Practical: 1

AIM: Write TCP or UDP program for CHAT Application.

PROGRAM:

Chatserver.java:

```
Import java.io. *;

Import java.net. *;

Public class Practical_2_TCP_Server

{

   Public static void main(String[] args) throws Exception

  {

      Serversocket sersock = new serversocket(3100);

      System.out.println("Server is connected");

         Socket sock = sersock.accept();

      Bufferedreader keyread = new bufferedreader(new inputstreamreader(System.in));

      Outputstream ostream = sock.getoutputstream();

      Printwriter pwrite = new printwriter(ostream, true);

      Inputstream istream = sock.getinputstream();

      Bufferedreader receiveread = new bufferedreader(new inputstreamreader(istream));

      String receivemessage, sendmessage;

      While (true) {

         If ((receivemessage = receiveread.readline()) != null) {

            System.out.println("Client: " + receivemessage);

         }

         Sendmessage = keyread.readline();

         Pwrite.println(sendmessage);

         Pwrite.flush();
```

```
    }
  }
}
```

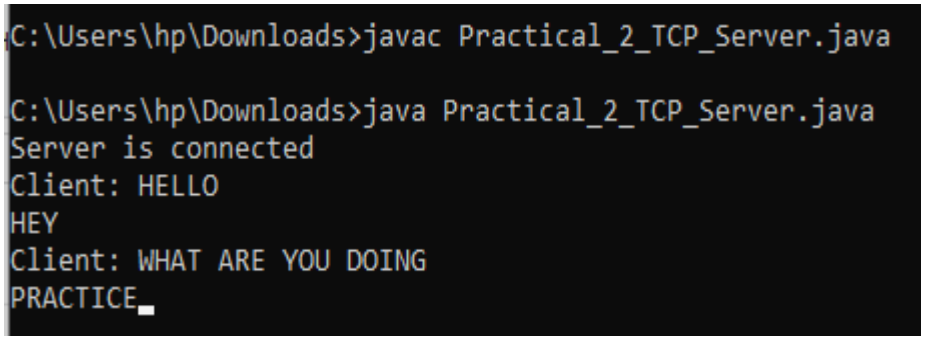
Chatclient.java:

```
Import java.io. *;
Import java.net. *;

Public class Practical_2_TCP_Client {
        Public static void main(String[] args) throws Exception {
                Socket sock = new Socket("127.0.0.1", 3100);
                Bufferedreader keyread = new bufferedreader(new
inputstreamreader(System.in));
                Outputstream ostream = sock.getoutputstream();
                Printwriter pwrite = new printwriter(ostream, true);
                Inputstream istream = sock.getinputstream();
                Bufferedreader receiveread = new bufferedreader(new
inputstreamreader(istream));
                System.out.println("Start the Chat, type and press Enter ");
                String receivemessage, sendmessage;
                While (true) {
                        Sendmessage = keyread.readline();
                        Pwrite.println(sendmessage);
                        Pwrite.flush();
                        {
                                If ((receivemessage = receiveread.readline()) != null) {
                                        System.out.println("Server: " + receivemessage);
```

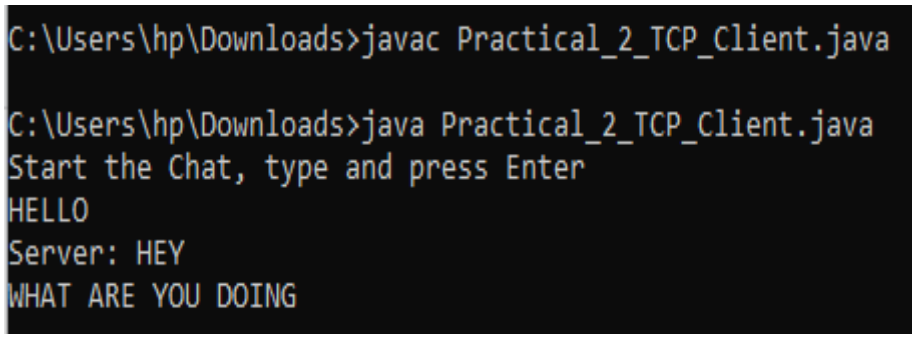
```
                    }
                }
            }
        }
                    }
```

OUTPUT:

Server:



Client:



Practical: 2

AIM: Implement TCP Server for transferring files using Socket and Server-Socket.

PROGRAM:

Server side Practical_2_Server.java:

```
Import java.io. *;
Import java.net. *;
Class Practical_2_Server
{
Public static void main (String args[]) throws Exception
{
Serversocket ss = new serversocket(7777);
Socket s = ss.accept();
System.out.println("connected..... ");
Fileinputstream fin = new fileinputstream("C://Users//hp//Downloads//ajp//send.text");
Dataoutputstream dout = new dataoutputstream(s.getoutputstream());
Int r;
While ((r = fin.read()) != -1) {
Dout.write(r);
}
System.out.println("\nfiletranfer Completed");
S.close();
Ss.close();
}
}
```

Client side Practical_2_Client.java:

```
Import java.io.*;
Import java.net.*;
Public class Practical_2_Client
```

```
{
Public static void main(String[] args) throws Exception {
Socket s = new Socket("localhost", 7777);
If (s.isconnected())
{
        System.out.println("Connected to server");
}
Fileoutputstream fout = new fileoutputstream("C://Users//hp//Downloads//ajp//received.txt");
Datainputstream din = new datainputstream(s.getinputstream());
Int r;
While ((r = din.read()) != -1)
{
        Fout.write((char) r);
}
}
}
```

OUTPUT :

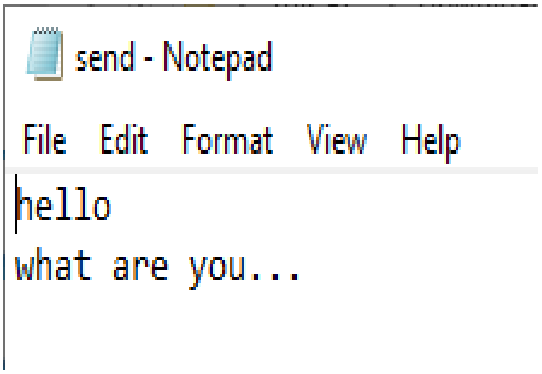
Server side :

```
C:\Users\hp\Downloads\ajp>javac Practical_2_Server.java

C:\Users\hp\Downloads\ajp>java Practical_2_Server.java
connected.........

Filetranfer Completed
```

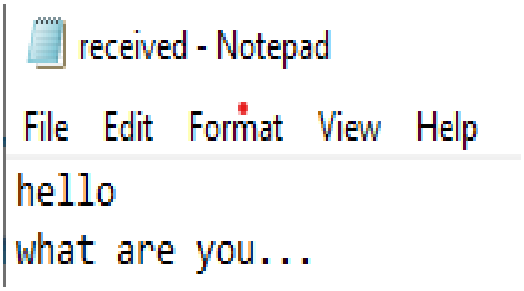
Client side:

```
C:\Users\hp\Downloads\ajp>javac Practical_2_Client.java

C:\Users\hp\Downloads\ajp>java Practical_2_Client.java
Connected to server
```

Send file :



Received file :



Practical: 3

1) Aim: Fetching results using JDBC.

Practical1: 1. Use of JDBC prepared statements with ResultSet.

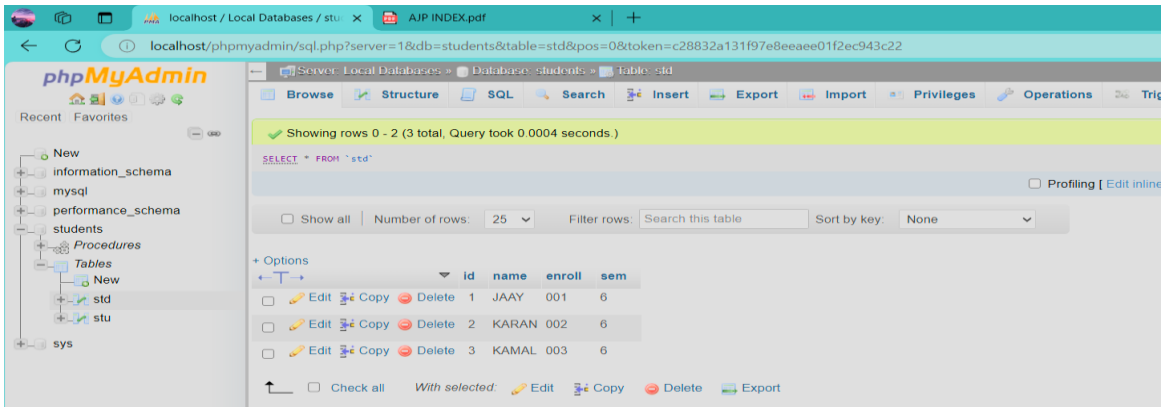
Code :

```
package project1;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Arrays;
public class PRACTICAL1 {
public static void main(String[] args) throws ClassNotFoundException, SQLException {
try{
Class.forName("com.mysql.jdbc.Driver");
System.out.println("connecting Database... ");
Connection con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/students","root","");
System.out.println("Database Connection Successful..!!");
Statement st=(Statement) con.createStatement();
String query = " update std set name=?,sem=? where id=? ";
PreparedStatement ps=con.prepareStatement(query);
ps.setString(1, "jagruti patel");
ps.setString(2,"4th");
ps.setInt(3,3);
ps.addBatch();
ps.setString(1, "rutalmam");
ps.setString(2,"4th");
```

```
ps.setInt(3,1);
ps.addBatch();
ps.setString(1,"riteshsir");
ps.setString(2,"6th");
ps.setInt(3,2);
ps.addBatch();
int[] i = ps.executeBatch();
ResultSet rs;
   rs = st.executeQuery("select * from std");
   while(rs.next())
   {
      System.out.print("Student Id:"+rs.getInt(1)+"\n");
      System.out.print("Student Name:"+rs.getString(2)+"\n");
      System.out.print("Student Enroll:"+rs.getString(3)+"\n");
      System.out.print("Student Semester:"+rs.getString(4)+"\n");
      System.out.print("\n");
   }
st.close();
con.close();
}catch(Exception e)
{
System.out.println(e.toString());
} } }
```

