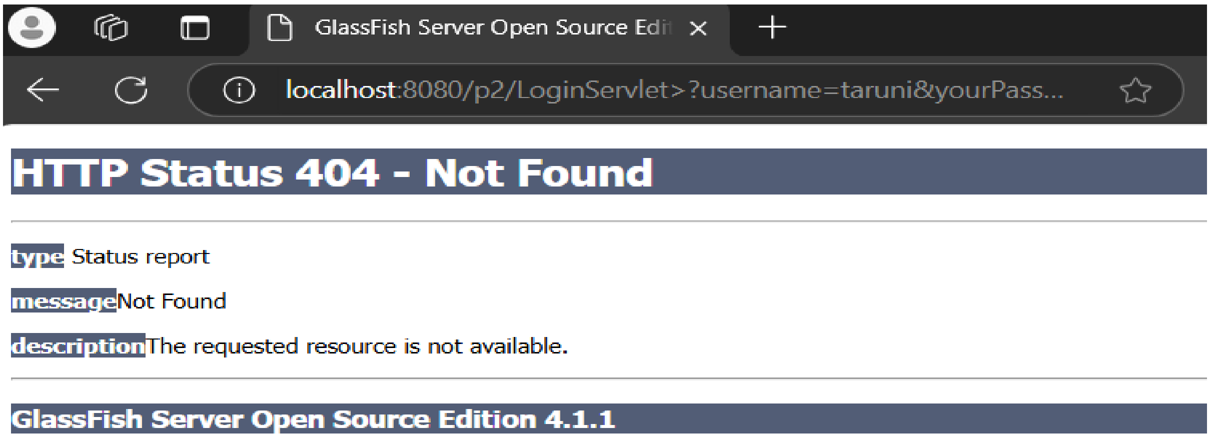
Output:



This is servlet program

Username: taruni

Password: •••


GlassFish Server Open Source Edi
localhost:8080/p2/LoginServlet>?username=taruni&yourPass...

HTTP Status 404 - Not Found

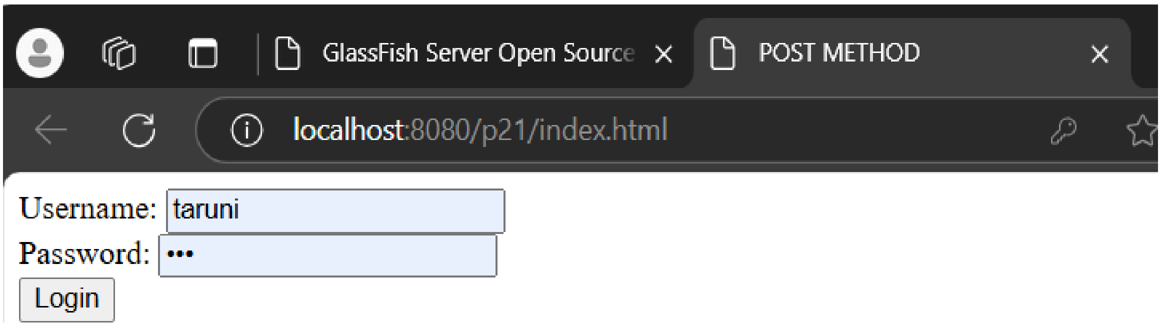
type Status report

messageNot Found

descriptionThe requested resource is not available.

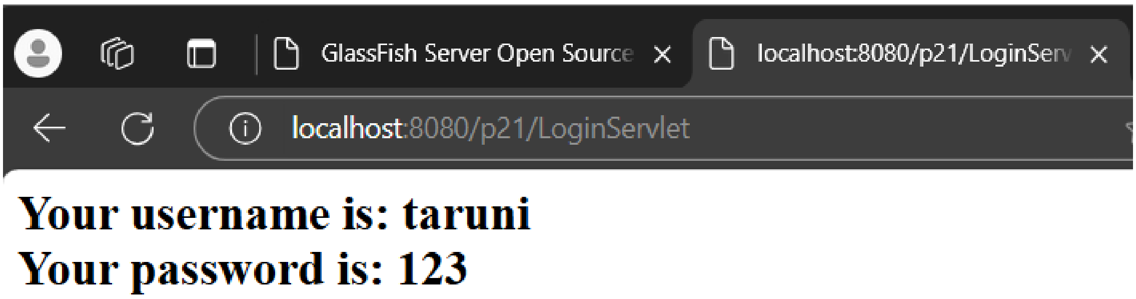
GlassFish Server Open Source Edition 4.1.1

Post method:



Username: taruni

Password: •••


GlassFish Server Open Source
localhost:8080/p21/LoginServ
localhost:8080/p21/LoginServlet

Your username is: taruni
Your password is: 123

EXPERIMENT 03:

Aim: Write an application to demonstrate Cookie & Sessions.

Execution Steps:

1. **Create a Project:**
    - Open NetBeans → **File** → **New Project** → **Java Web** → **web Application** → Enter **Project Name** → Click **Finish**.
2. **Add Servlets:**
    - Right-click **Source Packages** → **New** →**Servlet.**
    - Name the servlets: NewServlet, NewServlet1, brand, checkedout, NewServlet2, NewServlet3, pay_money,payment_page.
    - Paste the corresponding source codes into each servlet file.
3. **Set Up Database Connection:**
    - Install **GlassFish Server** and place it in your Java folder.
    - Add **Java DB library** under Libraries.
    - Confirm the paths for **Java DB** and the database location in Properties.
4. **Create a Database:**
    - In **Services** → **Databases**, start the **Java DB server.**
    - Create a database with username, password, and name.
5. **Create Tables:**
    - userdata table: Columns - username, password, amount, id.
    - cookie table: Columns - name, model, version, amount, id.
6. **Connect to the Database:**
    - Right-click your database in **Services** → **Databases** → Select **Connect.**
7. **Build and Run:**
    - Right-click your **project** → **Select Clean and Build**
    - Run the project and check the output in the Output section.
8. **View Data:**
    - Expand Tables under your database in **Services** → **Databases**

- Right-click a table → Select **View Data** to see its content.

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/Y22ACM473","taruni",
"taruni");
      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";
    }
}
```

NewServlet1.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/Y22ACM473","taruni",
"taruni");
        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 cookies 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);
        }
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @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/Y22ACM473","taruni",
"taruni");
        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/Y22ACM473","taruni",
"taruni");
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";
    }
}
```

<u>NewServlet2.java:</u>

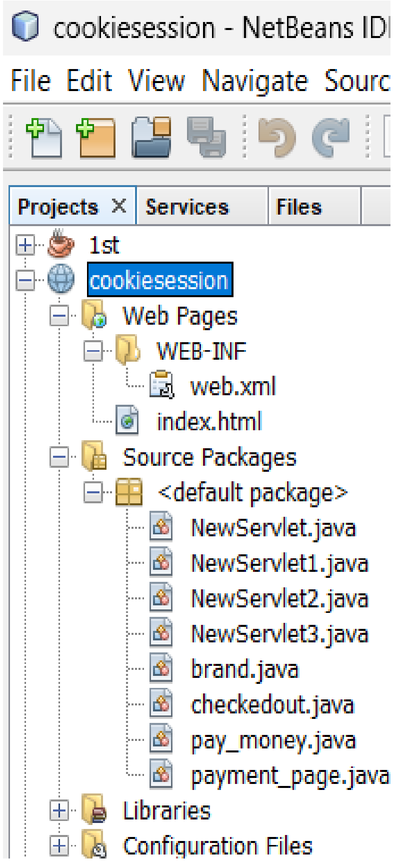
```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
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";
    }
}
```

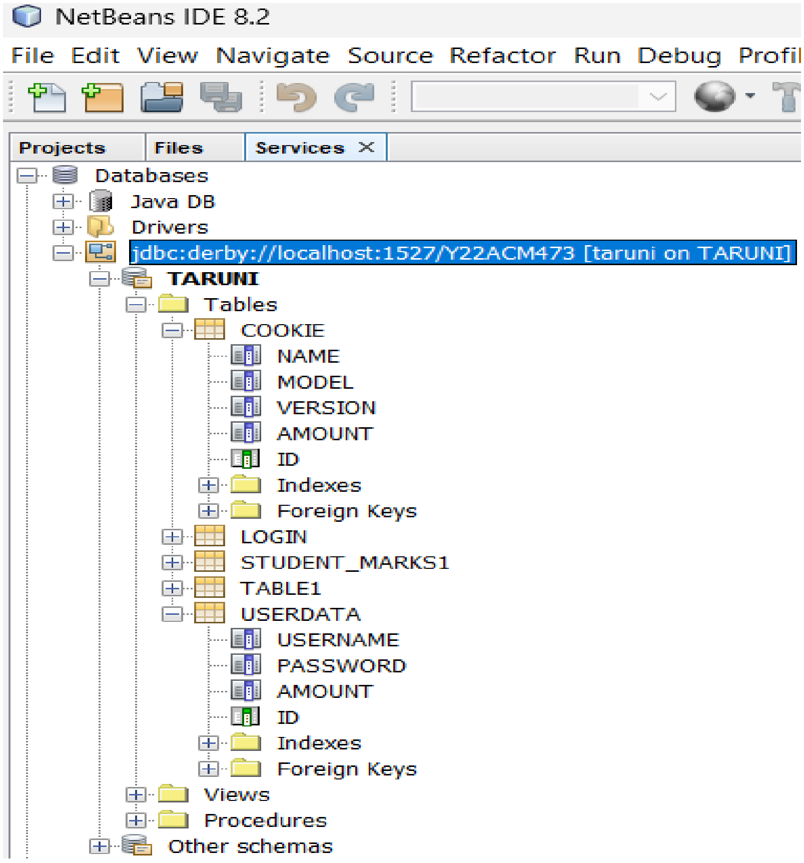
NewServlet3.java:

```
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";
    }

}
```

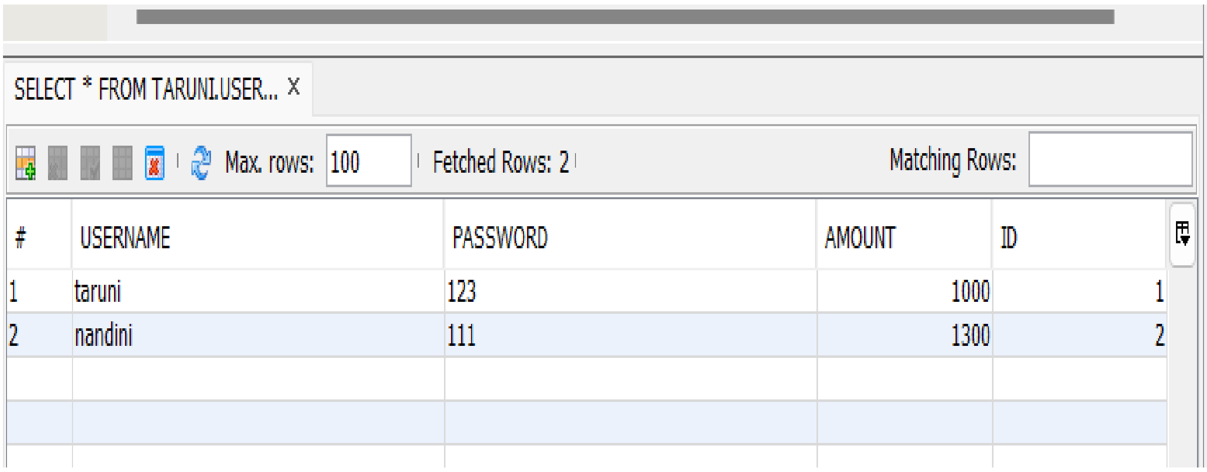
Output:


cookiesession - NetBeans IDI
File Edit View Navigate Sourc
Projects × Services Files

```
⊞ 1st
⊟ cookiesession
   ⊟ Web Pages
      ⊟ WEB-INF
            web.xml
         index.html
   ⊟ Source Packages
      ⊟ <default package>
            NewServlet.java
            NewServlet1.java
            NewServlet2.java
            NewServlet3.java
            brand.java
            checkedout.java
            pay_money.java
            payment_page.java
   ⊞ Libraries
   ⊞ Configuration Files
```

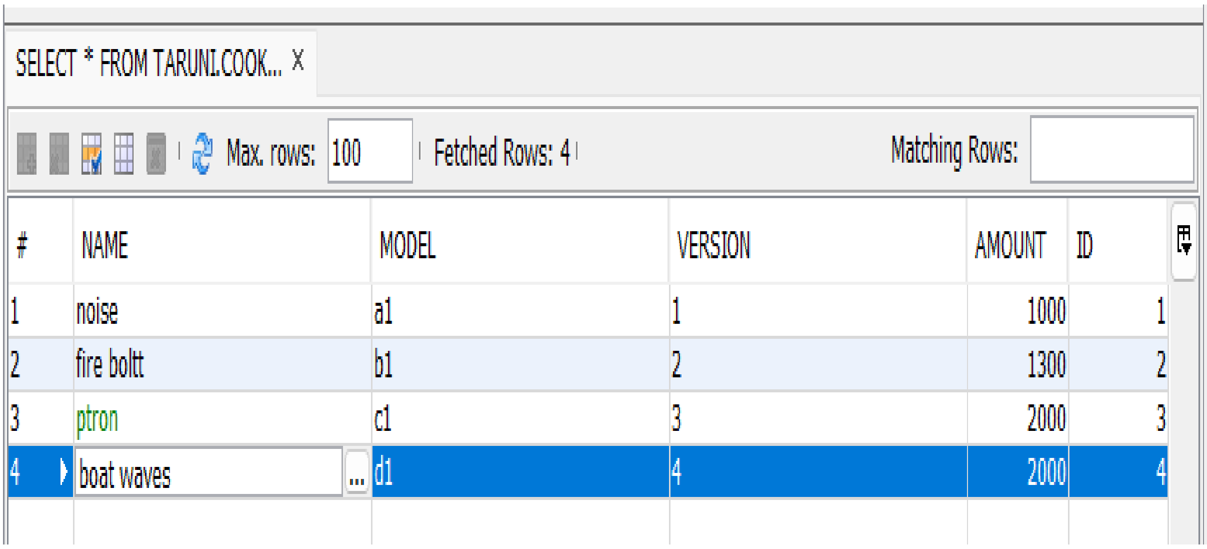

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profil
Projects Files Services ×

```
⊟ Databases
   ⊞ Java DB
   ⊞ Drivers
   ⊟ jdbc:derby://localhost:1527/Y22ACM473 [taruni on TARUNI]
      ⊟ TARUNI
         ⊟ Tables
            ⊟ COOKIE
                  NAME
                  MODEL
                  VERSION
                  AMOUNT
                  ID
               ⊞ Indexes
               ⊞ Foreign Keys
            ⊞ LOGIN
            ⊞ STUDENT_MARKS1
            ⊞ TABLE1
            ⊟ USERDATA
                  USERNAME
                  PASSWORD
                  AMOUNT
                  ID
               ⊞ Indexes
               ⊞ Foreign Keys
         ⊞ Views
         ⊞ Procedures
      ⊞ Other schemas
```



