PRACTICAL 1

AIM : Write down basic steps to establish database connection from java. Also write the connection code for different db.

THEORY :--

To establish a database connection from Java, you typically follow these basic steps:

1. **Load the Database Driver**: First, you need to load the appropriate JDBC driver for the database you're connecting to. This step is necessary to register the driver with the DriverManager.
2. **Set Up Connection URL**: Define the connection URL for your database. The URL typically includes information such as the database type, host, port, and database name.
3. **Provide Database Credentials**: You need to provide the username and password to authenticate and gain access to the database.
4. **Establish Connection**: Use the DriverManager to establish a connection to the database by passing the connection URL, username, and password.
5. **Execute SQL Queries**: Once the connection is established, you can create statements and execute SQL queries, updates, or other database operations.
6. **Process Results**: If you execute a query that returns a result set, you can iterate over the results and process them as needed.
7. **Close Resources**: After you finish working with the database, it's important to close the connection, statement, and result set to release resources and prevent memory leaks.

CODE :--

Index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Lab 1</title>
</head>
<body>
	<form action="Lab1" method="post">
		<input type="submit" value="submit">
	</form>
</body>
</html>
```

Lab1.java

```
package ajtLab1;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
```

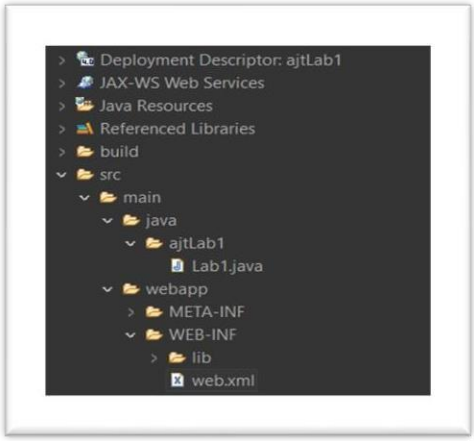
```
public class Lab1 extends HttpServlet{
private static final long serialVersionUID = 1L;
public void doPost(HttpServletRequest req, HttpServletResponse res) throws
IOException {
        Connection connection = null;
        PrintWriter out = res.getWriter();
        try {
          Class.forName("com.mysql.cj.jdbc.Driver");
          String url = "jdbc:mysql://localhost:3306/studentdb";
          String username = "root";
          String password = " ";
        connection = DriverManager.getConnection(url, username, password);
        out.println("Connected to the database!");
        }
        catch (ClassNotFoundException | SQLException e) {
                e.printStackTrace();
                out.println("Failed to connect to the database.");
         } finally {
                if (connection != null) {
                try {
                    connection.close();
                     } catch (SQLException e) {
                                e.printStackTrace();
                                    }
                        }
                }
        }
}
```

web.xml

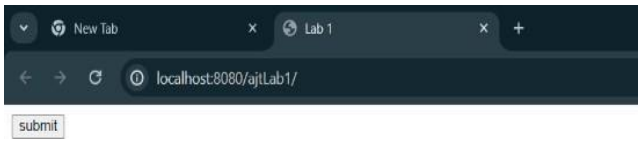
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
  <display-name>ajtLab1</display-name>



  <servlet>
<servlet-name>Lab1</servlet-name>
        <servlet-class>ajtLab1.Lab1</servlet-class>
  </servlet>
  <servlet-mapping>
      <servlet-name>Lab1</servlet-name>
      <url-pattern>/Lab1</url-pattern>
    </servlet-mapping>
</web-app>
```

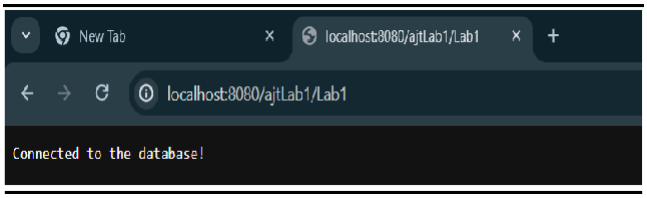
OUTPUT :--

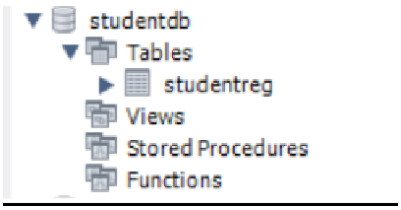
File Structure



Web Output :







PRACTICAL 2

AIM : Users can create a new database and also create a new table under that database. Once a database has been created then the user can perform database operation by calling above functions. Use following Java Statement interface to implement program:
1. Statement 2. Prepared statement 3. Callable statement

Write a JDBC application which will perform CRUD operation on the student table.

CODE :--

Index.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Lab2 CRUD</title>
</head>
<body>
    <h2>Lab2 CRUD Operations</h2>
    <form action="Lab2" method="post">
        <input type="hidden" name="action" value="create">
        Name: <input type="text" name="name"><br>
        Age: <input type="number" name="age"><br>
        <input type="submit" value="Create">
    </form>
    <br>
```

```
<form action="Lab2" method="post">
        <input type="hidden" name="action" value="read">
        <input type="submit" value="Read">
    </form>
    <br>
    <form action="Lab2" method="post">
        <input type="hidden" name="action" value="update">
        ID of student to update: <input type="number" name="id"><br>
        New Name: <input type="text" name="name"><br>
        New Age: <input type="number" name="age"><br>
        <input type="submit" value="Update">
    </form>
    <br>
    <form action="Lab2" method="post">
        <input type="hidden" name="action" value="delete">
        ID of student to delete: <input type="number" name="id"><br>
        <input type="submit" value="Delete">
    </form>
</body>
</html>
```

Lab2.java

```
package com.java;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
public class Lab2 extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String action = request.getParameter("action");
        String url = "jdbc:mysql://localhost:3306/lab2db";
        String user = "root";
        String password = " ";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection(url, user, password);
            if (action.equals("create")) {
                String name = request.getParameter("name");
                int age = Integer.parseInt(request.getParameter("age"));
                // CallableStatement
                CallableStatement cstmt = conn.prepareCall("{CALL insert_record(?, ?)}");
```

```
            cstmt.setString(1, name);
            cstmt.setInt(2, age);
            cstmt.executeUpdate();
            out.println("Record inserted successfully!");
        } else if (action.equals("read")) {
            //Statement
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM lab2table");
            out.println("<table border=1>");
            out.println("<tr><th>Name</th><th>Age</th></tr>");
            while (rs.next()) {
                out.println("<tr><td>" + rs.getString("name") + "</td><td>" + rs.getInt("age")
+ "</td></tr>");
            }
            out.println("</table>");
        } else if (action.equals("update")) {
            int id = Integer.parseInt(request.getParameter("id"));
            String name = request.getParameter("name");
            int age = Integer.parseInt(request.getParameter("age"));
            //Prepared Statement
            PreparedStatement pstmt = conn.prepareStatement("UPDATE lab2table SET
name=?, age=? WHERE id=?");
            pstmt.setString(1, name);
            pstmt.setInt(2, age);
            pstmt.setInt(3, id);
            pstmt.executeUpdate();
            out.println("Record updated successfully!");
        }
```

```
else if (action.equals("delete")) {
            int id = Integer.parseInt(request.getParameter("id"));
            PreparedStatement pstmt = conn.prepareStatement("DELETE FROM
lab2table WHERE id=?");
            pstmt.setInt(1, id);
            pstmt.executeUpdate();
            out.println("Record deleted successfully!");
        }
        conn.close();
    } catch (Exception e) {
        out.println(e);
    }

    out.close();
  }
}
```

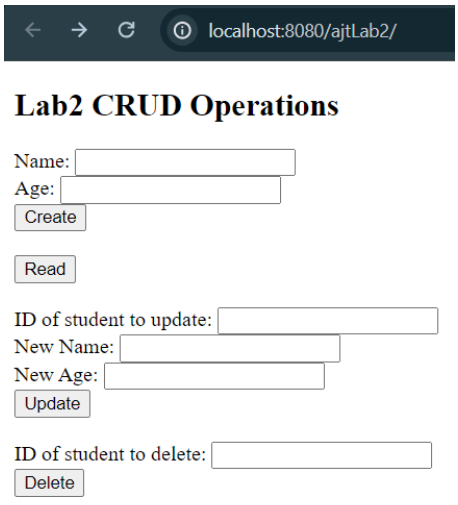
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <display-name>ajtLab2</display-name>
  <servlet>
    <servlet-name>Lab2</servlet-name>
    <servlet-class>com.java.Lab2</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Lab2</servlet-name>
    <url-pattern>/Lab2</url-pattern>
  </servlet-mapping>
</web-app>
```

OUTPUT :--

File Structure

Web Output :



CRUD Operations :

Create :

Name: A
Age: 11
Create

localhost:8080/ajtLab2/Lab2

Record inserted successfully!

Read :

localhost:8080/ajtLab2/Lab2

Name	Age
megh	10
ved	19
aaditya	20
A	11

Update :

ID of student to update: 6
New Name: Akhil
New Age: 21
Update

localhost:8080/ajtLab2/Lab2

Record updated successfully!