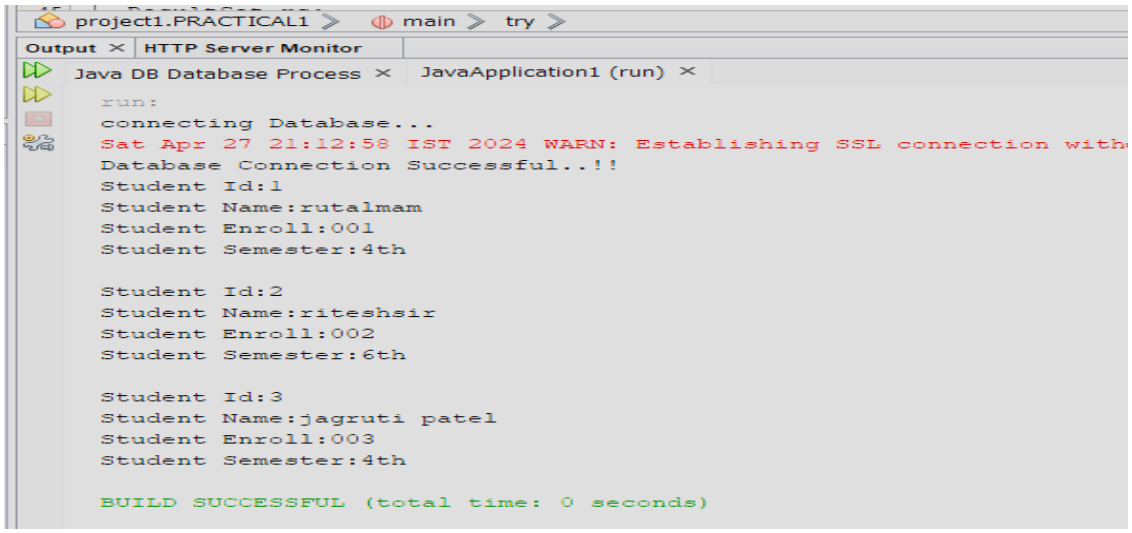
Output :

Before code run :

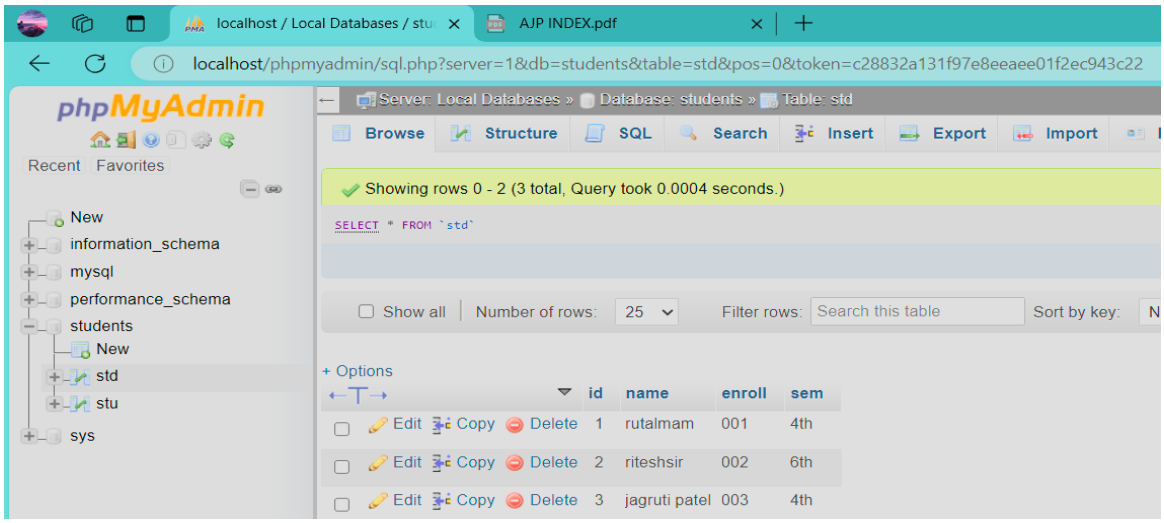


After code run:

Netbean:



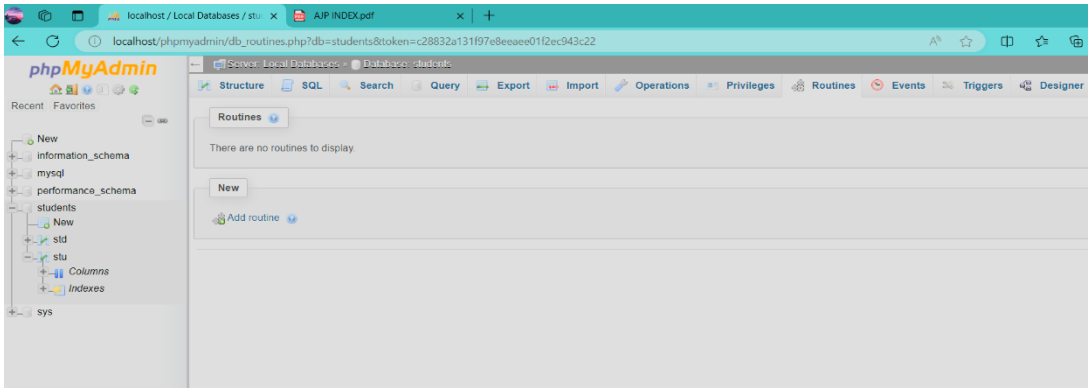
Wampserver:



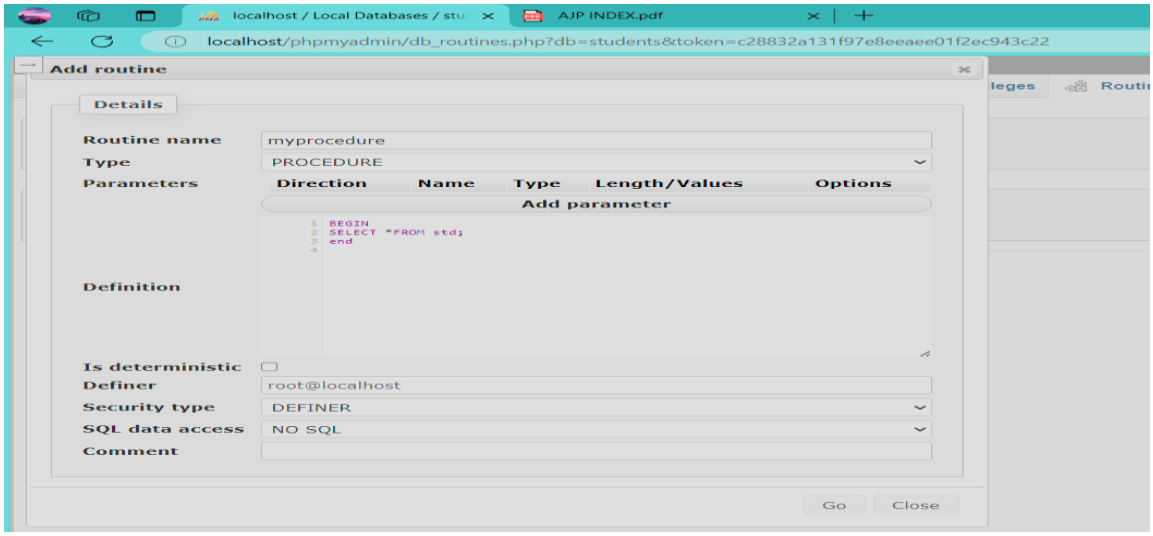
Practical 2: 2. To execute stored procedures using CallableStatement Statements.

To wemp server step:

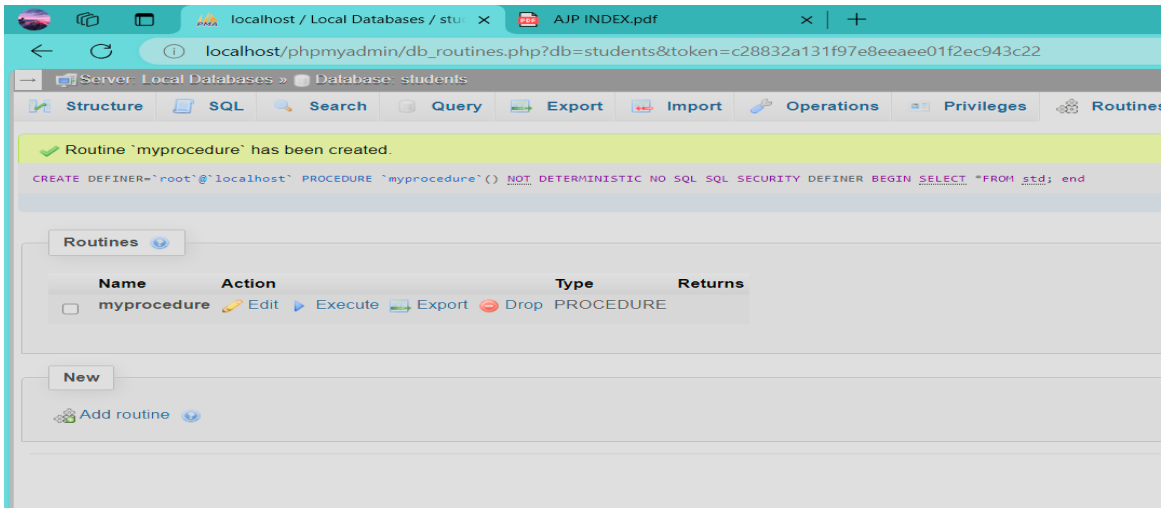
Step1: go to the routine.



Step 2: add routine.



Step3: added routine.

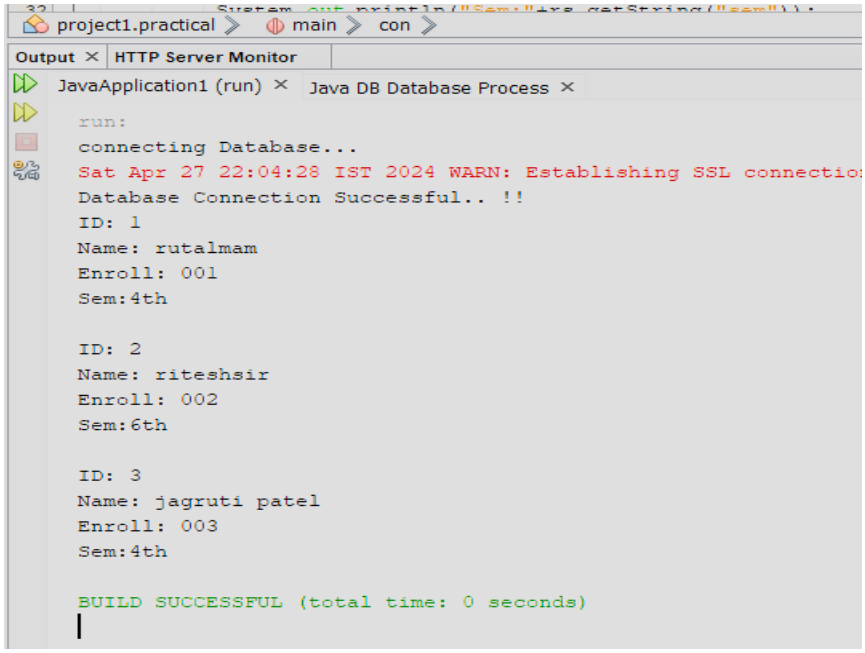


Code :

```
package project1;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
/**
 *
 * @author Admin
 */
public class practical {
public static void main(String[] args) throws ClassNotFoundException, SQLException {
Class.forName("com.mysql.jdbc.Driver");
System.out.println("connecting Database... ");
Connection con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/students","root","");
System.out.println("Database Connection Successful.. !!");
CallableStatement cs = con.prepareCall("{call myprocedure()}");
ResultSet rs = cs.executeQuery();
```

```
    while(rs.next()) {
      System.out.println("ID: "+rs.getInt("id"));
      System.out.println("Name: "+rs.getString("name"));
      System.out.println("Enroll: "+rs.getString("enroll"));
      System.out.println("Sem:"+rs.getString("sem"));
      System.out.println();
    }
//System.out.println("Name is Changed using Callable Statement");
}
}
```

Output :



Practical 3: 3 Batch update using Statement.

