

MIT COLLEGE OF MANAGEMENT BCA SEM V

Index			
SR. No	Assignment Name	Marks	Sign
	<u>1: Implementation of JDBC</u>		
Q1:	Write java program for display Employee information from data base (use JDBC and MySQL) Assume suitable database and table		
Q2:	Write java program for insert student information in database (use JDBC and MySQL) Assume suitable database and table.		
	<u>2: Implementation of Servlet with JDBC, session and Cookies</u>		
Q3:	Write a servlet program to display current date and time of server.		
Q4:	Design a servlet to display “Welcome IP address of client” to first time visitor. Display “Welcome-back IP address of client” if the user is revisiting the page.		
Q5:	Design the table User (username, password) Design HTML login screen. Accept the username and password from the user. Write a servlet program to accept the login name and password and validates it from the database you have created. If it is correct then display Welcome.html otherwise display Error.html.		
Q6:	Write a servlet program to create a shopping mall. User must be allowed to do purchase from two pages. Each page should have a page total. The third page should display a bill, which consists of a page total of whatever the purchase has been done and print the total. (Use Http Session)		

	<u>3: Implementation of JSP with JDBC, session and Cookies</u>		
Q7:	Write a JSP program to display number of times user has visited the page. (Use cookies)		
Q8:	Write a JSP program to display details of products from database in tabular format. Design Product table in database as (pid, pname, qty, price)		
Q9:	Write a program, using servlet and JDBC which takes students roll number and provides student information, which includes the name of the student.		
Q10:	Write Username and password validation from database to jsp view page		
	<u>4: Implementation of RMI</u>		
Q11:	Creating a Simple RMI application involves following steps Define a remote interface. Implementing remote interface. create and start remote application create and start client application		
	<u>5: Socket Programming</u>		
Q12:	Build a Simple Chat Application Using Sockets in Java.		
	<u>6: Java Spring Framework</u>		
Q13:	Design spring example print hello world in screen.		
Q14:	Design Spring MVC CRUD Example.		
Q15:	Building web application with Spring Boot		

Topic 1: Implementation of JDBC

Q1: Write java program for display Employee information from data base (use JDBC and MySQL) Assume suitable database and table:

File: JDBC_example.java

```
import java.sql.*;

public class JDBC_example {

    public static void main(String args[]) {

        try {

            Class.forName("com.mysql.jdbc.Driver");

            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "root", "");

            Statement stmt = con.createStatement();

            ResultSet rs = stmt.executeQuery("select * from emp");

            while (rs.next())

                System.out.println(rs.getInt(1) + " " + rs.getString(2) + " " +
rs.getString(3));

            con.close();

        } catch (Exception e) {

            System.out.println(e);

        }

    }

}
```

Output:

File: Java_codes/JDBC_example.java

C:\xampp\tomcat\webapps\Java_codes>java JDBC_example.java

- 1 Prashant Mahto
- 2 Sushil Kumar
- 3 Sakshi Thorat
- 4 Prem Gane

Q2: Write java program for insert student information in database (use JDBC and MySQL) Assume suitable database and table.

File: JDBC_example_2.java

```
import java.sql.*;

public class JDBC_example_2 {

    public static void main(String args[]) {

        try {

            Class.forName("com.mysql.jdbc.Driver");

            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "root", "");

            Statement stmt = con.createStatement();

            System.out.println("Inserting records into the table...");

            String sql = "INSERT INTO student VALUES (100, 'Zara', 'Ali', 18)";
            stmt.executeUpdate(sql);

            sql = "INSERT INTO student VALUES (101, 'Mahnaz', 'Fatma', 25)";
            stmt.executeUpdate(sql);

            sql = "INSERT INTO student VALUES (102, 'Zaid', 'Khan', 30)";
            stmt.executeUpdate(sql);

            sql = "INSERT INTO student VALUES(103, 'Sumit', 'Mittal', 28)";
            stmt.executeUpdate(sql);

            System.out.println("Inserted records into the table...");

            con.close();

        } catch (Exception e) {

            System.out.println(e);

        }

    }

}
```

```
}
```

Output:

File: Java_codes/JDBC_example_2.java

id	name	surname	age
100	Zara	Ali	18
101	Mahnaz	Fatma	25
102	Zaid	Khan	30
103	Sumit	Mittal	28

Topic 2: Implementation of Servlet with JDBC, session and Cookies:

Q3: Write a servlet program to display current date and time of server.

File: Curr_date.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class DateTimeServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html");

        Date currentDate = new Date();

        PrintWriter out = response.getWriter();
```

```
        out.println("<html>");
        out.println("<head><title>Current Date and Time</title></head>");
        out.println("<body>");
        out.println("<h1>Current Date and Time on Server</h1>");
        out.println("<p>" + currentDate.toString() + "</p>");
        out.println("</body>");
        out.println("</html>");

        out.close();
    }
}
```

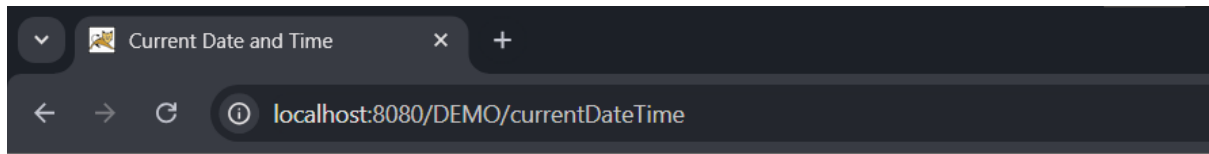
File: web.xml

```
<servlet>
    <servlet-name>DateTimeServlet</servlet-name>
    <servlet-class>DateTimeServlet</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>DateTimeServlet</servlet-name>
    <url-pattern>/currentDateTime</url-pattern>
</servlet-mapping>
```


Output:

File: DEMO/WEB_INF/classes/DateTimeServlet.class



Current Date and Time on Server

Fri Oct 25 16:02:44 IST 2024

Q4: Design a servlet to display “Welcome IP address of client” to first time visitor. Display “Welcome-back IP address of client” if the user is revisiting the page.

File: Welcome_Servlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Welcome_Servlet extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

        response.setContentType("text/html");

        String clientIp = request.getRemoteAddr()

        HttpSession session = request.getSession(true);

        String message;
        if (session.isNew()) {
            message = "Welcome " + clientIp;
        } else {
            message = "Welcome-back " + clientIp;
        }

        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>Welcome Page</title></head>");
        out.println("<body>");
```

```
        out.println("<h1>" + message + "</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

File: web.xml

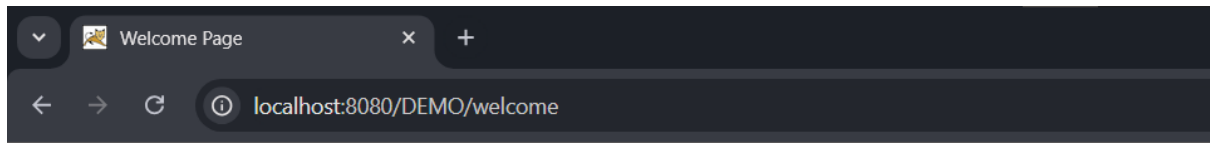
```
<?xml version="1.0" encoding="UTF-8"?>