Delete :

ID of student to delete: 6
Delete

localhost:8080/ajtLab2/Lab2

Record deleted successfully!

MySQL Database :

```
1 •  SELECT * FROM lab2db.lab2table;
```

Result Grid | Filter Rows:

id	name	age
3	megh	10
4	ved	19
5	aaditya	20
NULL	NULL	NULL

insert_record Procedure :

Name: insert_record
The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
1 •  CREATE DEFINER=`root`@`localhost` PROCEDURE `insert_record`(IN p_name VARCHAR(255), IN p_age INT)
2    BEGIN
3        INSERT INTO lab2table (name, age) VALUES (p_name, p_age);
4    END
```

PRACTICAL 3

AIM : Write a JDBC application to display records from the database using metadata.

CODE :--

Index.html

```
<html>
    <head>
        <title>Practical 3</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="StudentServlet">
            <input type="submit" name="Go to Table Data">
        </form>
    </body>
</html>
```

StudentServlet.java

```
package com.java;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
```

```
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
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("/StudentServlet")
public class StudentServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        String url = "jdbc:derby://localhost:1527/StudentRegistration";
        String username = "admin1";
        String password = "admin1";
        try {
            Connection connection = DriverManager.getConnection(url, username,
password);
            Statement statement = connection.createStatement();
            ResultSet result = statement.executeQuery("SELECT * FROM studentreg");
            ResultSetMetaData rsMeta = result.getMetaData();
            int numberOfColumns = rsMeta.getColumnCount();
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
```

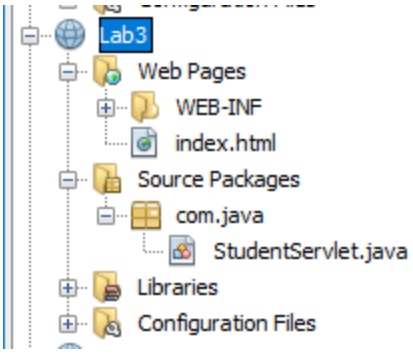
```
            out.println("<table border=1 width=50% height=50%>");
            out.println("<tr>");

for (int i = 1; i <= numberOfColumns; i++) {
                out.println("<th>" + rsMeta.getColumnName(i) + "</th>");
            }
            out.println("</tr>");

            while (result.next()) {
                out.println("<tr>");
                for (int i = 1; i <= numberOfColumns; i++) {
                    out.println("<td>" + result.getObject(i) + "</td>");
                }
                out.println("</tr>");
            }

            out.println("</table>");
            out.close();

            result.close();
            statement.close();
            connection.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

web.xml

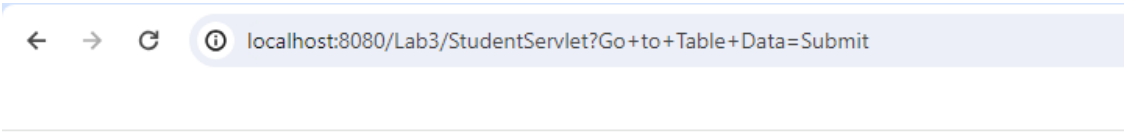
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
    <servlet>
        <servlet-name>StudentServlet</servlet-name>
        <servlet-class>  com.java.StudentServlet  </servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name> StudentServlet </servlet-name>
        <url-pattern>/StudentServlet  </url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>
```

OUTPUT :--

File Structure



Web Output :



NAME	EMAIL	PASSWORD
Megh Bhatt	meghbhatt.21.cs@iite.indusuni.ac.in	
Aaditya Puranmalka	puranmalkaaaditya.cs@iite.indusuni.ac.in	
Ved Desai	veddesai.21.cs@iite.indusuni.ac.in	

PRACTICAL 4

AIM : Create a web application for servlet and study web descriptor files. Write a servlet code which performs servlet context and servlet config object.

CODE :--

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>lab 4</title>
</head>
<body>
    <h2>Lab 4 AJT</h2>
    <form action="Lab4" method="get">
        <input type="submit" value="Get Addition">
    </form>
</body>
</html>
```

Lab4.java

```
package com.java;
import jakarta.servlet.ServletConfig;
import jakarta.servlet.ServletContext;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
public class Lab4 extends HttpServlet {
        private static final long serialVersionUID = 1L;
        private String servletName;

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        servletName = config.getServletName();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        ServletContext context = getServletContext();
        int contextParamValue = Integer.parseInt(context.getInitParameter("Num2"));
```

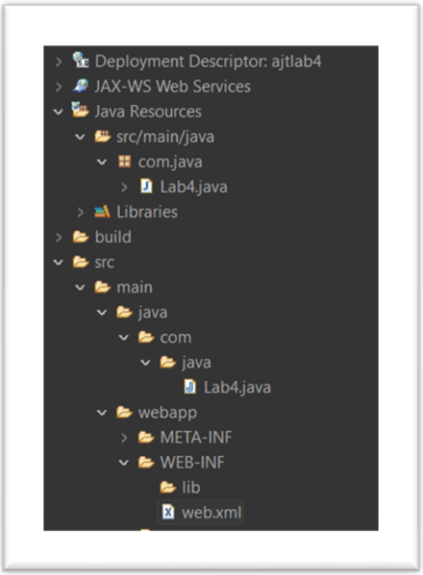
```
        int servletParamValue =
Integer.parseInt(getServletConfig().getInitParameter("Num1"));
        int num1 = contextParamValue;
        int num2 = servletParamValue;
        int sum = num1 + num2;
        out.println("<html><body>");
        out.println("<h3>Servlet Context and Servlet Config Example with Addition</h3>");
        out.println("<p>Servlet Name: " + servletName + "</p>");
        out.println("<p>Context Parameter Value: " + contextParamValue + "</p>");
        out.println("<p>Servlet Parameter Value: " + servletParamValue + "</p>");
        out.println("<p>Sum of " + num1 + " and " + num2 + " is: " + sum + "</p>");
        out.println("</body></html>");
        out.close();
    }
}
```

web.xml

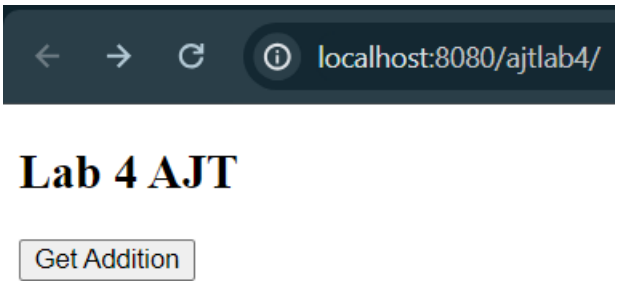
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="https://jakarta.ee/xml/ns/jakartaee"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd" id="WebApp_ID" version="6.0">
  <display-name>ajtlab4</display-name>
  <servlet>
    <description></description>
    <display-name>Lab4</display-name>
    <servlet-name>Lab4</servlet-name>
    <servlet-class>com.java.Lab4</servlet-class>
    <init-param>
          <param-name>Num1</param-name>
          <param-value>10</param-value>
        </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>Lab4</servlet-name>
    <url-pattern>/Lab4</url-pattern>
  </servlet-mapping>
   <context-param>
       <param-name>Num2</param-name>
       <param-value>20</param-value>
    </context-param>
</web-app>
```

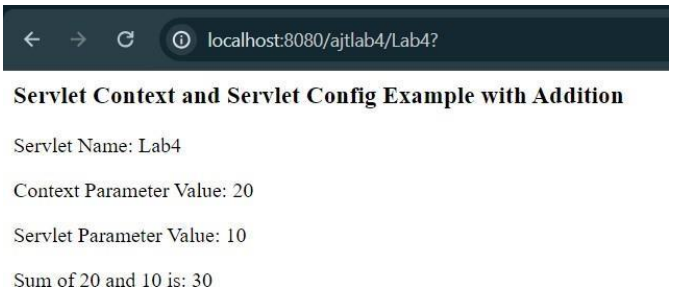
OUTPUT :--

File Structure :



Web Output :





PRACTICAL 5

AIM : Implement login form and perform session management using different methods.

1. Using Cookie :

CODE :-

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>
</head>
<body>
    <h2>Login Form with Cookie</h2>
    <form action="Lab5" 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>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

Lab5.java

```
package com.java;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.sql.*;
public class Lab5 extends HttpServlet {
	private static final long serialVersionUID = 1L;
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		try {
			Class.forName("com.mysql.cj.jdbc.Driver");
			Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/logindb","root"," ");
			String username = request.getParameter("username");
			String password = request.getParameter("password");
			 PreparedStatement stmt = conn.prepareStatement("SELECT * FROM login_info WHERE username=? AND password=?");
			 stmt.setString(1, username);
		stmt.setString(2, password);
		ResultSet rs = stmt.executeQuery();
			if (rs.next()) {
				Cookie usernameCookie = new Cookie("username", username);
				usernameCookie.setMaxAge(3600);
				response.addCookie(usernameCookie);
```

