

Assignment.5

Implement the sample program demonstrating the use of Servlet. e.g., Create a database table ebookshop (book_id, book_title, book_author, book_price, quantity) using database like Oracle/MySQL etc. and display (use SQL select query) the table content using servlet.

CODE:

```
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("/ShowBookInfo")
public class ShowBookInfo extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
    {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        out.print("<center><h2>Book Information</h2></center>");
        try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/ebooks","root","prajwal");
            Statement st=con.createStatement();
            ResultSet rs=st.executeQuery("select * from bookinfo");
            out.print("<center><table border=\"1\">");

            out.print("<tr><th>BookId</th><th>BookName</th><th>Author</th><th>Prize</th
><th>Quantity</th></tr>");
            while(rs.next())
            {
                out.print("<tr>");
                out.print("<td>" +rs.getString(1)+"</td>");
                out.print("<td>" +rs.getString(2)+"</td>");
                out.print("<td>" +rs.getString(3)+"</td>");
```

```

        out.print("<td>" + rs.getInt(4) + "</td>");
        out.print("<td>" + rs.getInt(5) + "</td>");
        out.print("</tr>");
    }
    out.print("</table></center>");
}
catch(Exception e)
{
    out.print("<center><h2>Unable to connect to
database</h2></center>");
}
}

```

MYSQL

```
mysql> CREATE TABLE bookinfo(id INT NOT NULL ,title VARCHAR(30) NOT NULL,author
VARCHAR(30) NOT NULL, price INT NOT NULL, quantity INT NOT NULL ,PRIMARY KEY(id));
```

```
Query OK, 0 rows affected (0.14 sec)
```

```
mysql> desc bookinfo
```

```
-> ;
+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| id   | int    | NO  | PRI | NULL   |      |
| title | varchar(30) | NO |     | NULL   |      |
| author | varchar(30) | NO |     | NULL   |      |
| price | int    | NO  |     | NULL   |      |
| quantity | int    | NO  |     | NULL   |      |
+-----+-----+-----+-----+
```

```
5 rows in set (0.03 sec)
```

```
mysql> insert into bookinfo values (101,"cpp program","xyz",345,2);
```

```
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from bookinfo;
```

id	title	author	price	quantity
101	cpp program	xyz	345	2

```
1 row in set (0.00 sec)
```

```
mysql> insert into bookinfo values (102,"python program","abc",555,5);
```

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into bookinfo values (103,"java program","pqr",225,1);
```

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from bookinfo;
```

id	title	author	price	quantity
101	cpp program	xyz	345	2
102	python program	abc	555	5
103	java program	pqr	225	1

```
3 rows in set (0.00 sec)
```

OUTPUT:

localhost:8080/ShowBookInfo

HackerRank General (DYPCOE C... Courses IN-ACE Track-D Y P... Practice | Geeksfor... Trailhead | The fun...

Book Information

BookId	BookName	Author	Prize	Quantity
101	cpp program	xyz	345	2
102	python program	abc	555	5
103	java program	pqr	225	1

Assignment.6

Implement the program demonstrating the use of JSP. e.g., Create a database table students_info (stud_id, stud_name, class, division, city) using database like Oracle/MySQL etc. and display (use SQL select query) the table content using JSP.

CODE:

showinfo.jsp

```
<%@pageimport="java.sql.*"language="java"contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPEhtml>
<html>
<head>
<metacharset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
out.print("<h2>Student Information Table</h2>");
try
{
    Class.forName("com.mysql.cj.jdbc.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/student","root","prajwal");
    Statement st=con.createStatement();
    ResultSet rs=st.executeQuery("select * from studentinfo");
    out.print("<center><table border=\"1\">");
    out.print("<tr><th>Id</th><th>Student
Name</th><th>Class</th><th>Division</th><th>City</th></tr>");

```

```

while(rs.next())
{
    out.print("<tr>");
    out.print("<td>" + rs.getString(1) + "</td>");
    out.print("<td>" + rs.getString(2) + "</td>");
    out.print("<td>" + rs.getString(3) + "</td>");
    out.print("<td>" + rs.getString(3) + "</td>");
    out.print("<td>" + rs.getString(5) + "</td>");
    out.print("</tr>");
}
out.print("</table></center>");
}

catch(Exception e)
{
    e.printStackTrace();
}

%>

</body>
</html>

```

Sql table

```

Database changed
mysql> select * From studentinfo;
+----+-----+-----+-----+-----+
| id | name | class | division | city |
+----+-----+-----+-----+-----+
| TECOMP-A99 | Ajinkya Raje | TECOMP | A | Pune |
| TECOMP-A100 | Mansi Jagtap | TECOMP | A | Ravet |
+----+-----+-----+-----+-----+
1 rows in set (0.00 sec)

mysql> []

```

OUTPUT:

Student Information Table

Id	Student Name	Class	Division	City
TECOMP-A99	Ajinkya Raje	TECOMP	TECOMP	Pune
TECOMP-A100	Mansi Jagtap	TECOMP	TECOMP	Ravet