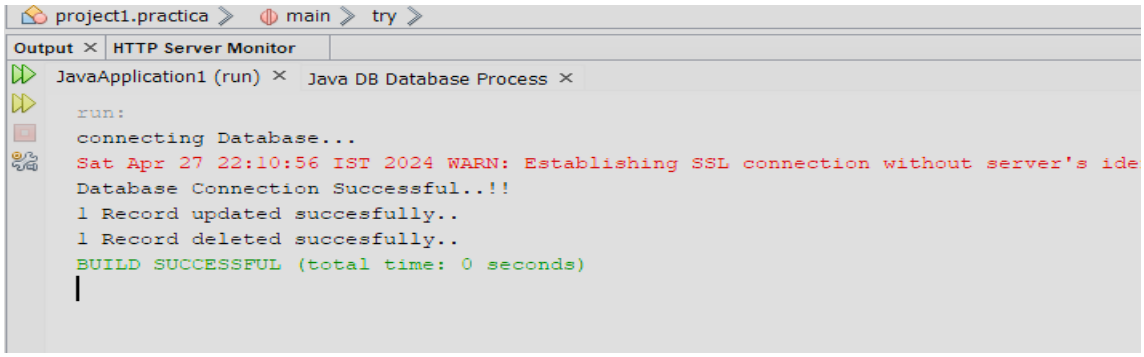
Code :

```
package project1;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
/**
 *
 * @author Admin
 */
public class practica {
public static void main(String[] args) throws ClassNotFoundException, SQLException { // TODO code application logic here
try
{
String query,query1;
Class.forName("com.mysql.jdbc.Driver");
System.out.println("connecting Database... ");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/students","root","");
System.out.println("Database Connection Successful..!!");
Statement st=(Statement) con.createStatement();
query = " update std set sem='5' where name='riteshsir'";
query1="delete from std where name='jagruti patel'";
st.executeUpdate(query);
st.executeUpdate(query1);
System.out.println("1 Record updated succesfully.. ");
System.out.println("1 Record deleted succesfully.. ");
st.close();
```

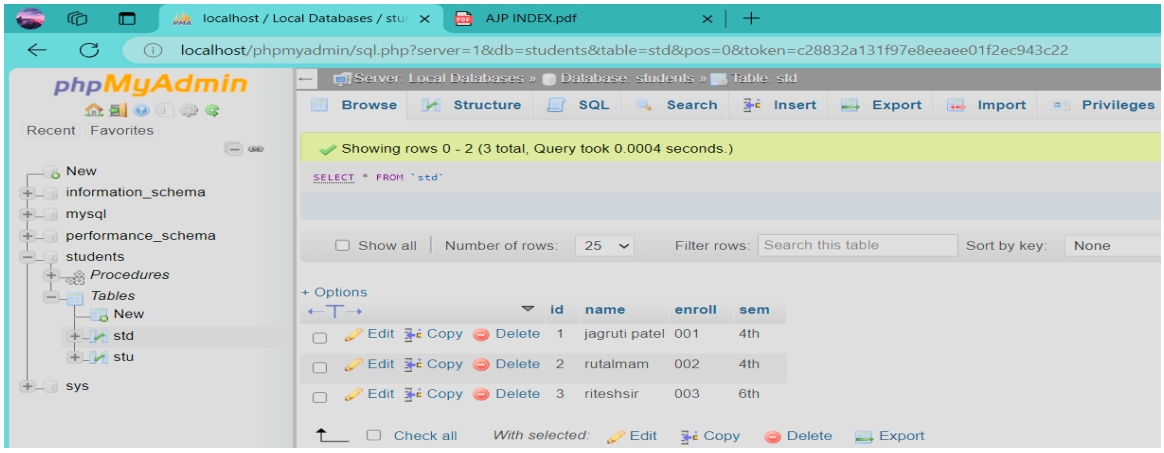
```
con.close();
}catch(Exception e)
{
System.out.println(e.toString());}
} }
```

Output:

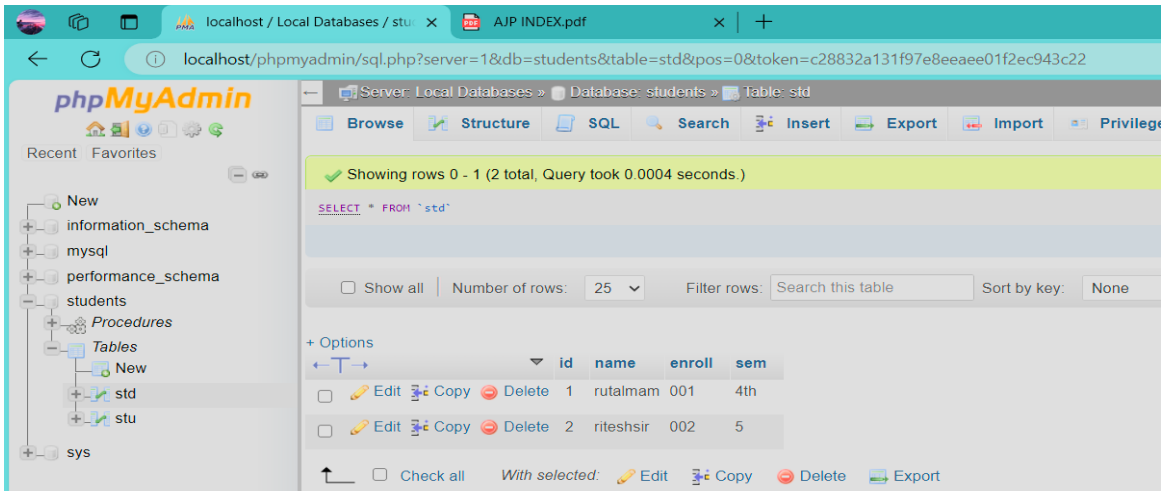
Netbeans :



Wemp server before code run:



Wemp server after code run:



Practical 4: Batch update using Prepared Statement.

Code :

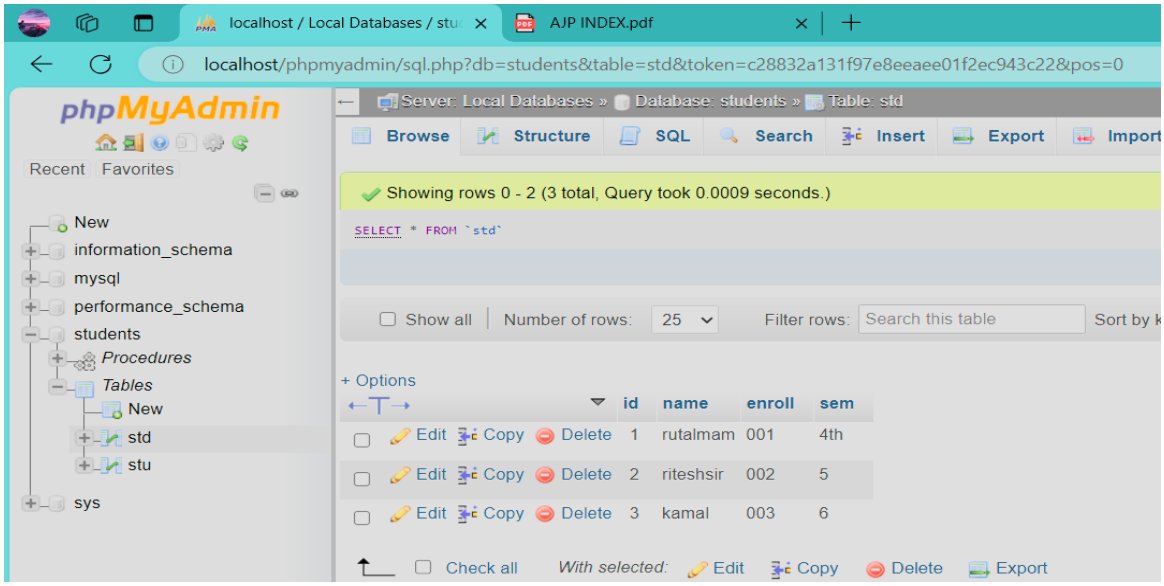
```
package project1;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Arrays;
/**
 *
 * @author Admin
 */
public class prac {
public static void main(String[] args) throws ClassNotFoundException, SQLException {
try{
Class.forName("com.mysql.jdbc.Driver");
System.out.println("connecting Database... ");
Connection con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/students","root","");
System.out.println("Database Connection Successful..!!");
```

```
Statement st=(Statement) con.createStatement();
String query = " update std set name=?,sem=? where id=? ";
PreparedStatement ps=con.prepareStatement(query);
ps.setString(1, "jagruti patel");
ps.setString(2,"4th");
ps.setInt(3,1);
ps.addBatch();
ps.setString(1, "rutalmam");
ps.setString(2,"4th");
ps.setInt(3,2);
ps.addBatch();
ps.setString(1,"riteshsir");
ps.setString(2,"6th");
ps.setInt(3,3);
ps.addBatch();
int[] i = ps.executeBatch();
System.out.println("Affected Rows are:-" +Arrays.toString(i));
st.close();
con.close();
}catch(Exception e)
{
System.out.println(e.toString());
}
}
}
```