```
                        response.sendRedirect("welcome.jsp");
                    } else {
                        response.sendRedirect("error.jsp");
                    }
                }
                catch(Exception e) {
                    e.printStackTrace();
                    System.out.println(e);
                }

        }
}
```

welcome.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome With Cookie</title>
</head>
<body>

    <%
        String username = null;
        Cookie[] cookies = request.getCookies();
```

```
if (cookies != null) {
        for (Cookie cookie : cookies) {
            if (cookie.getName().equals("username")) {
                username = cookie.getValue();
                break;
            }
        }
    }

    if (username != null) {
        out.println("<h2>Welcome, " + username + "!</h2>");
    } else {
        out.println("<p>No username found in cookies.</p>");
    }
  %>
</body>
</html>
```

error.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
        pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Error </title>
```

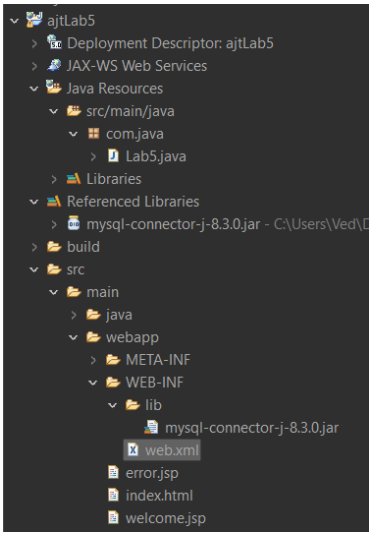
```
</head>
<body>
        <h2>Error</h2>
        <p>Invalid username or password!</p>
</body>
</html>
```

web.xml

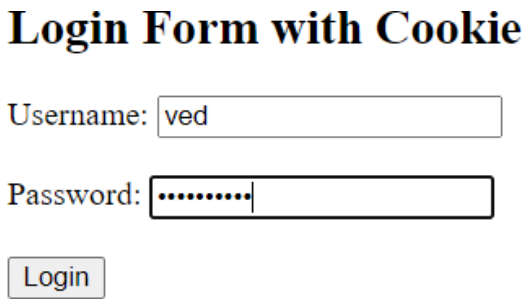
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app >
  <display-name>ajtLab5</display-name>
  <servlet>
    <display-name>Lab5</display-name>
    <servlet-name>Lab5</servlet-name>
    <servlet-class>com.java.Lab5</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Lab5</servlet-name>
    <url-pattern>/Lab5</url-pattern>
  </servlet-mapping>
</web-app>
```

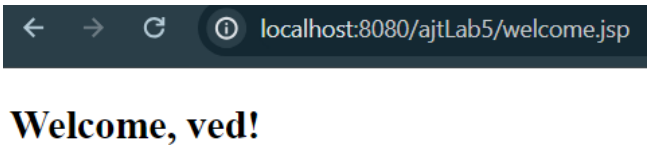
OUTPUT :--

File Structure :



Web Output :





2. Using Hidden Form Field :

CODE :-

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form </title>
</head>
<body>
    <h2>Login Form with Hidden Form Field</h2>
    <form action="Lab5" 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>
        <input type="hidden" id="sessionID" name="sessionID">
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

Lab5.java

```
package com.java;

import jakarta.servlet.ServletException;

import jakarta.servlet.http.*;

import java.io.IOException;

import java.sql.*;

public class Lab5 extends HttpServlet {

	private static final long serialVersionUID = 1L;

	protected void doPost(HttpServletRequest request, HttpServletResponse
response)

throws ServletException, IOException {

		try {

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

			Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/logindb", "root", "raag1995");

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

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

			PreparedStatement stmt = conn.prepareStatement("SELECT *
FROM login_info WHERE username=? AND password=?");

			stmt.setString(1, username);

			stmt.setString(2, password);

			ResultSet rs = stmt.executeQuery();

			if (rs.next()) {

				String sessionID = request.getSession().getId();

				request.setAttribute("sessionID", sessionID);

				request.setAttribute("username", username);

			request.getRequestDispatcher("welcome.jsp").forward(request,
response);

			} else {
```

```
                    response.sendRedirect("error.jsp");
                }
            } catch (Exception e) {
                e.printStackTrace();
                System.out.println(e);
            }
        }
}
```

welcome.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome With Hidden Form Field</title>
</head>
<body>
   <h2>Welcome, <%= request.getAttribute("username") %></h2>
</body>
</html>
```

error.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
        pageEncoding="UTF-8"%>
```

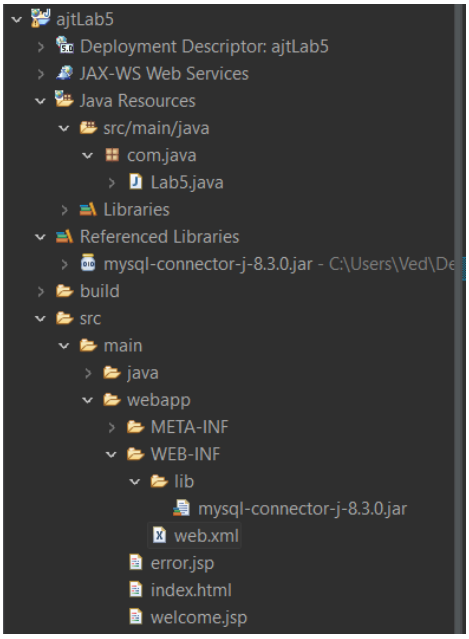
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Error </title>
</head>
<body>
        <h2>Error</h2>
        <p>Invalid username or password!</p>
</body>
</html>
```

web.xml

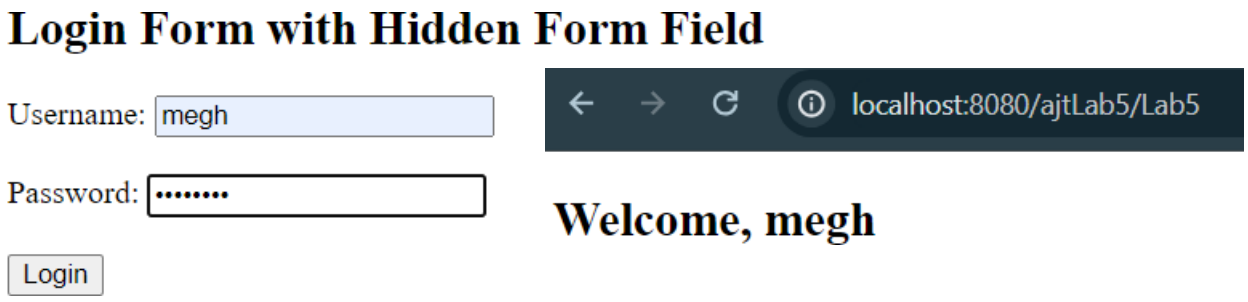
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <display-name>ajtLab5</display-name>
  <servlet>
    <servlet-name>Lab5</servlet-name>
    <servlet-class>com.java.Lab5</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Lab5</servlet-name>
    <url-pattern>/Lab5</url-pattern>
  </servlet-mapping>
</web-app>
```

OUTPUT :--

File Structure :



Web Output :



3. With Url Rewriting :

CODE :-

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>
</head>
<body>
    <h2>Login Form with URL Rewriting</h2>
    <form action="Lab5" 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>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

Lab5.java

```
package com.java;

import jakarta.servlet.ServletException;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.sql.*;

public class Lab5 extends HttpServlet {

	private static final long serialVersionUID = 1L;

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

		try {

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

			Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/logindb", "root", "raag1995");

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

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

			PreparedStatement stmt = conn.prepareStatement("SELECT *
FROM login_info WHERE username=? AND password=?");

			stmt.setString(1, username);

			stmt.setString(2, password);

			ResultSet rs = stmt.executeQuery();

			if (rs.next()) {

			response.sendRedirect("welcome.jsp?username=" + username);

			}

			else {
```

```
                response.sendRedirect("error.jsp");
                }
            } catch (Exception e) {
                e.printStackTrace();
                System.out.println(e);
            }
        }
```

Welcome.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome With Hidden Form Field</title>
</head>
<body>
    <h2>Welcome <%= request.getParameter("username") %></h2>
</body>
</html>
```

error.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
        pageEncoding="UTF-8"%>
<!DOCTYPE html>
```

