

In-Lab:

Q1. You are a database security consultant for a company. The company has given you the task of creating a login page which takes username and password as inputs and hashes the password and compares it with the previously stored hash value (from previous lab) in the database table and validates it. 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 compare it with the previously stored value in the database.

Sol)

1. Switch on the Eclipse IDE.
Go to 'File' → New → Other → Scroll down to 'Web' → Expand the folder and choose 'Dynamic Web Project'.
2. Set project name as something of your choice.
3. Choose 'Dynamic web module version' as 2.5.
4. Click Finish.
5. 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.
6. 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.

login.html

```
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
```

```
<form action="rverify.jsp" method="post">  
  
Username <input type="text" name="uname">  
  
<br>  
Password<input type="password" name="pwd">  
  
<br>  
<input type="submit">  
  
</form>  
</body>  
</html>
```

Save the file.

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

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

rverify.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 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();

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

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

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

    PreparedStatement ps=con.prepareStatement("select * from dbusers where
username=?");
    ps.setString(1,u);

    ResultSet rs=ps.executeQuery();

    rs.next();

    String dbpwd=rs.getString(3);

    if(dbpwd.equals(passw))

out.println(" Valid Credentials");
    else
        out.println(" INVALID PWD");
} catch (Exception e)
{
    out.println(e);
}
```

```
}
```

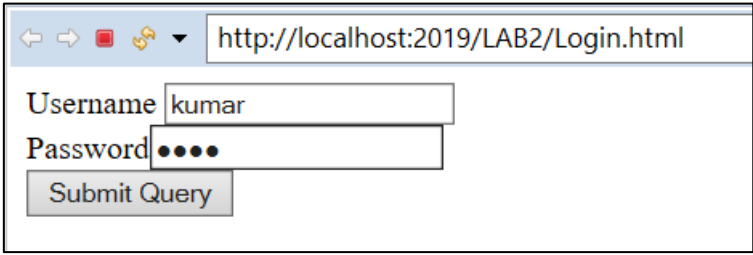
```
%>
```

```
</body>
```

```
</html>
```

Save the file.

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



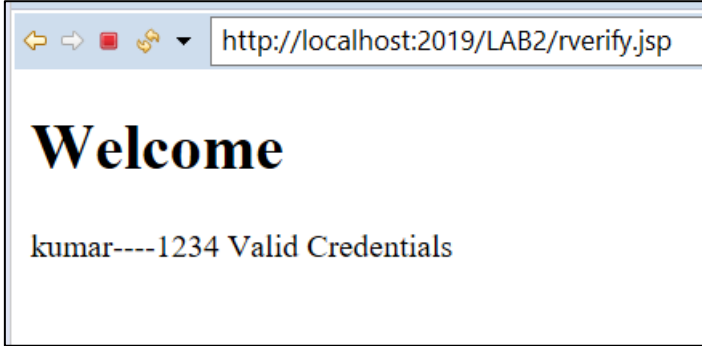
http://localhost:2019/LAB2/Login.html

Username

Password

9. Give your username and password.

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



http://localhost:2019/LAB2/rverify.jsp

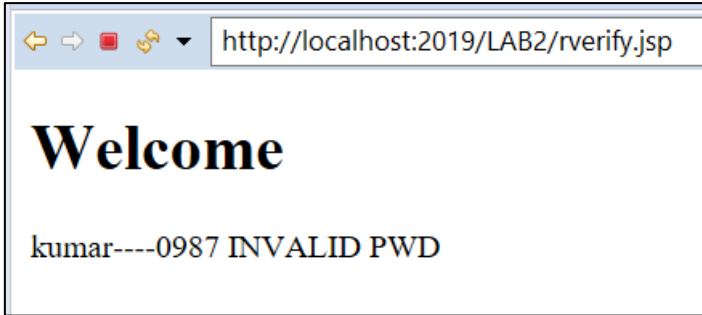
Welcome

kumar----1234 Valid Credentials

10. The given credentials are valid.

11. Lets try giving the wrong credentials.

The inputs I am giving are 'kumar' as username and '0987' as password.



http://localhost:2019/LAB2/rverify.jsp

Welcome

kumar----0987 INVALID PWD

Post-Lab:

Q1. Toggle Between Roles

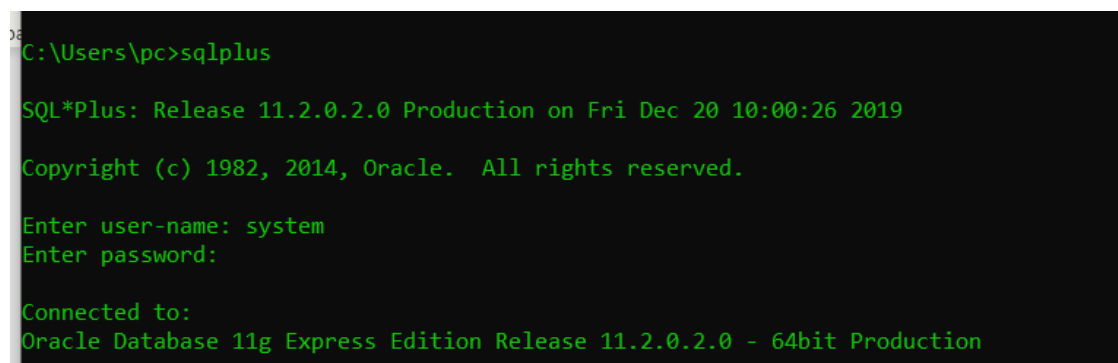
You can see the effects of running SET ROLE by observing your current session roles as you toggle back and forth between create_session_role and secadm_role. As secadm user, do this and observe the list of roles when you do each SELECT query:

- i) SELECT * FROM sys.session_roles;
- ii) SET ROLE create_session_role;
- iii) SELECT * FROM sys.session_roles;
- iv) EXECUTE sys.p_check_secadm_access;
- v) SELECT * FROM sys.session_roles;

Sol)

Step1: open commandPrompt and type sqlplus

Step2: Connect to oracle 11g with username as system and password that is given during installation.



