<u>In-Lab:</u>

Q1. You are a database security consultant for a company. The company has given you the task of creating a registration page which takes username and password as inputs and stores the hashed value of the password in a database table. You plan to store the hash value of the password in the database. The hashing is done using MD5. You are to implement this using Javascript.

First, create a html page with the username and password fields. Once the credentials are given as input and submit button is pressed, the javascript program in the background will hash the password and store it in the database along with the username.

Create a table called 'dbusers' to store the credentials.

Sol)

1. Connect as system to sqlplus and create the 'dbusers' table.
   create table dbusers(username varchar2(20), password varchar2(20), epwd varchar2(128));
2. Switch on the Eclipse IDE.
   Go to 'File' → New → Other → Scroll down to 'Web' → Expand the folder and choose 'Dynamic Web Project'.
3. Set project name as something of your choice.
4. Choose 'Dynamic web module version' as 2.5.
5. Click Finish.
6. Then expand the project file in 'Project Explorer'. Expand 'WebContent' → WEB-INF → Right click on 'lib' → Build Path… → Configure Build Path… → Click on 'Classpath' → Click 'Add External JARs…' → Navigate to the 'ojdbc14.jar' and add it.
7. Then right click on 'WebContent'. This is where you make your html and jsp files.
   Right click on 'WebContent' → New → HTML File.
   Give a filename and end it with '.html' extension. Press Finish.
   register.html

   --

```
<html> /* Entire Html doc*/
<head> /*head/prologue*/
<meta charset="ISO-8859-1">/*specify char encoding for html doc*/
<title>Insert title here</title>/*title of odc*/
</head>
<body> /*all other content in html doc*/
<form action="rlogin.jsp" method="post"> /*specifies where to send data whn a
form is submitted*/ /*Method specifies how to send data*/

Username <input type="text" name="uname"> /*Input fields*/

<br> /*Line break or carriage return*/
Password<C="password" name="pwd">

<br>
<input type="submit">

</form> /*HTML  form for user input*/
</body>
</html>
```

Save the file.

8. Right click on 'WebContent' → New → JSP File.

Give a filename and end it with '.jsp' extension. Press Finish.

rlogin.jsp

```
<%@

page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>/* The page directive is used to provide instructions to the container that
pertain to the current JSP page*/

<%@ page  import="java.sql.*" %> /* The static import statement is used to import read only live bindings which
are exported by another module
<%@page import=" java.security.MessageDigest"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>  Welcome    </h1>

<%
```

String u=request.getParameter("uname"); / * **Returns the value of a request parameter as a String , or null if the parameter does not exist*/**

String p=request.getParameter("pwd");

String algorithm="";
byte[] unencodedPassword = p.getBytes(); **/* The Java String getBytes() method encodes the string into a sequence of bytes and stores it in a byte array**.*/

MessageDigest md = **null;/* method is called to calculate message digest. // of an input digest() return array of byte*/**

try {
md = MessageDigest.getInstance("MD5");
} **catch** (Exception e) {}
md.reset(); **/* It is used to clear all the values of the form elements. It can be used to set the values to default. */**
md.update(unencodedPassword);
byte[] encodedPassword = md.digest();
StringBuffer buf = **new** StringBuffer**();/*StringBuilder allocates an array of char into which it copies the strings you append.*/**
for (**int** i = 0; i < encodedPassword.length; i++) {
if (((**int**) encodedPassword[i] & 0xff) < 0x10) {

/*0xff is a number represented in the hexadecimal numeral system (base 16). It's composed of two F numbers in hex.*/

buf.append("0");/* **an inbuilt method in Java which is used to append the string representation of the boolean argument to a given sequence**.*/
}
buf.append(Long.toString((**int**) encodedPassword[i] & 0xff, 16));
}
String passw=buf.toString();
/* **turns a string containing the source text segment which was used to define the function**. */

out.println(u +"----"+p);
try
{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","system");**/* Java DriverManager class attempts to establish a connection to the database by using the given database URL. */**

PreparedStatement ps=con.prepareStatement("insert into dbusers values(?,?,?)");/***The PreparedStatement interface extends the Statement interface it represents a precompiled SQL statement which can be executed multiple times**.*/
ps.setString(1,u);
ps.setString(2,p);
ps.setString(3,passw);

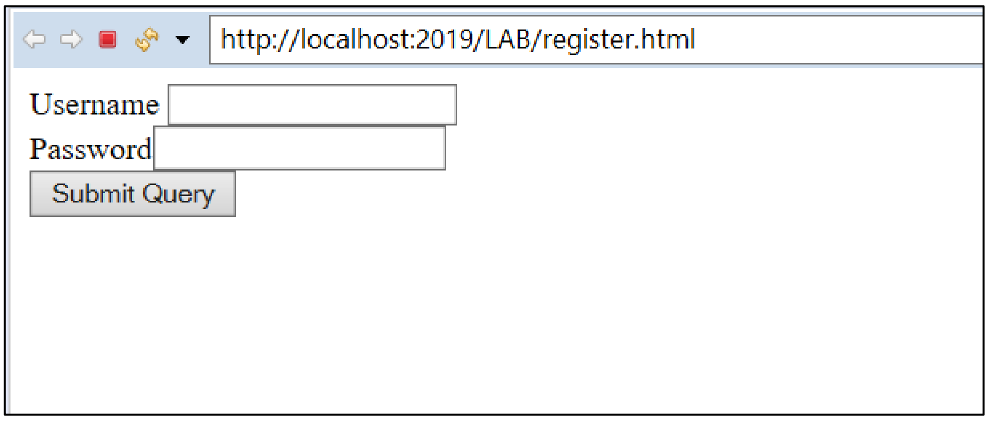
int a=ps.executeUpdate();

out.println(a+ " Record Inserted Successfully");
/* The executeUpdate() method returns the number of rows affected by the SQL statement (an INSERT typically affects one row, but an UPDATE or DELETE statement can affect more)*/
}
catch(Exception e)
{
out.println(e);

```
}

%>

</body>
        </html>
```

Save the file.

9. Right click on the html file and press 'Run As' → 1 Run on Server → (The Tomcat server should already be selected) → Press Finish.



http://localhost:2019/LAB/register.html

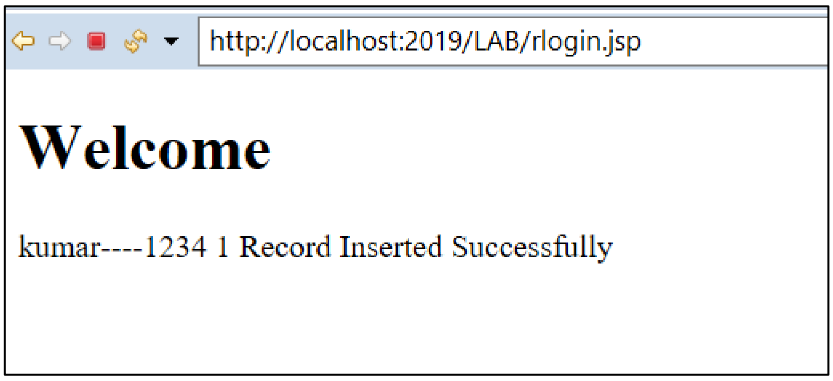
Username

Password

Submit Query

10. Give your username and password.

    The inputs I am giving are 'kumar' as username and '1234' as password.



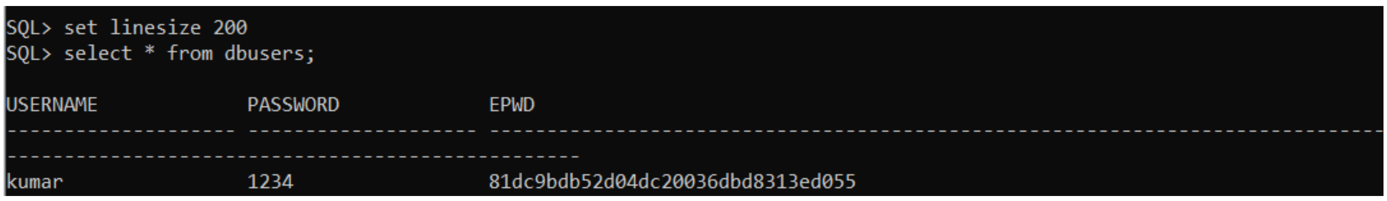
http://localhost:2019/LAB/rlogin.jsp

Welcome

kumar----1234 1 Record Inserted Successfully

11. The hashed password must be stored in the table 'dbusers' now.

    Display the table using '**select * from dbusers;**'.



```
SQL> set linesize 200
SQL> select * from dbusers;

USERNAME             PASSWORD             EPWD
-------------------- -------------------- --------------------------------------------------------------------------------
--------------------------------------------------
kumar                1234                 81dc9bdb52d04dc20036dbd8313ed055
```

Post-Lab:

Q1. You are a database security consultant for a company. The company has given you the task of creating a registration page which takes username, password and gender as inputs and stores the hashed value of the password, gender in a database table. You plan to store the hash values of the password, gender in the database. The hashing is done using MD5. You are to implement this using Javascript.

First, create a html page with the username, password and gender fields. Once the details are given as input and submit button is pressed, the javascript program in the background will hash the password, gender and store it in the database along with the username and original value of gender.

Create a table called 'dbusers3' to store the credentials.

Sol)

1. Connect as system to sqlplus and create the 'dbusers3' table.
   create table dbusers3(username varchar2(20), password varchar2(20), epwd varchar2(50), gender varchar2(10), egender varchar2(50));
2. Switch on the Eclipse IDE.
   Go to 'File' → New → Other → Scroll down to 'Web' → Expand the folder and choose 'Dynamic Web Project'.
3. Set project name as something of your choice.
4. Choose 'Dynamic web module version' as 2.5.
5. Click Finish.
6. Then expand the project file in 'Project Explorer'. Expand 'WebContent' → WEB-INF → Right click on 'lib' → Build Path… → Configure Build Path… → Click on 'Classpath' → Click 'Add External JARs…' → Navigate to the 'ojdbc14.jar' and add it.
7. Then right click on 'WebContent'. This is where you make your html and jsp files.
   Right click on 'WebContent' → New → HTML File.
   Give a filename and end it with '.html' extension. Press Finish.
   register.html

   --

```
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="rlogin.jsp" method="post">

Username <input type="text" name="uname">

<br>
Password<input type="password" name="pwd">

<br>
Gender<input type="text" name="gender">

<br>
<input type="submit">



</form>
</body>
</html>
```

Save the file.

8. Right click on ‘WebContent’ → New → JSP File.

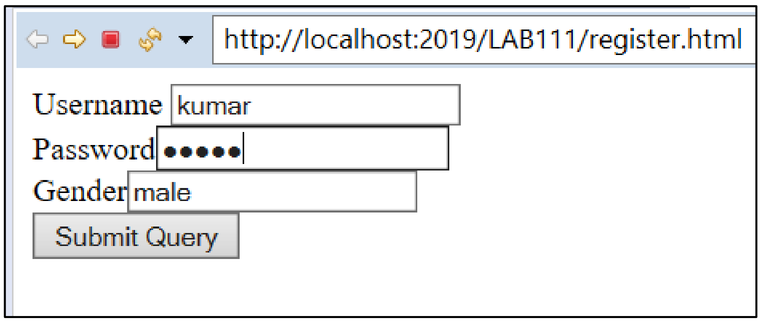
Give a filename and end it with ‘.jsp’ extension. Press Finish.