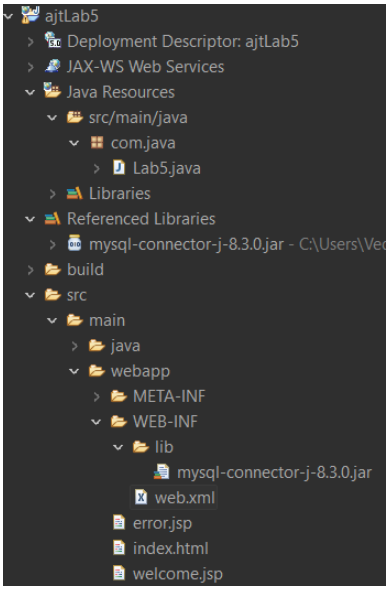
```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Error </title>
</head>
<body>
        <h2>Error</h2>
        <p>Invalid username or password!</p>
</body>
</html>
```

web.xml

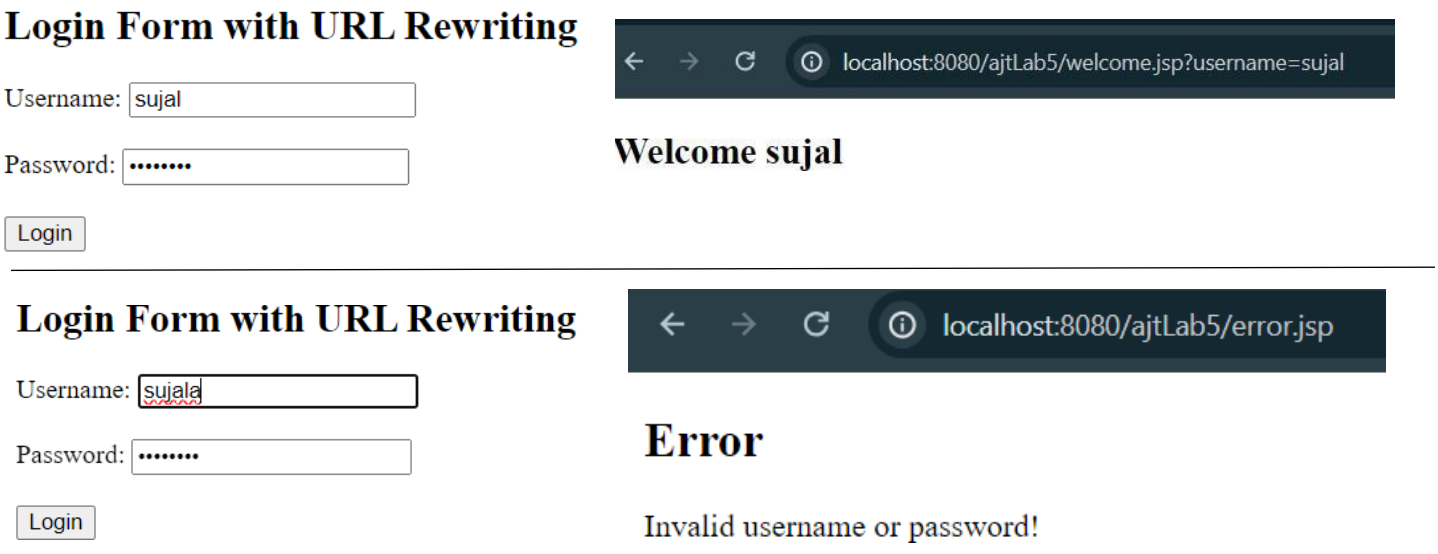
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <servlet>
    <display-name>Lab5</display-name>
    <servlet-name>Lab5</servlet-name>
    <servlet-class>com.java.Lab5</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Lab5</servlet-name>
    <url-pattern>/Lab5</url-pattern>
  </servlet-mapping>
</web-app>
```

OUTPUT :--

File Structure :



Web Output :



4. Using HttpSession :

CODE :-

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>
</head>
<body>
    <h2>Login Form with HttpSession</h2>
    <form action="Lab5" 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>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

Lab5.java

```
package com.java;

import jakarta.servlet.ServletException;

import jakarta.servlet.http.*;

import java.io.IOException;

import java.sql.*;

public class Lab5 extends HttpServlet {

	private static final long serialVersionUID = 1L;

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

		try {

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

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

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

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

			PreparedStatement stmt = conn.prepareStatement("SELECT * FROM login_info WHERE username=? AND password=?");

			stmt.setString(1, username);

			stmt.setString(2, password);

			ResultSet rs = stmt.executeQuery();

			if (rs.next()) {

				HttpSession session = request.getSession(true);

				session.setAttribute("username", username);

				response.sendRedirect("welcome.jsp");

			} else {

				response.sendRedirect("error.jsp");

			}
```

```
                } catch (Exception e) {
                        e.printStackTrace();
                        System.out.println(e);
                }

        }
}
```

welcome.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome with HttpSession</title>
</head>
<body>
    <h2>Welcome <%= session.getAttribute("username") %></h2>
</body>
</html>
```

error.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
        pageEncoding="UTF-8"%>
<!DOCTYPE html>
```

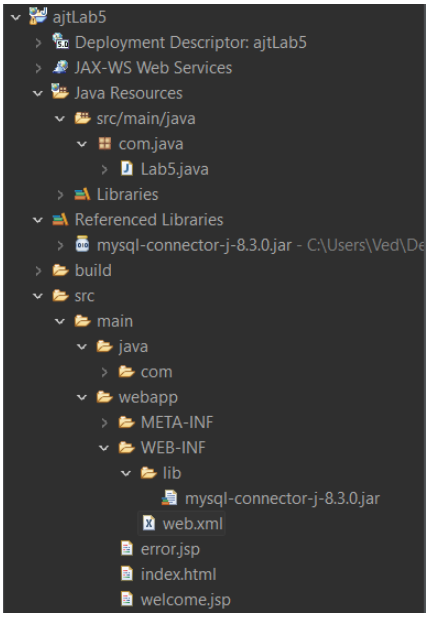
```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Error </title>
</head>
<body>
        <h2>Error</h2>
        <p>Invalid username or password!</p>
</body>
</html>
```

web.xml

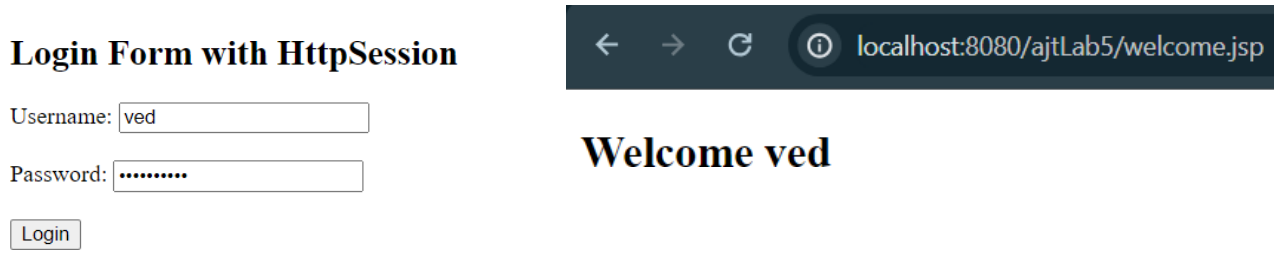
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <servlet>
    <display-name>Lab5</display-name>
    <servlet-name>Lab5</servlet-name>
    <servlet-class>com.java.Lab5</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Lab5</servlet-name>
    <url-pattern>/Lab5</url-pattern>
  </servlet-mapping>
</web-app>
```

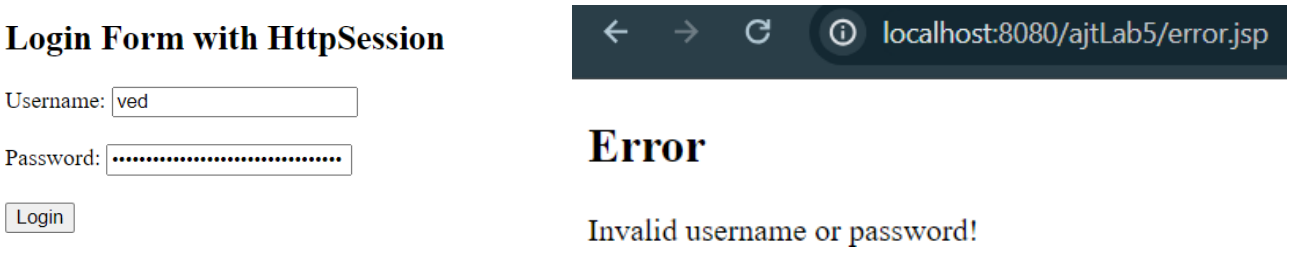
OUTPUT :--

File Structure :



Web Output :





Login Database :

```
1 •   SELECT * FROM logindb.login_info;
```

Result Grid | Filter Rows: | Edit:

id	username	password
1	ved	helloworld
2	megh	megh@123
3	sujal	sujal057
4	aaditya	adityaisakid
NULL	NULL	

Lab 5

Aim: Implement login form and perform session management using different methods.