Assignment.7

Build a dynamic web application using PHP and MySQL. a. Create database tables in MySQL and create connection with PHP. b. Create the add, update, delete and retrieve functions in the PHP web app interacting with MySQL database

CODE:

Index.html

```
<html>
<head>
<link rel = "stylesheet" type = "text/css" href = "style.css"/>
</head>
<style>
input[type="submit"]
{
    background-color: #4CAF50;
    color: white;
    height: 42px;
    margin: 10px 0 0 36px;
    width: 250px;
    padding-left: 10px;
}
</style>
<body>
<form action="deletedata.php">
    <input type="submit" value="Delete">
</form>
<form action="updatedata.php">
    <input type="submit" value="Update">
</form>
<form action="showdata.php">
    <input type="submit" value="Show">
</form>
<form action="insertdata.php">
    <input type="submit" value="Insert">
</form>
</body>
</html>
```

showdata.php

```
<?php
```

```
$db_name = 'student';  
$db_user = 'root';  
$db_pass = " ";  
$db_host = 'localhost';  
  
$conn = mysqli_connect($db_host, $db_user, $db_pass,$db_name);  
if (!$conn )  
{  
    $result= $conn->connect_error;  
}  
else  
{  
    $sql = "SELECT * FROM studentinfo";  
    $result = $conn->query($sql);  
    if ($result->num_rows >0)  
    {  
        echo'<table border="1">';  
        echo  
"<tr><th>Name</th><th>Roll</th><th>Class</th><th>Address</th></tr>";  
        while($row = $result->fetch_assoc())  
        {  
            $name = $row['name'];  
            $roll = $row['roll'];  
            $class = $row['class'];  
            $address = $row['address'];
```

```
        echo "<tr><td>".$name
."</td><td>".$roll."</td><td>".$class."</td><td>".$address."</td></tr>";
    }
echo"</table>";
}

$conn->close();
}
```

```
?>
<html>
<head>
</head>
<body>
<h2><a href="index.html">Click Here !!! fro Home Page</a?</h2>
</body>
</html>
```

insertdata.php

```
<?php

if($_SERVER['REQUEST_METHOD'] == "POST")
{
    $name = $_POST['name'];
    $roll = $_POST['roll'];
    $class =$_POST['class'];
    $address=$_POST['address'];

    $db_name = 'student';
    $db_user = 'root';
    $db_pass = "";
    $db_host = 'localhost';
```

```
$conn = mysqli_connect($db_host, $db_user, $db_pass,$db_name);

if (!$conn )
{
    $result= $conn->connect_error;

}

else
{

    $sql = "insert into studentinfo values('$name','$roll','$class','$address')";



if ($conn->query($sql) === TRUE)
{
    echo"<center><font color=\"red\">SData Inserted
Successfully</font></center>";

}
else
{
    echo "Error: ". $sql . "<br>". $conn->error;
    echo"<center><font color=\"red\">Problem in Inserting
data</font></center>";
}

$conn->close();
}

?>
<html>
```

```
<head>
<style>

input[type="password"], input[type="text"]
{
    border:1px solid;
    background-color:rgba(0,0,0,0);
    height: 33px;
    width:240px;
    margin: 2px 0 0 36px;
    padding-left: 10px;

}

input[type="submit"]
{
    background-color: #4CAF50;
    color: white;
    height: 42px;
    margin: 10px 0 0 36px;
    width: 250px;
    padding-left: 10px;

}
</style>
</head>
<body>

<form action=""name="myform"method="post">
<label><b>StudentName:</b></label>
<br><input type="text"name="name"id="name"placeholder=
"Username"required="required"onfocus="this.value = ";">
```

```
<br><label><b>Roll No:</b></label>
<br><input type="text" name="roll" id="roll" placeholder= "Roll
No" required="required" onfocus="this.value = ";">
<br><label><b>Class:</b></label>
<br><input type="text" name="class" id="class" placeholder=
"Class" required="required" onfocus="this.value = ";">
<br><label><b>Address:</b></label>
<br><input type="text" name="address" id="address" placeholder=
"Address" required="required" onfocus="this.value = ";">
```

```
<br><input type="submit" value="Insert Data">
<h2><a href="index.html">Click Here !!! fro Home Page</a?</h2>
</body>
</html>
```