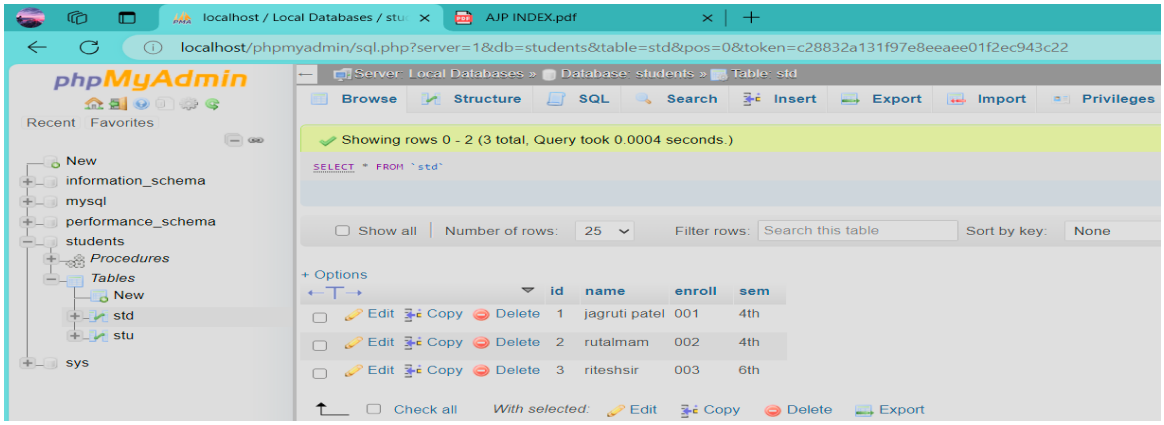
Output :

```
project1.prac > main > try > query >
Output | HTTP Server Monitor
JavaApplication1 (run) | Java DB Database Process
run:
connecting Database...
Sat Apr 27 22:16:50 IST 2024 WARN: Establishing SSL connection wi
Database Connection Successful..!!
Affected Rows are:-[1, 1, 1]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Wampserver before run code:



Wampserver after run code:



Practical: 4

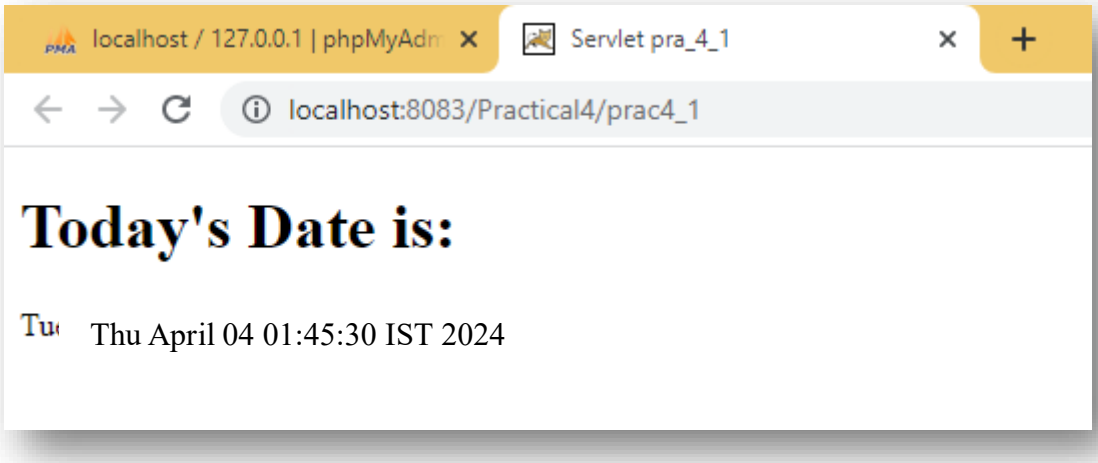
Aim: Write a Servlet program to print system date and time.

PROGRAM:

```
Import java.io.ioexception;
Import java.io.printwriter;
Import java.util.Date;
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 = {"/prac4_1"})
Public class prac4_1 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 pra_4_1</title>");
       Out.println("</head>");
       Out.println("<body>");
       Out.println("<h1>Today's Date is: " + "</h1>");
       Date Today=new Date();
       Out.println( Today );
       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";
    }// </editor-fold>

}
```

Output:



Practical 5

Aim: Implement student registration form with enrollment number, first name, last name, semester, contact number. Store the details in database. Also implement search, delete andmodify facility for student records.

Index.html

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

<h1><b>Perform Following Operation</b></h1>

<table border="">
<tr><td><a href="insert.html" with="100"

height="100">REGISTRATION</a></td></tr>

<tr><td><a href="search.html" with="100"

height="100">SEARCH</a></td></tr>

<tr><td><a href="update.html" with="100"

height="100">UPDATE</a></td></tr>

<tr><td><a href="delete.html" with="100"
```

```
height="100">DELETE</a></td></tr>
</table>
</center>
</body>
</html>
```

Insert.html

```
<html>
<head>
<title>Student Details</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<center>
<form action="Servlet1" method="post">
<h1><b>REGISTRATION FORM</b></h1>
<table border="">
<tr><td>First Name:</td><td><input type="text"
name="fname"value=""></td></tr>
<tr><td>Last Name:</td><td><input type="text"
name="lname"value=""></td></tr>
<tr><td>Semester:</td><td><input type="number"
name="sem"value=""></td></tr>
<tr><td>Contact No:</td><td><input type="text" name="contact"
value=""></td></tr>
<tr><td>Enrollment:</td><td><input type="text"
name="enroll"value=""></td></tr>
</table>
<table border="">
<br>
```

```
<tr><td><input type="submit" name="Save" value="Save"></td></tr>
</table>
</form>
</center>
</body>
</html>
```

Search.html

```
<html>
<head>
<title>Student Details</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<center>
<form action="Servlet41">
<h1><b>SEARCH RECORD</b></h1>
<table border="">
<tr><td>Enter Name You want to search a record :</td><td><input type="text" name="fname" value=""></td></tr>
</table>
<table border="">
<br>
<tr><td><input type="Submit" name="Save" value="Search"></td></tr>
</table>
</form>
</center>
</body>
</html>
```

Update.html

```
<html>
```

```
<head>
<title>Student Details</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<center>
<form action="Servlet21">
<h1><b>UPDATE RECORD</b></h1>
<table border="">
<tr><td>Enter Enrollment You want to update a record
:</td><td><inputtype="text" name="enroll" value=""></td></tr>

<tr><td>Enter Name of your Enrollment You want to update a

record :</td><td><input type="text" name="fname" value=""></td></tr>

</table>
<table border="">
<br>
<tr><td><input type="Submit" name="Save" value="UPDATE"></td></tr>
</table>
</form>
</center>
</body>
</html>
```

Delete.html

```
<html>
```

```
<head>
<title>Student Details</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<center>
<form action="Servlet31">
<h1><b>DELETE RECORD</b></h1>
<table border="">
<tr><td>Enter Name You want to delete record :</td><td><input
type="text"name="fname" value=""></td></tr>
</table>
<table border="">
<br>
<tr><td><input type="Submit" name="Save" value="DELETE"></td></tr>
</table>
</form>
</center>
</body>
</html>
```

Servlet1.java

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

```
javax.servlet.annotation.WebServlet;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import
javax.servlet.http.HttpServletRequest;
import
javax.servlet.http.HttpServletResponse;

@WebServlet("/Servlet1")
public class Servlet1 extends
HttpServlet {Statement st=null;
Connection con=null;
static final String
DB_URL="jdbc:mysql://localhost:3306/students";static final
String USER="root";
static final String PASS="";

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-
8");try (PrintWriter out = response.getWriter()) {

String s1,s2,s3,s4,s5,sql;
s1=request.getParameter("fname");
s2=request.getParameter("lname");
s3=request.getParameter("sem");
s4=request.getParameter("contact");
s5=request.getParameter("enroll");
```

```
out.println("<!DOCTYPE
html>");out.println("<html>");
out.println("<head>");
out.println("<title>Servlet Data
Insertion</title>");out.println("</head>");
out.println("<body>");
out.println("<h3>First Name: " + s1 +
"</h3>");out.println("<h3>Last Name: " +
s2 + "</h3>");out.println("<h3>Semester: "
+ s3 + "</h3>"); out.println("<h3>Contact: "
+ s4 + "</h3>");
out.println("<h3>Enrollment: " + s5 +
"</h3>");out.println("</body>");
out.println("</html>");

Class.forName("com.mysql.jdbc.Driver");
con =
DriverManager.getConnection(DB_URL,USER,PASS);st
= con.createStatement();
sql=" insert into
student_18(fname,lname,enroll,contact,sem)values('"+s1+"','"+s2+"','"+s5+"','"+s4+"','"+s3+"')";

st.executeUpdate(sql);
System.out.println("Record Inserted Sucessfuly...");

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

```
@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";
}

}
```

Servlet2.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import
java.sql.DriverManager;
import
java.sql.ResultSet;
import
java.sql.Statement;
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("/Servlet2")
public class Servlet2 extends
HttpServlet {Statement st=null;
Connection
con=null;
ResultSet rs;
static final String
DB_URL="jdbc:mysql://localhost:3306/students";static final
String USER="root";
static final String PASS="";
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-
8");try (PrintWriter out = response.getWriter()) {
String s1,s2,sql;
s1=request.getParameter("fnam
e");
s2=request.getParameter("enrol
l");out.println("<!DOCTYPE
html>");out.println("<html>");
out.println("<head>");
out.println("<title>Record
Updation</title>");out.println("</head>");
out.println("<body>");
```

```
out.println("<h3>Name Changed Succesfully
</h3>");out.println("</body>");
out.println("</html>");

Class.forName("com.mysql.jdbc.Driver");
con =
DriverManager.getConnection(DB_URL,USER,PASS);st
= con.createStatement();
sql=" update student_18 set fname= '"+s1+"' where enroll='"+s2+"'
";st.executeUpdate(sql);
System.out.println("Record Updated Sucessfuly...");

rs = st.executeQuery("select * from student_18 where fname='"+s1+"'");
while(rs.next())
{
out.println("<h5>First Name:-"+rs.getString(1)+
"</h5>"); out.println("<h5>Last Name:-
"+rs.getString(2)+ "</h5>");
out.println("<h5>Enrollment:-"+rs.getString(3)+
"</h5>"); out.println("<h5>Contact No:-
"+rs.getString(4)+ "</h5>"); out.println("<h5>Sem:-
"+rs.getString(5)+ "</h5>");

}
out.println("</body
>");
out.println("</html
>");
}catch(Exception e)
{
```

```
System.out.println(e.toString());
}
}
@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";
}// </editor-fold>

}
```

Servlet3.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import
java.sql.DriverManager;
import java.sql.Statement;
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("/Servlet3")

public class Servlet3 extends

HttpServlet {Statement st=null;

Connection con=null;

static final String

DB_URL="jdbc:mysql://localhost:3306/students";static final

String USER="root";

static final String PASS="";


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. */


String s1,sql;


s1=request.getParameter("fnam

e");out.println("<!DOCTYPE

html>");out.println("<html>");

out.println("<head>");

out.println("<title>Record

Updation</title>");out.println("</head>");

out.println("<body>");

out.println("<h3>Record Deleted of name: " + s1 +

"</h3>");out.println("</body>");
```

```
out.println("</html>");

Class.forName("com.mysql.jdbc.Driver");
con =
DriverManager.getConnection(DB_URL,USER,PASS);st
= con.createStatement();
sql=" delete from student_18 where
fname='"+s1+"' ";st.executeUpdate(sql);
System.out.println("Record Deleted Sucessfuly...");

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

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";
```

```
}
}
```

Servlet4.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import
java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
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("/Servlet4")
public class Servlet4 extends
HttpServlet {Statement st=null;
Connection con=null;

static final String
DB_URL="jdbc:mysql://localhost:3306/students";static final
String USER="root";
static final String PASS="";
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)throws ServletException, IOException {
```