LOGIN using different session methods:

(1) HTTP session management:

CODE:
index.html:

```
<!DOCTYPE html>
<html>
<head>
   <title>Login</title>
</head>
<body>
   <h2>Login</h2>
   <form action="LoginServlet" method="post">
      Username: <input type="text" name="username"><br>
      Password: <input type="password" name="password"><br>
      <input type="submit" value="Login">
   </form>
</body>
</html>
```

LoginServlet.java

```
package com.java;

import java.io.*;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.*;
import java.sql.*;

public class LoginServlet extends HttpServlet {

        private static final long serialVersionUID = 1L;
        private static final String JDBC_URL = "jdbc:mysql://localhost:3306/LoginDB";
   private static final String DB_USERNAME = "root";
   private static final String DB_PASSWORD = "Root";
```

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

      try {
         Connection connection = DriverManager.getConnection(JDBC_URL,
DB_USERNAME, DB_PASSWORD);
         PreparedStatement preparedStatement = connection.prepareStatement("SELECT *
FROM users WHERE username = ? AND password = ?");
         preparedStatement.setString(1, username);
         preparedStatement.setString(2, password);
         ResultSet resultSet = preparedStatement.executeQuery();

         if (resultSet.next()) {
            HttpSession session = request.getSession();
            session.setAttribute("username", username);
            response.sendRedirect("Welcome.jsp");
         } else {
            response.sendRedirect("error.html");
         }

         resultSet.close();
         preparedStatement.close();
         connection.close();
      } catch (SQLException e) {
         e.printStackTrace();
         response.sendRedirect("error.html");
      }
   }
}
```

Welcome.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
   <title>Welcome</title>
</head>
<body>
   <h2>Welcome, <%= session.getAttribute("username") %></h2>
```

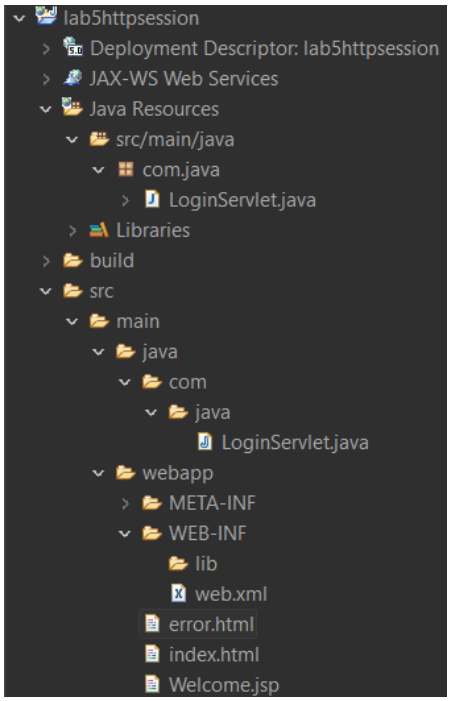
```
    <p>You are logged in!</p>
</body>
</html>
```

error.html:

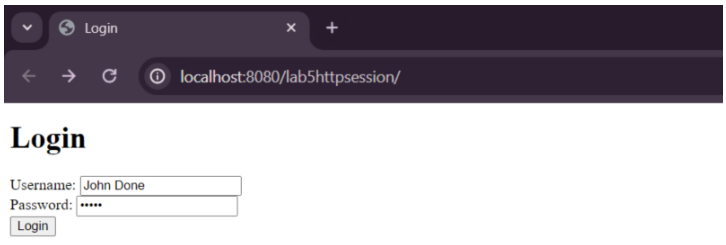
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
        <h3> Something went wrong. Try again later. </h3>
        <a href="index.html">try again</a>
</body>
</html>
```

OUTPUT:

File Structure:

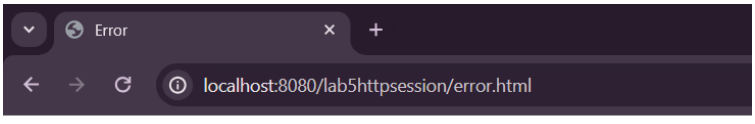


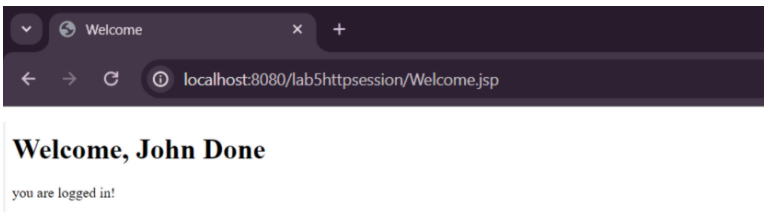
Web Output:



Welcome.jsp:

error.html





(2) Cookies:

CODE:

index.html:

```
<!DOCTYPE html>
<html>
<head>
   <title>Login</title>
</head>
<body>
   <h2>Login</h2>
   <form action="LoginServlet" method="post">
      Username: <input type="text" name="username"><br>
      Password: <input type="password" name="password"><br>
      <input type="submit" value="Login">
   </form>
</body>
</html>
```

LoginServlet.java:

```
package com.java;
import java.io.*;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.sql.*;

public class LoginServlet extends HttpServlet {

        private static final long serialVersionUID = 1L;
        private static final String JDBC_URL = "jdbc:mysql://localhost:3306/LoginDB";
   private static final String DB_USERNAME = "root";
   private static final String DB_PASSWORD = "Root";

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

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

        try {
            Connection connection = DriverManager.getConnection(JDBC_URL,
DB_USERNAME, DB_PASSWORD);
            PreparedStatement preparedStatement = connection.prepareStatement("SELECT *
FROM users WHERE username = ? AND password = ?");
            preparedStatement.setString(1, username);
            preparedStatement.setString(2, password);
            ResultSet resultSet = preparedStatement.executeQuery();

            if (resultSet.next()) {
                Cookie cookie = new Cookie("username", username);
                cookie.setMaxAge(60 * 60 * 24); // Cookie lasts for 24 hours
                response.addCookie(cookie);
                response.sendRedirect("welcome.jsp");
            } else {
                response.sendRedirect("index.html");
            }

            resultSet.close();
            preparedStatement.close();
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
            response.sendRedirect("error.html");
        }
    }
}
```

Welcome.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
```

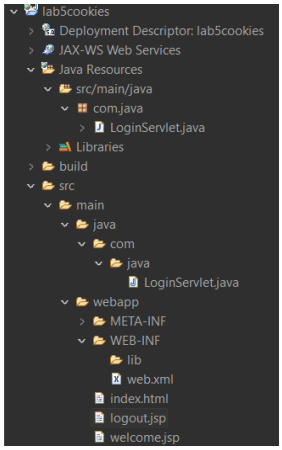
```
<html>
<head>
   <title>Welcome</title>
</head>
<body>
   <h2>Welcome, <%= request.getCookies()[0].getValue() %></h2>
   <p>You are logged in!</p>
   <a href="logout.jsp">Logout</a>
</body>
</html>
```

logout.jsp:

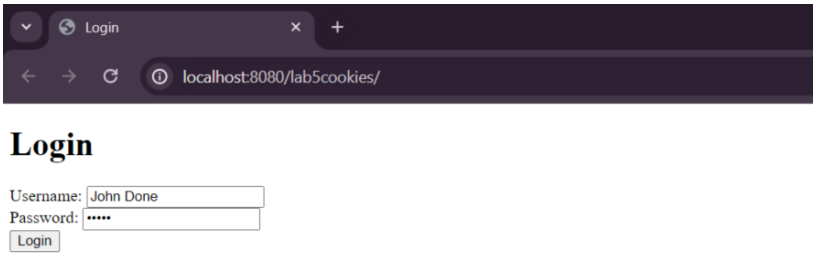
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
   <title>Logout</title>
</head>
<body>
   <h2>Logout</h2>
   <p>You have been logged out successfully.</p>
   <p><a href="index.html">Click here to login again</a></p>
</body>
</html>
```

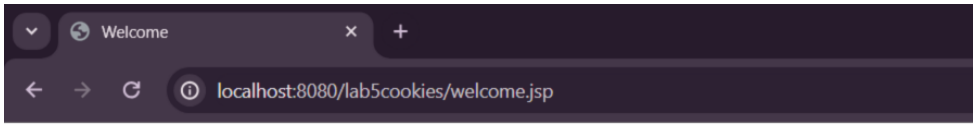
OUTPUT:

File Structure:



Web Output:

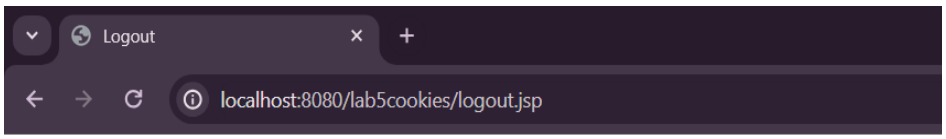




Welcome, John Done

you are logged in!

Logout



Logout

You have been logged out successfully.

Click here to login again

(3) Hidden Object:

CODE:

index.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
</head>
<body>
    <h2>Login</h2>
    <form action="LoginServlet" method="post">
        <input type="hidden" name="action" value="login">
        Username: <input type="text" name="username"><br>
        Password: <input type="password" name="password"><br>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

LoginServlet.java:

```
package com.java;
import java.io.*;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.sql.*;

public class LoginServlet extends HttpServlet {

        private static final long serialVersionUID = 1L;
        private static final String JDBC_URL = "jdbc:mysql://localhost:3306/LoginDB";
   private static final String DB_USERNAME = "root";
   private static final String DB_PASSWORD = "Root";

   protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
      String action = request.getParameter("action");

      if ("login".equals(action)) {
         String username = request.getParameter("username");
         String password = request.getParameter("password");

         try {
            Connection connection = DriverManager.getConnection(JDBC_URL,
DB_USERNAME, DB_PASSWORD);
            PreparedStatement preparedStatement = connection.prepareStatement("SELECT *
FROM users WHERE username = ? AND password = ?");
            preparedStatement.setString(1, username);
            preparedStatement.setString(2, password);
            ResultSet resultSet = preparedStatement.executeQuery();

            if (resultSet.next()) {
               // Create hidden form field for session management
               String hiddenField = "<input type=\"hidden\" name=\"username\" value=\"" +
username + "\">";
```