deletedata.php

```
<?php
```

```
if($_SERVER['REQUEST_METHOD'] == "POST")
{
```

```
$roll = $_POST['roll'];
```

```
$db_name = 'student';
$db_user = 'root';
$db_pass = "";
$db_host = 'localhost';
```

```
$conn = mysqli_connect($db_host, $db_user, $db_pass,$db_name);
if (!$conn )
```

```
{  
    $result= $conn->connect_error;  
}  
else  
{  
  
    $sql = "delete from studentinfo where roll='$roll"';  
  
if ($conn->query($sql) === TRUE)  
{  
    echo"<center><font color=\"red\">SData Deleted  
Successfully</font></center>";  
}  
  
else  
{  
    echo "Error: ". $sql . "<br>". $conn->error;  
    echo"<center><font color=\"red\">Problem in Updating  
data</font></center>";  
}  
  
}  
$conn->close();  
}  
  
?>  
<html>  
  
<head>  
<style>  
input[type="password"], input[type="text"]  
{
```

```
border:1px solid;
background-color:rgba(0,0,0,0.0);
height: 33px;
width:240px;
margin: 2px 0 0 36px;
padding-left: 10px;

}

input[type="submit"]
{
background-color: #4CAF50;
color: white;
height: 42px;
margin: 10px 0 0 36px;
width: 250px;
padding-left: 10px;
}

</style>
</head>
<body>

<form action=<?php echo $_SERVER['PHP_SELF'];?>"name="myform"method="post">

<br><label><b>Roll No:</b></label>
<br><input type="text" name="roll" id="roll"placeholder= "Roll
No"required="required"onfocus="this.value = ";">
```

```
<br><input type="submit" value="Delete Record">  
<h2><a href="index.html">Click Here !!! fro Home Page</a?</h2>  
</body>  
</html>
```

updatedata.php

```
<?php  
  
if($_SERVER['REQUEST_METHOD'] == "POST")  
{  
    $name = $_POST['name'];  
    $roll = $_POST['roll'];  
    $class =$_POST['class'];  
    $address=$_POST['address'];  
  
    $db_name = 'student';  
    $db_user = 'root';  
    $db_pass = '';  
    $db_host = 'localhost';  
  
    $conn = mysqli_connect($db_host, $db_user, $db_pass,$db_name);  
    if (!$conn )  
    {  
        $result= $conn->connect_error;  
    }  
    else  
    {  
  
        $sql = "update studentinfo set  
name='$name',class='$class',address='$address'where roll='$roll"';
```

```
if ($conn->query($sql) === TRUE)
{
    echo "<center><font color=\"red\">SData Updated
Successfully</font></center>";
}

else
{
    echo "Error: ". $sql . "<br>". $conn->error;
    echo "<center><font color=\"red\">Problem in Updating
data</font></center>";
}

$conn->close();
}

?>

<html>

<head>
<style>
input[type="password"], input[type="text"]
{
    border:1px solid;
    background-color:rgba(0,0,0,0.0);
    height: 33px;
    width:240px;
    margin: 2px 0 0 36px;
    padding-left: 10px;
}
```

```

}

input[type="submit"]
{
background-color: #4CAF50;
color: white;
height: 42px;
margin: 10px 0 0 36px;
width: 250px;
padding-left: 10px;
}

</style>
</head>
<body>

<form action=""name="myform"method="post">
<label><b>StudentName:</b></label>
<br><input type="text" name="name" id="name" placeholder="Username" required="required" onfocus="this.value = '';">
<br><label><b>Roll No:</b></label>
<br><input type="text" name="roll" id="roll" placeholder="Roll No" required="required" onfocus="this.value = '';">
<br><label><b>Class:</b></label>
<br><input type="text" name="class" id="class" placeholder="Class" required="required" onfocus="this.value = '';">
<br><label><b>Address:</b></label>
<br><input type="text" name="address" id="address" placeholder="Address" required="required" onfocus="this.value = '';">
<br><input type="submit" value="Insert Data">

```

```

<h2><a href="index.html">Click Here !!! fro Home Page</a?</h2>
</body>
</html>

```

OUTPUT:

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions
	Apache	23508 10572	80, 443	Stop Admin Config Logs Shell
	MySQL	21840	3306	Stop Admin Config Logs Explorer Services Help
	FileZilla			Start Admin Config Logs
	Mercury			Start Admin Config Logs
	Tomcat			Start Admin Config Logs

```

28.4.1.01 [main] Initializing Control Panel
28.4.1.01 [main] Windows Version: Home 64-bit
28.4.1.02 [main] XAMPP Version: 8.1.4
28.4.1.02 [main] Control Panel Version: 3.3.0 [ Compiled: Apr 6th 2021 ]
28.4.1.02 [main] You are not running with administrator rights! This will work for
28.4.1.02 [main] most application stuff but whenever you do something with services
28.4.1.02 [main] there will be a security dialogue or things will break! So think
28.4.1.02 [main] about running this application with administrator rights!
28.4.1.02 [main] XAMPP Installation Directory: "c:\xampp\"
28.4.1.02 [main] Checking for prerequisites
28.4.1.02 [main] All prerequisites found
28.4.1.02 [main] Initializing Modules
28.4.1.02 [main] Starting Check Timer
28.4.1.02 [main] Control Panel Ready
28.4.1.04 [Apache] Attempting to start Apache app...
28.4.1.04 [Apache] Status change detected: running
28.4.1.04 [mysql] Attempting to start MySQL app...