Max. rows: 100 Fetched Rows: 2 Matching Rows:

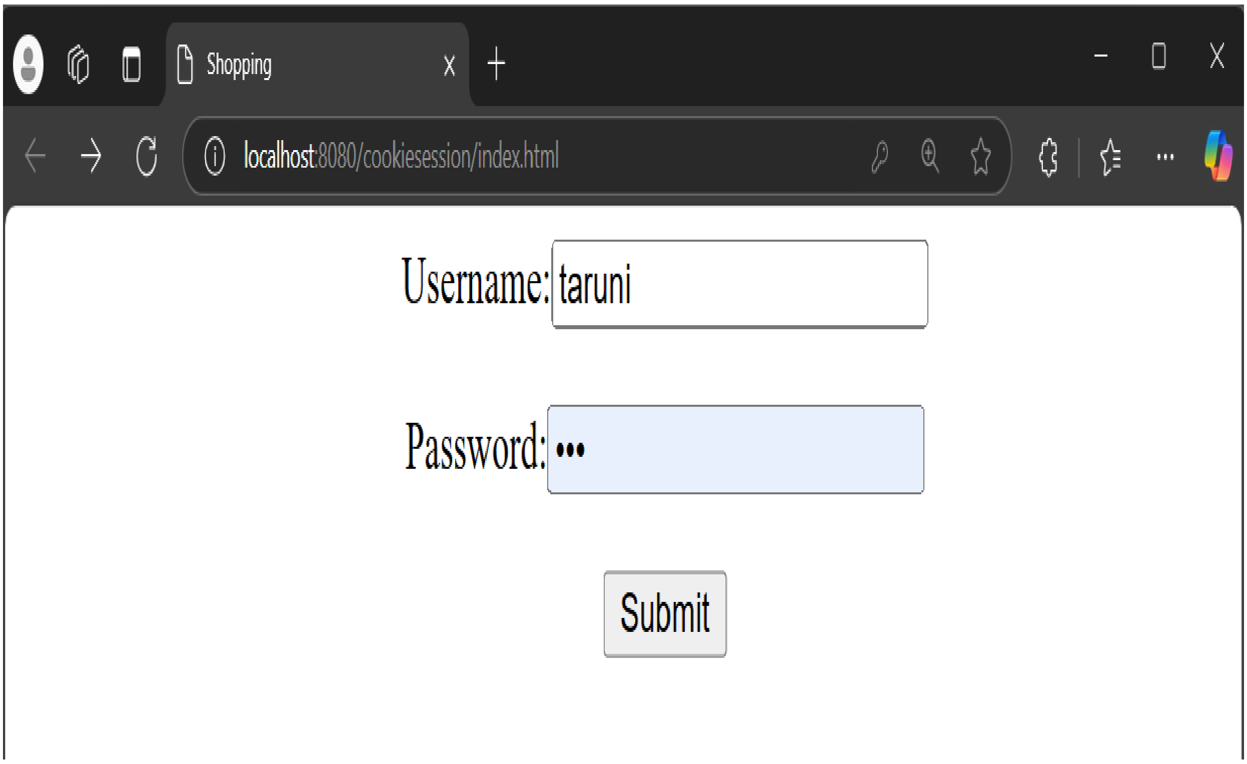
#	USERNAME	PASSWORD	AMOUNT	ID
1	taruni	123	1000	1
2	nandini	111	1300	2



SELECT * FROM TARUNI.COOK... X

Max. rows: 100 Fetched Rows: 4 Matching Rows:

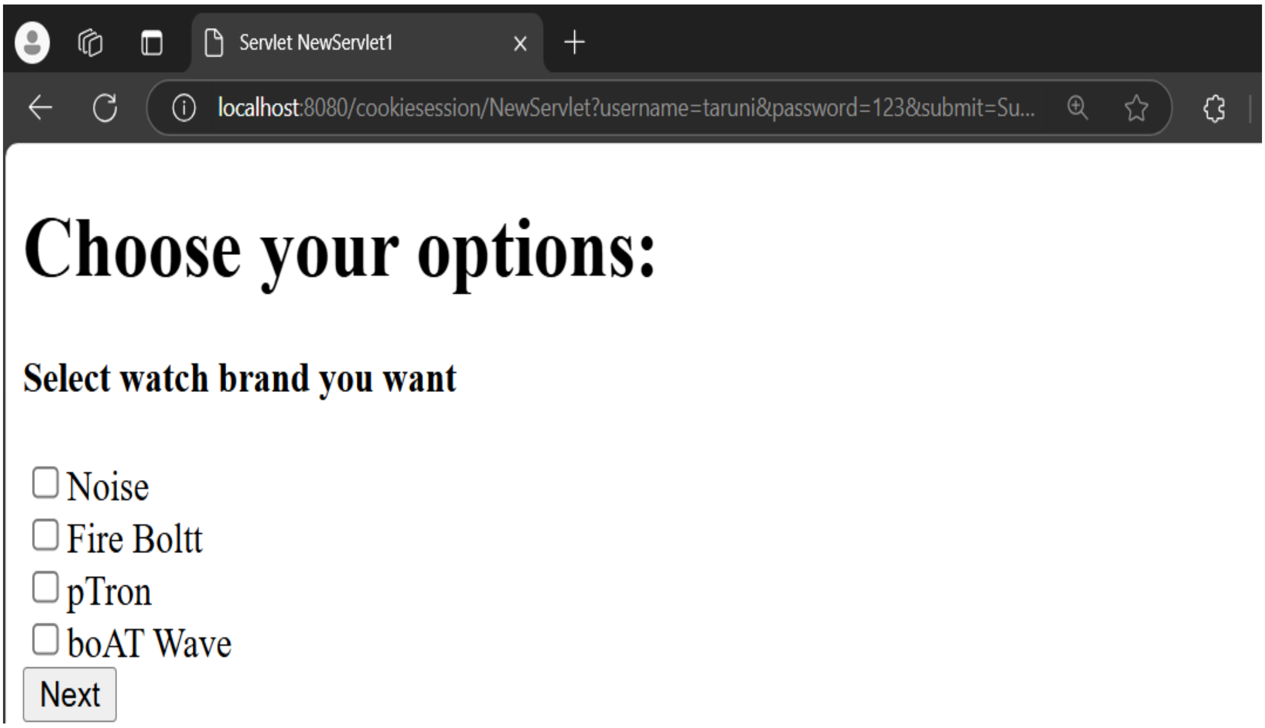
#	NAME	MODEL	VERSION	AMOUNT	ID
1	noise	a1	1	1000	1
2	fire boltt	b1	2	1300	2
3	ptron	c1	3	2000	3
4	boat waves	d1	4	2000	4



localhost:8080/cookiesession/index.html

Username: taruni

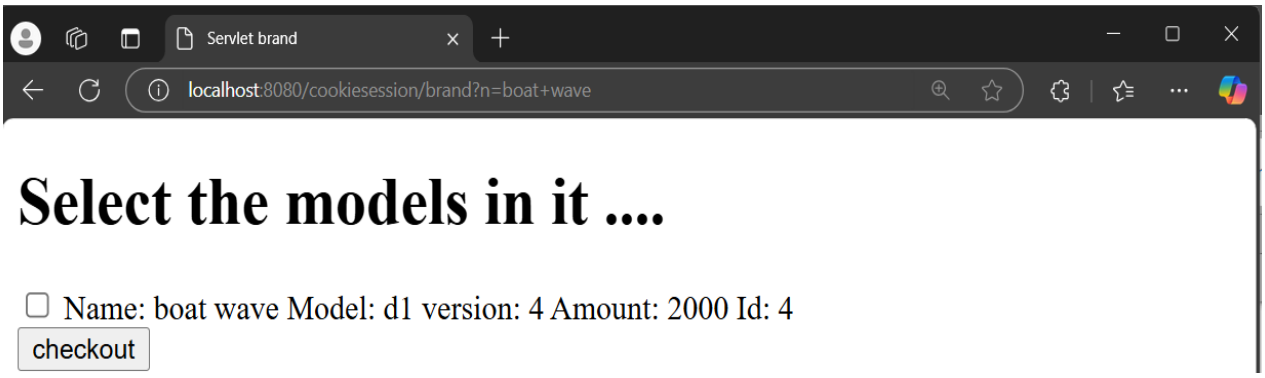
Password: •••


Servlet NewServlet1
localhost:8080/cookiesession/NewServlet?username=taruni&password=123&submit=Su...

Choose your options:

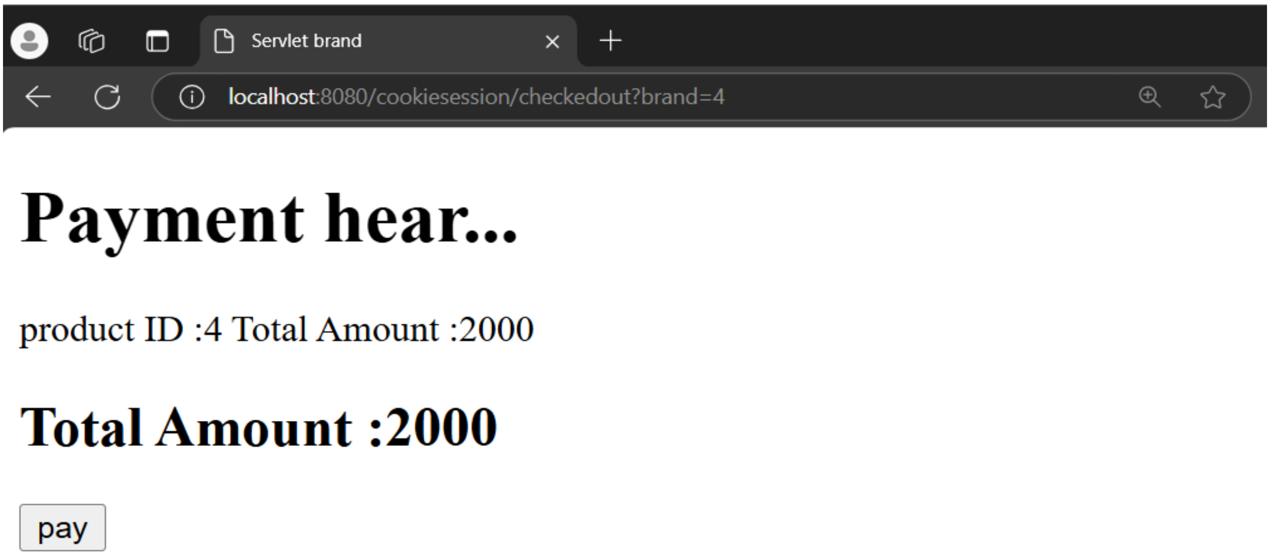
Select watch brand you want

□ Noise
□ Fire Boltt
□ pTron
□ boAT Wave
Next



Select the models in it ....

□ Name: boat wave Model: d1 version: 4 Amount: 2000 Id: 4
checkout

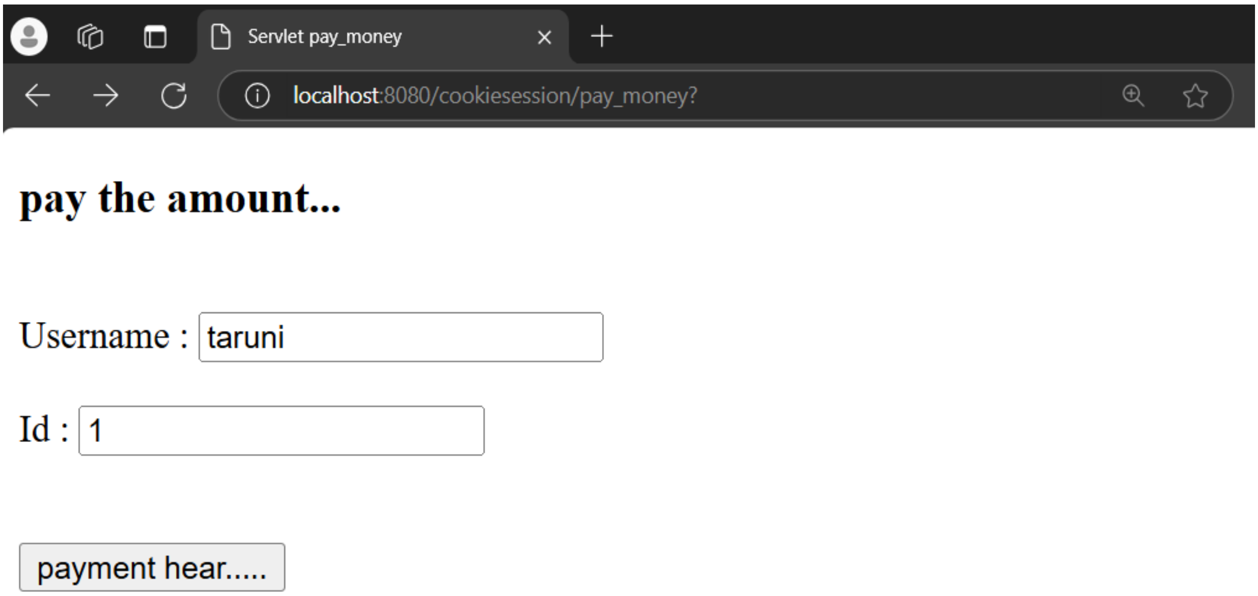


Payment hear...

product ID :4 Total Amount :2000

Total Amount :2000

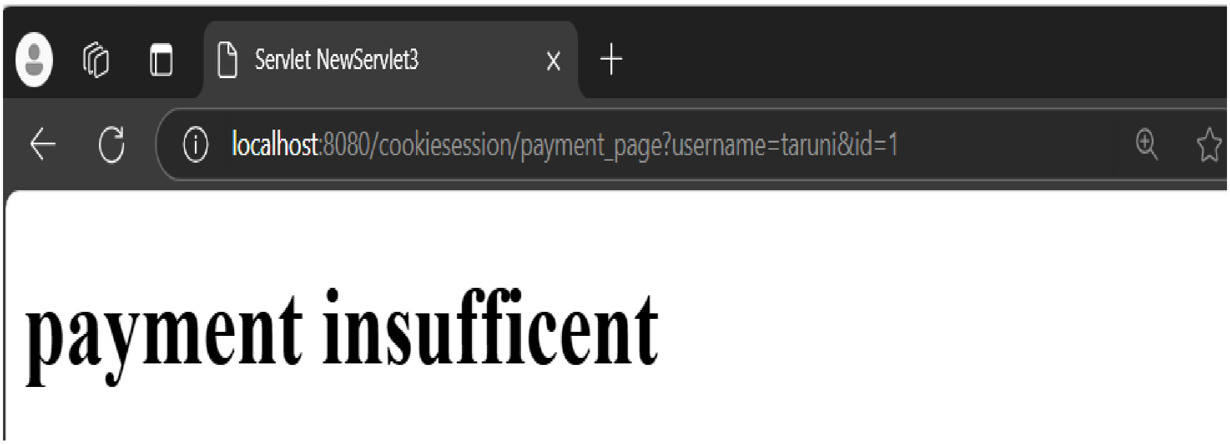
pay



pay the amount...