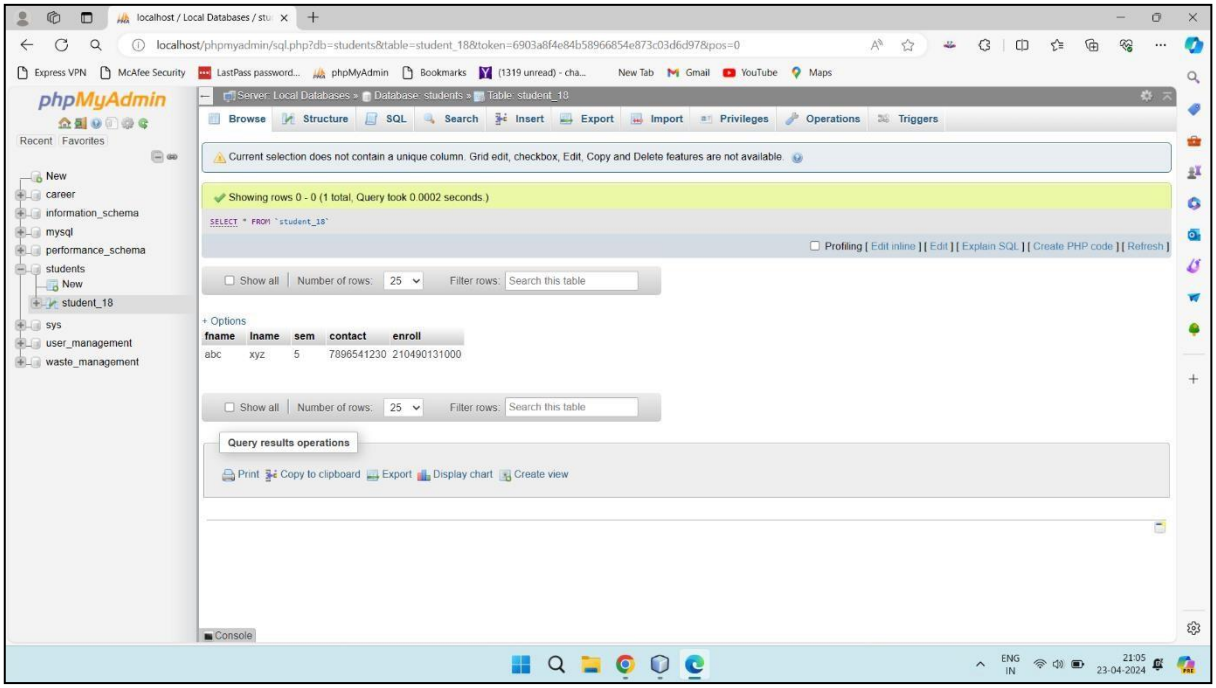
```
response.setContentType("text/html;charset=UTF-
8");try (PrintWriter out = response.getWriter()) {
String s1,sql;
s1=request.getParameter("fnam
e");out.println("<!DOCTYPE
html>");out.println("<html>");
out.println("<head>");
out.println("<title>Record
Updation</title>");out.println("</head>");
out.println("<body>");
out.println("<h3>Record Searched Value of name: " + s1 + "</h3>");

Class.forName("com.mysql.jdbc.Driver");
con =
DriverManager.getConnection(DB_URL,USER,PASS);st
= con.createStatement();
System.out.println("Record Search
Sucessfuly...");ResultSet rs;
rs = st.executeQuery("select * from student_18 where fname="'+s1+'"");
while(rs.next())
{
out.println("<h5>First Name:-"+rs.getString(1)+
"</h5>"); out.println("<h5>Last Name:-
"+rs.getString(2)+ "</h5>");
out.println("<h5>Enrollment:-"+rs.getString(3)+
"</h5>"); out.println("<h5>Contact No:-
"+rs.getString(4)+ "</h5>"); out.println("<h5>Sem:-
"+rs.getString(5)+ "</h5>");

}
out.println("</body
```

```
>");
out.println("</html
>");
}catch(Exception e)
{
System.out.println(e.toString());
}
}
@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:

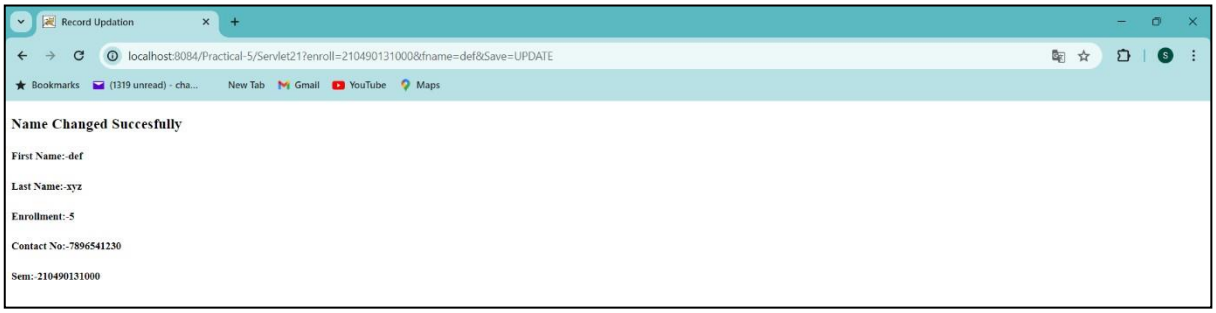
Insert Record