```

localhost/phpmyadmin/index.php?route=/sql&db=student...

phpMyAdmin

Showing rows 0 - 1 (2 total). Query took 0.0005 seconds.

```

SELECT * FROM `studentinfo`

```

Options

name	roll	class	address
prajwal	comp1102	TE	pune
yash	comp1101	TE	shirdi

(no subject) - prajsa99@gmail.com

localhost/assignment_07/index.html

localhost / 127.0.0.1 / student / s

Delete

Update

Show

Insert

A screenshot of a web browser window. The address bar shows the URL `localhost/assignment_07/showdata.php?`. The browser has multiple tabs open, including Gmail, YouTube, HackerRank, General (DYP COE C...), Courses, IN-ACE Track-D Y P..., Practice | Geeksfor..., and Trailhead | The fun... . The main content area displays a table with four columns: Name, Roll, Class, and Address. The data in the table is:

Name	Roll	Class	Address
prajwal	comp1102	TE	pune
yash	comp1101	TE	shirdi

[Click Here !!! fro Home Page](#)

StudentName:

Roll No:

Class:

Address:

[Click Here !!! fro Home Page](#)

Name	Roll	Class	Address
prajwal	comp1102	TE	pune
yash	comp1101	TE	shirdi
ashish	comp1103	TE	nashik

[**Click Here !!! fro Home Page**](#)

Assignment.8

Design a login page with entries for name, mobile number email id and login button. Use struts and perform following validations

- a. Validation for correct names**
- b. Validation for mobile numbers**
- c. Validation for email id**
- d. Validation if no entered any value**
- e. Re-display for wrongly entered values with message**
- f. Congratulations and welcome page upon successful entries**

CODE:

web.xml

```
<?xmlversion="1.0"encoding="UTF-8"?>
<web-appxmlns: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>AAStrutsApplication</display-name>
    <welcome-file-list>
        <welcome-file>Registration.jsp</welcome-file>
    </welcome-file-list>
    <filter>
        <filter-name>Struts2</filter-name>
        <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-
class>
    </filter>

    <filter-mapping>
        <filter-name>Struts2</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
</web-app>
```

struts.xml

```
<?xmlversion="1.0"encoding="UTF-8"?>
<!DOCTYPEstrutsPUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">

    <struts>
        <includefile="struts-default.xml"></include>
            <packagename="default"namespace="/"extends="struts-default">
                <actionname="register"class="nilesh.korade.RegistrationAction">
```

```

<resultname="success">/Welcome.jsp</result>
<resultname="input">/Registration.jsp</result>

</action>
</package>
```

RegistrationAction.java

```

packagenilesh.korade;
importcom.opensymphony.xwork2.ActionSupport;
publicclassRegistrationActionextendsActionSupport{
    String emailid;
    String name;
    longcontactno;
    publicString execute() throwsException
    {
        returnSUCCESS;
    }
    publicString getEmailid() {
        returnemailid;
    }
    publicvoidsetEmailid(String emailid) {
        this.emailid= emailid;
    }
    publicString getName() {
        returnname;
    }
    publicvoidsetName(String name) {
        this.name= name;
    }
    publiclonggetContactno() {
        returncontactno;
    }
    publicvoidsetContactno(longcontactno) {
        this.contactno= contactno;
    }
}
```

RegistrationAction-validation.xml

```

<!DOCTYPEValidatorsPUBLIC"-//Apache Struts//XWork Validator 1.0.3//EN"
 "http://struts.apache.org/dtds/xwork-validator-1.0.3.dtd">
<validators>
    <fieldname="name">
        <field-validatortype="requiredstring">
            <message>You must enter Your Name</message>
        </field-validator>
    </field>
    <fieldname="emailid">
        <field-validatortype="requiredstring">
            <message>You must enter Your Emil-ID</message>
```

```

        </field-validator>
    </field>

    <fieldname="contactno">
        <field-validatortype="regex">
            <paramname="regex"><![CDATA[\d{10}]]></param>
                <message>Phone number must be entered as 9999999999.</message>
            </field-validator>
        </field>
    </validators>

```

Registration.jsp

```

<%@page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@taglib prefix="s" uri="/struts-tags"%>
<!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>Registration</title>
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>
    <s:form action="register" method="post">

        <s:textfield label="Username " name="name"/>
        <s:textfield label="Email-ID " name="emailid"/>
        <s:textfield label="Contact No " name="contactno"/>
        <s:submit value="Submit"/>
    </s:form>
</div>
</body>
</html>