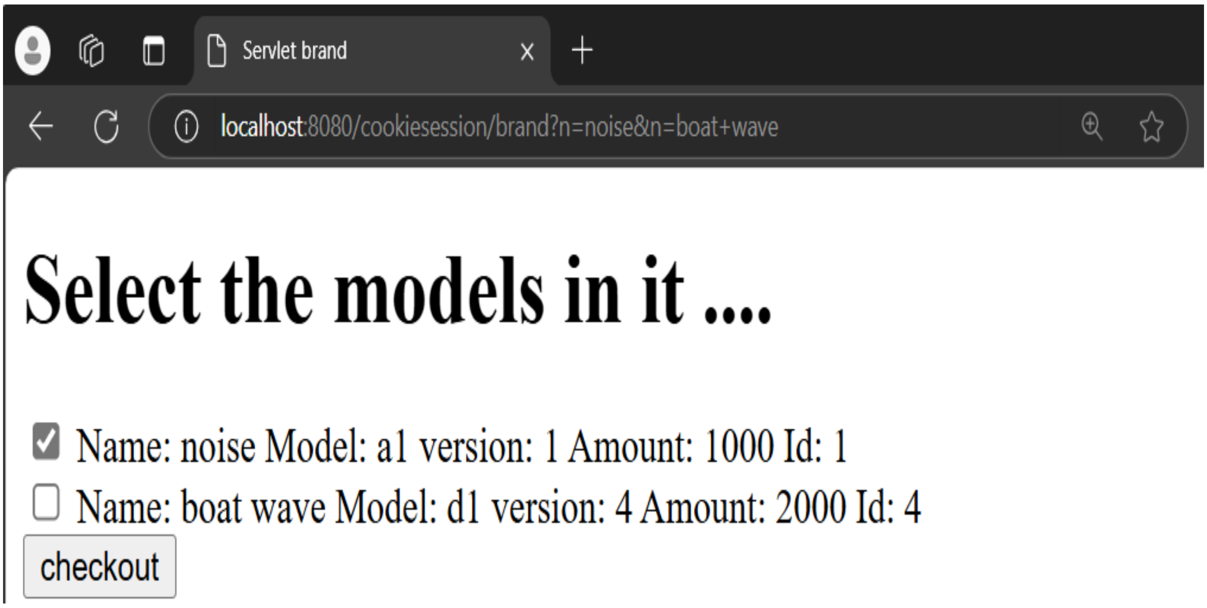
Username : taruni

Id : 1


Servlet NewServlet3
localhost:8080/cookiesession/payment_page?username=taruni&id=1

payment insufficent

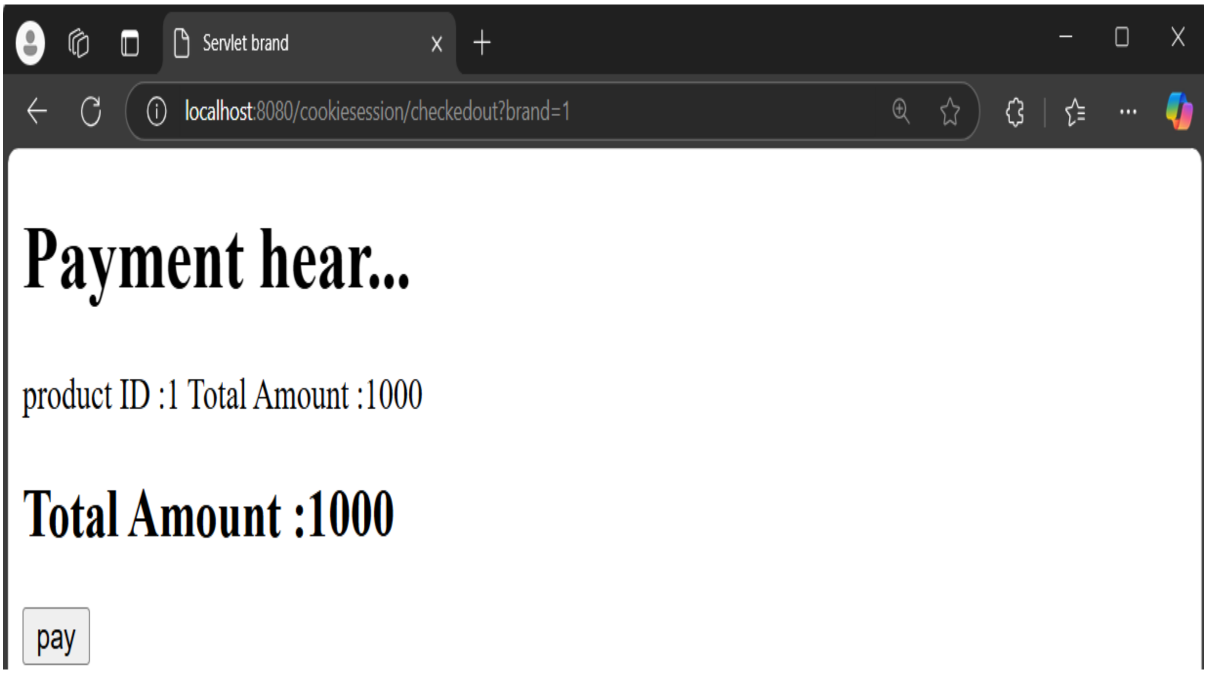
Output2:



Select the models in it ....

☑ Name: noise Model: a1 version: 1 Amount: 1000 Id: 1

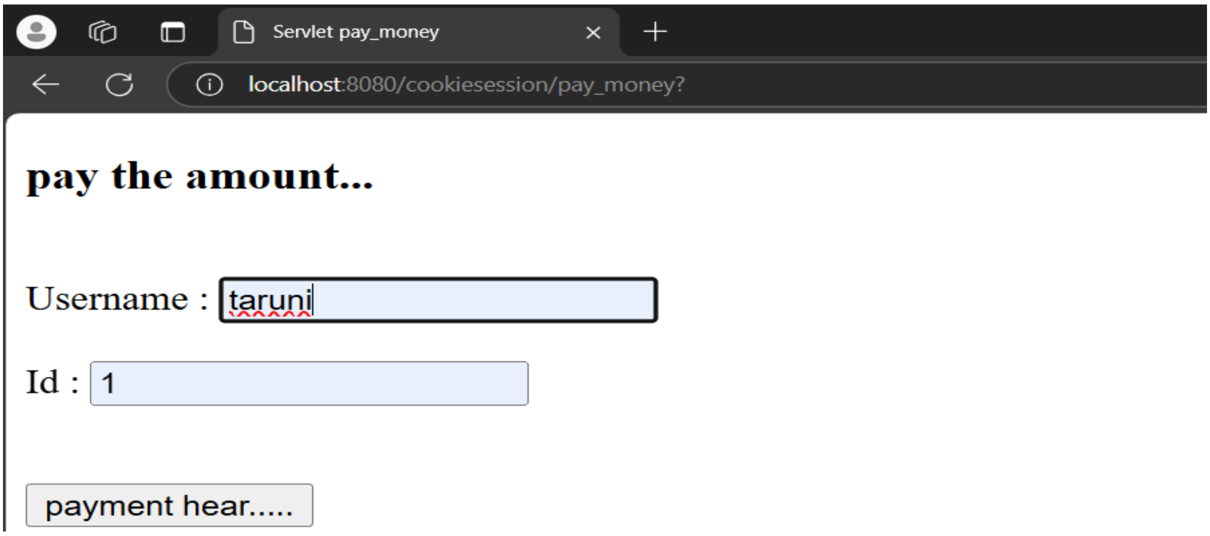
□ Name: boat wave Model: d1 version: 4 Amount: 2000 Id: 4



Payment hear...

product ID :1 Total Amount :1000

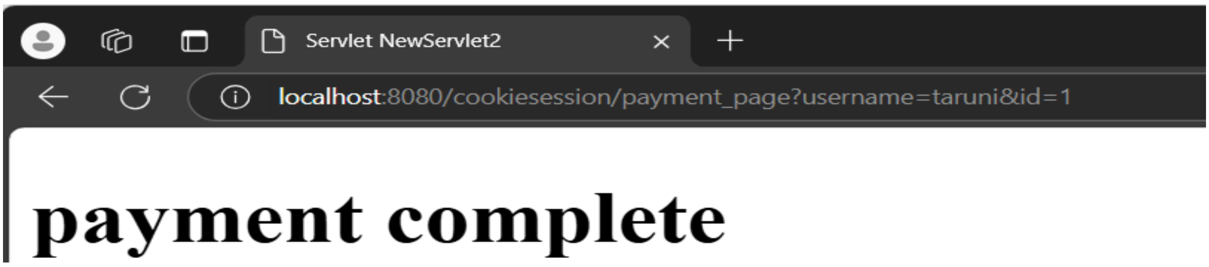
Total Amount :1000



pay the amount...

Username : taruni

Id : 1


Servlet NewServlet2
localhost:8080/cookiesession/payment_page?username=taruni&id=1

payment complete

EXPERIMENT 04:

Aim: Write an application to integrate JSP & Servlets.

Execution Steps:

1. **Create a Project:**
   - Open **NetBeans → File → New Project → Java Web →Web Application →** Enter a **Project Name** → Click **Finish.**
2. **Add Servlets and JSP Files:**
   - Right-click **Source Packages →New →Servlet** →Name it as **Login.**
   - Create another servlet named **NewServlet.** Paste the provided source code.
   - Right-click **Source Packages →New →JSP File** →Name them as **details, index, and register.**
   - Create a **new package** named **t** → Add a **servlet** named **bean.**
3. **Set Up Database Connection:**
   - Install **GlassFish Server** and place it in your Java folder.
   - Add **Java DB library** under **Libraries.**
   - Verify the paths for **Java DB** and the database location in Properties.
4. **Create and Connect Database:**
   - In **Services →Databases,** start the **Java DB server.**
   - Create a database with a username, password, and name.
   - Create a **Login table** with columns: name, username, password, regd, gender, branch.
   - Right-click the **database** and select **Connect.**
5. **Build and Run the Project:**
   - Right-click your **project** → Select **Clean and Build.**
   - Click **Run** to view the output in the Output section.
6. **View Data in the Table:**
   - In **Services→Databases,** expand your database then Expand Tables.
   - Right-click the **Login table** → Select **View Data** to display its content**.**

Source Code:

index.jsp:

```
<%@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="Servlet">
         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/Y22ACM473", "taruni",
"taruni");
```

```
         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:

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

bean.java:

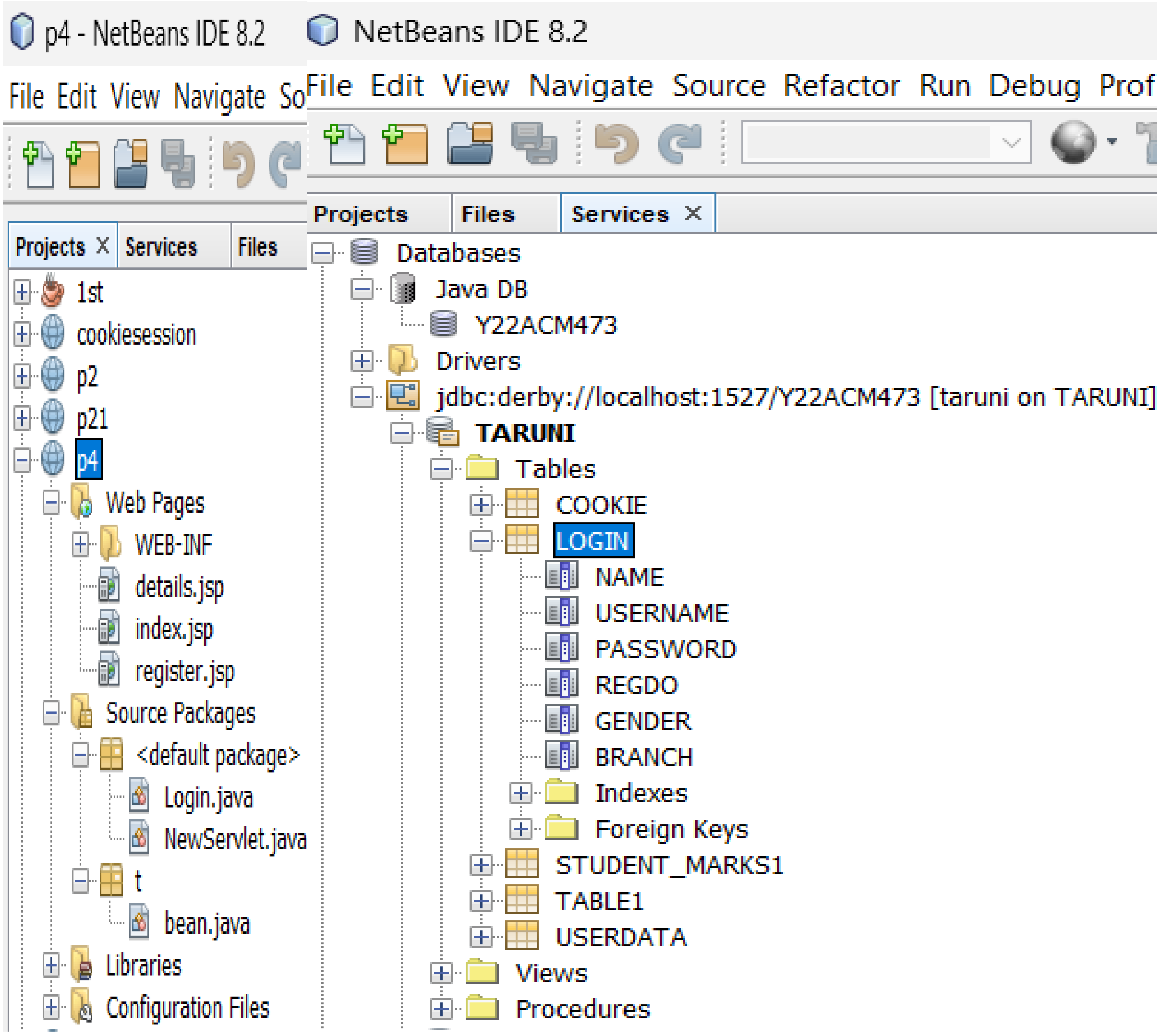
```
package t;
public class bean
{
    private String name,username,pass,regd,gender,branch;
    public String getName()
    {
        return name;
    }
    public void setName(String name)
    {
        this.name=name;
    }
    public String getUname()
    {
        return username;
    }
    public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