```
                request.setAttribute("hiddenField", hiddenField);
                RequestDispatcher rd = request.getRequestDispatcher("welcome.jsp");
                rd.forward(request, response);
            } else {
                response.sendRedirect("error.html");
            }

            resultSet.close();
            preparedStatement.close();
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
            response.sendRedirect("error.html");
        }
    } else {
        // Handle invalid action
        response.sendRedirect("index.html");
    }
  }
}
```

Welcome.jsp:

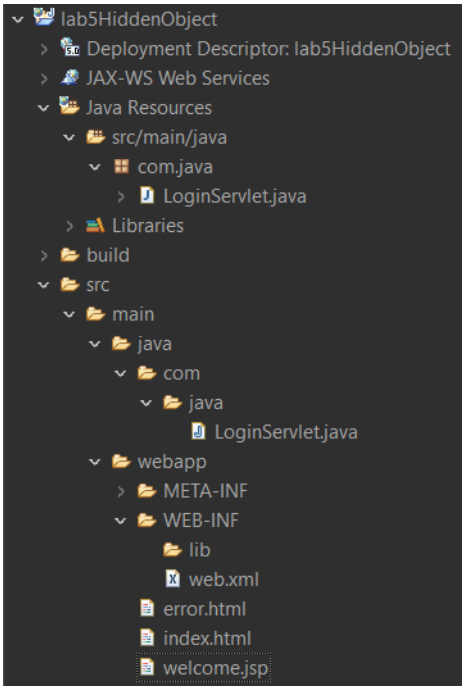
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
  pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
  <title>Welcome</title>
</head>
<body>
  <h2>Welcome, <%= request.getParameter("username") %></h2>
  <p>You are logged in!</p>
</body>
</html>
```

error.html:

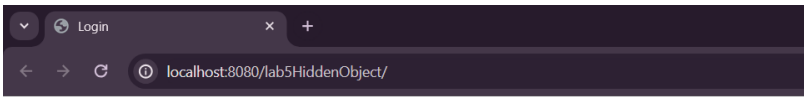
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>error</title>
</head>
<body>
        <h3> Something went wrong. Try again later. </h3>
        <a href="index.html">try again</a>
</body>
</html>
```

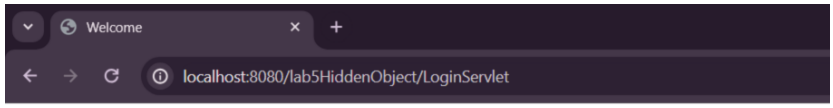
OUTPUT:

File Structure:



Web Output:





(4) URL Rewriting:

CODE:

index.html:

```
<!DOCTYPE html>
<html>
<head>
   <title>Login</title>
</head>
<body>
   <h2>Login</h2>
   <form action="LoginServlet" method="post">
      Username: <input type="text" name="username"><br>
      Password: <input type="password" name="password"><br>
      <input type="submit" value="Login">
   </form>
</body>
</html>
```

LoginServlet.java:

```
package com.java;
import java.io.*;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.sql.*;

public class LoginServlet extends HttpServlet {

        private static final long serialVersionUID = 1L;
        private static final String JDBC_URL = "jdbc:mysql://localhost:3306/LoginDB";
   private static final String DB_USERNAME = "root";
   private static final String DB_PASSWORD = "Root";

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

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

        try {
            Connection connection = DriverManager.getConnection(JDBC_URL,
DB_USERNAME, DB_PASSWORD);
            PreparedStatement preparedStatement = connection.prepareStatement("SELECT *
FROM users WHERE username = ? AND password = ?");
            preparedStatement.setString(1, username);
            preparedStatement.setString(2, password);
            ResultSet resultSet = preparedStatement.executeQuery();

            if (resultSet.next()) {
                // Append username to the URL for URL rewriting session management
                String encodedUsername = response.encodeURL(username);
                response.sendRedirect("welcome.jsp?username=" + encodedUsername);
            } else {
                response.sendRedirect("error.html");
            }

            resultSet.close();
            preparedStatement.close();
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
            response.sendRedirect("error.html");
        }
    }
}
```

Welcome.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
```

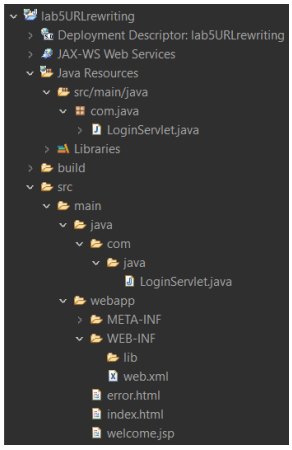
```
<head>
    <title>Welcome</title>
</head>
<body>
    <h2>Welcome, <%= request.getParameter("username") %></h2>
    <p>You are logged in!</p>
</body>
</html>
```

error.html:

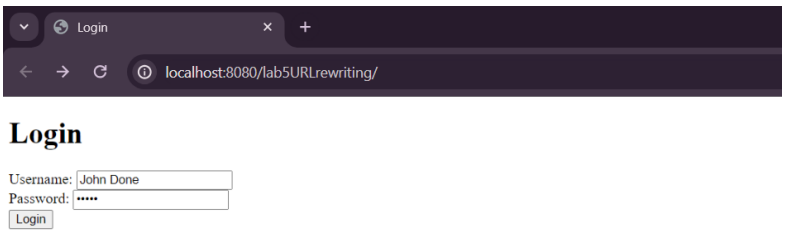
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
        <h3> Something went wrong. Try again later. </h3>
        <a href="index.html">try again</a>
</body>
</html>
```

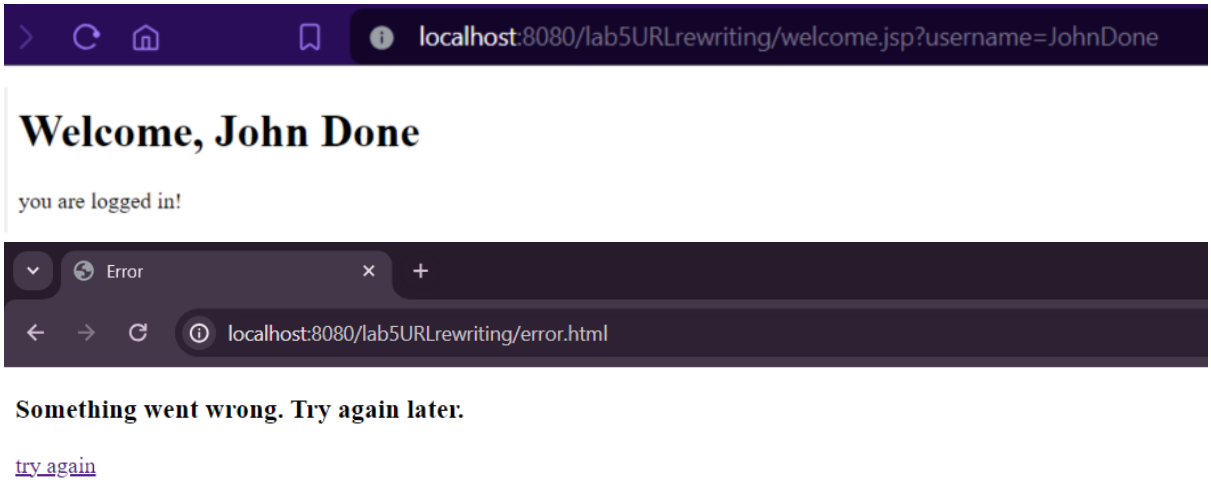
OUTPUT:

File Structure:



Web Output:





(5) RequestDispatcher:

CODE:

index.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
</head>
<body>
    <h2>Login</h2>
    <form action="LoginServlet" method="post">
        Username: <input type="text" name="username"><br>
        Password: <input type="password" name="password"><br>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

LoginServlet.java

```
package com.java;

import java.io.*;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.sql.*;

public class LoginServlet extends HttpServlet {

        private static final long serialVersionUID = 1L;
        private static final String JDBC_URL = "jdbc:mysql://localhost:3306/LoginDB";
   private static final String DB_USERNAME = "root";
   private static final String DB_PASSWORD = "Root";

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

      try {
         Connection connection = DriverManager.getConnection(JDBC_URL,
DB_USERNAME, DB_PASSWORD);
         PreparedStatement preparedStatement = connection.prepareStatement("SELECT *
FROM users WHERE username = ? AND password = ?");
         preparedStatement.setString(1, username);
         preparedStatement.setString(2, password);
         ResultSet resultSet = preparedStatement.executeQuery();

         if (resultSet.next()) {
            // Set session attribute and forward to welcome page using RequestDispatcher
            HttpSession session = request.getSession();
            session.setAttribute("username", username);
            RequestDispatcher rd = request.getRequestDispatcher("welcome.jsp");
            rd.forward(request, response);
         } else {
```