Database Table after Record Registered



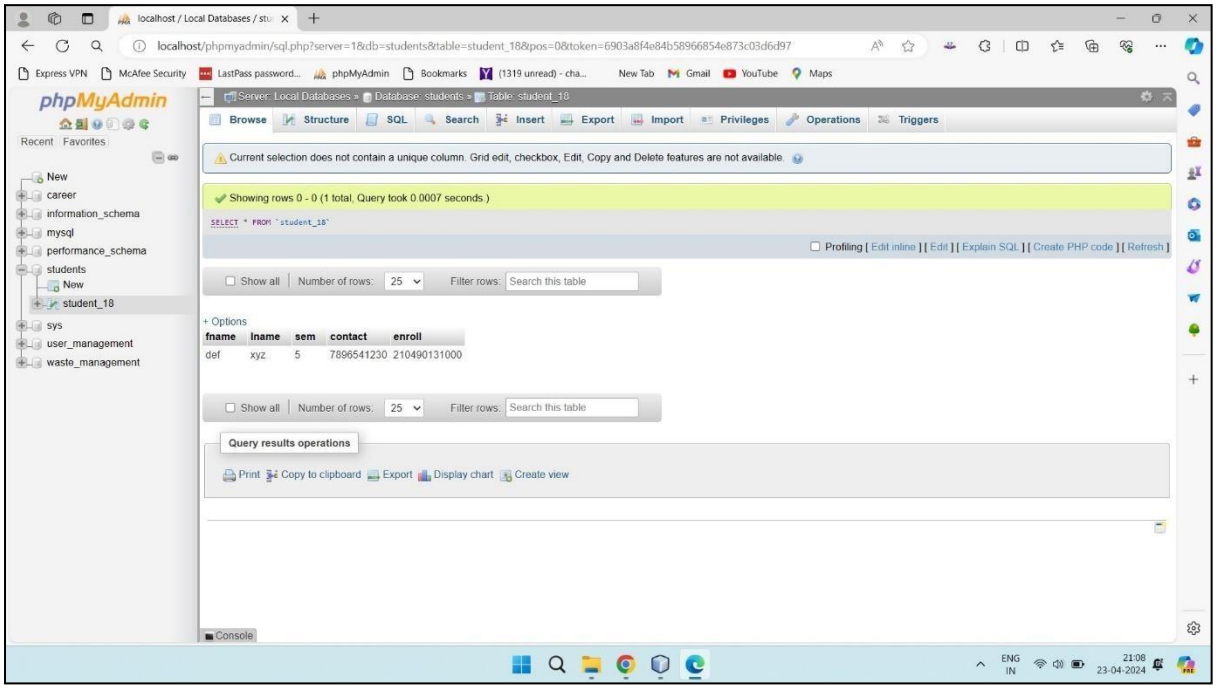
Search Record

After Searching Data in Database



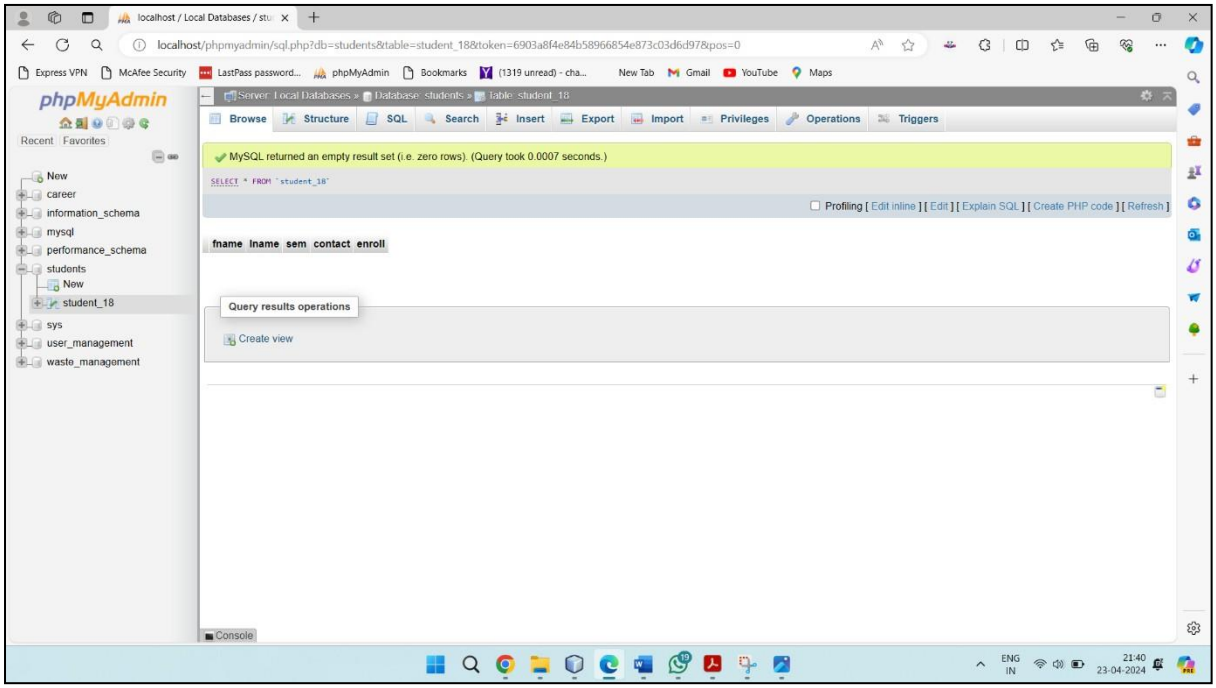
Update Record

After Updating Record



Delete Record

After Deleting record



Practical: 6

Aim: Implement cookies to store firstname and lastname using Java server pages.

PROGRAM:

Cookie.html

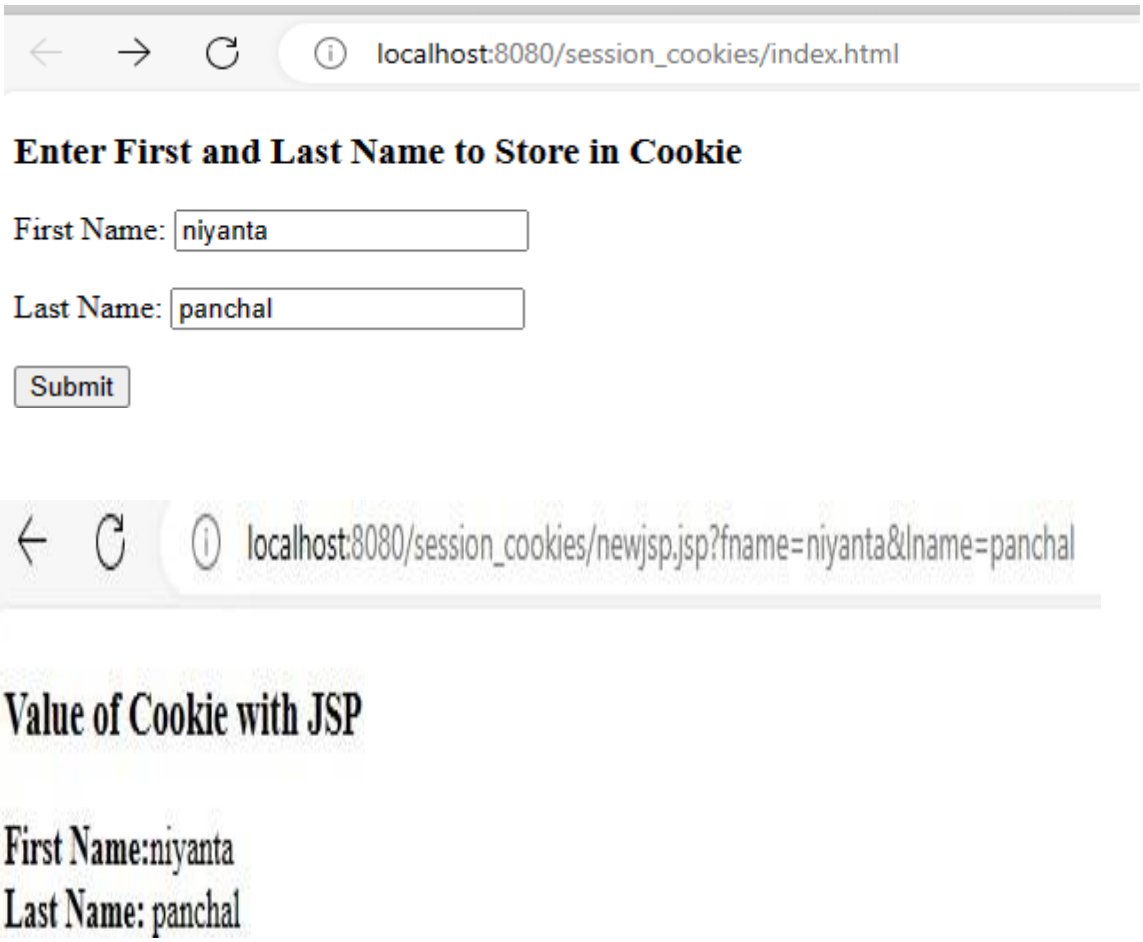
```
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <h3>Enter First and Last Name to Store in Cookie</h3>
        <form action="cookie.jsp" method="GET">
            First Name: <input type="text" name="fname"><br><br>
            Last Name: <input type="text" name="lname" /><br><br>
            <input type="submit" value="Submit" />
        </form>
    </body>
</html>
```

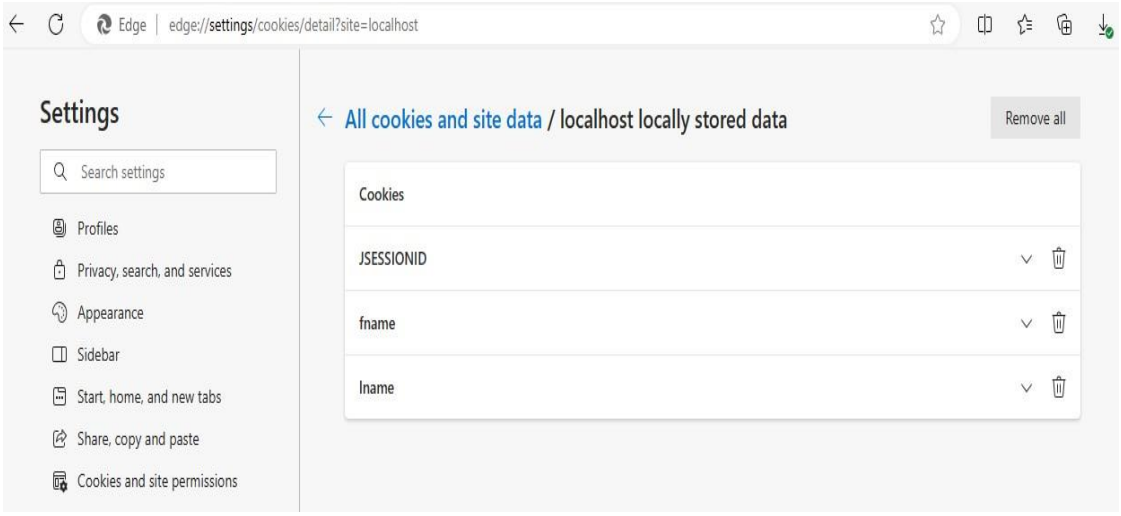
Cookie.jsp

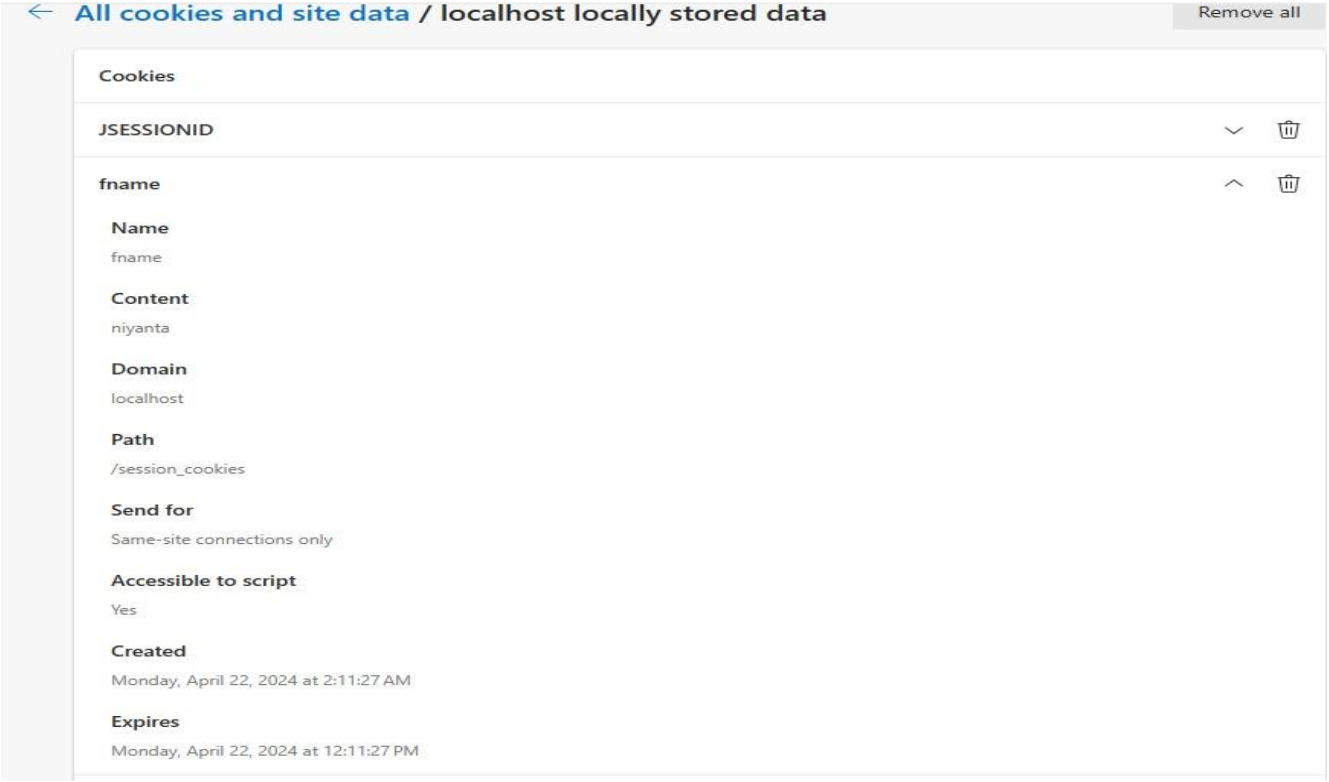
```
<%@page contenttype="text/html" pageencoding="UTF-8"%>
<!DOCTYPE html>
<%
    Cookie fname = new Cookie("fname", request.getparameter("fname"));
    Cookie lname = new Cookie("lname", request.getparameter("lname"));
    Fname.setmaxage(60 * 60 * 10);
    Lname.setmaxage(60 * 60 * 10);
    Response.addcookie(fname);
    Response.addcookie(lname);
%>
<html>
```