<web-app 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"
  version="3.1"
  metadata-complete="true">

  <servlet>
    <servlet-name>Welcome_Servlet</servlet-name>
    <servlet-class>Welcome_Servlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>Welcome_Servlet</servlet-name>
    <url-pattern>/welcome</url-pattern>
  </servlet-mapping>
</web-app>
```

Output:



Welcome-back 0:0:0:0:0:0:0:1

Q5: Design the table User (username, password) Design HTML login screen. Accept the username and password from the user. Write a servlet program to accept the login name and password and validates it from the database you have created. If it is correct then display Welcome.html otherwise display Error.html.

File: login.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login</title>

</head>

<body>

    <h2>Login Page</h2>

    <form action="LoginServlet" method="post">

        <label for="username">Username:</label>

        <input type="text" id="username" name="username" required><br><br>

        <label for="password">Password:</label>

        <input type="password" id="password" name="password"
required><br><br>

        <button type="submit">Login</button>

    </form>

</body>

</html>
```

File: LoginServlet.java

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
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("/LoginServlet")
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    private final String dbURL = "jdbc:mysql://localhost:3306/test";
    private final String dbUser = "root";
    private final String dbPassword = "";

    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {

        String username = request.getParameter("username");
        String password = request.getParameter("password");
```

```
boolean isValidUser = false;

String sql = "SELECT * FROM User WHERE username = ? AND
password = ?";

try {
    Class.forName("com.mysql.cj.jdbc.Driver");

    try (Connection conn = DriverManager.getConnection(dbURL, dbUser,
dbPassword);

        PreparedStatement statement = conn.prepareStatement(sql)) {

        statement.setString(1, username);
        statement.setString(2, password);

        ResultSet resultSet = statement.executeQuery();

        if (resultSet.next()) {
            isValidUser = true;
        }
    }
} catch (Exception e) {
    e.printStackTrace();
}

if (isValidUser) {
    response.sendRedirect("Welcome.html");
} else {
    response.sendRedirect("Error.html");
}
```

```
}  
}  
}
```

File: Welcome.html

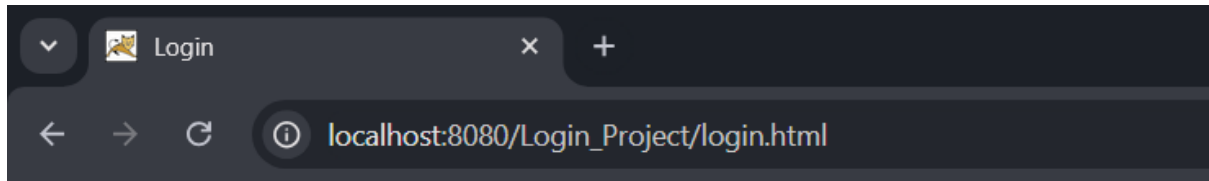
```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Welcome</title>  
</head>  
<body>  
  <h2>Welcome to the System!</h2>  
</body>  
</html>
```

File: Database Structure

<input type="checkbox"/> Show all Number of rows: 25 ▼ Filter rows: <input type="text" value="Search this table"/>	
Extra options	
username	password
admin	password
test	test
<input type="checkbox"/> Show all Number of rows: 25 ▼ Filter rows: <input type="text" value="Search this table"/>	

Output:

File: Login_Project\login.html

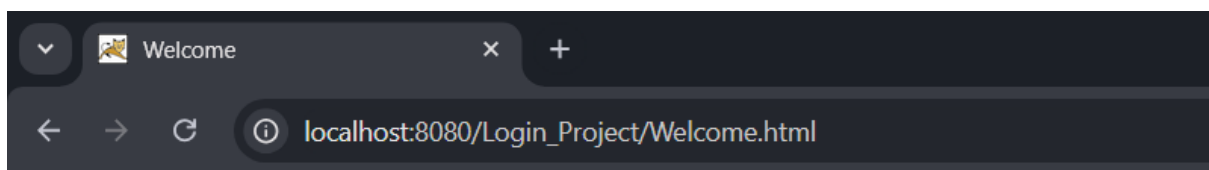


Login Page

Username:

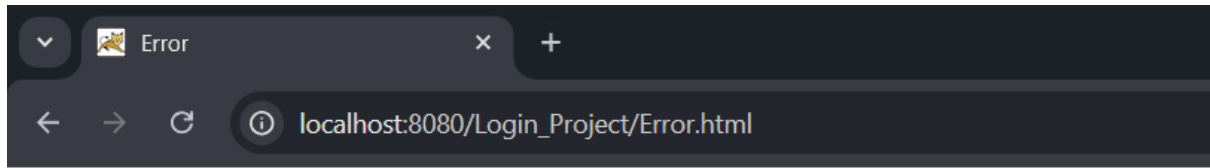
Password:

File: Login_Project\Welcome.html



Welcome to the System!

File: Login_Project\Error.html



Invalid username or password. Please try again.

Q6: Write a servlet program to create a shopping mall. User must be allowed to do purchase from two pages. Each page should have a page total. The third page should display a bill, which consists of a page total of whatever the purchase has been done and print the total. (Use Http Session)

File: ShoppinPage1.java

```
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/shoppingPage1")
public class ShoppingPage1 extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        HttpSession session = request.getSession();
        int page1Total = 0;
        String[] items = request.getParameterValues("item1");
        if (items != null) {
            for (String item : items) {
                page1Total += Integer.parseInt(item);
            }
        }
    }
}
```

```

        session.setAttribute("page1Total", page1Total);
        response.sendRedirect("shoppingPage2");
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<h2>Shopping Page 1</h2>");
        out.println("<form action='shoppingPage1' method='post'>");
        out.println("<input type='checkbox' name='item1' value='100'> Item A -
$100<br>");
        out.println("<input type='checkbox' name='item1' value='200'> Item B -
$200<br>");
        out.println("<input type='checkbox' name='item1' value='300'> Item C -
$300<br>");
        out.println("<input type='submit' value='Continue to Page 2'>");
        out.println("</form>");
    }
}

```

File: ShoppingPage2.java

```

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

```

```

import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/shoppingPage2")
public class ShoppingPage2 extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        HttpSession session = request.getSession();
        int page2Total = 0;
        String[] items = request.getParameterValues("item2");
        if (items != null) {
            for (String item : items) {
                page2Total += Integer.parseInt(item);
            }
        }
        session.setAttribute("page2Total", page2Total);
        response.sendRedirect("bill");
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<h2>Shopping Page 2</h2>");
        out.println("<form action='shoppingPage2' method='post'>");
        out.println("<input type='checkbox' name='item2' value='150'> Item D -
$150<br>");
        out.println("<input type='checkbox' name='item2' value='250'> Item E -
$250<br>");
    }
}

```