```
            response.sendRedirect("error.html");
        }

        resultSet.close();
        preparedStatement.close();
        connection.close();
      } catch (SQLException e) {
        e.printStackTrace();
        response.sendRedirect("error.html");
      }
   }
}
```

Welcome.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
   <title>Welcome</title>
</head>
<body>
   <h2>Welcome, <%= request.getSession().getAttribute("username") %></h2>
   <p>You are logged in!</p>
</body>
</html>
```

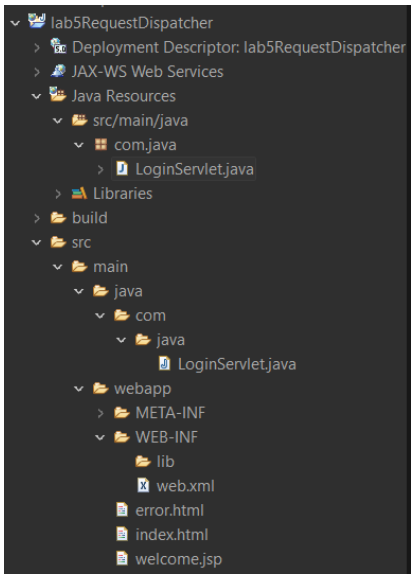
error.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Error</title>
</head>
<body>
        <h3> Something went wrong. Try again later. </h3>
```

```
        <a href="index.html">try again</a>
</body>
</html>
```

OUTPUT:

File Structure:



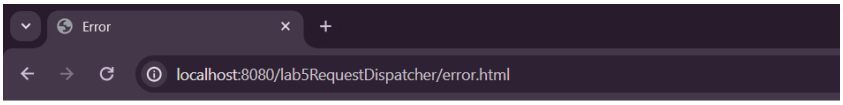
Web Output:

Login

Username: John Done
Password: •••••
Login

Welcome, John Done

you are logged in!



Something went wrong. Try again later.

try again

PRACTICAL 6

AIM : Develop a web application for a bank system which performs the following task :

1. Create database and master table for bank
2. Perform insert, update and delete operation.
3. Validate the attributes.

CODE :--

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Banking System</title>
   <style>
        body{
                background-color:red;
        }
        header{
                text-align: center;
                margin-bottom:2rem;
                padding:1rem;
                background-color:blue;
                border-radius: 0.5rem;
```

```
            box-shadow: rgba(50, 50, 105, 0.15) 0px 2px 5px 0px, rgba(0, 0, 0, 0.05) 0px 1px
1px 0px;
        }
          main{
            display:flex;
            justify-content: center;
          }
      section{
        margin: 2rem;
        padding: 2rem;
        background-color: #40A2D8cc;
        border-radius: 0.5rem;
        box-shadow: rgba(50, 50, 105, 0.15) 0px 2px 5px 0px, rgba(0, 0, 0, 0.05) 0px 1px 1px
0px;

      }
      input{
        border:0;
        border-radius:0.5rem;
        width:100%;
        height:1.5rem;
      }
      h2{
        text-align:center;
      }

  </style>
</head>
```

```
<body>

    <header>
        <h1>lab 6 - Banking System</h1>
    </header>

    <main>
        <section id="add-account">
            <h2>Add New Account</h2>
            <form id="add-account-form" action="AccountServlet?action=insert" method="POST">
                <label for="accountNumber">Account Number:</label><br>
                <input type="text" id="accountNumber" name="accountNumber" required><br>
                <label for="accountHolder">Account Holder:</label><br>
                <input type="text" id="accountHolder" name="accountHolder" required><br>
                <label for="balance">Balance:</label><br>
                <input type="number" id="balance" name="balance" required><br><br>
                <button type="submit">Add Account</button>
            </form>
        </section>

        <section id="edit-account">
            <h2>Edit Account</h2>
            <form id="edit-account-form" action="AccountServlet?action=update"
method="POST">
                    <label for="edit-accountId">Account Id:</label><br>
                <input type="text" id="edit-accountId" name="accountId" required><br>
                <label for="edit-accountNumber">Account Number:</label><br>
```

```
            <input type="text" id="edit-accountNumber" name="accountNumber" required><br>
            <label for="edit-accountHolder">Account Holder:</label><br>
            <input type="text" id="edit-accountHolder" name="accountHolder" required><br>
            <label for="edit-balance">Balance:</label><br>
            <input type="number" id="edit-balance" name="balance" required><br><br>
            <button type="submit">Update Account</button>
        </form>
    </section>
    <section id="delete-account">
        <h2>Delete Account</h2>
        <form id="delete-account-form" action="AccountServlet?action=delete"
method="POST">
                <label for="delete-accountId">Account Id:</label><br>
            <input type="text" id="delete-accountId" name="accountId" required><br><br>
            <button type="submit">Delete Account</button>
        </form>
    </section>
    <section id="account-list">
        <h2>List Accounts</h2>
        <form id="edit-account-form" action="AccountServlet?action=list" method="POST">
            <button type="submit">View</button><br>
        </form>
    </section>
  </main>
</body> </html>
```

AccountServlet.java

```
package com.java;
import jakarta.servlet.*;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;
public class AccountServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private Connection conn;
    public void init() {
        String jdbcUrl = "jdbc:mysql://localhost:3306/bank_system";
        String jdbcUsername = "root";
        String jdbcPassword = "Krupal@123 ";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(jdbcUrl, jdbcUsername, jdbcPassword);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        doGet(request, response);
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String action = request.getParameter("action");
        if (action == null) {
            action = "list";
        }
        switch (action) {
            case "list":
                listAccounts(request, response);
                break;
            case "insert":
                insertAccount(request, response);
                break;
            case "update":
                updateAccount(request, response);
                break;
            case "delete":
                deleteAccount(request, response);
                break;
            default:
                listAccounts(request, response);
        }
    }
```

```
    private void listAccounts(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        List<Account> accounts = new ArrayList<>();
        String sql = "SELECT * FROM accounts";
        try {
            Statement statement = conn.createStatement();
            ResultSet resultSet = statement.executeQuery(sql);
            while (resultSet.next()) {
                int accountId = resultSet.getInt("account_id");
                String accountNumber = resultSet.getString("account_number");
                String accountHolder = resultSet.getString("account_holder");
                double balance = resultSet.getDouble("balance");
                Account account = new Account(accountId, accountNumber, accountHolder, balance);
                accounts.add(account);
            }
            StringBuilder tableHtml = new StringBuilder();
            tableHtml.append("<table border='1'>");
            tableHtml.append("<tr><th>Account ID</th><th>Account Number</th><th>Account
Holder</th><th>Balance</th></tr>");
            for (Account account : accounts) {
                tableHtml.append("<tr>");
                tableHtml.append("<td>").append(account.getAccountId()).append("</td>");
          tableHtml.append("<td>").append(account.getAccountNumber()).append("</td>");
          tableHtml.append("<td>").append(account.getAccountHolder()).append("</td>");
                tableHtml.append("<td>").append(account.getBalance()).append("</td>");
                tableHtml.append("</tr>");
            }
```

```
            tableHtml.append("</table>");
            request.setAttribute("accountTable", tableHtml.toString());
            RequestDispatcher dispatcher = request.getRequestDispatcher("accounts.jsp");
            dispatcher.forward(request, response);
        } catch (SQLException e) {
            e.printStackTrace();
            response.sendRedirect("error.jsp");
        }
    }
    private void insertAccount(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String accountNumber = request.getParameter("accountNumber");
        String accountHolder = request.getParameter("accountHolder");
        double balance = Double.parseDouble(request.getParameter("balance"));
        String sql = "INSERT INTO accounts (account_number, account_holder, balance)
VALUES (?, ?, ?)";
        try {
            PreparedStatement preparedStatement = conn.prepareStatement(sql);
            preparedStatement.setString(1, accountNumber);
            preparedStatement.setString(2, accountHolder);
            preparedStatement.setDouble(3, balance);
            int rowsInserted = preparedStatement.executeUpdate();
            if (rowsInserted > 0) {
                    PrintWriter out = response.getWriter();
               out.println("Account inserted successfully!");
            }
        } catch (SQLException e) {
```

```
            e.printStackTrace();


            response.sendRedirect("error.jsp");
        }
    }
    private void updateAccount(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        int accountId = Integer.parseInt(request.getParameter("accountId"));
        String accountNumber = request.getParameter("accountNumber");
        String accountHolder = request.getParameter("accountHolder");
        double balance = Double.parseDouble(request.getParameter("balance"));
        String sql = "UPDATE accounts SET account_number=?, account_holder=?, balance=?
WHERE account_id=?";
        try {
            PreparedStatement statement = conn.prepareStatement(sql);
            statement.setString(1, accountNumber);
            statement.setString(2, accountHolder);
            statement.setDouble(3, balance);
            statement.setInt(4, accountId);
            int rowsUpdated = statement.executeUpdate();
            if (rowsUpdated > 0) {
                    PrintWriter out = response.getWriter();
               out.println("Account updated successfully!");
            }
        } catch (SQLException e) {
            e.printStackTrace();
            response.sendRedirect("error.jsp");
        }
```

```
    }


    private void deleteAccount(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        int accountId = Integer.parseInt(request.getParameter("accountId"));
        String sql = "DELETE FROM accounts WHERE account_id=?";
        try {
            PreparedStatement statement = conn.prepareStatement(sql);
            statement.setInt(1, accountId);
            int rowsDeleted = statement.executeUpdate();
            if (rowsDeleted > 0) {
                    PrintWriter out = response.getWriter();
                out.println("Account deleted successfully!");
            }
        } catch (SQLException e) {
            e.printStackTrace();
            response.sendRedirect("error.jsp");
        }
    }
    public void destroy() {
        try {
            if (conn != null && !conn.isClosed()) {
                conn.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
```

```
        }
    }
}
```