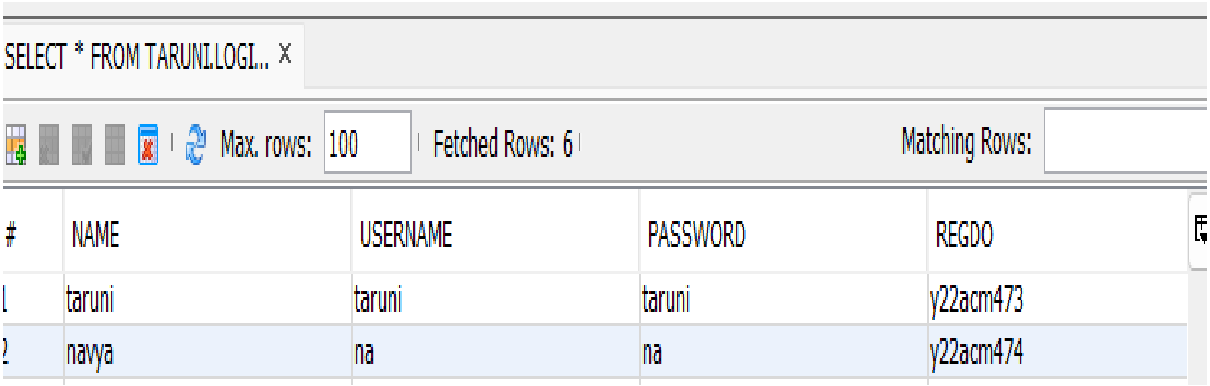
details.jsp:

```
<%@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/Y22ACM473","taruni",
"taruni");
        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>
```

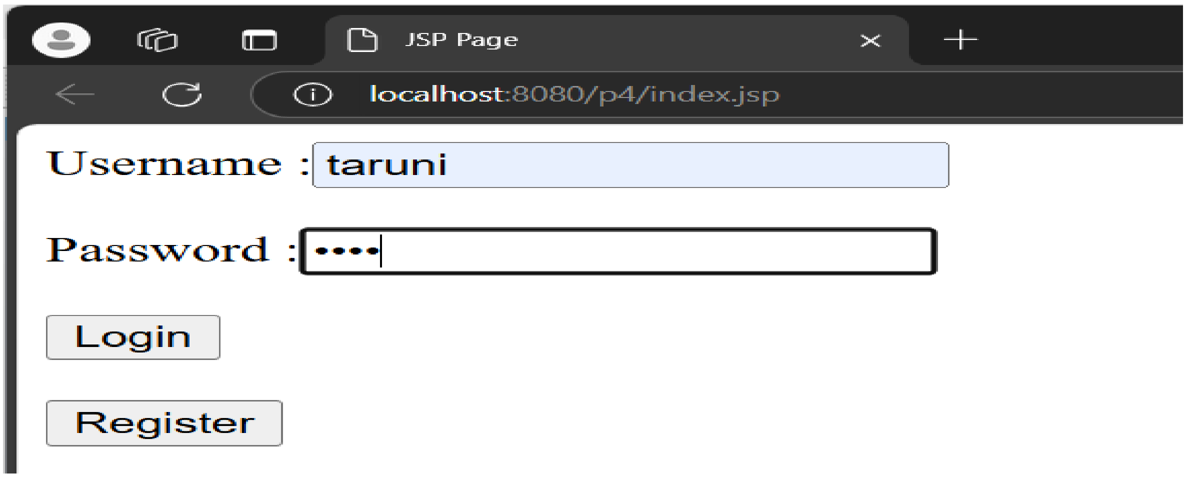
Output:





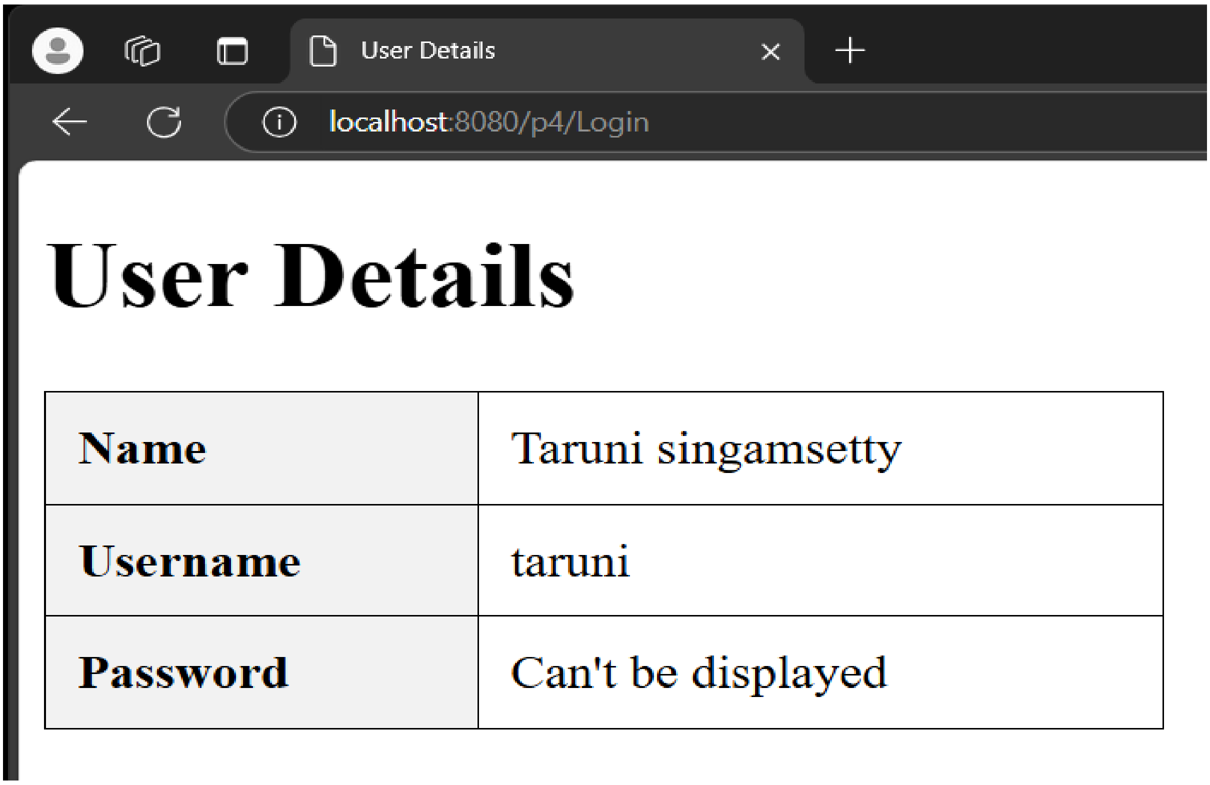
Max. rows: 100 Fetched Rows: 6 Matching Rows:

#	NAME	USERNAME	PASSWORD	REGDO
1	taruni	taruni	taruni	y22acm473
2	navya	na	na	y22acm474



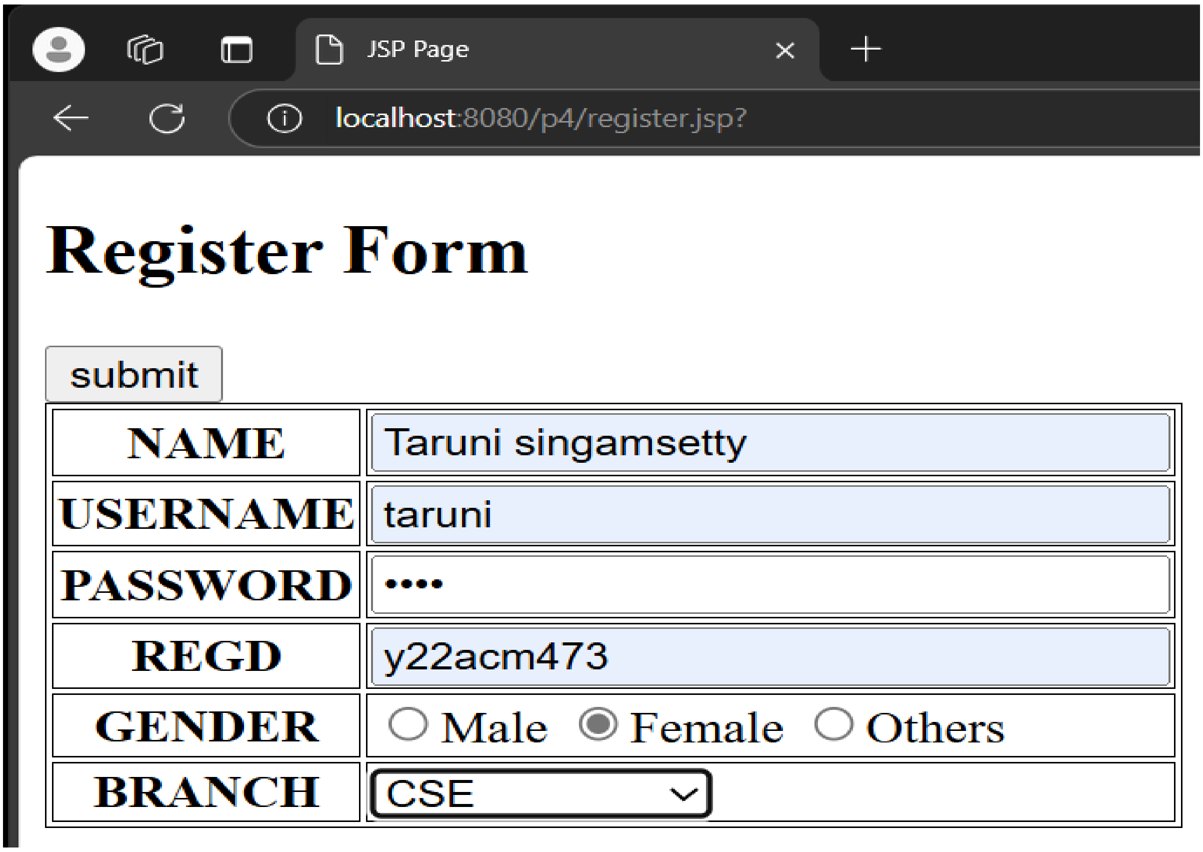
Username : taruni

Password : ••••



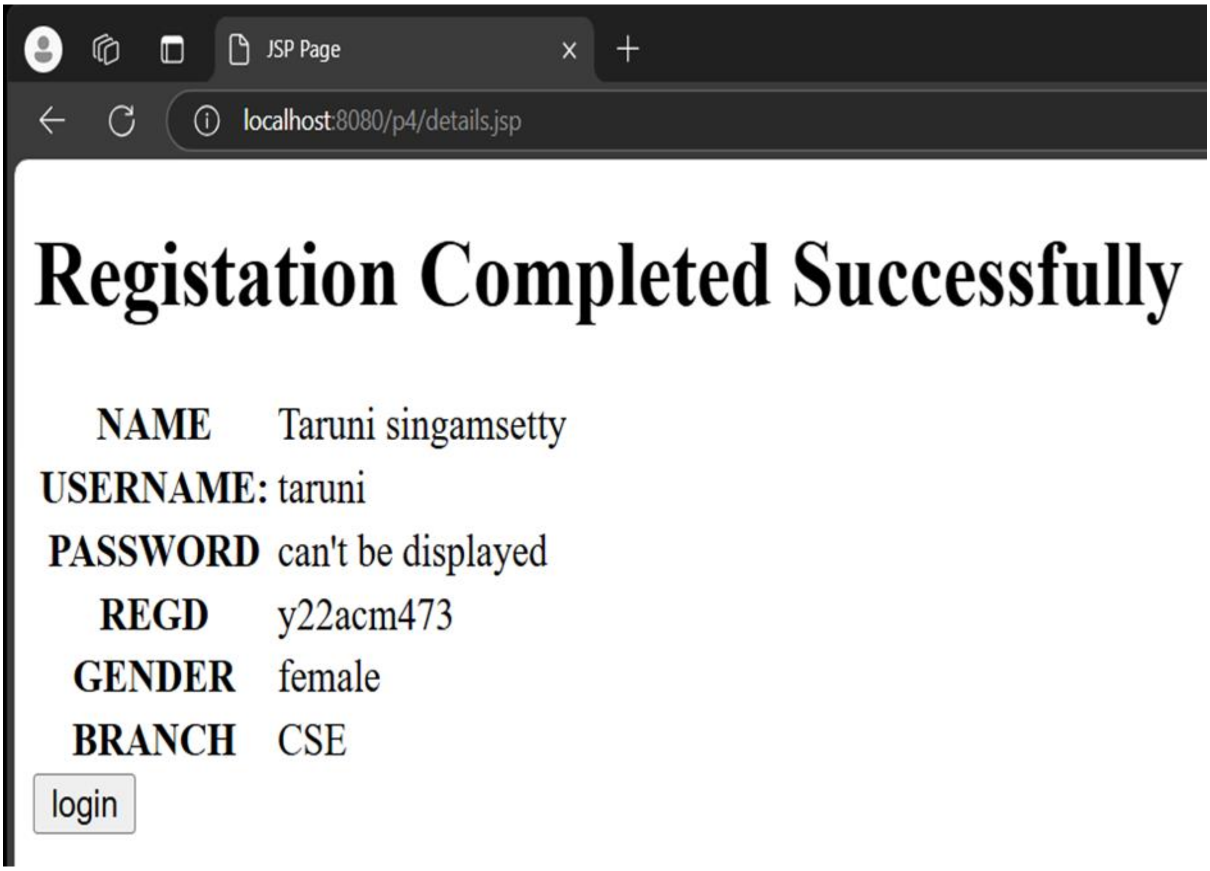
User Details

Name	Taruni singamsetty
Username	taruni
Password	Can't be displayed



Register Form

NAME	Taruni singamsetty
USERNAME	taruni
PASSWORD	••••
REGD	y22acm473
GENDER	○ Male ● Female ○ Others
BRANCH	CSE



Registation Completed Successfully

NAME	Taruni singamsetty
USERNAME:	taruni
PASSWORD	can't be displayed
REGD	y22acm473
GENDER	female
BRANCH	CSE

EXPERIMENT 05:

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

Execution Steps:

1. **Create a Project:**
   - **File→New Project →Java Web→Web Application** → Enter **Project** Name → **Finish.**
2. **Add Java File:**
   - Right-click **Source Packages→New →Java File** → Paste the provided source code.
3. **Set Up Database:**
   - Install **GlassFish Server** and add **Java DB** under **Libraries.**
   - Verify paths in Properties.
4. **Create Database and Table:**
   - Start **Java DB server** in **Services→Databases.**
   - Create a database and a table with columns: regno, name, s1, s2, s3, s4, s5, s6, total=0, grade=0.
   - **Connect** to the **database.**
5. **Add Tag Handler:**
   - **Project→New→Others→Taghandler** → Set Package to **aaa** → **Path: Web-INF/tlds → Finish.**
6. **Build and Run:**
   - Right-click the project → **Clean and Build → Run.**
7. **View Data:**
   - Expand Tables under your database →Right-click the **project →View Data.**

Source Code:

index.html:

```
<%@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:

```
<%@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/Y22ACM473" user="taruni" password="taruni”/>
    <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:NewTagHandler 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.html">
      <input type='submit' value="Login page">
    </form>
 </body>
```

```
</html>
```

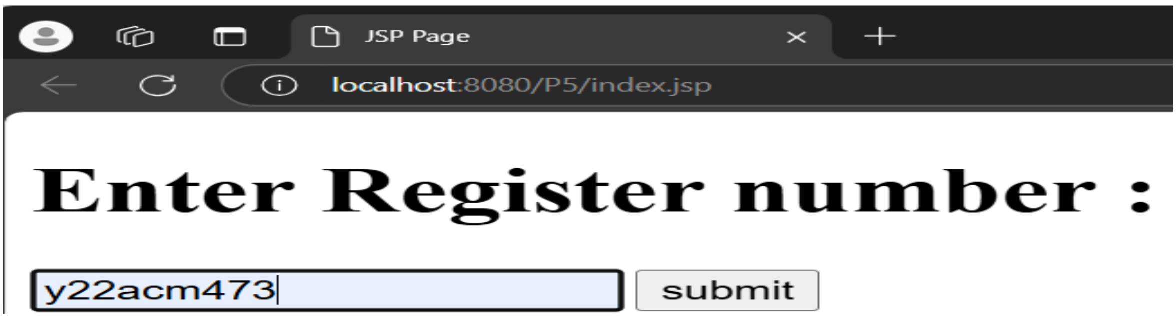
newtag_lib :

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

TagNewHandler1 :

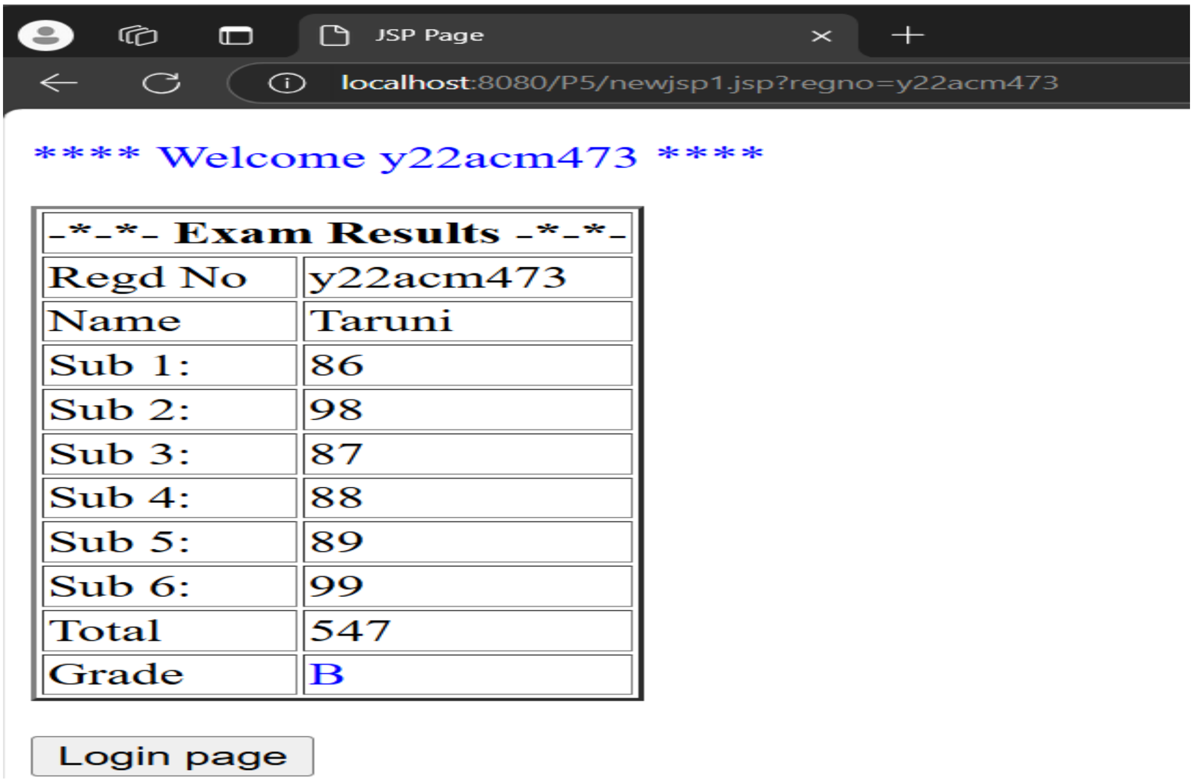
```
package aaa;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.JspException;
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;
    /**
     * Called by the container to invoke this tag. The implementation of this
     * method is provided by the tag library developer, and handles all tag
     * processing, body iteration, etc.
     * @throws javax.servlet.jsp.JspException
     * @throws java.io.IOException
     */
    @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/Y22ACM473","taruni",
"taruni");
            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:



Enter Register number :

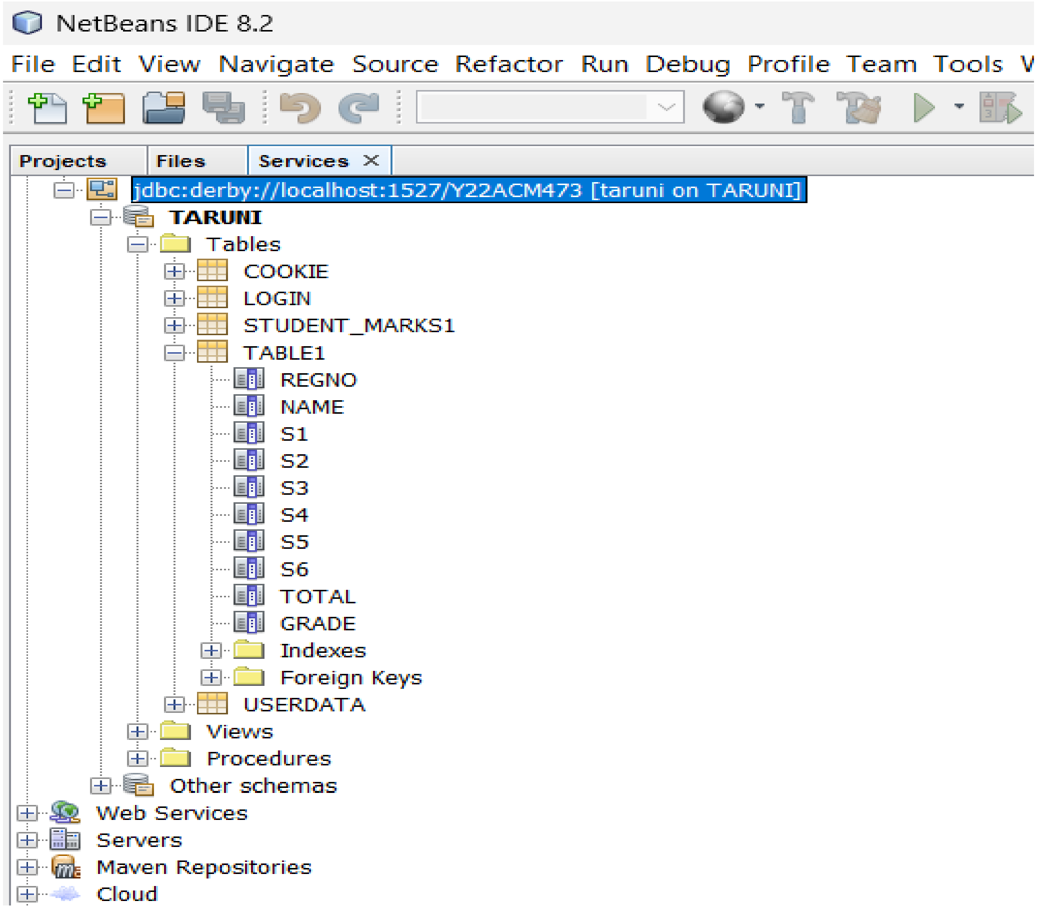
y22acm473 submit



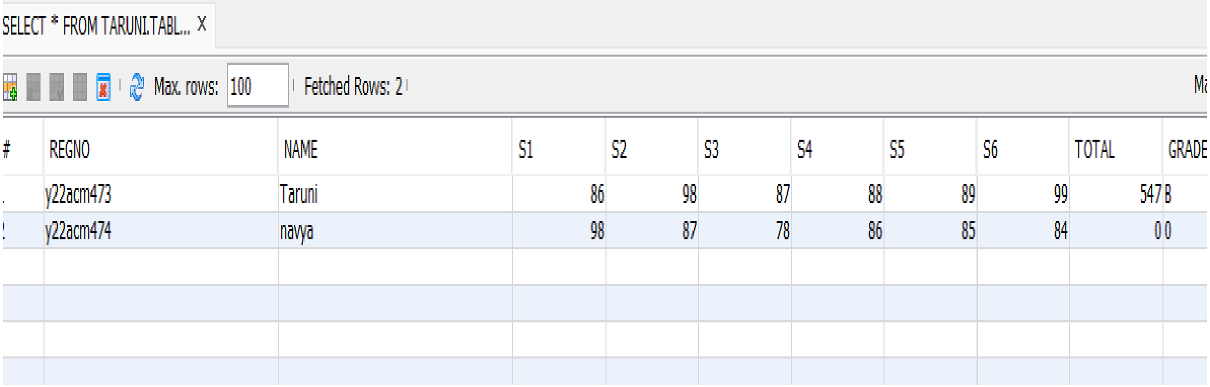
**** Welcome y22acm473 ****

-*-*- Exam Results -*-*-	
Regd No	y22acm473
Name	Taruni
Sub 1:	86
Sub 2:	98
Sub 3:	87
Sub 4:	88
Sub 5:	89
Sub 6:	99
Total	547
Grade	B

Login page


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools V
Projects
Files
Services

```
jdbc:derby://localhost:1527/Y22ACM473 [taruni on TARUNI]
    TARUNI
        Tables
            COOKIE
            LOGIN
            STUDENT_MARKS1
            TABLE1
                REGNO
                NAME
                S1
                S2
                S3
                S4
                S5
                S6
                TOTAL
                GRADE
                Indexes
                Foreign Keys
            USERDATA
        Views
        Procedures
    Other schemas
Web Services
Servers
Maven Repositories
Cloud
```



#	REGNO	NAME	S1	S2	S3	S4	S5	S6	TOTAL	GRADE
1	y22acm473	Taruni	86	98	87	88	89	99	547	B
2	y22acm474	navya	98	87	78	86	85	84	0	0

EXPERIMENT 06:

Aim: Write an application to demonstrate Java Server Faces Validators , Event handlers and convertors.

Execution Steps:

1. **Create Project:**
   - **File→New Project→ Java Web→ Web Application** → Enter **Project Name → Finish.**
2. **Add Servlets:**
   - Right-click **Source Packages→ New→ Servlet→** Name them **email, phn converter, regno** → Paste the code**.**
3. **Set Up Database:**
   - Install **GlassFish Server** and add **Java DB library.**
   - Verify paths in Properties.
4. **Add JSP Files:**
   - Create **index** and **preview** files → Paste the code.
5. **Build and Run:**
   - Right-click the project → **Clean and Build → Run.**
6. **View Data:**
   - Expand Tables in Services→Databases → View your table.

Source code:

index.xhtml :

```
<?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>Registration Page</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>
```