```

        out.println("<input type='checkbox' name='item2' value='350'> Item F -
$350<br>");
        out.println("<input type='submit' value='Generate Bill'>");
        out.println("</form>");
    }
}

```

File: Bill.java

```

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/bill")
public class Bill extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        HttpSession session = request.getSession();
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Integer page1Total = (Integer) session.getAttribute("page1Total");
        Integer page2Total = (Integer) session.getAttribute("page2Total");
    }
}

```

```

        int finalTotal = (page1Total != null ? page1Total : 0) + (page2Total != null
? page2Total : 0);

        out.println("<h2>Final Bill</h2>");

        out.println("Page 1 Total: $" + (page1Total != null ? page1Total : 0) +
"<br>");

        out.println("Page 2 Total: $" + (page2Total != null ? page2Total : 0) +
"<br>");

        out.println("<strong>Total Amount: $" + finalTotal + "</strong>");

    }
}

```

File: web.xml

```

<!DOCTYPE web-app PUBLIC
    "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
    <servlet>
        <servlet-name>ShoppingPage1</servlet-name>
        <servlet-class>ShoppingPage1</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>ShoppingPage1</servlet-name>
        <url-pattern>/shoppingPage1</url-pattern>
    </servlet-mapping>
    <servlet>
        <servlet-name>ShoppingPage2</servlet-name>
        <servlet-class>ShoppingPage2</servlet-class>
    </servlet>

```

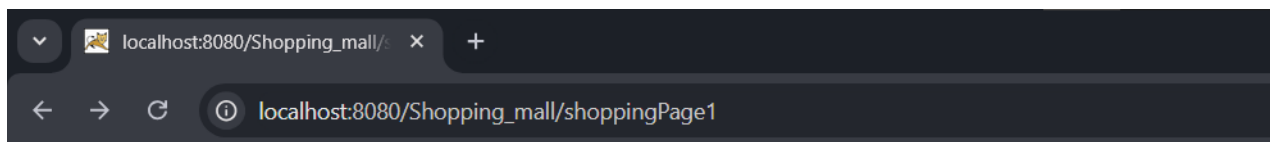
```

</servlet>
<servlet-mapping>
    <servlet-name>ShoppingPage2</servlet-name>
    <url-pattern>/shoppingPage2</url-pattern>
</servlet-mapping>
<servlet>
    <servlet-name>Bill</servlet-name>
    <servlet-class>Bill</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>Bill</servlet-name>
    <url-pattern>/bill</url-pattern>
</servlet-mapping>
<session-config>
    <session-timeout>30</session-timeout>
</session-config>
</web-app>

```

Output:

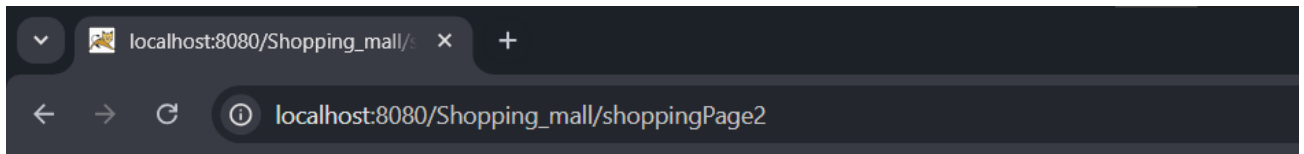
File: Shopping_mall\ShoppinPage1.java



Shopping Page 1

- ☐ Item A - \$100
 - ☐ Item B - \$200
 - ☐ Item C - \$300
- [Continue to Page 2](#)

File: Shopping_mall\ShoppingPage2.java

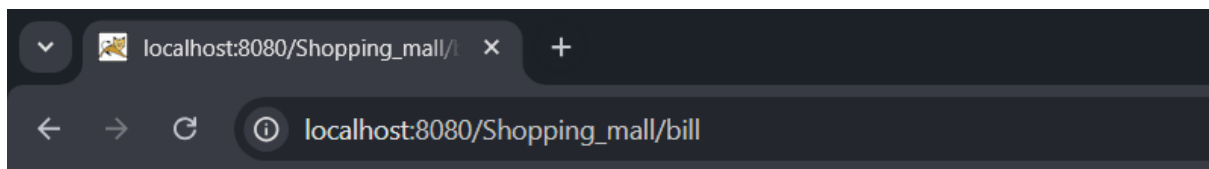


Shopping Page 2

- ☐ Item D - \$150
- ☒ Item E - \$250
- ☒ Item F - \$350

Generate Bill

File: Bill.java



Final Bill

Page 1 Total: \$200

Page 2 Total: \$600

Total Amount: \$800

Topic: 3: Implementation of JSP with JDBC, session and Cookies

Q7: Write a JSP program to display number of times user has visited the page. (Use cookies)

File: Count_view.jsp

```
<% @ page import="javax.servlet.http.Cookie" %>
<% @ page import="java.io.*" %>
<%
    int visitCount = 0;
    boolean isNewUser = true;
    String visitCountCookieName = "visitCount";
    Cookie[] cookies = request.getCookies();
    if (cookies != null) {
        for (Cookie cookie : cookies) {
            if (cookie.getName().equals(visitCountCookieName)) {
                visitCount = Integer.parseInt(cookie.getValue());
                isNewUser = false;
                break;
            }
        }
    }
    visitCount++;
    Cookie visitCookie = new Cookie(visitCountCookieName,
String.valueOf(visitCount));
    visitCookie.setMaxAge(60 * 60 * 24 * 365); // Cookie expiration time: 1 year
    response.addCookie(visitCookie);
%>
```

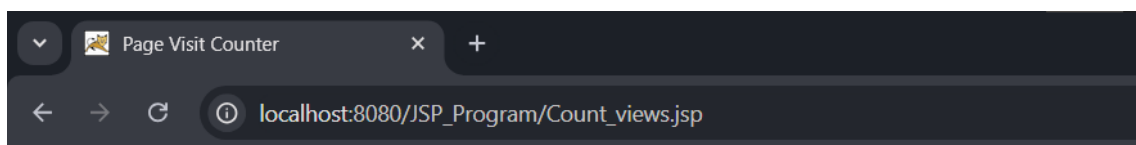
```

%>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Page Visit Counter</title>
</head>
<body>
    <h2>Welcome to the page!</h2>
    <%
        if (isNewUser) {
            out.println("<p>This is your first visit to this page!</p>");
        } else {
            out.println("<p>You have visited this page " + visitCount + "
times.</p>");
        }
    %>
</body>
</html>

```

Output: JSP_Programs\Count_view.jsp



Welcome to the page!

You have visited this page 3 times.

Q8: Write a JSP program to display details of products from database in tabular format. Design Product table in database as (pid, pname, qty, price)

File: Display_Program.jsp