```
        <head>
            <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
            <title>JSP Page</title>
        </head>
        <body>
            <h3>Value of Cookie with JSP </h3>
            <b>First Name:</b><%= request.getparameter("fname")%><br>
            <b>Last Name:</b> <%= request.getparameter("lname")%>
        </body>
    </html>
```

Output:



- 
Edge | edge://settings/cookies/detail?site=localhost
Settings
Search settings
Profiles
Privacy, search, and services
Appearance
Sidebar
Start, home, and new tabs
Share, copy and paste
Cookies and site permissions
All cookies and site data / localhost locally stored data
Remove all
Cookies
JSESSIONID
fname
lname


All cookies and site data / localhost locally stored data
Remove all
Cookies
JSESSIONID
fname
Name
fname
Content
niyanta
Domain
localhost
Path
/session_cookies
Send for
Same-site connections only
Accessible to script
Yes
Created
Monday, April 22, 2024 at 2:11:27 AM
Expires
Monday, April 22, 2024 at 12:11:27 PM

Practical: 7

Aim: Implement the shopping cart for users for online shopping. Apply the concept Of session.

PROGRAM:

Index.jsp

```
<%@ page session="true" %>
<!DOCTYPE html>
<html>
  <head>
    <title>Online Shopping</title>
  </head>
  <body>
  <center>
    <h1>Products</h1>
    <table border="3px" borderpadding="0">
      <tr>
        <th>Book</th>
        <th>Image</th>
        <th>Add into Cart</th>
      </tr>
      <tr>
        <td><span>Soul</span></td>
        <td><img src="b2.jpg" width="200" height="250"></td>
        <td>Rs.100 <br><button onclick="addtocart('Book')">Add to
            Cart</button></td>
      </tr>
      <tr></tr>
      <tr>
        <td><span>Anubhuti</span></td>
        <td><img src="b3.jpg" width="200" height="250"></td>
        <td>Rs.150 <br><button onclick="addtocart('Book')">Add to
            Cart</button></td>
      </tr>
```

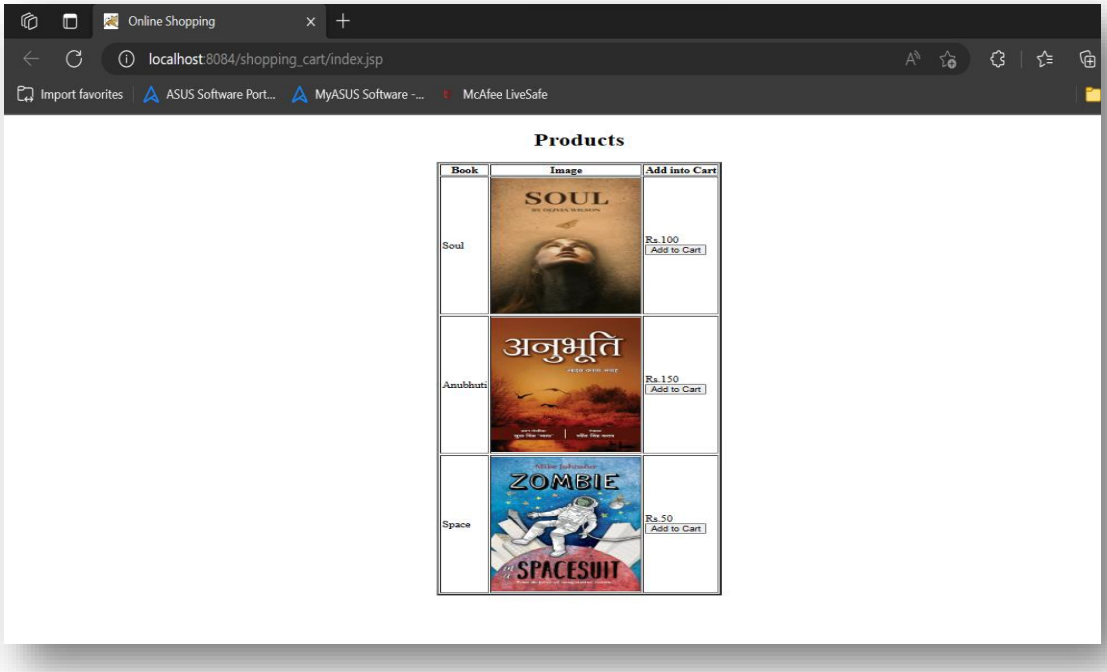
```
            <tr></tr>
            <tr>
              <td><span>Space</span></td>
              <td><img src="b4.jpg" width="200" height="250"></td>
              <td>Rs.50 <br><button onclick="addtocart('Book')">Add to
                    Cart</button></td>
            </tr>
          </table>
          <script>
            Function addtocart(productid) {
              Location.href = "cart.jsp?Productid=" + productid;
            }
          </script>
        </center>
      </body>
      </html>
```

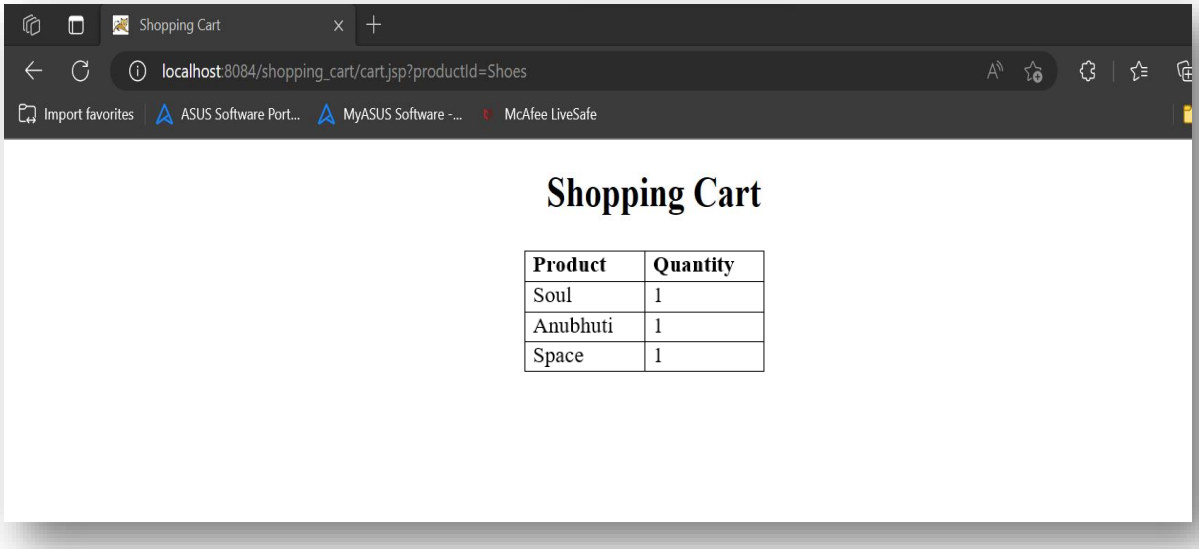
Cart.jsp

```
      <%@ page session="true" %>
      <!DOCTYPE html>
      <html>
        <head>
          <title>Shopping Cart</title>
        </head>
        <body>
        <center>
          <h1>Shopping Cart</h1>
          <%-- Check if the cart session attribute exists --%>
          <%
            If (session.getattribute("cart") == null) {
              Session.setattribute("cart", new java.util.hashmap<String, Integer>());
            }
```

```
            Java.util.hashmap<String, Integer> cart = (java.util.hashmap<String, Integer>)
                    session.getattribute("cart");

            String productid = request.getparameter("productid");
            If (productid != null) {
               If (cart.containskey(productid)) {
                  Int quantity = cart.get(productid);
                  Cart.put(productid, quantity + 1);
               } else {
                  Cart.put(productid, 1);
               }
            }
         %>
         <table border="1px">
            <thead>
               <tr>
                  <th>Product</th>
                  <th>Quantity</th>
               </tr>
            </thead>
            <tbody>
               <%-- Iterate over the cart items and display them --%>
               <% for (java.util.Map.Entry<String, Integer> entry : cart.entryset()) {%>
               <tr>
                  <td><%= entry.getkey()%></td>
                  <td><%= entry.getvalue()%></td>
               </tr>
               <% }%>
            </tbody>
         </table>
      </center>
      </body>
   </html>
```

Output:


Online Shopping
localhost:8084/shopping_cart/index.jsp
Import favorites
ASUS Software Port...
MyASUS Software -...
McAfee LiveSafe
Products
Book
Image
Add into Cart
Soul
Rs. 100
Add to Cart
Anubhuti
Rs. 150
Add to Cart
Space
Rs. 50
Add to Cart


Shopping Cart
localhost:8084/shopping_cart/cart.jsp?productId=Shoes
Import favorites
ASUS Software Port...
MyASUS Software -...
McAfee LiveSafe
Shopping Cart
Product
Quantity
Soul
1
Anubhuti
1
Space
1

Practical: 8

Aim: Design a web page that takes the Username from the user and if it is a Valid username prints “Welcome Username”. Use JSF to implement.

PROGRAM:

Index.xhtml

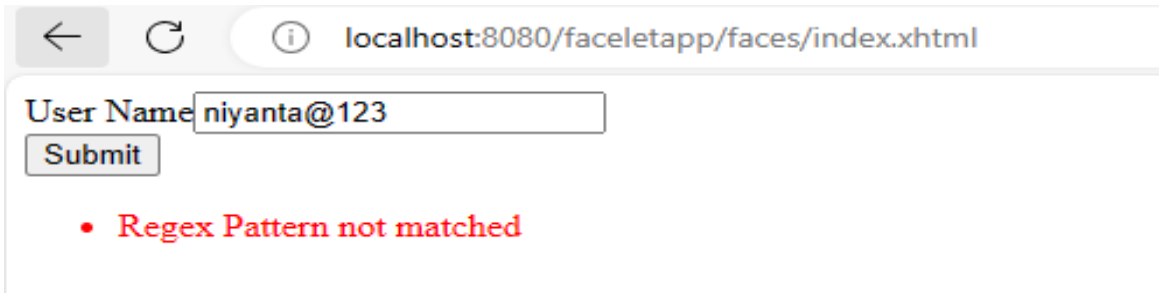
```
<?Xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
Xmlns:h="http://xmlns.jcp.org/jsf/html"
Xmlns:f="http://xmlns.jcp.org/jsf/core">
<h:head>
<title>validation</title>
</h:head>
<h:body>
<h:form>
<h:outputlabel for="username">User Name</h:outputlabel>
<h:inputtext id="username" value="#{user.name}" required="true"
Requiredmessage="Enter Username First...!!" >
<f:validaterequired/>
<f:validatelength minimum="5" maximum="20" />
<f:validateregex pattern="^([a-za-Z]+(.)?[\s]*)$" />
</h:inputtext><br></br>
<h:commandbutton id="submit-button" value="Submit" action="response.xhtml"/>
</h:form>
</h:body>
</html>
```

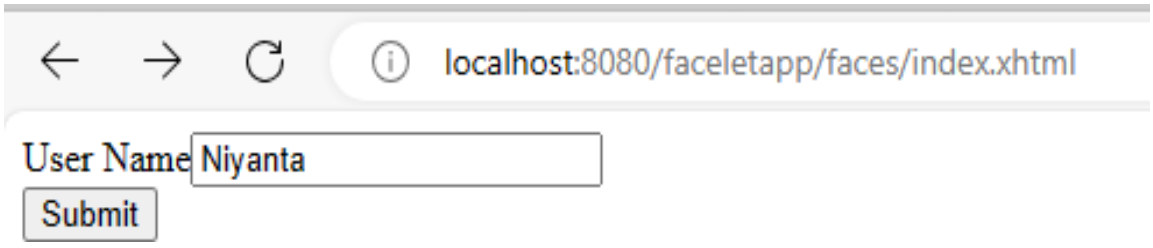
Responce.xhtml

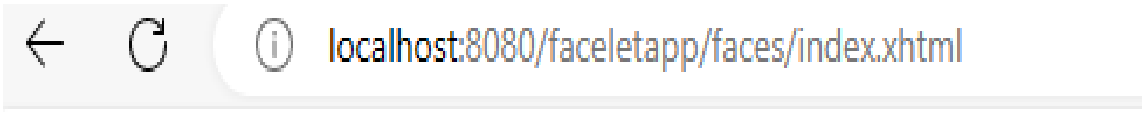
```
<?Xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
Xmlns:h="http://xmlns.jcp.org/jsf/html">
```

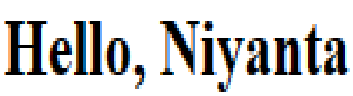
```
<h:head>
<title>Welcome Page</title>
</h:head>
<h:body>
<h2>Hello, <h:outputtext value="#{user.name}"></h:outputtext></h2>
</h:body>
</html>
```

Output:









Practical: 9

Aim: Write a Hibernate application to store customer records and retrieve the Customer record including name, contact number, address.

PROGRAM:

Hibernate.cfg.xml

```
<?Xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration
        DTD
3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
 <session-factory>
 <property name="hibernate.dialect">org.hibernate.dialect.mysqldialect</property>
 <property
        name="hibernate.connection.driver_class">com.mysql.jdbc.Driver
        </property>
 <property
        name="hibernate.connection.url">jdbc:mysql://localhost:3306/gtu
        </property>
 <property name="hibernate.connection.username">root</property>
 <mapping resource="test/Customer.hbm.xml"/>
 </session-factory>
</hibernate-configuration>
```

Hibernate.reveng.xml

```
<?Xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-reverse-engineering PUBLIC "-//Hibernate/Hibernate Reverse
        Engineering
DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-reverse-engineering-
        3.0.dtd">
<hibernate-reverse-engineering>
 <schema-selection match-catalog="gtu"/>
 <table-filter match-name="customer "/>
</hibernate-reverse-engineering>
```

Customer.hbm.xml