```

Welcome.jsp

```

<%@page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@taglib prefix="s" uri="/struts-tags"%>
<!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=UTF-8">
<title>Insert title here</title>
</head>
<body>
    <h1>Welcome</h1>
    Hello: <s:property value="name"/>
    Your Email-ID: <s:property value="emailid"/>
    Your Contact No: <s:property value="contactno"/>
</body>
</html>

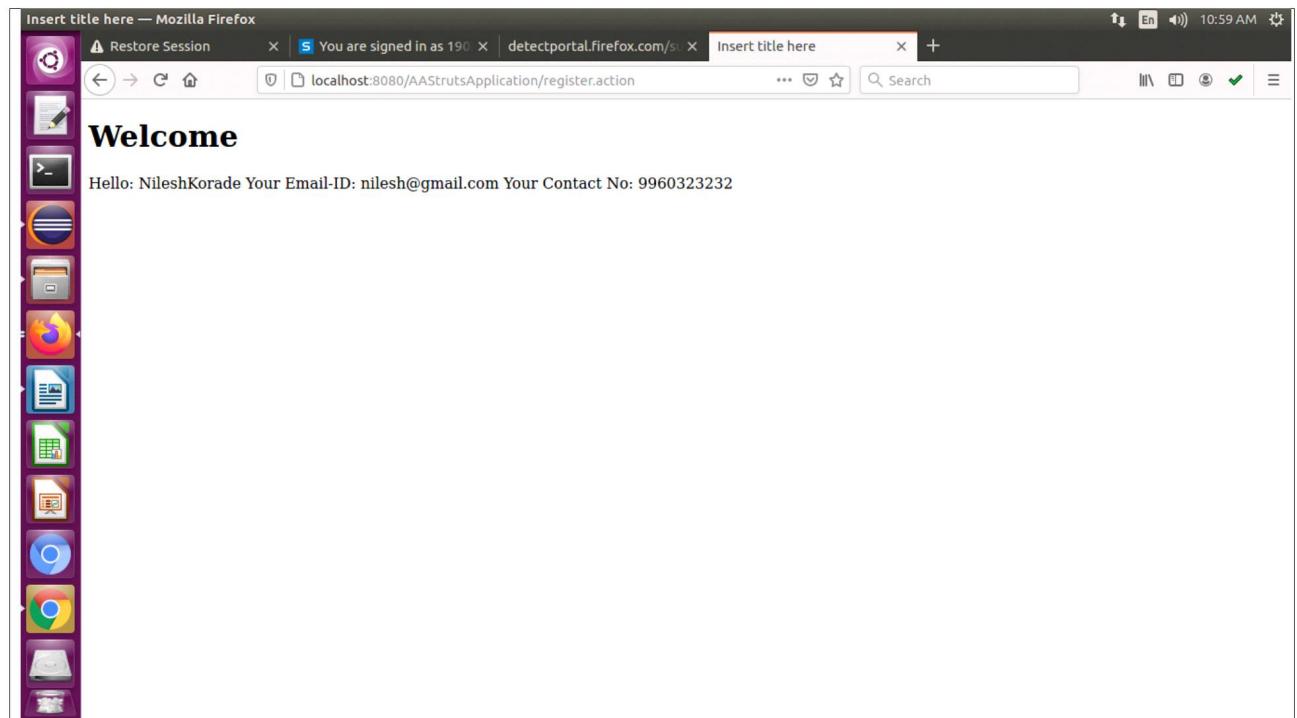
```

OutPut:

The screenshot displays two windows side-by-side. The left window is the Eclipse IDE interface, specifically the Project Explorer and Editor panes. The Project Explorer shows a Java EE project structure with various packages like AAStrutsApplication, JAX-WS Web Services, Java Resources, and WebContent. The Editor pane shows the file `RegistrationAction-validation.xml` containing Struts validation code. The right window is a Mozilla Firefox browser window titled "Registration". It shows a simple HTML form with four fields: "Username", "Email-ID", "Contact No.", and a "Submit" button. Each field has an associated validation message: "You must enter Your Name" for the username, "You must enter Your Emil-ID" for the email, and "Contact No." for the contact number.

```
<!DOCTYPE validators PUBLIC "-//Apache Struts//XWork Validator 1.0.3//EN"
"http://struts.apache.org/dtds/xwork-validator-1.0.3.dtd">
<validators>
    <field name="name">
        <field-validator type="requiredstring">
            <message>You must enter Your Name</message>
        </field-validator>
    </field>
    <field name="emailid">
        <field-validator type="requiredstring">
            <message>You must enter Your Emil-ID</message>
        </field-validator>
    </field>
    <field name="contactno">
        <field-validator type="regex">
            <param name="regex"><![CDATA[\d{10}]]></param>
        </field-validator>
    </field>

```



Assignment.9

Design an application using Angular JS.

CODE:

Userhome.html

```
<!DOCTYPE html>

<html>

<style>

table, th , td

{

border: 1px solid grey;

border-collapse: collapse;

padding: 5px;

}

table tr:nth-child(odd)

{

background-color: #f1f1f1;

}

table tr:nth-child(even)

{

background-color: #ffffff;

}

</style>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min.js"></script>

<body>

<div ng-app="myApp" ng-controller="courseCtrl">

<center><h3><font color="red"> {{welcome}}</font></h3>

<table>

<tr>

<th>Qualification</th>

<th>Course</th>
```

```

<th>Duration</th>
</tr>

<tr ng-repeat="x in names">
<td>{{ x.qual }}</td>
<td>{{ x.course }}</td>
<td>{{ x.duration }}</td>
</tr>
</table>

</div>
</center>
<script>
var app = angular.module('myApp', []);
app.controller('courseCtrl', function($scope, $http)
{
    $scope.welcome = "Welcome in Course Counceling Website";
    $http.get("course.php")
        .then(function (response)
        {
            $scope.names = response.data.records;
        });
</script>
</body>
</html>

```

Course.php

```

{ "records": [
    {"qual":"SSC","course":"MSCIT","duration":"3 months"},

    {"qual":"SSC","course":"DTP","duration":"3 months"},

    {"qual":"HSC","course":"HTML","duration":"1 months"},

    {"qual":"HSC","course":"C Programming","duration":"3 months"},

    {"qual":"Diploma","course":"OCJP","duration":"3 months"},

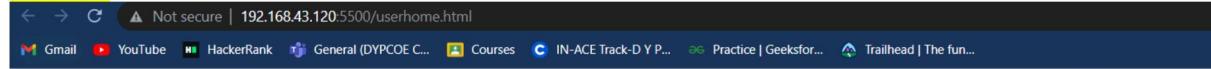
    {"qual":"Diploma","course":"CCNA","duration":"3 months"},

]
}

```

```
{"qual":"BE","course":"Software Development","duration":"6 months"},  
{"qual":"BE","course":"Software Testing","duration":"6 months"}  
]}{
```

OutPut:



Welcome in Course Cunceling Website

Qualification	Course	Duration
SSC	MSCIT	3 months
SSC	DTP	3 months
HSC	HTML	1 months
HSC	C Programming	3 months
Diploma	OCJP	3 months
Diploma	CCNA	3 months
BE	Software Development	6 months
BE	Software Testing	6 months

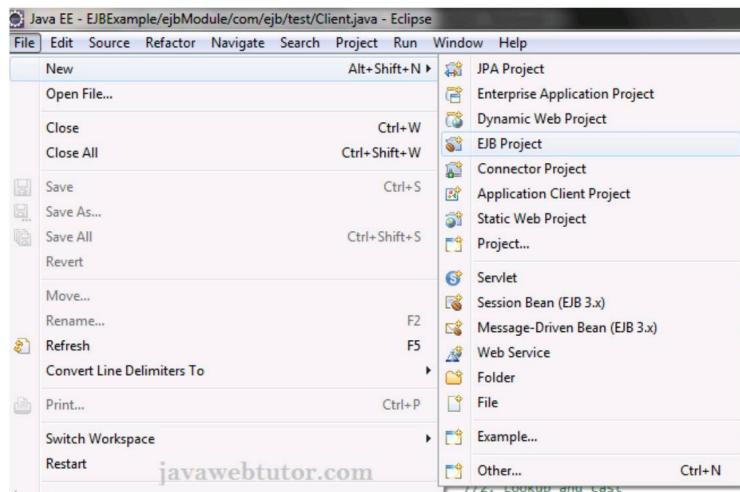
Assignment.10

Design an application using Angular JS.

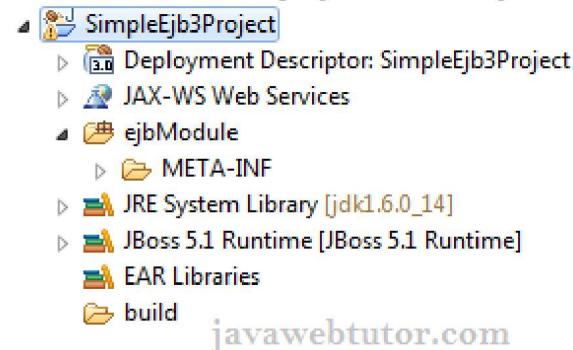
CODE:

Create a new EJB Project :

Open Eclipse IDE and create a new EJB project which can be done by clicking on, File menu -> New -> EJB Project. Name the Project as SimpleEjb3Project. Make sure that Target Runtime as JBoss 5.1 Runtime EJB Module version as 3.0 and click on finish.

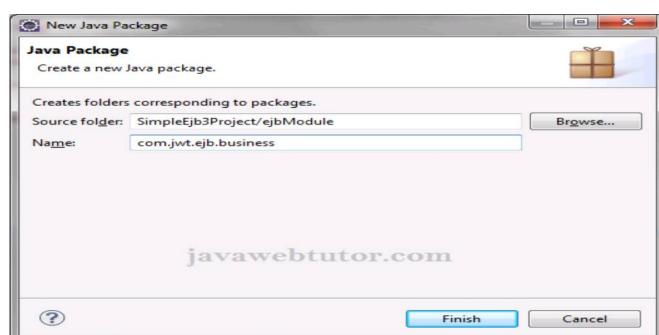


You will see an EJB project in the Project Explorer view as shown below.