```
<% @ page import="java.sql.*" %>
<% @ page import="java.io.*" %>
<%
    String jdbcURL = "jdbc:mysql://localhost:3306/test";
    String jdbcUsername = "root";
    String jdbcPassword = "";
    Connection conn = null;
    Statement stmt = null;
    ResultSet rs = null;
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        conn = DriverManager.getConnection(jdbcURL, jdbcUsername,
jdbcPassword);
        String sql = "SELECT * FROM Product";
        stmt = conn.createStatement();
        rs = stmt.executeQuery(sql);
    %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Product List</title>
```

```

<style>
  table {
    width: 50%;
    border-collapse: collapse;
    margin: 20px auto;
  }
  table, th, td {
    border: 1px solid black;
    padding: 10px;
    text-align: center;
  }
  th {
    background-color: #f2f2f2;
  }
</style>
</head>
<body>
<h2 style="text-align: center;">Product Details</h2>
<table>
  <tr>
    <th>Product ID</th>
    <th>Product Name</th>
    <th>Quantity</th>
    <th>Price</th>
  </tr>
  <%
    while (rs.next()) {

```










```

%>
<tr>
    <td><%= rs.getInt("pid") %></td>
    <td><%= rs.getString("pname") %></td>
    <td><%= rs.getInt("qty") %></td>
    <td><%= rs.getDouble("price") %></td>
</tr>
<%
    }
%>
</table>
<%
    } catch (Exception e) {
        out.println("<p>Error: " + e.getMessage() + "</p>");
    } finally {
        if (rs != null) try { rs.close(); } catch (SQLException ignore) {}
        if (stmt != null) try { stmt.close(); } catch (SQLException ignore) {}
        if (conn != null) try { conn.close(); } catch (SQLException ignore) {}
    }
%>
</body>
</html>

```

Database:

Extra options

			pid	pname	qty	price
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Product A	100 29.99
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	Product B	150 49.99
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	Product C	200 19.99

Output: JSP_Program/Display_Product.jsp



Product Details			
Product ID	Product Name	Quantity	Price
1	Product A	100	29.99
2	Product B	150	49.99
3	Product C	200	19.99

Q9: Write a program, using servlet and JDBC which takes students roll number and provides student information, which includes the name of the student

File: index.jsp

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Student Information</title>
</head>
<body>
<h2>Enter Student Roll Number</h2>
<form method="post" action="StudentInfoServlet">
    Roll Number: <input type="text" name="roll_no">
    <input type="submit" value="Get Student Info">
</form>
</body>
</html>
```

File: StudentInfoServlet.java

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.servlet.RequestDispatcher;
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("/StudentInfoServlet")
public class StudentInfoServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String JDBC_URL = "jdbc:mysql://localhost:3306/test";
    private static final String JDBC_USER = "root";
    private static final String JDBC_PASSWORD = "";

    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String rollNo = request.getParameter("roll_no");
        String studentName = null;
        if (rollNo != null && !rollNo.trim().isEmpty()) {
            Connection conn = null;
            PreparedStatement pstmt = null;
            ResultSet rs = null;
            try {
                Class.forName("com.mysql.cj.jdbc.Driver");
                conn = DriverManager.getConnection(JDBC_URL, JDBC_USER,
JDBC_PASSWORD);

                String sql = "SELECT name FROM Student WHERE roll_no = ?";
                pstmt = conn.prepareStatement(sql);
                pstmt.setInt(1, Integer.parseInt(rollNo));
                rs = pstmt.executeQuery();
                if (rs.next()) {
                    studentName = rs.getString("name");
                }
            }
        }
    }
}

```

```

        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try { if (rs != null) rs.close(); } catch (Exception e) {
e.printStackTrace(); }

            try { if (pstmt != null) pstmt.close(); } catch (Exception e) {
e.printStackTrace(); }

            try { if (conn != null) conn.close(); } catch (Exception e) {
e.printStackTrace(); }

        }
    }

    request.setAttribute("rollNo", rollNo);
    request.setAttribute("studentName", studentName);

    RequestDispatcher dispatcher =
request.getRequestDispatcher("studentInfo.jsp");
    dispatcher.forward(request, response);
}
}

```

File: studentInfo.jsp

```

<% @ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Student Information</title>
</head>
<body>
<%

```

```

String rollNo = (String) request.getAttribute("rollNo");
String studentName = (String) request.getAttribute("studentName");
%>
<h2>Student Information</h2>
<%
    if (rollNo != null && studentName != null) {
%>
        <p><strong>Roll Number:</strong> <%= rollNo %></p>
        <p><strong>Name:</strong> <%= studentName %></p>
    <%
        } else if (rollNo != null) {
%>
        <p>No student found with Roll Number: <%= rollNo %></p>
    <%
        } else {
%>
        <p>Please enter a roll number.</p>
    <%
        }
%>
</body>
</html>

```

File: web.xml

```
<!DOCTYPE web-app PUBLIC
    "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

<servlet>

    <servlet-name>StudentInfoServlet</servlet-name>

    <servlet-class>StudentInfoServlet</servlet-class>

</servlet>

<servlet-mapping>











    <servlet-name>StudentInfoServlet</servlet-name>

    <url-pattern>/StudentInfoServlet</url-pattern>

</servlet-mapping>

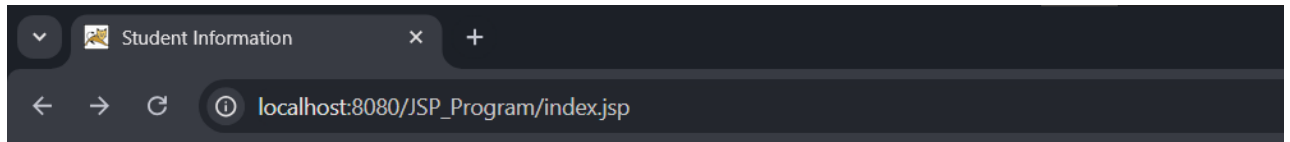
</web-app>
```

Database:

				
			roll_no	name
<input type="checkbox"/>	 Edit	 Copy	 Delete	101 Prashant
<input type="checkbox"/>	 Edit	 Copy	 Delete	102 Tushar
<input type="checkbox"/>	 Edit	 Copy	 Delete	103 Amit

Output:

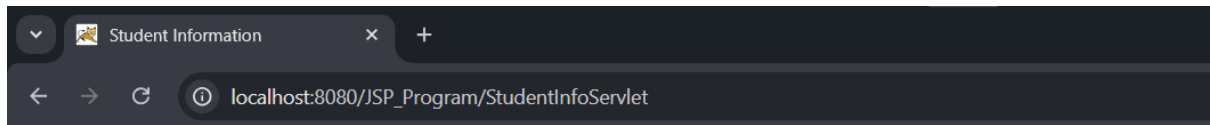
JSP_Program\index.jsp :



Enter Student Roll Number

Roll Number:

JSP_Program\StudentInfoServlet :



Student Information

Roll Number: 101

Name: Prashant

Q10: Write Username and password validation from database to jsp view page

File: login.jsp

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login</title>
</head>
<body>
<h2>Login</h2>
<form method="post" action="LoginServlet">
    Username: <input type="text" name="username" required><br>
    Password: <input type="password" name="password" required><br>
    <input type="submit" value="Login">
</form>
</body>
</html>
```

File: LoginServlet.java

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.servlet.RequestDispatcher;
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("/LoginServlet")
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    private static final String JDBC_URL = "jdbc:mysql://localhost:3306/test";
    private static final String JDBC_USER = "root";
    private static final String JDBC_PASSWORD = "";

    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        boolean isValidUser = false;

        if (username != null && password != null) {
            Connection conn = null;
            PreparedStatement pstmt = null;
            ResultSet rs = null;

            try {
                Class.forName("com.mysql.cj.jdbc.Driver");
                conn = DriverManager.getConnection(JDBC_URL, JDBC_USER,
JDBC_PASSWORD);

```

```

        String sql = "SELECT * FROM user WHERE username = ? AND
password = ?";

        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, username);
        pstmt.setString(2, password);
        rs = pstmt.executeQuery();
        if (rs.next()) {
            isValidUser = true;
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try { if (rs != null) rs.close(); } catch (Exception e) {
e.printStackTrace(); }

        try { if (pstmt != null) pstmt.close(); } catch (Exception e) {
e.printStackTrace(); }

        try { if (conn != null) conn.close(); } catch (Exception e) {
e.printStackTrace(); }

    }
}

request.setAttribute("isValidUser", isValidUser);
request.setAttribute("username", username);

RequestDispatcher dispatcher =
request.getRequestDispatcher("loginResult.jsp");
dispatcher.forward(request, response);
}
}

```


File: loginResult.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login Result</title>
</head>
<body>
<%
    Boolean isValidUser = (Boolean) request.getAttribute("isValidUser");
    String username = (String) request.getAttribute("username");
%>
<h2>Login Result</h2>
<%
    if (isValidUser != null && isValidUser) {
%>
        <p>Welcome, <strong><%= username %></strong>! You have
successfully logged in.</p>
<%
    } else {
%>
        <p>Invalid username or password. Please try again.</p>
<%
    }
%>
</body>
</html>
```

File: web.xml

```
<!DOCTYPE web-app PUBLIC
    "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

<servlet>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>LoginServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/LoginServlet</url-pattern>
</servlet-mapping>

</web-app>
```

Database:

✓ Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)

SELECT * FROM `user`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

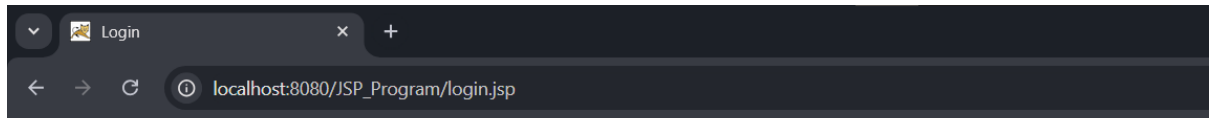
☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

username	password
admin	password
test	test
user1	pass1
user2	pass2

Output:

JSP_Program\login.jsp :

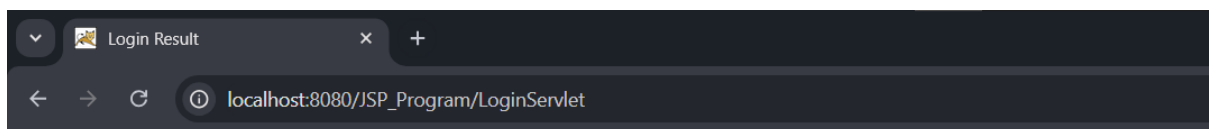


Login

Username:

Password:

JSP_Program\LoginServlet.java :



Login Result

Welcome, **admin**! You have successfully logged in.

4: Implementation of RMI

Q11: Creating a Simple RMI application involves following steps Define a remote interface. Implementing remote interface. create and start remote application create and start client application

File: server/RMIServer.java

```
package server;

import java.rmi.Naming;
import java.rmi.registry.LocateRegistry;

public class RMIServer {
    public static void main(String[] args) {
        try {
            LocateRegistry.createRegistry(1100);

            HelloImpl hello = new HelloImpl();

            Naming.rebind("rmi://localhost:1100/HelloService", hello);

            System.out.println("Server is ready.");
        } catch (Exception e) {
            System.out.println("Server exception: " + e.toString());
            e.printStackTrace();
        }
    }
}
```

File: server/Hello.java

```
package server;

import java.rmi.Remote;
import java.rmi.RemoteException;

public interface Hello extends Remote {
    String sayHello() throws RemoteException;
}
```

File: server/HelloImpl.java

```
package server;

import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class HelloImpl extends UnicastRemoteObject implements Hello {
    private static final long serialVersionUID = 1L;

    public HelloImpl() throws RemoteException {
        super();
    }
    @Override
    public String sayHello() throws RemoteException {
        return "Hello, world!";
    }
}
```

File: client/RMIClient.java

```
package client;

import server.Hello;

import java.rmi.Naming;

public class RMIClient {
    public static void main(String[] args) {
        try {

            Hello hello = (Hello)
Naming.lookup("rmi://localhost:1100/HelloService");

            String response = hello.sayHello();
            System.out.println("Response from server: " + response);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output:

```
C:\Program Files\Apache Software Foundation\Tomcat
9.0\webapps\RMI_Programs\build>java -cp . server.RMIServer
```

Server is ready.

Output:

C:\Program Files\Apache Software Foundation\Tomcat
9.0\webapps\RMI_Programs\build>java -cp . client.RMIClient

Response from server: Hello, world!

Topic 5: Socket Programming

Q12: Build a Simple Chat Application Using Sockets in Java

File: MyServre1.java

```
import java.net.*;
import java.io.*;
class MyServer1 {
    public static void main(String args[])throws Exception{
        ServerSocket ss=new ServerSocket(3333);
        Socket s=ss.accept();
        DataInputStream din=new DataInputStream(s.getInputStream());
        DataOutputStream dout=new DataOutputStream(s.getOutputStream());
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        String str="",str2="";
        while(!str.equals("stop")){
            str=din.readUTF();
            System.out.println("client says: "+str);
            str2=br.readLine();
            dout.writeUTF(str2);
            dout.flush();
        }
        din.close();
        s.close();
        ss.close();
    }
}
```


File: MyClient1.java

```
import java.net.*;
import java.io.*;
class MyClient1 {
    public static void main(String args[]) throws Exception {
        Socket s=new Socket("localhost",3333);
        DataInputStream din=new DataInputStream(s.getInputStream());
        DataOutputStream dout=new DataOutputStream(s.getOutputStream());
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        String str="",str2="";
        while(!str.equals("stop")){
            str=br.readLine();
            dout.writeUTF(str);
            dout.flush();
            str2=din.readUTF();
            System.out.println("Server says: "+str2);
        }
        dout.close();
        s.close();
    }
}
```

Output:

```
C:\Program Files\Apache Software Foundation\Tomcat
  9.0\webapps\Socket_Programming>java MyServer1.java
```

client says: hi

hello

Output:

```
C:\Program Files\Apache Software Foundation\Tomcat  
9.0\webapps\Socket_Programming>java MyClient1.java
```

hi

Server says: hello

Topic 6: Java Spring Framework

Q13: Design spring example print hello world in screen

File: HelloSpring/src/com.example/HelloWorld.java

```
package com.example;

public class HelloWorld {

    private String message;

    public void setMessage(String message){
        this.message = message;
    }

    public void getMessage(){
        System.out.println("Your Message : " + message);
    }

}
```

File: HelloSpring/src/com.example/MainApp.java

```
package com.example;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

    public static void main(String[] args) {

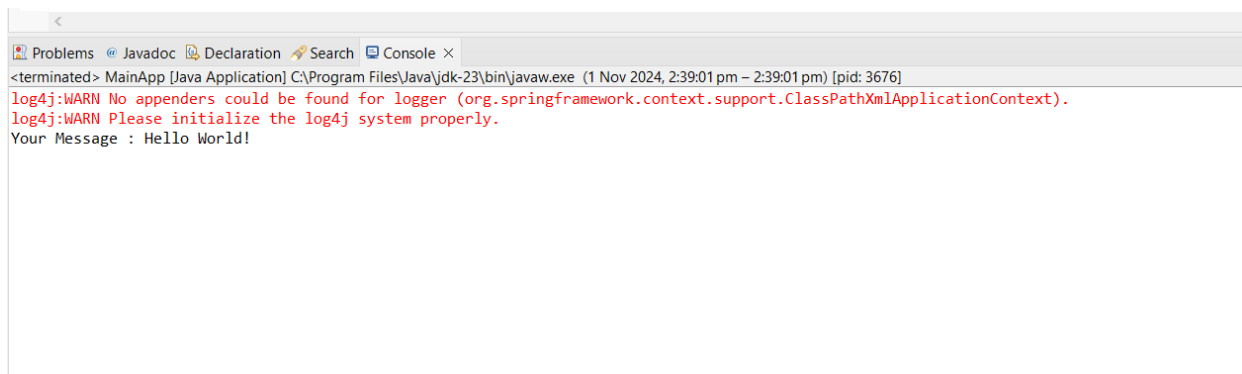
        ApplicationContext context = new
            ClassPathXmlApplicationContext("Beans.xml");
```

```
    HelloWorld obj = (HelloWorld) context.getBean("helloWorld");  
    obj.getMessage();  
}  
}
```

File: HelloSpring/src/Beans.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<beans xmlns = "http://www.springframework.org/schema/beans"  
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation = "http://www.springframework.org/schema/beans  
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">  
  
    <bean id = "helloWorld" class = "com.example.HelloWorld">  
        <property name = "message" value = "Hello World!"/>  
    </bean>  
</beans>
```

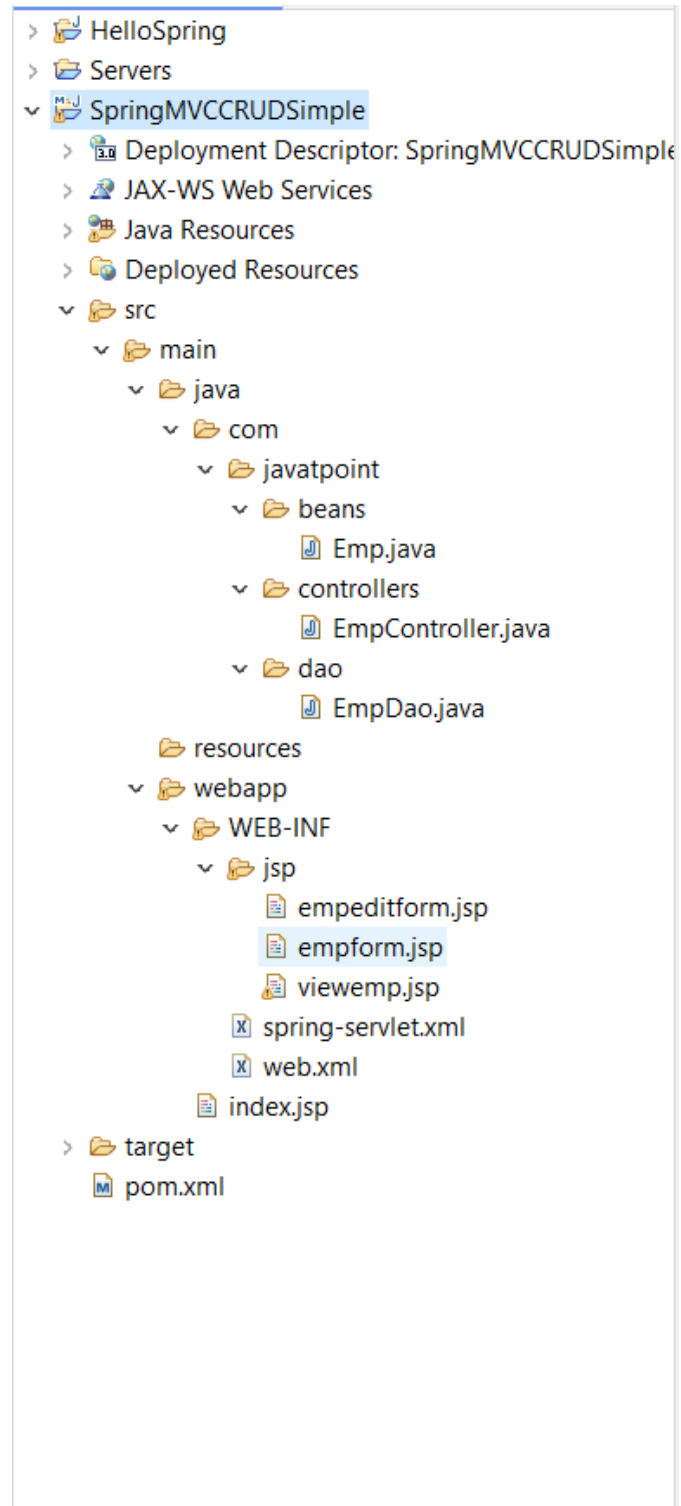
Output :



```
<terminated> MainApp [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (1 Nov 2024, 2:39:01 pm - 2:39:01 pm) [pid: 3676]  
log4j:WARN No appenders could be found for logger (org.springframework.context.support.ClassPathXmlApplicationContext).  
log4j:WARN Please initialize the log4j system properly.  
Your Message : Hello World!
```

Q14: Design Spring MVC CRUD Example.

File Structure:



File: src/main/java/com/javatpoint/beans/Emp.java

```
package com.javatpoint.beans;
```

```
public class Emp {  
    private int id;  
    private String name;  
    private float salary;  
    private String designation;
```

```
    public int getId() {  
        return id;  
    }
```

```
    public void setId(int id) {  
        this.id = id;  
    }
```

```
    public String getName() {  
        return name;  
    }
```

```
    public void setName(String name) {  
        this.name = name;  
    }
```

```
    public float getSalary() {  
        return salary;  
    }
```

```
    public void setSalary(float salary) {  
        this.salary = salary;  
    }
```

```

public String getDesignation() {
    return designation;
}

public void setDesignation(String designation) {
    this.designation = designation;
}

}

```

File: src/main/java/com/javatpoint/controllers/EmpController.java

```

package com.javatpoint.controllers;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import com.javatpoint.beans.Emp;
import com.javatpoint.dao.EmpDao;

@Controller
public class EmpController {

    @Autowired
    EmpDao dao;//will inject dao from xml file

    @RequestMapping("/empform")
    public String showform(Model m){

```

```

        m.addAttribute("command", new Emp());
        return "empform";
    }

    @RequestMapping(value="/save",method = RequestMethod.POST)
    public String save(@ModelAttribute("emp") Emp emp){
        dao.save(emp);
        return "redirect:/viewemp";//will redirect to viewemp request mapping
    }

    @RequestMapping("/viewemp")
    public String viewemp(Model m){
        List<Emp> list=dao.getEmployees();
        m.addAttribute("list",list);
        return "viewemp";
    }

    @RequestMapping(value="/editemp/{id}")
    public String edit(@PathVariable int id, Model m){
        Emp emp=dao.getEmpById(id);
        m.addAttribute("command",emp);
        return "empeditform";
    }

    @RequestMapping(value="/editsave",method = RequestMethod.POST)
    public String editsave(@ModelAttribute("emp") Emp emp){
        dao.update(emp);
        return "redirect:/viewemp";
    }

    @RequestMapping(value="/deleteemp/{id}",method = RequestMethod.GET)
    public String delete(@PathVariable int id){

```



```

        dao.delete(id);
        return "redirect:/viewemp";
    }
}

```

File: src/main/java/com/javatpoint/dao/EmpDao.java

```

package com.javatpoint.dao;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;
import org.springframework.jdbc.core.BeanPropertyRowMapper;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.RowMapper;
import com.javatpoint.beans.Emp;

public class EmpDao {
    JdbcTemplate template;

    public void setTemplate(JdbcTemplate template) {
        this.template = template;
    }

    public int save(Emp p){
        String sql="insert into Emp99(name,salary,designation)
values('"+p.getName()+"','"+p.getSalary()+"','"+p.getDesignation()+"')";
        return template.update(sql);
    }

    public int update(Emp p){

```

```

        String sql="update Emp99 set name='"+p.getName()+"",
salary='"+p.getSalary()+"",designation='"+p.getDesignation()+" where
id='"+p.getId()+"";

        return template.update(sql);
    }

    public int delete(int id){

        String sql="delete from Emp99 where id="+id+"";

        return template.update(sql);
    }

    public Emp getEmpById(int id){

        String sql="select * from Emp99 where id=?";

        return template.queryForObject(sql, new Object[]{id},new
        BeanPropertyRowMapper<Emp>(Emp.class));
    }

    public List<Emp> getEmployees(){

        return template.query("select * from Emp99",new RowMapper<Emp>(){

            public Emp mapRow(ResultSet rs, int row) throws SQLException {

                Emp e=new Emp();

                e.setId(rs.getInt(1));

                e.setName(rs.getString(2));

                e.setSalary(rs.getFloat(3));

                e.setDesignation(rs.getString(4));

                return e;
            }

        });
    }
}

```

File: src/main/webapp/index.jsp

```
<a href="empform">Add Employee</a>
```

```
<a href="viewemp">View Employees</a>
```

File: src/main/webapp/WEB-INF/jsp/empform.jsp

```
<% @ taglib uri="http://www.springframework.org/tags/form"
prefix="form"%>
```

```
<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

```
<h1>Add New Employee</h1>
```

```
<form:form method="post" action="save">
```

```
<table >
```

```
<tr>
```

```
<td>Name : </td>
```

```
<td><form:input path="name" /></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Salary :</td>
```

```
<td><form:input path="salary" /></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Designation :</td>
```

```
<td><form:input path="designation" /></td>
```

```
</tr>
```

```
<tr>
```

```
<td> </td>
```

```
<td><input type="submit" value="Save" /></td>
```

```
</tr>
</table>
</form:form>
```

File: src/main/webapp/WEB-INF/jsp/empeditform.jsp

```
<% @ taglib uri="http://www.springframework.org/tags/form"
prefix="form"%>
<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

```

    <h1>Edit Employee</h1>
    <form:form method="POST"
action="/SpringMVCCRUDSimple/editsave">
        <table >
            <tr>
                <td></td>
                <td><form:hidden path="id" /></td>
            </tr>
            <tr>
                <td>Name : </td>
                <td><form:input path="name" /></td>
            </tr>
            <tr>
                <td>Salary :</td>
                <td><form:input path="salary" /></td>
            </tr>
            <tr>
                <td>Designation :</td>
```

```

        <td><form:input path="designation" /></td>
    </tr>

    <tr>
        <td> </td>
        <td><input type="submit" value="Edit Save" /></td>
    </tr>
</table>
</form:form>

```

File: src/main/webapp/WEB-INF/jsp/viewmap.jsp

```

<% @ taglib uri="http://www.springframework.org/tags/form"
prefix="form"%>

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

    <h1>Employees List</h1>

    <table border="2" width="70%" cellpadding="2">
        <tr><th>Id</th><th>Name</th><th>Salary</th><th>Designation</th><th>
Edit</th><th>Delete</th></tr>
        <c:forEach var="emp" items="${list}">
            <tr>
                <td>${emp.id}</td>
                <td>${emp.name}</td>
                <td>${emp.salary}</td>
                <td>${emp.designation}</td>
                <td><a href="editemp/${emp.id}">Edit</a></td>
                <td><a href="deleteemp/${emp.id}">Delete</a></td>
            </tr>
        </c:forEach>
    </table>

```

```

</tr>
</c:forEach>
</table>
<br/>
<a href="empform">Add New Employee</a>

```

File: src/main/webapp/WEB-INF/spring-servlet.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.0.xsd">

    <context:component-scan base-
package="com.javatpoint.controllers"></context:component-scan>

    <bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/jsp/"></property>
    <property name="suffix" value=".jsp"></property>
    </bean>

    <bean id="ds"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">

```

```

<property name="driverClassName"
value="com.mysql.jdbc.Driver"></property>

<property name="url" value="jdbc:mysql://localhost:3306/test"></property>

<property name="username" value="root"></property>

<property name="password" value=""></property>

</bean>

<bean id="jt" class="org.springframework.jdbc.core.JdbcTemplate">
<property name="dataSource" ref="ds"></property>
</bean>