```
<?Xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD
                    3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<!-- Generated 17 May, 2023 10:53:53 PM by Hibernate Tools 4.3.1 -->
<hibernate-mapping>
   <class name="test.Customer " table="customert" catalog="gtu" optimistic-
                    lock="version">
      <id name="name " type="string">
         <column name="Name " length="20" />
         <generator class="assigned" />
      </id>
      <property name="contactno " type="string">
         <column name="contactno " length="20" not-null="true" />
      </property>
      <property name="address " type="string">
         <column name="Address " length="20" not-null="true" />
      </property>
   </class>
</hibernate-mapping>
```

Customer.java

```
Package test;
Public class Customer  implements java.io.Serializable {
    Private String name;
    Private String contactno;
    Private String address;

   Public Customer() {
   }

    Public (String name, String contactno, String address,  String sem) {
```

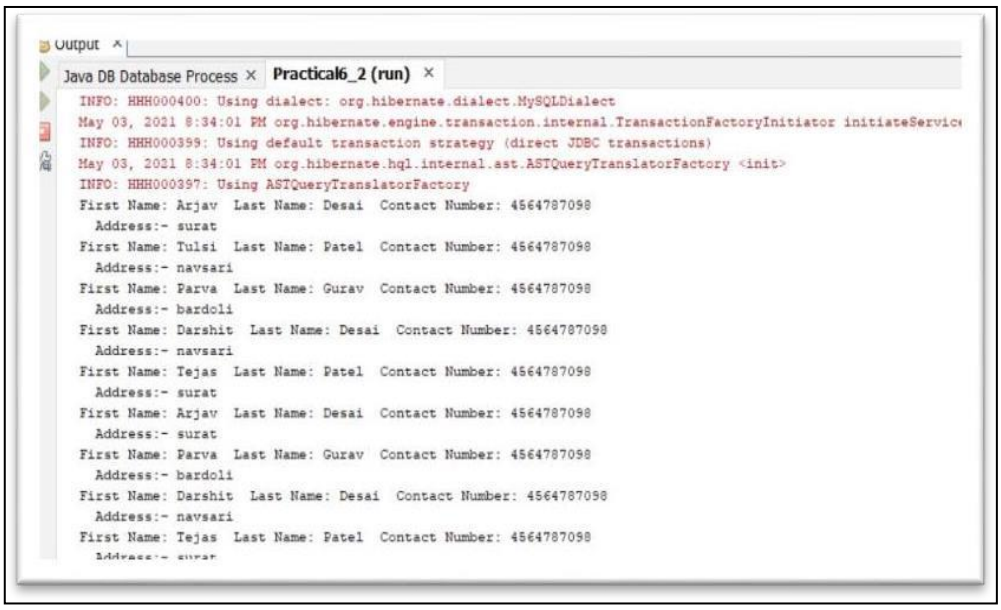
```
        This.name = name;
        This.contactno = contactno;
        This.address = address;
    }

    Public String getname() {
        Return this.name;
    }

    Public void setname(String name) {
        This.name = name;
    }
    Public String getcontactno() {
        Return this.contactno;
    }

    Public void setcontactno(String contactno) {
        This.contactno = contactno;
    }
    Public String getaddress() {
        Return this.address;
    }

    Public void setaddress(String address) {
        This.address = address;
    }
}
```

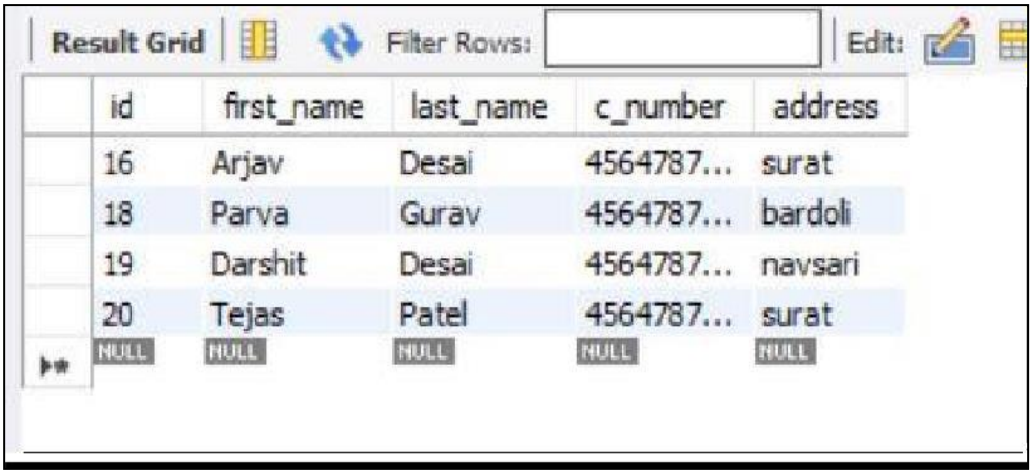
Storedata.java

```
Import org.hibernate.Session;
Import org.hibernate.sessionfactory;
Import org.hibernate.Transaction;
Import org.hibernate.cfg.Configuration;
Import java.util.*;
```

```
Import org.hibernate.Query;
Import test.Customer;
Public class storedata {
   Public static void main(String[] args) {
      Configuration cfg = new Configuration();
      Cfg.configure("hibernate.cfg.xml");//populates the data of the configuration file
      Sessionfactory factory = cfg.buildsessionfactory();
      Session session = factory.opensession();
      String hql = "FROM Customer";
      Query query = session.createquery(hql);
      List results = query.list();
      Iterator it = results.iterator();
      System.out.println("contactno\tname \taddress ");
      System.out.println("==================================");
      While (it.hasnext()) {
         Customer e1 = (Customer) it.next();
         System.out.print(e1.getname() + "\t");
         System.out.print(e1.getcontactno() + "\t");
         System.out.print(e1.getaddress() + "\t");
         System.out.println("");
      }
      Transaction t = session.begintransaction();
      Session.close();
   }}
```

Output:





id	first_name	last_name	c_number	address
16	Arjav	Desai	4564787...	surat
18	Parva	Gurav	4564787...	bardoli
19	Darshit	Desai	4564787...	navsari
20	Tejas	Patel	4564787...	surat
NULL	NULL	NULL	NULL	NULL

Practical: 10

Aim: Write an application to keep records and retrieve records of students. The record includes student id, enrollment number, semester, SPI. Use MVC architecture.

PROGRAM:

Index.html:

```
<?Xml version="1.0" encoding="ISO-8859-1" ?>
<!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://java.sun.com/jsf/html"
Xmlns:f="http://java.sun.com/jsf/core">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Index</title>
</head>
<h:body>
<f:view>
<h3>Enter Details</h3>
<h:form>
<table>
<tr>
<td>ID:</td><td><h:inputtext value="#{bean.id}"/></td>
</tr>
<tr>
<td>Enrollment Number:</td><td><h:inputtext value="#{bean.en}"/></td>
</tr>
<tr>
<td>Semester:</td><td><h:inputtext value="#{bean.sem}"/></td>
</tr>
<tr>
<td>SPI:</td><td><h:inputtext value="#{bean.spi}"/></td>
</tr>
```

```
</table>
<h:commandbutton value="Submit" action="#{bean.submit}"/>
</h:form>
<br></br>
<h:outputtext value="#{bean.data}" escape="false" />
</f:view>
</h:body>
</html>
```

Bean.java:

```
Package jsfpackage;
Import
Java.io.Serializable;
Import javax.faces.bean.managedbean;
Import java.sql.*;
@managedbean
Public class bean implements Serializable {
Private static final long serialversionuid = 6529685098267757690L;private int id;
Private String en;
Private int sem;
Private float spi;
Private String data = "";
Public String submit() throws sqlexception, classnotfoundexception{
Class.forname("com.mysql.jdbc.Driver");
Connection con =
Drivermanager.getconnection("jdbc:mysql://127.0.0.1:3306/test?Characterencodin
G=utf8&usessl=false&useunicode=true","root","root");
Preparedstatement st = con.preparestatement("insert into student
values(?,?,?,?)");st.setint(1,
Getid());
St.setstring(2,
Geten());st.setint(3,
```

```
Getsem());
St.setfloat(4, getspi());st.execute();
Statement stt = con.createstatement();
Resultset rs = stt.executequery("select * from student");
This.data = "<table style=\"width:50%\" bgcolor=\"cyan\" border=\"1px\"><tr
Bgcolor=\"teal\"><th>Id</th><th>Enrollment</th><th>Semester</th><th>SPI</th
></tr>";
While(rs.next())
{ int i = rs.getint(1);
String en = rs.getstring(2);
Int sm = rs.getint(3);
Float sp = rs.getfloat(4);
This.data                                                                    +=
"<tr><td>"+i+"</td><td>"+en+"</td><td>"+sm+"</td><td>"+sp+"</td></tr>";
}
This.data += "</table>";
Con.close();
Return "index.xhtml";
}
Public String getdata()
{
Return this.data;
}

Public int getid() {
Return id;
}
Public void setid(int id) {
This.id = id;
}
Public String geten(){
Return en;
}
```

```
Public void seten(String en) {
This.en = en;
}
Public int getsem(){
Return sem;
}
Public void setsem(int sem) {
This.sem = sem;
}
Public float getspi() {
Return spi;
}
Public void setspi(float spi) {

This.spi = spi;
}
}
```

Web.xml:

```
<?Xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/xmlschema-instance"
Xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemalocation="http://xmlns.jcp.org/xml/ns/javaee
;http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
id="webapp_ID"version="3.1">
<display-name>P11 JSF</display-name>
<welcome-file-list>
<welcome-file>index.xhtml</welcome-file>
<welcome-file>index.htm</welcome-file>
<welcome-file>index.jsp</welcome-file>
<welcome-file>default.html</welcome-file>
<welcome-file>default.htm</welcome-file>
<welcome-file>default.jsp</welcome-file>
```

```
</welcome-file-list>
<servlet>
<servlet-name>Faces Servlet</servlet-name>
<servlet-class>javax.faces.webapp.facesservlet</servlet-class>

<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>Faces Servlet</servlet-name>
<url-pattern>/faces/*</url-pattern>
</servlet-mapping>
<context-param>
<description>State saving method: 'client' or 'server' (=default). See jsfspecification
2.5.2</description>
<param-name>javax.faces.STATE_SAVING_METHOD</param-name>
<param-value>client</param-value>
</context-param>
<context-param>
<param-name>javax.servlet.jsp.jstl.fmt.localizationcontext</param-name>
<param-value>resources.application</param-value>
</context-param>
<listener>
<listener-class>com.sun.faces.config.configurelistener</listener-class>
</listener>
</web-app>
```

Faces-config.xml:

```
<?Xml version="1.0" encoding="UTF-8"?>
<faces-config
Xmlns="http://xmlns.jcp.org/xml/ns/javaee"
Xmlns:xsi="http://www.w3.org/2001/xmlschema-instance"
Xsi:schemalocation="http://xmlns.jcp.org/xml/ns/javaee
;http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd"version="2.2">
```

```
<navigation-rule>
<from-view-id>/index.xhtml</from-view-id>
<navigation-case>
<from-outcome>index</from-outcome>
<to-view-id>/index.xhtml</to-view-id>
</navigation-case>
</navigation-rule>
<managed-bean>
<managed-bean-name>bean</managed-bean-name>
<managed-bean-class>jsfpackage.bean</managed-bean-class>
<managed-bean-scope>session</managed-bean-scope>
</managed-bean>
</faces-config>
```

Output: