Experiment 10

Implementing Log in using JSP JDBC and Servlet

Name: Sai Harsha Vardhan AVN

Roll-no: BCSE1823

Batch: ‘A’

Aim: Implementing Log in using JSP, JDBC and Servlet.

Theory:

Servlets are Java classes which service HTTP requests and implement the **javax.servlet.Servlet** interface. Web application developers typically write servlets that extend javax.servlet.http.HttpServlet, an abstract class that implements the Servlet interface and is specially designed to handle HTTP requests.

Steps to create a servlet example

There are given 6 steps to create a **servlet example**. These steps are required for all the servers.

The servlet example can be created by three ways:

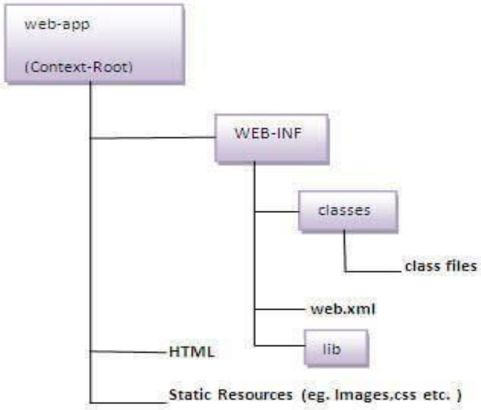
1. By implementing Servlet interface,
2. By inheriting GenericServlet class, (or)
3. By inheriting HttpServlet class

The mostly used approach is by extending HttpServlet because it provides http request specific method such as doGet(), doPost(), doHead() etc.

There are few server's to run your application , you can use either of those for example we have **apache tomcat server, Glassfishserver etc**. The steps are as follows:

1. Create a directory structure
2. Create a Servlet
3. Compile the Servlet
4. Create a deployment descriptor
5. Start the server and deploy the project
6. Access the servlet

Directory/File Structure for Servlet Programs to run is shown below:



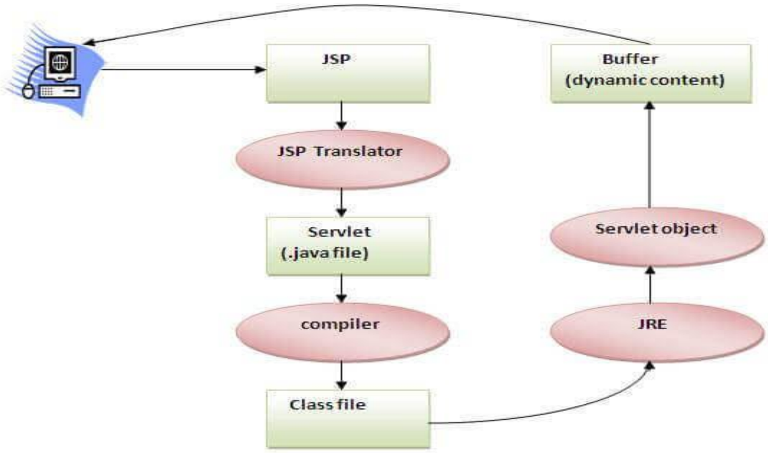
JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

The Lifecycle of a JSP Page

The JSP pages follow these phases:

- Translation of JSP Page
- Compilation of JSP Page
- Classloading (the classloader loads class file)
- Instantiation (Object of the Generated Servlet is created).
- Initialization (the container invokes jspInit() method).
- Request processing (the container invokes _jspService() method).
- Destroy (the container invokes jspDestroy() method).



Creating a simple JSP Page

To create the first JSP page, write some HTML code as given below, and save it by .jsp extension. We have saved this file as index.jsp. Put it in a folder and paste the folder in the web-apps directory in apache tomcat to run the JSP page.

index.jsp

Let's see the simple example of JSP where we are using the scriptlet tag to put Java code in the JSP page. We will learn scriptlet tag later.

```
<html>
<body>
<% out.print(2*5); %>
</body>
</html>
```

Code:

1) **Index.jsp Code:**

```
<%@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="<%= request.getContextPath() %>/Database" method="get">
            <fieldset>
                <legend>Student Information</legend>
                <table border="0" bgcolor="grey">
                    <tr>
                        <td>Student ID</td>
                        <td style="color:red">*</td>
                        <td>
                            <input type="text" name="txtStudentID">
                        </td>
                    </tr>
                    <tr>
                        <td>First Name</td>
                        <td style="color:red">*</td>
                        <td>
                            <input type="text" name="txtStudentFirstName">
                        </td>
                    </tr>
                    <tr>
                        <td></td>
                        <td></td>
                        <td>
                            <input type="submit" value="Submit">
                            <input type="reset" value="Reset">
                        </td>
                    </tr>
                </table>
            </fieldset>
        </form>
    </body>
</html>
```