<bean id="dao" class="com.javatpoint.dao.EmpDao">
<property name="template" ref="jt"></property>
</bean>

</beans>

```

File: src/main/webapp/WEV-INF/web.xml

```

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID"
version="3.0">

  <display-name>SpringMVC</display-name>

  <servlet>

    <servlet-name>spring</servlet-name>

    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>

```

```

    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>spring</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
</web-app>

```

File: target/pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.javatpoint</groupId>
    <artifactId>SpringMVCPagination</artifactId>
    <packaging>war</packaging>
    <version>0.0.1-SNAPSHOT</version>
    <name>SpringMVCPagination Maven Webapp</name>
    <url>http://maven.apache.org</url>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>

```



```
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-beans</artifactId>
<version>5.1.1.RELEASE</version>
</dependency>
```

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>5.1.1.RELEASE</version>
</dependency>
```

```
<dependency>
  <groupId>org.apache.tomcat</groupId>
  <artifactId>tomcat-jasper</artifactId>
  <version>9.0.12</version>
</dependency>
```

```
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>4.0.1</version>
  <scope>provided</scope>
</dependency>
```

```
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
</dependency>
```

```

<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.11</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>5.1.1.RELEASE</version>
</dependency>
</dependencies>
<build>
  <finalName>SpringMVCPagination</finalName>
</build>
</project>

```

Database:

Showing rows 0 - 1 (2 total, Query took 0.0001 seconds.)

SELECT * FROM emp99

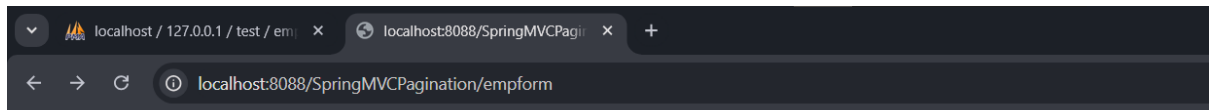
Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	name	salary	designation
<input type="checkbox"/>	1	Prashant Mahto	5000000.0	CEO
<input type="checkbox"/>	2	Vedant Rajankar	250000.0	Manager

Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations: Print Copy to clipboard Export Display chart Create view

Output:

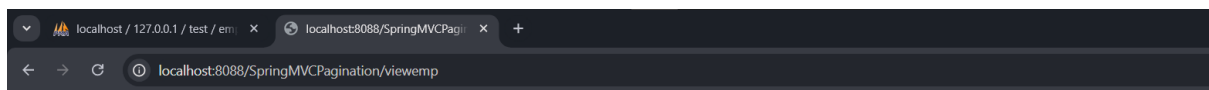


Add New Employee

Name :

Salary :

Designation :



Employees List

Id	Name	Salary	Designation	Edit	Delete
1	Prashant Mahto	5.0E7	CEO	Edit	Delete
2	Vedant Rajankar	250000.0	Manager	Edit	Delete
3	Amit	258000.0	Product Manager	Edit	Delete

[Add New Employee](#)

Q15: Building web application with Spring Boot

File:

src/main/java/com/example/hellospringboot/HelloSpringBootApplications.java

```
package com.example.hellospringboot;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
```

```
public class HelloSpringBootApplication {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(HelloSpringBootApplication.class, args);
```

```
    }
```

```
}
```

File:

src/main/java/com/example/hellospringboot/controller/HelloController.java

```
package com.example.hellospringboot.controller;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
```

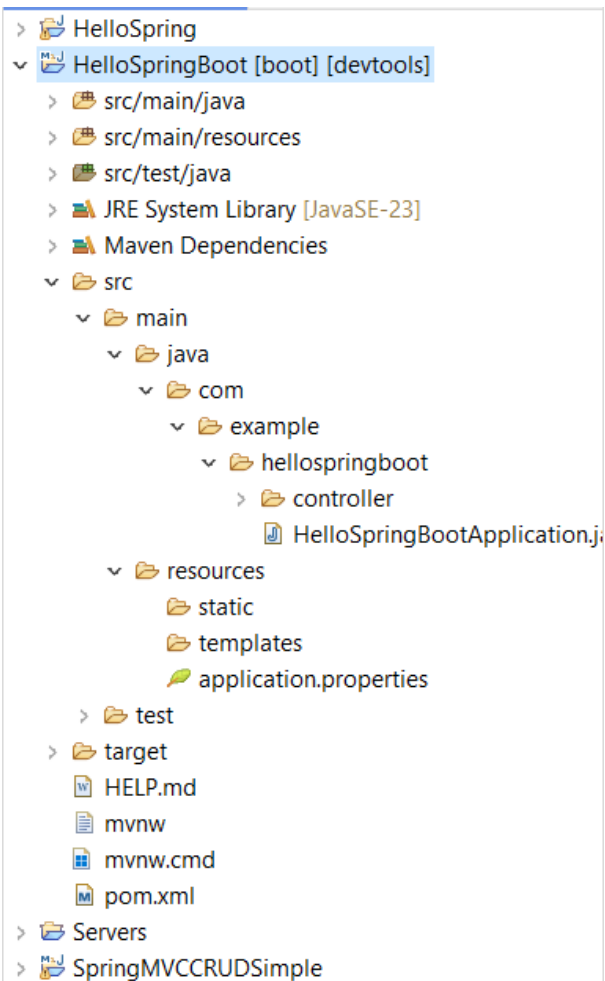
```
public class HelloController {
```

```

    @GetMapping("/hello")
    public String sayHello() {
        return "Hello, Spring Boot!";
    }
}

```

File Structure:

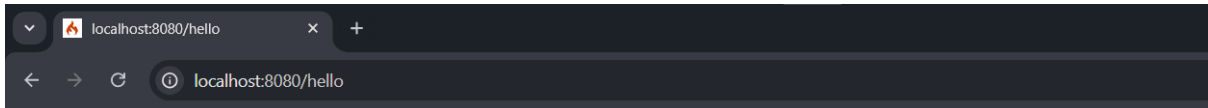


```

> HelloSpring
├── HelloSpringBoot [boot] [devtools]
│   ├── src/main/java
│   ├── src/main/resources
│   ├── src/test/java
│   ├── JRE System Library [JavaSE-23]
│   ├── Maven Dependencies
│   └── src
│       ├── main
│       │   ├── java
│       │   │   ├── com
│       │   │   │   ├── example
│       │   │   │   │   ├── hellospringboot
│       │   │   │   │   │   └── controller
│       │   │   │   │   │       HelloSpringBootApplication.java
│       │   │   └── resources
│       │   │       ├── static
│       │   │       ├── templates
│       │   │       └── application.properties
│       │   └── test
│       ├── target
│       ├── HELP.md
│       ├── mvnw
│       ├── mvnw.cmd
│       └── pom.xml
├── Servers
└── SpringMVCCRUDSimple

```


Output:



Hello, Spring Boot!