```
C:\Users\pc>sqlplus

SQL*Plus: Release 11.2.0.2.0 Production on Fri Dec 20 10:00:26 2019

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Enter user-name: system
Enter password:

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
```

Step3:

- i) SELECT * FROM sys.session_roles;

```
SQL> SELECT * FROM sys.session_roles;

ROLE
-----
DBA
SELECT_CATALOG_ROLE
HS_ADMIN_SELECT_ROLE
EXECUTE_CATALOG_ROLE
HS_ADMIN_EXECUTE_ROLE
DELETE_CATALOG_ROLE
EXP_FULL_DATABASE
IMP_FULL_DATABASE
DATAPUMP_EXP_FULL_DATABASE
DATAPUMP_IMP_FULL_DATABASE
GATHER_SYSTEM_STATISTICS

ROLE
-----
SCHEDULER_ADMIN
PLUSTRACE
XDBADMIN
XDB_SET_INVOKER
AQ_ADMINISTRATOR_ROLE

16 rows selected.
```

ii) First create the role and then set role

a) CREATE ROLE create_session_role;

```
SQL> CREATE ROLE create_session_role;

Role created.
```

b) SET ROLE create_session_role;

```
SQL> SET ROLE create_session_role;

Role set.
```

iii) `SELECT * FROM sys.session_roles;`

```
SQL> SELECT * FROM sys.session_roles;

ROLE
-----
CREATE_SESSION_ROLE
```

iv) First create a procedure `sys.p_check_secadm_access` and then execute it.

a) Create a procedure `sys.p_check_secadm_access`

```
Command Prompt - sqlplus

SQL> CREATE OR REPLACE PROCEDURE sys.p_check_secadm_access
2 AUTHID CURRENT_USER
3 AS
4 BEGIN
5 IF( SYS_CONTEXT( 'USERENV', 'IP_ADDRESS' ) = '127.0.0.1' )
6 THEN
7 EXECUTE IMMEDIATE 'SET ROLE secadm_role';
8 END IF;
9 END;
10 /

Procedure created.
```

b) `EXECUTE sys.p_check_secadm_access;`

```
SQL> EXECUTE sys.p_check_secadm_access;

PL/SQL procedure successfully completed.
```

v) `SELECT * FROM sys.session_roles;`


```
SQL> SELECT * FROM sys.session_roles;  
  
no rows selected
```