2) Web.xml Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" 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-app_3_1.xsd">
    <servlet>
        <servlet-name>Database</servlet-name>
        <servlet-class>Database</servlet-class>
        <servlet-name>Insertion</servlet-name>
        <servlet-class>Insertion</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>Database</servlet-name>
        <url-pattern>/Database</url-pattern>
        <servlet-name>Insertion</servlet-name>
        <url-pattern>/Insertion</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>
```

3) Database.java (Servlet Code):

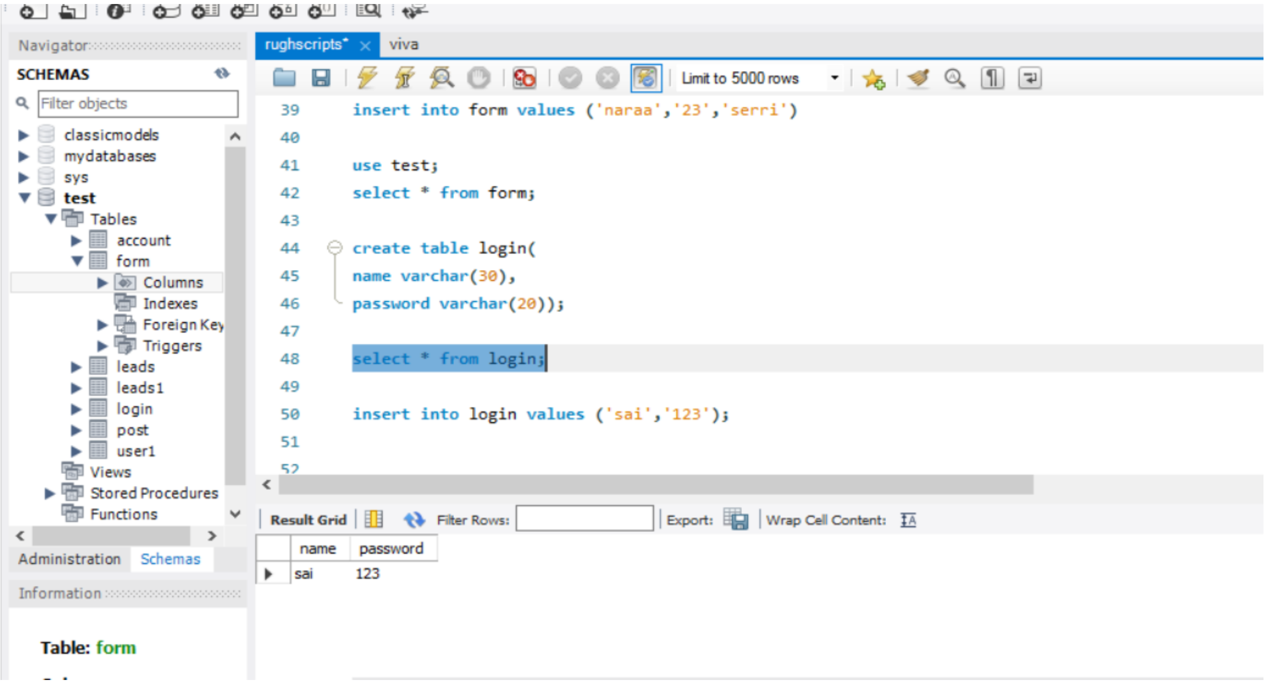
```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;

/**
 *
 * @author user new
 */
public class Database extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {

            String name = request.getParameter("txtname");
            String pass = request.getParameter("txtpass");
            //String address = request.getParameter("txtStudentaddress");
            String htmlRespone = "<html>";
            htmlRespone += "<form>";
            htmlRespone += "<h2>" + "Name is: " + name + "</h2>";
            htmlRespone += "<h2>" + "Password is: " + pass + "</h2>";
            //htmlRespone += "<h2>" + "Address is: " + address + "</h2>";
            htmlRespone += "</form>";
            htmlRespone += "</html>";
            out.println(htmlRespone);
            Connection con;
            Statement stmt;
           try {
                Class.forName("com.mysql.jdbc.Driver");
                con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test?useSSL=false",
"root", "mysql");
                stmt = con.createStatement();
                String query = "select * from login where name='"+name+"'";
                ResultSet rs = stmt.executeQuery(query);
                if(rs.next()){
                  out.print("sucess");
                }else{
                    out.print("not sucess!");
                }

            } catch (Exception ex) {
                System.out.print(ex);
            }

        }

    }
}
```

4) Insertion.java(Servlet Code for Insertion of data in database):

```
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.Statement;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author user new
 */
public class Insertion extends HttpServlet {

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

            String name = request.getParameter("txtname");
            String pass = request.getParameter("txtpass");
            //String address = request.getParameter("txtStudentaddress");
            String htmlRespone = "<html>";
            htmlRespone += "<form>";
            htmlRespone += "<h2>" + "Name is: " + name + "</h2>";
            htmlRespone += "<h2>" + "Password is: " + pass + "</h2>";
            //htmlRespone += "<h2>" + "Address is: " + address + "</h2>";
            htmlRespone += "</form>";
            htmlRespone += "</html>";
            out.println(htmlRespone);
            Connection con;
            Statement stmt;
            PreparedStatement pst;

            int cnt = 0;

            try {
                Class.forName("com.mysql.jdbc.Driver");
                con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test?useSSL=false",
"root", "mysql");

                pst = con.prepareStatement("insert into login values(?,?)");

                pst.setString(1, name);
                pst.setString(2, pass);
                //pst.setString(3, address);

                cnt = pst.executeUpdate();

                if (cnt > 0) {
                    System.out.print("name:\n" + name + "\n" + "pass:\n" + pass+ "\n" +
"address:\n");
                    System.out.println("updated succefully");
                } else {
                    System.out.println("updation failed");
                }
            } catch (Exception ex) {
                System.out.print(ex);
            }

        }

    }
}
```

Output:

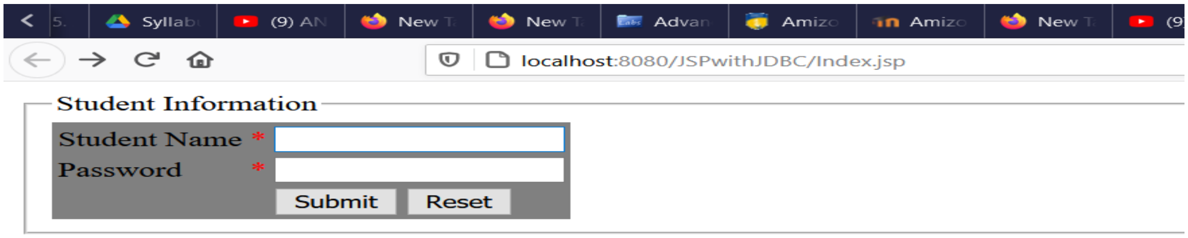
1) Number of records in the data base:



```
insert into form values ('naraa','23','serri')

use test;
select * from form;

create table login(
name varchar(30),
password varchar(20));

select * from login;

insert into login values ('sai','123');
```

name	password
sai	123

2) Login From

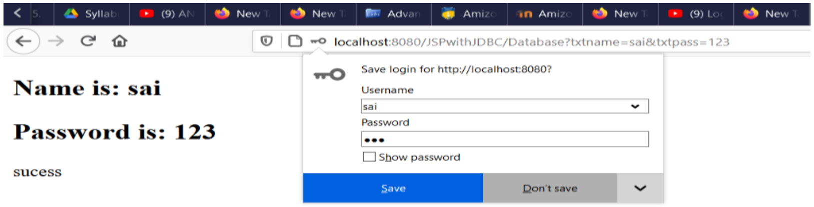


localhost:8080/JSPwithJDBC/Index.jsp

Student Information

Student Name *	
Password *	

3) The output when you give a correct password and username:



localhost:8080/JSPwithJDBC/Database?txtname=sai&txtpass=123

Name is: sai

Password is: 123

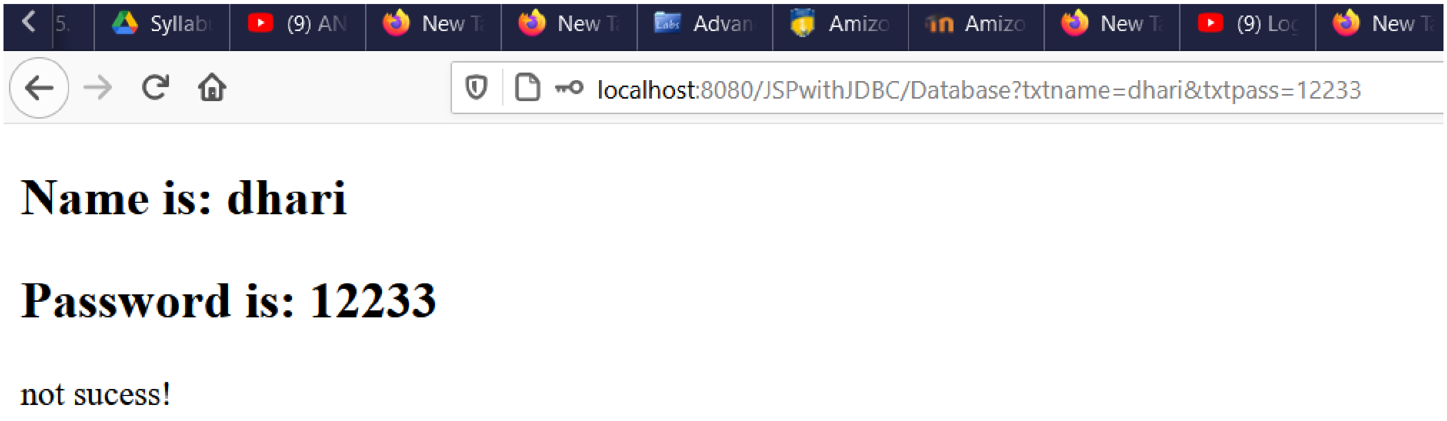
sucess

Save login for http://localhost:8080?

Username: sai

Password: •••

4) The output when you give a incorrect password and username which is not in database:



localhost:8080/JSPwithJDBC/Database?txtname=dhari&txtpass=12233

Name is: dhari

Password is: 12233

not sucess!

Conclusion: Log in using JSP, JDBC and Servlet is created successfully.