rlogin.jsp

```
<%@

page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>

<%@ page  import="java.sql.*" %>
<%@page import=" java.security.MessageDigest"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>  Welcome    </h1>

<%


String u=request.getParameter("uname");

String p=request.getParameter("pwd");
String g=request.getParameter("gender");

String algorithm="";
byte[] unencodedPassword = p.getBytes();
MessageDigest md = null;
try {
md = MessageDigest.getInstance("MD5");
} catch (Exception e) {}
md.reset();
md.update(unencodedPassword);
byte[] encodedPassword = md.digest();
StringBuffer buf = new StringBuffer();
for (int i = 0; i < encodedPassword.length; i++) {
if (((int) encodedPassword[i] & 0xff) < 0x10) {
buf.append("0");
}
buf.append(Long.toString((int) encodedPassword[i] & 0xff, 16));
}
String passw=buf.toString();

String algorithm1="";
byte[] unencodedPassword1 = g.getBytes();
MessageDigest md1 = null;
try {
md1 = MessageDigest.getInstance("MD5");
} catch (Exception e) {}
md1.reset();
md1.update(unencodedPassword1);
byte[] encodedPassword1 = md1.digest();
StringBuffer buf1 = new StringBuffer();
for (int i = 0; i < encodedPassword1.length; i++) {
if (((int) encodedPassword1[i] & 0xff) < 0x10) {
buf1.append("0");
}
buf1.append(Long.toString((int) encodedPassword1[i] & 0xff, 16));
}
String gend=buf1.toString();


out.println(u +"----"+p+"----"+g);
try
{

        Class.forName("oracle.jdbc.driver.OracleDriver");


        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","system");

        PreparedStatement ps=con.prepareStatement("insert into dbusers3 values(?,?,?,?,?)");
        ps.setString(1,u);
        ps.setString(2,p);
        ps.setString(3,passw);
        ps.setString(4,g);
        ps.setString(5, gend);

int a=ps.executeUpdate();

out.println(a+ "  Record Inserted Successfully");
}
catch(Exception e)
{
        out.println(e);

}

%>

</body>
</html>
```

Save the file.

9. Right click on the html file and press 'Run As' → 1 Run on Server → (The Tomcat server should already be selected) → Press Finish.



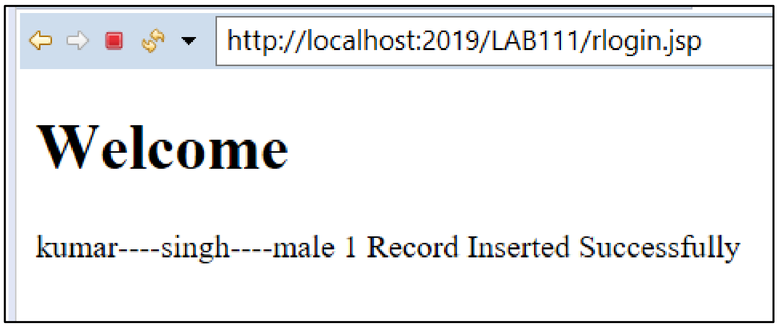
http://localhost:2019/LAB111/register.html

Username	kumar
Password	●●●●●
Gender	male

Submit Query

10. Give your username and password.

    The inputs I am giving are 'kumar' as username, 'singh' as password and 'male' as gender.



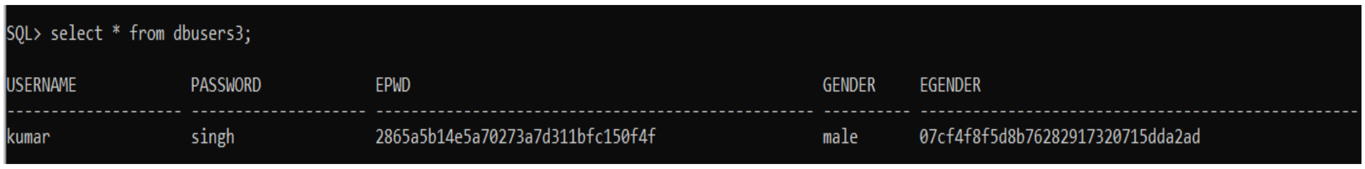
http://localhost:2019/LAB111/rlogin.jsp

Welcome

kumar----singh----male 1 Record Inserted Successfully

11. The hashed password must be stored in the table 'dbusers3' now.

    Display the table using '**select * from dbusers3;**'.



```
SQL> select * from dbusers3;

USERNAME             PASSWORD             EPWD                                               GENDER     EGENDER
-------------------- -------------------- -------------------------------------------------- ---------- --------------------------------------------------
kumar                singh                2865a5b14e5a70273a7d311bfc150f4f                   male       07cf4f8f5d8b76282917320715dda2ad
```