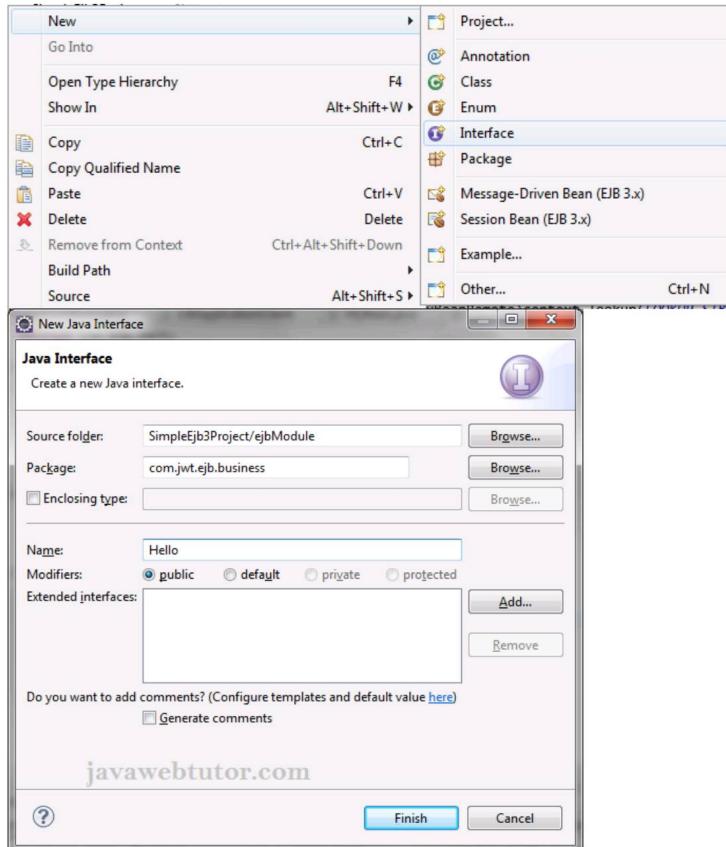


2.1 Creating Session Bean and Bean Interface :

Create a new package by Right click on ejbModule -> New -> Package and name this package to com.jwt.ejb.business and click on Finish.



Right click on this package and Create a new Interface and name it to Hello and click Finish



Interface can be either @Remote or @Local or without any annotation. In this example we have used @Remote .

Add following code into this Interface.

Hello.java

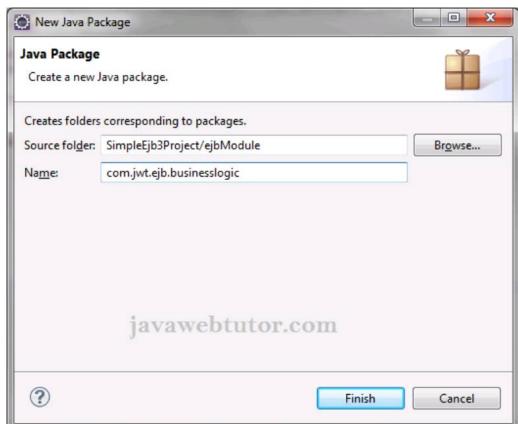
```
package com.jwt.ejb.business;

import javax.ejb.Remote;

@Remote
public interface Hello {

    public String sayHello();
}
```

Create a new package by Right click on ejbModule -> New -> Package and name this package to com.jwt.ejb.businesslogic and click on Finish.



Create a new Class HelloBean in this package which should implement Hello interface and add the following code.

This is a Bean class. Bean type can either be @Stateful or @Stateless . In this example we have used @Stateless .

Hello.java

```
package com.jwt.ejb.businesslogic;

import javax.ejb.Stateless;

import com.jwt.ejb.business.Hello;

@Stateless
public class HelloBean implements Hello {

    public HelloBean() {
    }

    public String sayHello() {
        return "Hello Boss Welcome to EJB";
    }
}
```

Creating Client :

The next step is to write a remote Java client application (with main()) for accessing and invoking the EJBs deployed on the server.

Client uses JNDI to lookup for a proxy of your bean and invokes method on that proxy.

Creating JNDI InitialContext :

All naming service operations are performed on implementation of javax.naming.Context interface. The starting point of interacting with the naming service is to obtain a Context by providing the properties specific to the server implementation being used. In our case it is, JBoss Application Server.

To create a javax.naming.InitialContext , we need to initialize it with properties from the environment. JNDI verifies each property's values from the following two sources,

- jndi.properties resource files found on the classpath.
- Using parametrized constructor of InitialContext which takes properties of supplied environment

We are going to use constructor for initializing the InitialContext.
For JBoss AS 5 we need to set following properties

```
Context.INITIAL_CONTEXT_FACTORY =
org.jnp.interfaces.NamingContextFactory<br>
Context.URL_PKG_PREFIXES = org.jboss.naming:org.jnp.interfaces<br>
Context.PROVIDER_URL = jnp://localhost:1099
```

Steps to create Client Class :

- Right click on ejbModule -> New -> Class
- Enter the class name as Client .
- Enter the package name as com.jwt.ejb.test and click on Finish.