preview.xhtml:

```
<?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>
```

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;
    }
}
```

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:

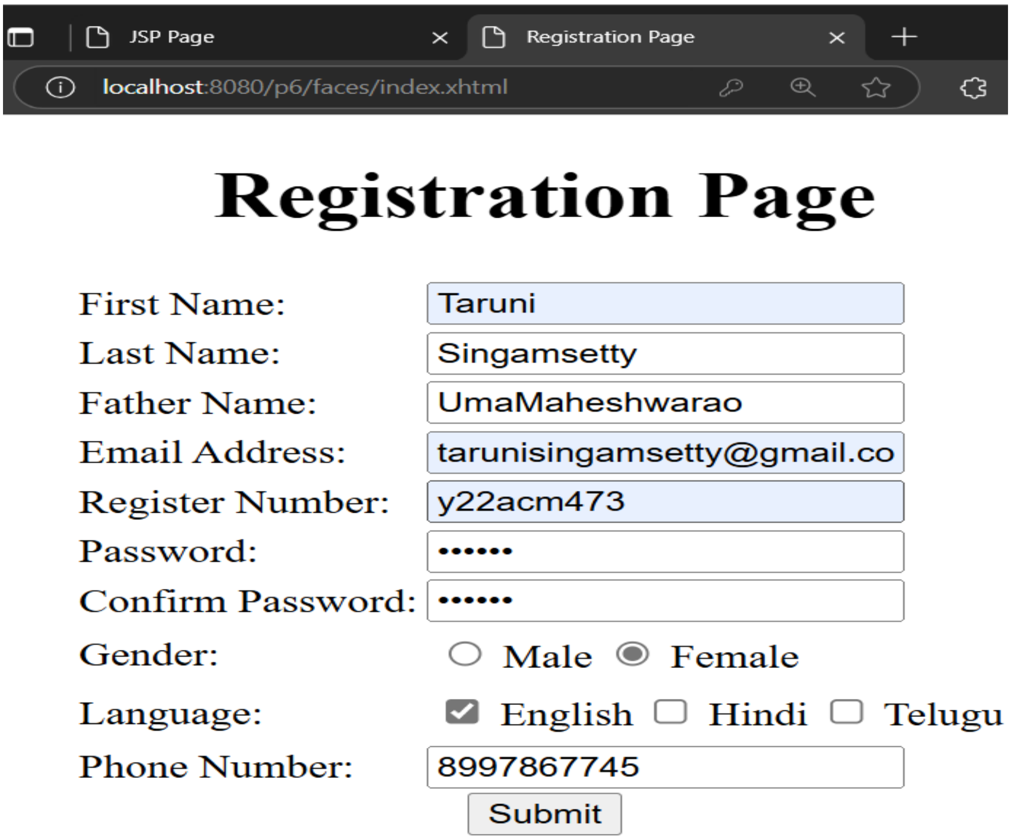
```
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import javax.faces.convert.FacesConverter;
import java.util.logging.Level;
import java.util.logging.Logger;
@FacesConverter("phn")
public class PhnConverter implements Converter {
   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}");
    }
}
```

regdno.java:

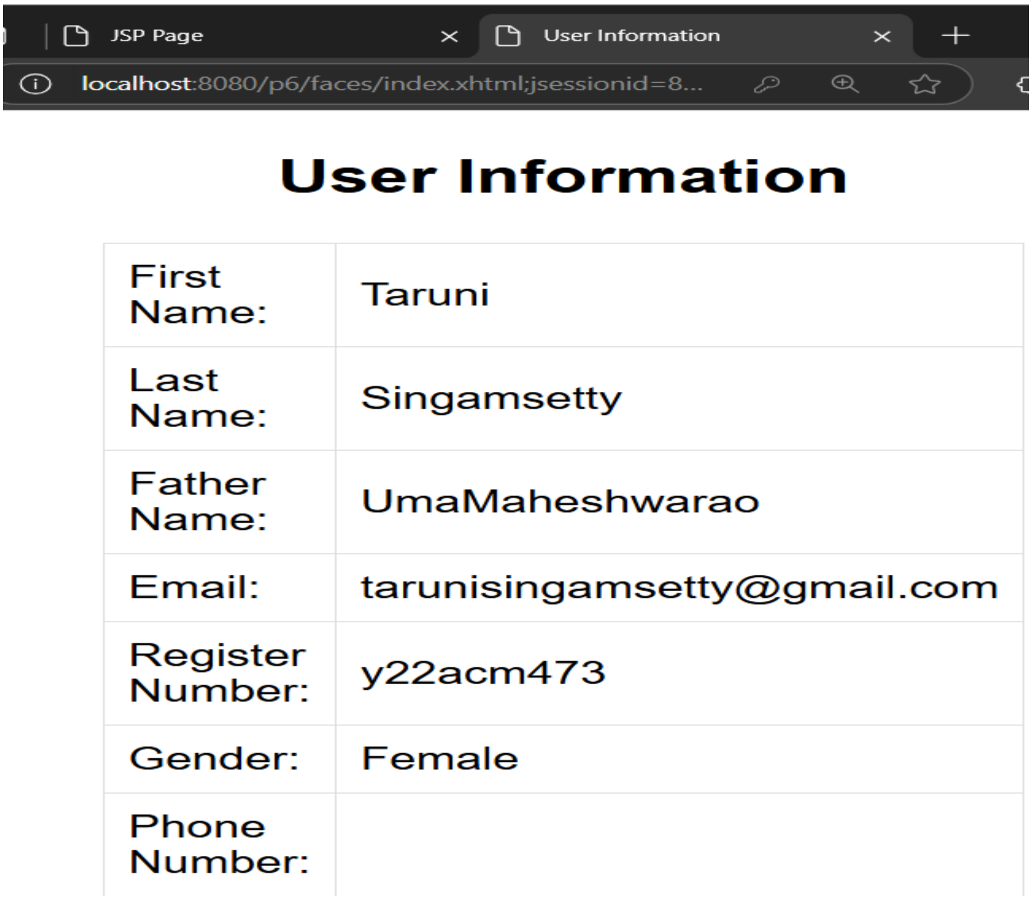
```
import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.validator.FacesValidator;
import javax.faces.validator.Validator;
import javax.faces.validator.ValidatorException;
@FacesValidator("regd")
public class 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("[A-Z]\\d{2}[A-Z]{3}\\d{3}")) {
            FacesMessage msg = new FacesMessage("Invalid Register Number");
            msg.setSeverity(FacesMessage.SEVERITY_ERROR);
            throw new ValidatorException(msg);
        }
    }
}
```

Output:



Registration Page

First Name:	Taruni
Last Name:	Singamsetty
Father Name:	UmaMaheshwarao
Email Address:	tarunisingamsetty@gmail.co
Register Number:	y22acm473
Password:	••••••
Confirm Password:	••••••
Gender:	○ Male ● Female
Language:	☑ English □ Hindi □ Telugu
Phone Number:	8997867745



User Information

First Name:	Taruni
Last Name:	Singamsetty
Father Name:	UmaMaheshwarao
Email:	tarunisingamsetty@gmail.com
Register Number:	y22acm473
Gender:	Female
Phone Number:	

EXPERIMENT 07:

Aim: Write an application to demonstrate Web Services.

Execution Steps:

1. **Create Project:**
   - **File→New Project→Java Web→ Web Application**→Enter **Project Name → Finish**.
2. **Add Servlets:**
   - Right-click **Source Packages→New →Servlet** → Name them **Application, GenericResources, myclasses** → Set the package name as **pkg** → Paste the provided code.
3. **Set Up Database:**
   - Install **GlassFish Server** and add **Java DB** library.
   - Verify paths in **Properties**.
4. **Add JSP Files:**
   - Create index and preview files → Paste the code.
5. **Build and Run:**
   - Right-click the project → **Clean and Build → Run** → Check output in the s**Output** section.

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/p7/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.java1:

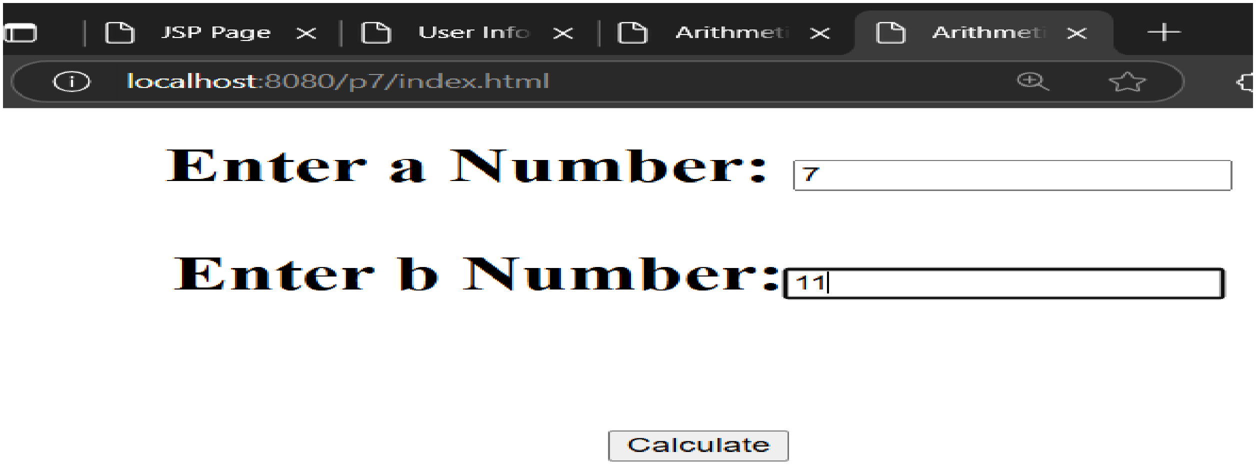
```
package pkg;
import javax.ws.rs.*;
import javax.ws.rs.core.*;
@Path("generic")
public class GenericResource {
    @Context
    private UriInfo context;
    public GenericResource() {
    }
    @GET
    @Produces(MediaType.TEXT_HTML)
    public String getHtml() {
        try {
            // Parse 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;
            double factorialA = factorial(a);
            double 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></tr
>");
        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 double factorial(int number) {
        if (number < 0) return -1; // Error for negative numbers
        double 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:



Enter a Number: 7

Enter b Number: 11

Results:

Basic Operations:

Operation	Result
Addition	18
Subtraction	-4
Multiplication	77
Division	0.6363636363636364
Modulo	7
Sum of Even Numbers	18
Sum of Odd Numbers	27

Perfect Number:

Number	Is Perfect?
7	No
11	No

Factorial:

Number	Factorial
7	5040
11	39916800

Prime Check:

Number	Is Prime?
7	Yes
11	Yes

Multiplication Tables

Table for 7:

7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70

Table for 11:

11 x 1 = 11
11 x 2 = 22
11 x 3 = 33
11 x 4 = 44
11 x 5 = 55
11 x 6 = 66
11 x 7 = 77
11 x 8 = 88
11 x 9 = 99
11 x 10 = 110

EXPERIMENT 08:

Aim: Write an application using web Sockets

Execution Steps

Create Project:

- **File → New Project → Java Web → Web Application →**
  Name: websocketHome → Select **GlassFish Server** → **Finish.**

1. **Add Files:**
   - **HTML:** Create **index.html** → Paste the provided code.
   - **CSS:** Create a **Cascading Style Sheet** → Paste the provided code.
   - **Java Socket:** Add a **Java Class** → Paste the provided code.
2. **Create Packages and Classes:**
   - **Model Package: org.example.model** → Add **Device.java.**
   - **WebSocket Package: org.example.websocket** → Add **DeviceSessionHandler.java** and **DeviceWebsocketServer.java.**
3. **Build and Run:**
   - Right-click **project → Clean and Build → Run.**

Source Code:

Index.html:

```
<!DOCTYPE html>
<html>
   <head>
      <title></title>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <script src="websocket.js"></script>
      <link rel="stylesheet" type="text/css" href="style.css">
      <link rel="stylesheet" href="style.css" type="text/css"/>
      <link rel="stylesheet" href="style.css" type="text/css">
      <link rel="stylesheet" href="style.css" type="text/css"/>
   </head>
   <body>
      <div id="wrapper">
         <h1>Java Websocket Home</h1>
         <p>Welcome to the Java WebSocket Home. Click the Add a device button to start
adding devices.</p>
         <br />
         <div id="addDevice">
```

```
        <div class="button"> <a href="#" OnClick="showForm()">Add a device</a>
</div>
        <form id="addDeviceForm">
          <h3>Add a new device</h3>
          <span>Name: <input type="text" name="device_name"
id="device_name"></span>
          <span>Type:
            <select id="device_type">
              <option name="type" value="Appliance">Appliance</option>
              <option name="type" value="Electronics">Electronics</option>
              <option name="type" value="Lights">Lights</option>
              <option name="type" value="Other">Other</option>
            </select></span>
          <span>Description:<br />
            <textarea name="description" id="device_description" rows="2"
cols="50"></textarea>
          </span>
          <input type="button" class="button" value="Add" onclick=formSubmit()>
          <input type="reset" class="button" value="Cancel" onclick=hideForm()>
        </form>
      </div>
      <br />
      <h3>Currently connected devices:</h3>
      <div id="content">
      </div>
    </div>
  </body>
</html>
```

Style.css:

```
body {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 80%;
  background-color: #1f1f1f;
}
#wrapper {
  width: 960px;
  margin: auto;
  text-align: left;
  color: #d9d9d9;
}
p {
  text-align: left;
}
.button {
  display: inline;
```

```
    color: #fff;
    background-color: #f2791d;
    padding: 8px;
    margin: auto;
    border-radius: 8px;
    -moz-border-radius: 8px;
    -webkit-border-radius: 8px;
    box-shadow: none;
    border: none;
}
.button:hover {
    background-color: #ffb15e;
}
.button a, a:visited, a:hover, a:active {
    color: #fff;
    text-decoration: none;
}
#addDevice {
    text-align: center;
    width: 960px;
    margin: auto;
    margin-bottom: 10px;
}
#addDeviceForm {
    text-align: left;
    width: 400px;
    margin: auto;
    padding: 10px;
}
#addDeviceForm span {
    display: block;
}
#content {
    margin: auto;
    width: 960px;
}
.device {
    width: 180px;
    height: 110px;
    margin: 10px;
    padding: 16px;
    color: #fff;
    vertical-align: top;
    border-radius: 8px;
    -moz-border-radius: 8px;
    -webkit-border-radius: 8px;
    display: inline-block;
}
.device.off {
    background-color: #c8cccf;
}
.device span {
    display: block;
}
.deviceName {
    text-align: center;
    font-weight: bold;
    margin-bottom: 12px;
}
.removeDevice {
    margin-top: 12px;
    text-align: center;
}
.device.Appliance {
    background-color: #5eb85e;
}
.device.Appliance a:hover {
    color: #a1ed82;
}
.device.Electronics {
    background-color: #0f90d1;
}
.device.Electronics a:hover {
    color: #4badd1;
}
.device.Lights {
    background-color: #c2a00c;
}
.device.Lights a:hover {
    color: #fad232;
}
.device.Other {
    background-color: #db524d;
}
.device.Other a:hover {
    color: #ff907d;
}
.device a {
    text-decoration: none;
}
```

```
.device a:visited, a:active, a:hover {
   color: #fff;
}
.device a:hover {
   text-decoration: underline;
}
```

websocket.js:

```
window.onload = init;
var socket = new WebSocket("ws://localhost:8080/WebsocketHome/actions");
socket.onmessage = onMessage;
function onMessage(event) {
   var device = JSON.parse(event.data);
   if (device.action === "add") {
      printDeviceElement(device);
   }
   if (device.action === "remove") {
      document.getElementById(device.id).remove();
      //device.parentNode.removeChild(device);
   }
   if (device.action === "toggle") {
      var node = document.getElementById(device.id);
      var statusText = node.children[2];
      if (device.status === "On") {
         statusText.innerHTML = "Status: " + device.status + " (<a href=\"#\"
OnClick=toggleDevice(" + device.id + ")>Turn off</a>)";
      } else if (device.status === "Off") {
         statusText.innerHTML = "Status: " + device.status + " (<a href=\"#\"
OnClick=toggleDevice(" + device.id + ")>Turn on</a>)";
      }
   }
}
function addDevice(name, type, description) {
   var DeviceAction = {
      action: "add",
      name: name,
      type: type,
      description: description
   };
   socket.send(JSON.stringify(DeviceAction));
}
function removeDevice(element) {
   var id = element;
   var DeviceAction = {
      action: "remove",
        id: id
    };
    socket.send(JSON.stringify(DeviceAction));
}
function toggleDevice(element) {
    var id = element;
    var DeviceAction = {
        action: "toggle",
        id: id
    };
    socket.send(JSON.stringify(DeviceAction));
}
function printDeviceElement(device) {
    var content = document.getElementById("content");
    var deviceDiv = document.createElement("div");
    deviceDiv.setAttribute("id", device.id);
    deviceDiv.setAttribute("class", "device " + device.type);
    content.appendChild(deviceDiv);
    var deviceName = document.createElement("span");
    deviceName.setAttribute("class", "deviceName");
    deviceName.innerHTML = device.name;
    deviceDiv.appendChild(deviceName);
    var deviceType = document.createElement("span");
    deviceType.innerHTML = "<b>Type:</b> " + device.type;
    deviceDiv.appendChild(deviceType);
    var deviceStatus = document.createElement("span");
    if (device.status === "On") {
        deviceStatus.innerHTML = "<b>Status:</b> " + device.status + " (<a href=\"#\"
OnClick=toggleDevice(" + device.id + ")>Turn off</a>)";
    } else if (device.status === "Off") {
        deviceStatus.innerHTML = "<b>Status:</b> " + device.status + " (<a href=\"#\"
OnClick=toggleDevice(" + device.id + ")>Turn on</a>)";
        //deviceDiv.setAttribute("class", "device off");
    }
    deviceDiv.appendChild(deviceStatus);
    var deviceDescription = document.createElement("span");
    deviceDescription.innerHTML = "<b>Comments:</b> " + device.description;
    deviceDiv.appendChild(deviceDescription);
    var removeDevice = document.createElement("span");
    removeDevice.setAttribute("class", "removeDevice");
    removeDevice.innerHTML = "<a href=\"#\" OnClick=removeDevice(" + device.id +
")>Remove device</a>";
    deviceDiv.appendChild(removeDevice);
}
function showForm() {
    document.getElementById("addDeviceForm").style.display = '';
}
function hideForm() {
    document.getElementById("addDeviceForm").style.display = "none";
}
function formSubmit() {
    var form = document.getElementById("addDeviceForm");
    var name = form.elements["device_name"].value;
    var type = form.elements["device_type"].value;
    var description = form.elements["device_description"].value;
    hideForm();
    document.getElementById("addDeviceForm").reset();
    addDevice(name, type, description);
}
function init() {
    hideForm();
}
```

<u>Device.java:</u>

```
package org.example.model;
public class Device {
    private int id;
    private String name;
    private String status;
    private String type;
    private String description;
    public Device() {
    }
    public int getId() {
        return id;
    }
    public String getName() {
        return name;
    }
    public String getStatus() {
        return status;
    }
    public String getType() {
        return type;
    }
    public String getDescription() {
        return description;
    }
    public void setId(int id) {
        this.id = id;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    public void setType(String type) {
        this.type = type;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

DeviceSessionHandler.java:

```
package org.example.websocket;
import java.io.IOException;
import java.util.ArrayList;
import javax.enterprise.context.ApplicationScoped;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.json.JsonObject;
import javax.json.spi.JsonProvider;
import javax.websocket.Session;
import org.example.model.Device;
@ApplicationScoped
public class DeviceSessionHandler {
        private int deviceId = 0;
    private final Set<Session> sessions = new HashSet<>();
    private final Set<Device> devices = new HashSet<>();
    public void addSession(Session session) {
        sessions.add(session);
        for (Device device : devices) {
            JsonObject addMessage = createAddMessage(device);
            sendToSession(session, addMessage);
        }
    }
    public void removeSession(Session session) {
        sessions.remove(session);
    }
    public List<Device> getDevices() {
        return new ArrayList<>(devices);
    }
```

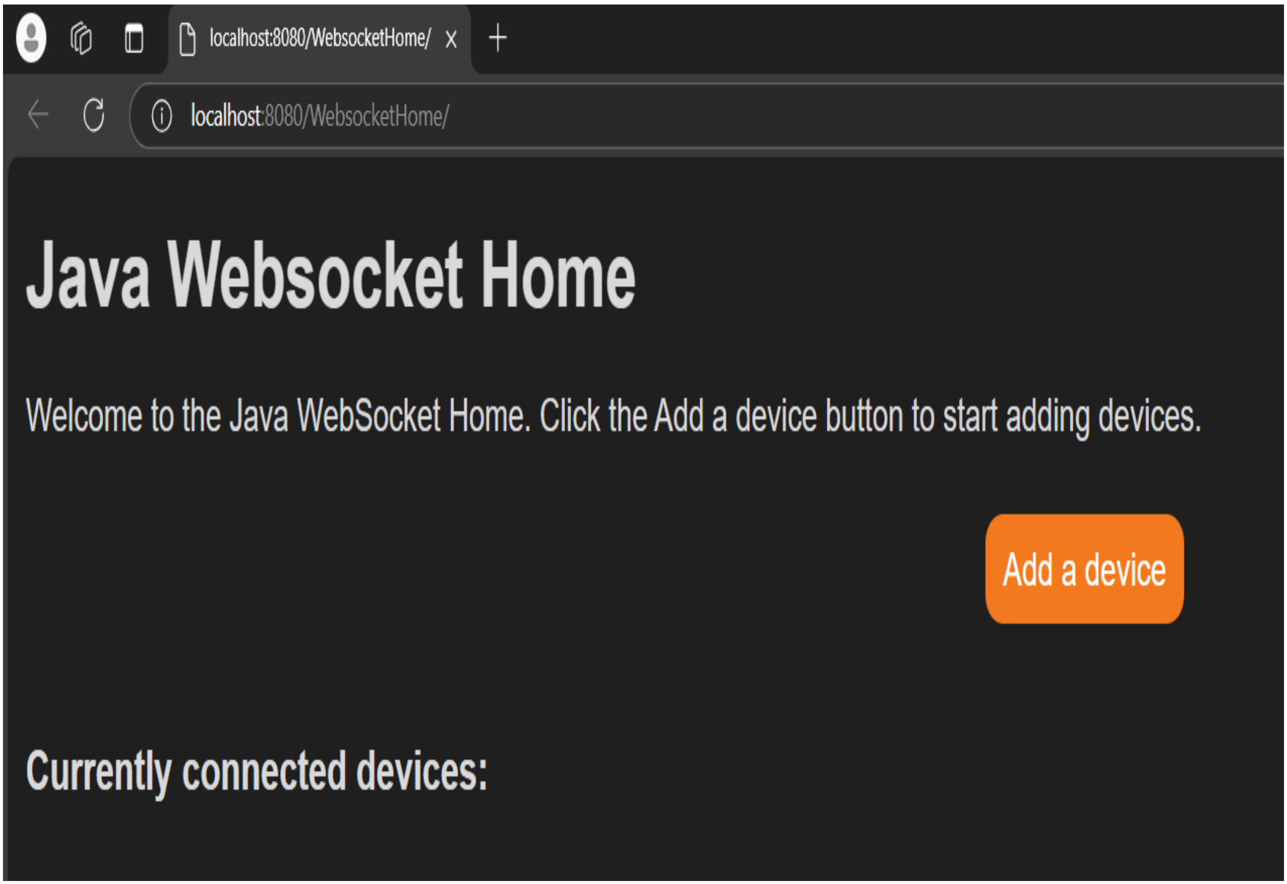
```
public void addDevice(Device device) {
     device.setId(deviceId);
    devices.add(device);
    deviceId++;
    JsonObject addMessage = createAddMessage(device);
    sendToAllConnectedSessions(addMessage);
}
public void removeDevice(int id) {
    Device device = getDeviceById(id);
    if (device != null) {
        devices.remove(device);
        JsonProvider provider = JsonProvider.provider();
        JsonObject removeMessage = provider.createObjectBuilder()
                .add("action", "remove")
                .add("id", id)
                .build();
        sendToAllConnectedSessions(removeMessage);
}
}
public void toggleDevice(int id) {
    JsonProvider provider = JsonProvider.provider();
    Device device = getDeviceById(id);
    if (device != null) {
        if ("On".equals(device.getStatus())) {
            device.setStatus("Off");
        } else {
            device.setStatus("On");
        }
        JsonObject updateDevMessage = provider.createObjectBuilder()
                .add("action", "toggle")
                .add("id", device.getId())
                .add("status", device.getStatus())
                .build();
        sendToAllConnectedSessions(updateDevMessage);
    }
}
private Device getDeviceById(int id) {
     for (Device device : devices) {
        if (device.getId() == id) {
            return device;
        }
    }
    return null;
}
private JsonObject createAddMessage(Device device) {
        JsonProvider provider = JsonProvider.provider();
        JsonObject addMessage = provider.createObjectBuilder()
                .add("action", "add")
                .add("id", device.getId())
                .add("name", device.getName())
                .add("type", device.getType())
                .add("status", device.getStatus())
                .add("description", device.getDescription())
                .build();
        return addMessage;
    }
    private void sendToAllConnectedSessions(JsonObject message) {
         for (Session session : sessions) {
            sendToSession(session, message);
        }
    }
    private void sendToSession(Session session, JsonObject message) {
         try {
            session.getBasicRemote().sendText(message.toString());
        } catch (IOException ex) {
            sessions.remove(session);
            Logger.getLogger(DeviceSessionHandler.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }
}
```

DeviceWebSocketServer.java:

```
package org.example.websocket;
import javax.websocket.OnClose;
import javax.websocket.OnError;
import javax.websocket.OnMessage;
import javax.websocket.OnOpen;
import javax.websocket.Session;
import javax.websocket.server.ServerEndpoint;
import javax.enterprise.context.ApplicationScoped;
import javax.inject.Inject;
import java.io.StringReader;
import javax.json.Json;
import javax.json.JsonObject;
import javax.json.JsonReader;
import org.example.model.Device;
import java.util.logging.Level;
import java.util.logging.Logger;
@ApplicationScoped
@ServerEndpoint("/actions")
```

```
public class DeviceWebSocketServer {
     @Inject
    private DeviceSessionHandler sessionHandler;
    @OnOpen
        public void open(Session session) {
             sessionHandler.addSession(session);
    }
    @OnClose
        public void close(Session session) {
            sessionHandler.removeSession(session);
    }
    @OnError
        public void onError(Throwable error) {
Logger.getLogger(DeviceWebSocketServer.class.getName()).log(Level.SEVERE, null,
error);
    }

    @OnMessage
        public void handleMessage(String message, Session session) {
          try (JsonReader reader = Json.createReader(new StringReader(message))) {
            JsonObject jsonMessage = reader.readObject();
            if ("add".equals(jsonMessage.getString("action"))) {
                Device device = new Device();
                device.setName(jsonMessage.getString("name"));
                device.setDescription(jsonMessage.getString("description"));
                device.setType(jsonMessage.getString("type"));
                device.setStatus("Off");
                sessionHandler.addDevice(device);
            }
            if ("remove".equals(jsonMessage.getString("action"))) {
                int id = (int) jsonMessage.getInt("id");
                sessionHandler.removeDevice(id);
            }
            if ("toggle".equals(jsonMessage.getString("action"))) {
                int id = (int) jsonMessage.getInt("id");
                sessionHandler.toggleDevice(id);
            }
        }
```

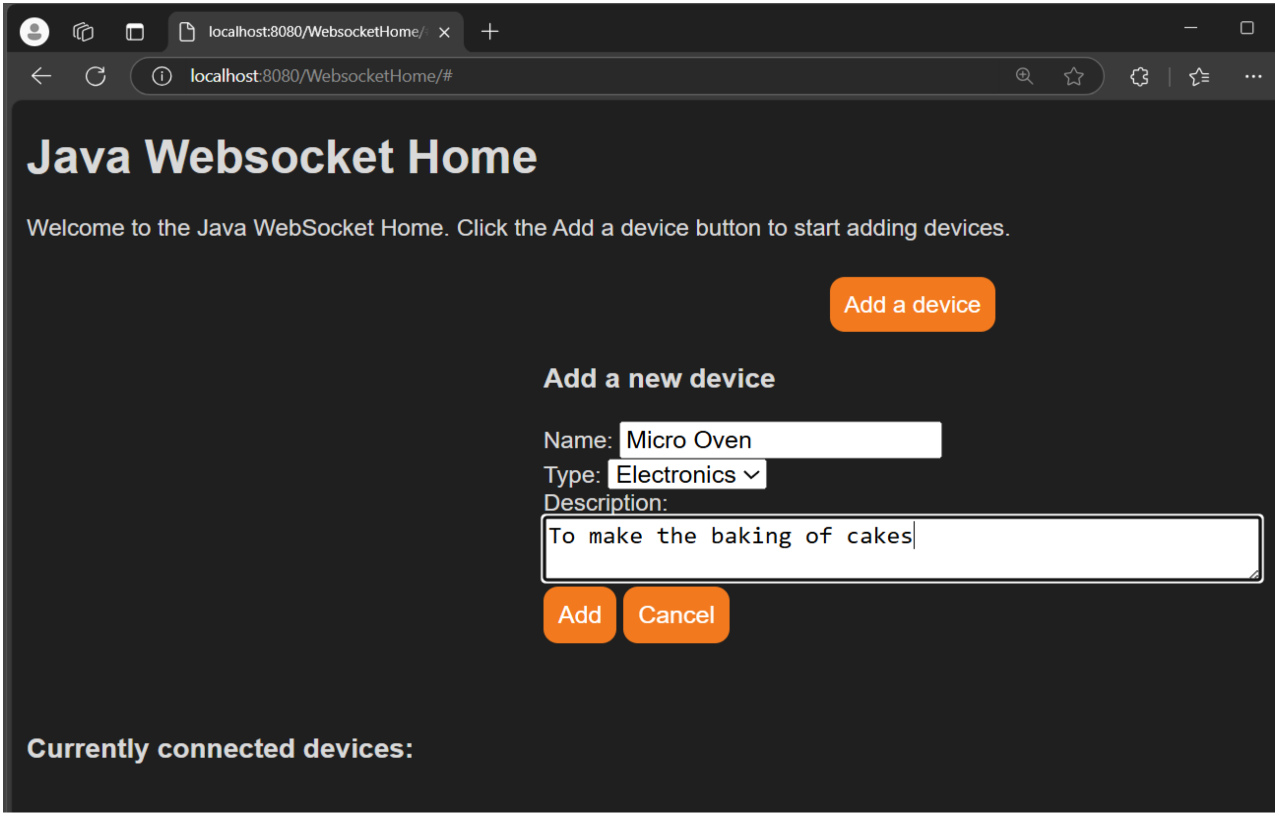
Output:



Java Websocket Home

Welcome to the Java WebSocket Home. Click the Add a device button to start adding devices.

Currently connected devices:



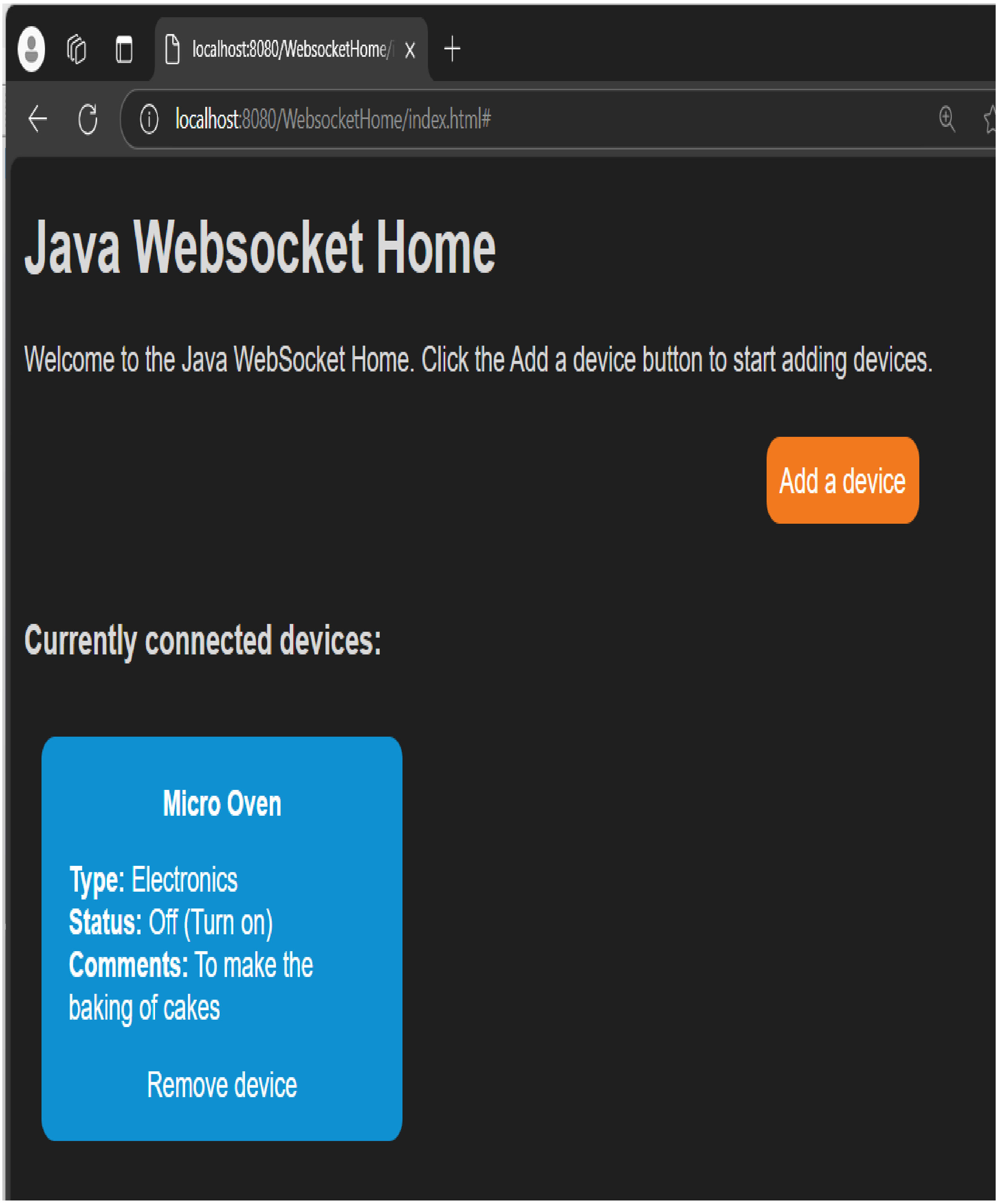
Java Websocket Home

Welcome to the Java WebSocket Home. Click the Add a device button to start adding devices.

Add a new device

Name: Micro Oven
Type: Electronics
Description:
To make the baking of cakes

Currently connected devices:


localhost:8080/WebsocketHome/i
localhost:8080/WebsocketHome/index.html#

Java Websocket Home

Welcome to the Java WebSocket Home. Click the Add a device button to start adding devices.

Currently connected devices:

Micro Oven

Type: Electronics
Status: Off (Turn on)
Comments: To make the baking of cakes

EXPERIMENT 09:

Aim: write an application to demonstrate Session Bean and Entity Bean(persistence)

Execution Steps:

1. **Create Project:**
   - **File → New Project → Java Web → Web Application → project Name →** Select **GlassFish Server → Finish.**
2. **Add Files:**
   - **HTML**: Create **index.html** → Paste the provided code.
3. **Create Servlets:**
   - Right-click the project → **New → Servlet → Class Name: Employe**e → Click the checkbox to enable required options → Finish.
   - Create the servlets: **EmployeeServlet** and **EmployeeSessionBean.**
4. **Build and Run:**
   - Right-click project → **Clean** and **Build → Run.**

Source code:

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Add Employee</title>
</head>
<body>
   <h2>Add Employee</h2>
   <form action="EmployeeServlet" method="POST">
      NAME: <input type="text" name="name" required /><br><br/>
      JOB ROLE: <input type="text" name="position" required /><br><br/>
      SALARY: <input type="text" name="salary" required /><br><br/>
      <input type="submit" value="Add Employee" />
   </form>
</body>
</html>
```

Employee.java:

```
import java.io.Serializable;
public class Employee implements Serializable {
   private Long id;
   private String name;
   private String position;
   private double salary;
```

```
    // Constructor
    public Employee(Long id, String name, String position, double salary) {
        this.id = id;
        this.name = name;
        this.position = position;
        this.salary = salary;
    }
    // Getters and Setters
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getPosition() {
        return position;
    }
    public void setPosition(String position) {
        this.position = position;
    }
    public double getSalary() {
        return salary;
    }
    public void setSalary(double salary) {
        this.salary = salary;
    }
}
```

EmployeeServlet.java:

```
import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;

@WebServlet("/EmployeeServlet")
public class EmployeeServlet extends HttpServlet {

    @EJB
    private EmployeeSessionBean employeeSessionBean;
```

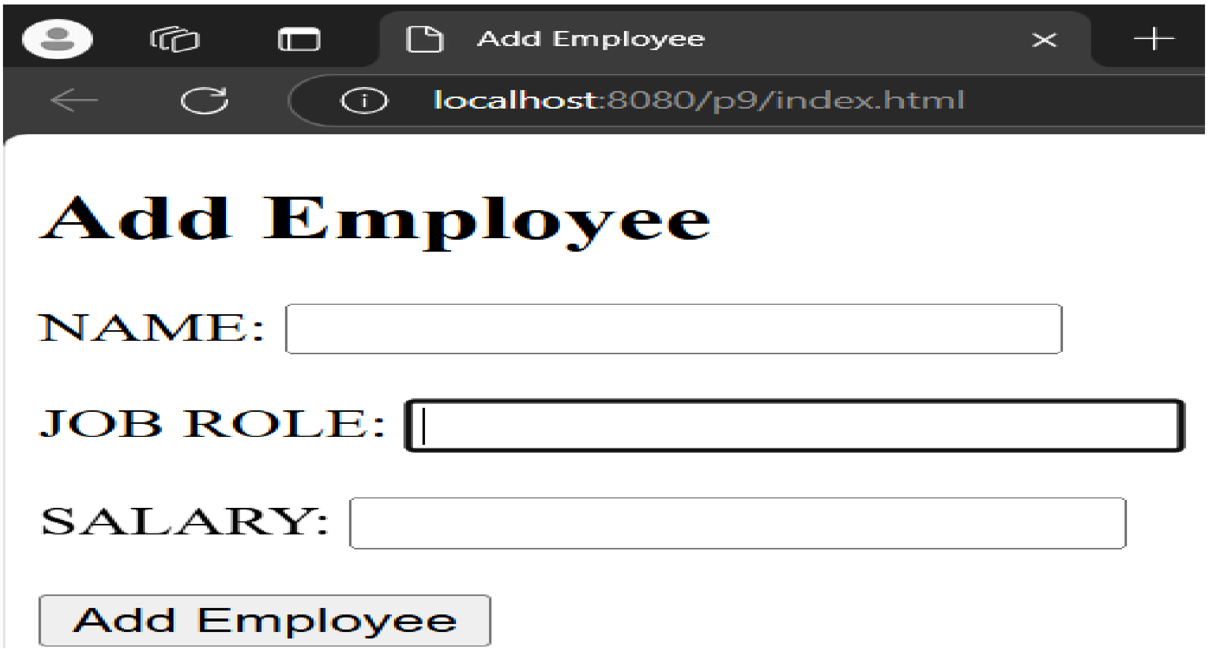
```
  @Override
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
   List<Employee> employees = employeeSessionBean.getAllEmployees();

   response.setContentType("text/html");
   PrintWriter out = response.getWriter();
   out.println("<html><body>");
   out.println("<h2>Employee List</h2>");

   // Start table
   out.println("<table border='1'>");
   out.println("<tr><th>Name</th><th>Position</th><th>Salary</th></tr>");

   // Loop through employees and create table rows
   for (Employee employee : employees) {
      out.println("<tr>");
      out.println("<td>" + employee.getName() + "</td>");
      out.println("<td>" + employee.getPosition() + "</td>");
      out.println("<td>" + employee.getSalary() + "</td>");
      out.println("</tr>");
   }
   // End table
   out.println("</table>");

   out.println("</body></html>");
}
   @Override
   protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
      String name = request.getParameter("name");
      String position = request.getParameter("position");
      double salary = Double.parseDouble(request.getParameter("salary"));

      employeeSessionBean.addEmployee(name, position, salary);

      response.sendRedirect("EmployeeServlet");
   }
}
```

EmployeeSessionBean.java:

```
import javax.ejb.Stateless;
import java.util.ArrayList;
import java.util.List;
@Stateless
public class EmployeeSessionBean {
   // In-memory list to mimic a database
   private List<Employee> employeeList = new ArrayList<>();
   private static Long idCounter = 1L;
```

```
    // Method to add employee
    public void addEmployee(String name, String position, double salary) {
        Employee employee = new Employee(idCounter++, name, position, salary);
        employeeList.add(employee);
    }
    // Method to get all employees
    public List<Employee> getAllEmployees() {
        return employeeList;
    }
}
```

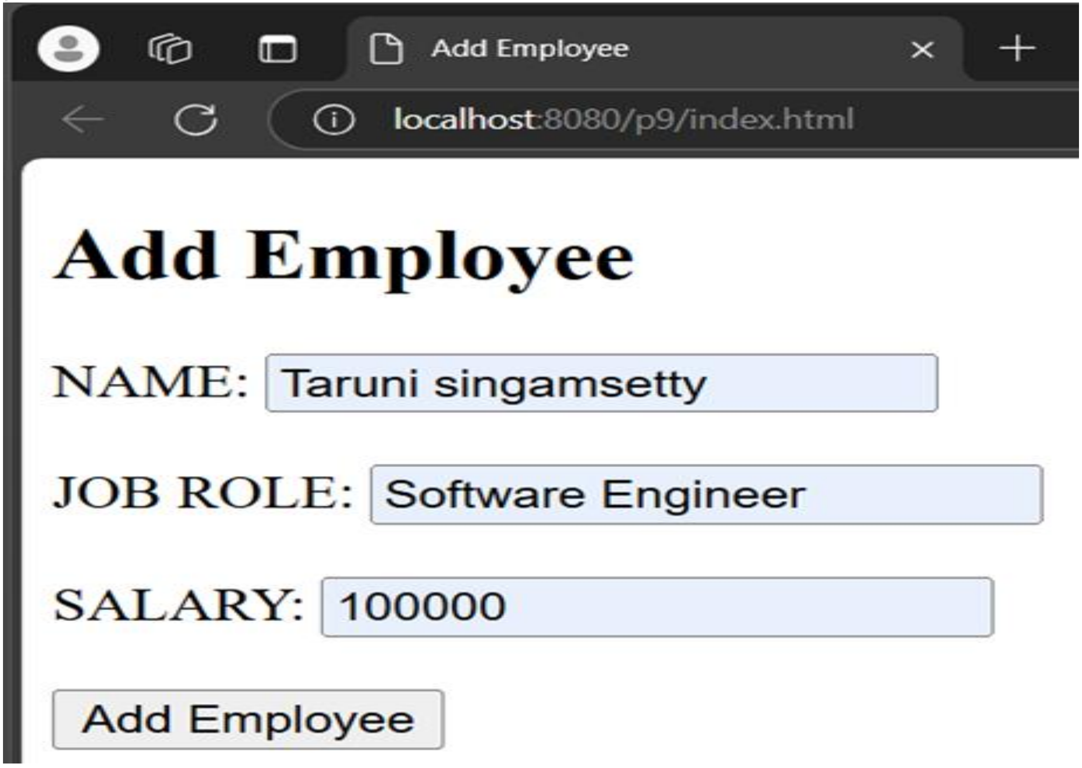
Output:



NAME:

JOB ROLE:

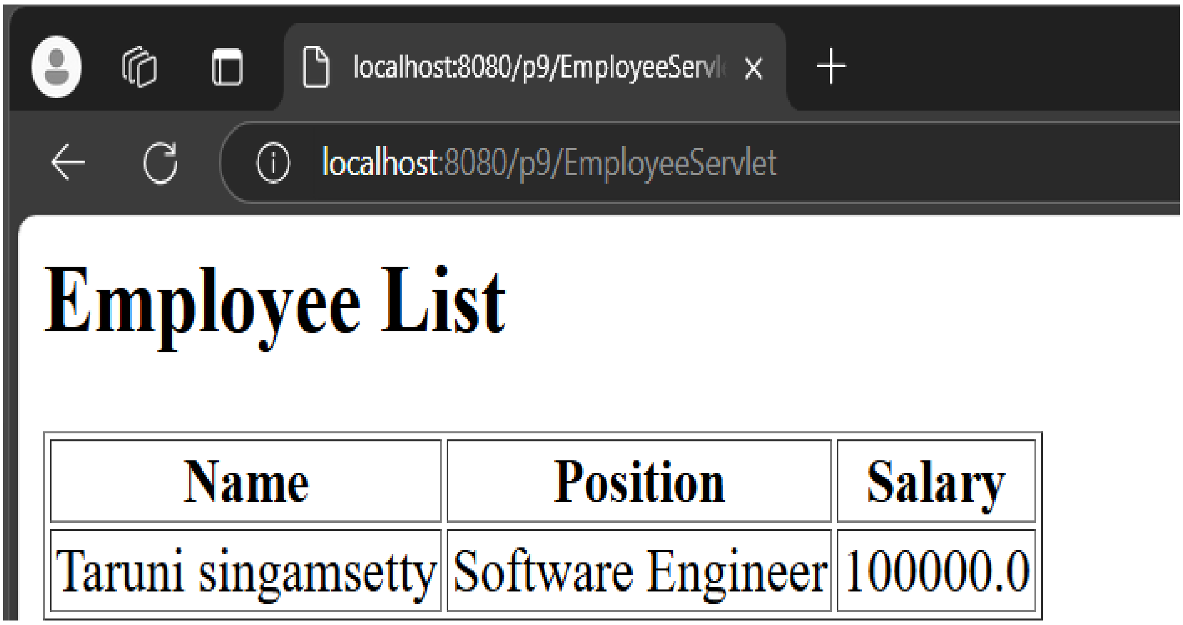
SALARY:



NAME: Taruni singamsetty

JOB ROLE: Software Engineer

SALARY: 100000



localhost:8080/p9/EmployeeServl
localhost:8080/p9/EmployeeServlet

Employee List

Name	Position	Salary
Taruni singamsetty	Software Engineer	100000.0