Client.java

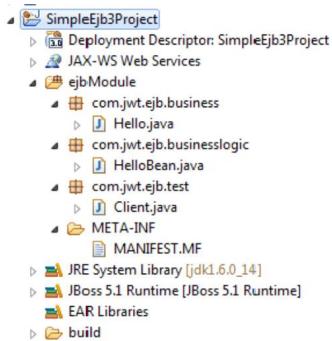
```
package com.jwt.ejb.test;
import java.util.Properties;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import com.jwt.ejb.business.Hello;
public class Client {
    public static void main(String[] args) {
        Hello bean = doLookup();
        System.out.println(bean.sayHello()); // 3. Call business logic
    }
    private static Hello doLookup() {
        Context context = null;
        Hello bean = null;
        try {
            // 1. Obtaining Context
            context = getInitialContext();
            // 2. Lookup and cast
            bean = (Hello) context.lookup(LOOKUP_STRING);
        } catch (NamingException e) {
            e.printStackTrace();
        }
        return bean;
    }
    private static final String LOOKUP_STRING = "HelloBean/remote";
```

```

private static final String PROVIDER_URL = "jnp://localhost:1099";
private static final String JNP_INTERFACES =
"org.jboss.naming:org.jnp.interfaces";
private static final String INITIAL_CONTEXT_FACTORY
= "org.jnp.interfaces.NamingContextFactory";
private static Context initialContext;
public static Context getInitialContext() throws NamingException {
    if (initialContext == null) {
        // Properties extends HashTable
        Properties prop = new Properties();
        prop.put(Context.INITIAL_CONTEXT_FACTORY,
INITIAL_CONTEXT_FACTORY);
        prop.put(Context.URL_PKG_PREFIXES, JNP_INTERFACES);
        prop.put(Context.PROVIDER_URL, PROVIDER_URL);
        initialContext = new InitialContext(prop);
    }
    return initialContext;
}
}

```

Directory Structure of Project

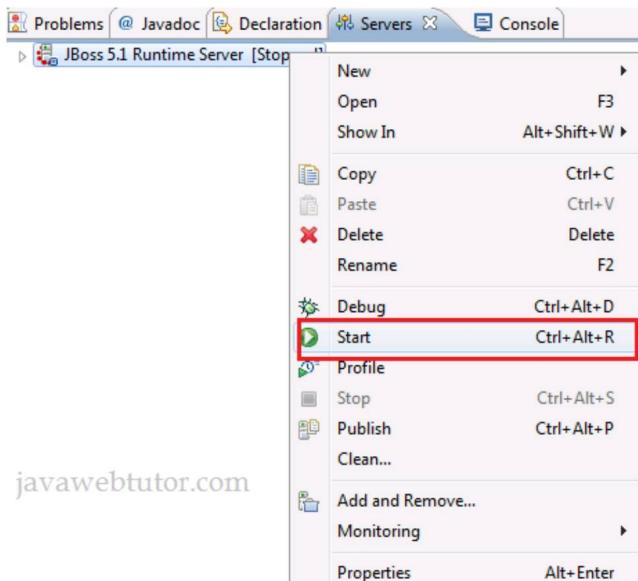


Deploying the project :

Now we need to deploy the our EJB "SimpleEjb3Project" on server.Follow the steps mentioned bellow to deploy this project on server.

Strat the server

Right click on "JBoss 5.1 Runtime Server" from Servers view and click on Start.



Deploy the application :

Right click on "JBoss 5.1 Runtime Server" available in Servers view -> Add and Remove. -> Select the SimpleEjb3Project JAR file from the left pane and click Add-> and then Finish.



If the project is deployed properly with global JNDI mapping then you will see the following message in the console.

```

22:09:56,170 INFO [JBossASKernel] jndi:HelloBean/remote
22:09:56,170 INFO [JBossASKernel] jndi:HelloBean/remote-com.jwt.ejb.business.Hello
22:09:56,170 INFO [JBossASKernel] Class:com.jwt.ejb.business.Hello
22:09:56,171 INFO [JBossASKernel] Added bean(jboss.j2ee:jar=SimpleEjb3Project.jar,name=HelloBean,service=EJB3) to KernelDeployment of:
22:09:56,172 INFO [EJB3EndpointDeployer] Deploy AbstractBeanMetaData@d56991{name=jboss.j2ee:jar=SimpleEjb3Project.jar,name=HelloBean,service=EJB3}
22:09:56,248 INFO [SessionSpecContainer] Starting jboss.j2ee:jar=SimpleEjb3Project.jar,name=HelloBean,service=EJB3
22:09:56,249 INFO [EJBContainer] STARTED EJB: com.jwt.ejb.businesslogic.HelloBean ejbName: HelloBean
22:09:56,265 INFO [JndiSessionRegistrarBase] Binding the following Entries in Global JNDI:

HelloBean/remote - EJB3.x Default Remote Business Interface
HelloBean/remote-com.jwt.ejb.business.Hello - EJB3.x Remote Business Interface

```



Hello	Boss	Welcome	to EJB
-------	------	---------	--------