Account.java

```
package com.java;
public class Account {
    private int accountId;
    private String accountNumber;
    private String accountHolder;
    private double balance;
    public Account(int accountId, String accountNumber, String accountHolder, double balance) {
        this.accountId = accountId;
        this.accountNumber = accountNumber;
        this.accountHolder = accountHolder;
        this.balance = balance;
    }
    public int getAccountId() {
        return accountId;
    }
    public void setAccountId(int accountId) {
        this.accountId = accountId;
    }
    public String getAccountNumber() {
        return accountNumber;
    }
    public void setAccountNumber(String accountNumber) {
        this.accountNumber = accountNumber;
    }
```

```
    public String getAccountHolder() {
        return accountHolder;
    }

    public void setAccountHolder(String accountHolder) {
        this.accountHolder = accountHolder;
    }
    public double getBalance() {
        return balance;
    }
    public void setBalance(double balance) {
        this.balance = balance;
    }
}
```

accounts.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Accounts List</title>
    <style type="text/css">
            body{
                    background-color: #40A2D8cc;
            }
    </style>
</head>
<body>
    <h1>Accounts List</h1>
```

```
   <%= request.getAttribute("accountTable") %>
</body>
</html>
```

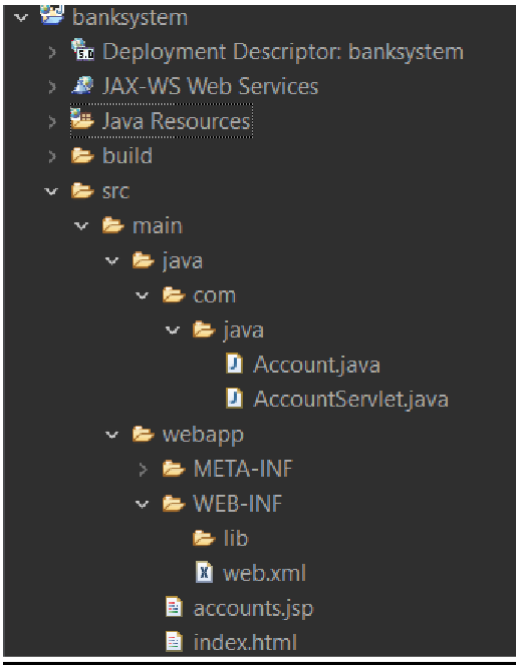
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="https://jakarta.ee/xml/ns/jakartaee"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd" id="WebApp_ID" version="6.0">
  <display-name>ajtlab6</display-name>
  <servlet>
    <display-name>AccountServlet</display-name>
    <servlet-name>AccountServlet</servlet-name>
    <servlet-class>com.java.AccountServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>AccountServlet</servlet-name>

    <url-pattern>/AccountServlet</url-pattern>
  </servlet-mapping>
  <servlet>
    <description></description>
    <display-name>Account</display-name>
    <servlet-name>Account</servlet-name>
    <servlet-class>com.java.Account</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Account</servlet-name>
```

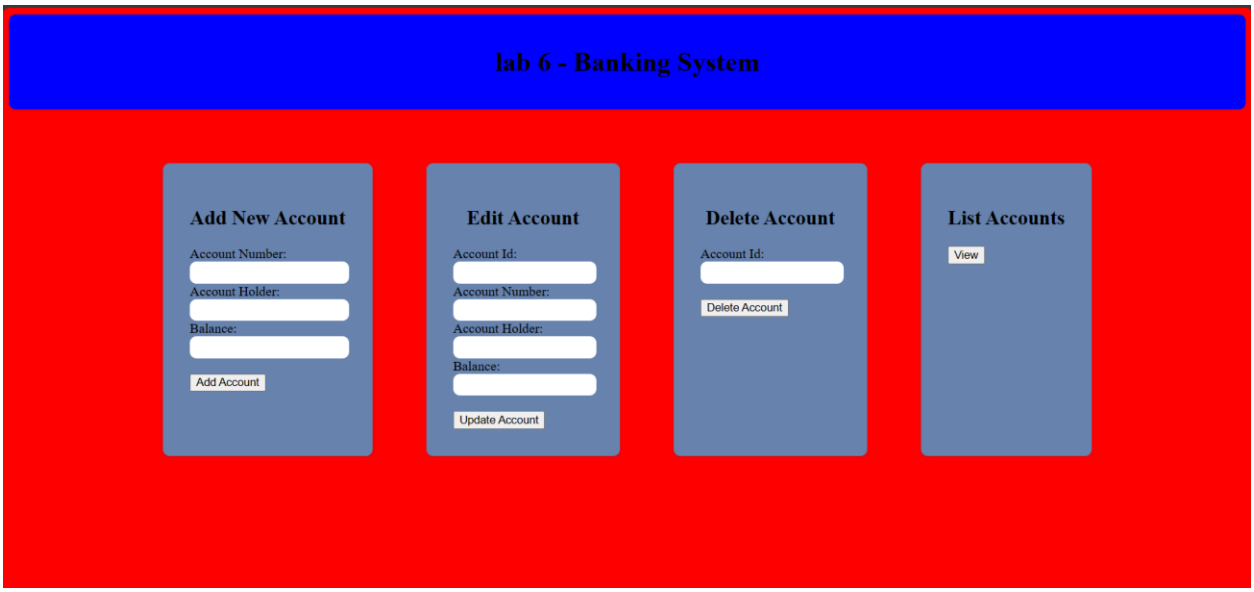
```
    <url-pattern>/Account</url-pattern>
  </servlet-mapping>
</web-app>
```

OUTPUT :--

File Structure



Web Output :


lab 6 - Banking System
Add New Account
Account Number:
Account Holder:
Balance:
Add Account
Edit Account
Account Id:
Account Number:
Account Holder:
Balance:
Update Account
Delete Account
Account Id:
Delete Account
List Accounts
View

Insert :


Add New Account
Account Number:
12345
Account Holder:
Anshu
Balance:
15234
Add Account


Add New Account
Account Number:
123456
Account Holder:
harshil
Balance:
24578
Add Account

Accounts List

Account ID	Account Number	Account Holder	Balance
5	12345	Anshu	15234.0
6	123456	harshil	24578.0

Update:

Edit Account

Account Id:
5
Account Number:
12345
Account Holder:
Anshu
Balance:
78787

Update Account

Accounts List

Account ID	Account Number	Account Holder	Balance
5	12345	Anshu	78787.0
6	123456	harshil	24578.0

Delete:


Delete Account
Account Id:
5
Delete Account


Accounts List
Account ID
Account Number
Account Holder
Balance
6
123456
harshil
24578.0

Database :

```
1 •  select * from bank_system.accounts;
2
```

Result Grid | Filter Rows: | Edit: | Export/

Account_ID	Account_number	Account_Holder	Balance
6	123456	harshil	24578
NULL	NULL	NULL	NULL

PRACTICAL 7

AIM : Write down the program for testing the include action and forward action for servlet collaboration.

CODE :--

Index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Lab 7</title>
</head>
<body>
	<h2>lab 7|IU182</h2>
	<form action="Lab7" method="Get">
		<input type="submit" value="SUBMIT">
	</form>
</body>
</html>
```

Lab7.java

```
package com.kru;
import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
```

```
import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;

public class Lab7 extends HttpServlet {
        private static final long serialVersionUID = 1L;
        @Override
   protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
      resp.setContentType("text/html");

      RequestDispatcher includeDispatcher = req.getRequestDispatcher("/TargetServlet");
      includeDispatcher.include(req, resp);

      RequestDispatcher forwardDispatcher = req.getRequestDispatcher("/TargetServlet");
      forwardDispatcher.forward(req, resp);
   }

}
```

TargetServlet.java

```
package com.kru;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
```

```
import java.io.PrintWriter;
public class TargetServlet extends HttpServlet {
        private static final long serialVersionUID = 1L;
   @Override
   protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
      resp.setContentType("text/html");
      PrintWriter out = resp.getWriter();
      out.println("<html><body>");
      out.println("<h1>This is the Target Servlet</h1>");
      out.println("</body></html>");
   }
}
```

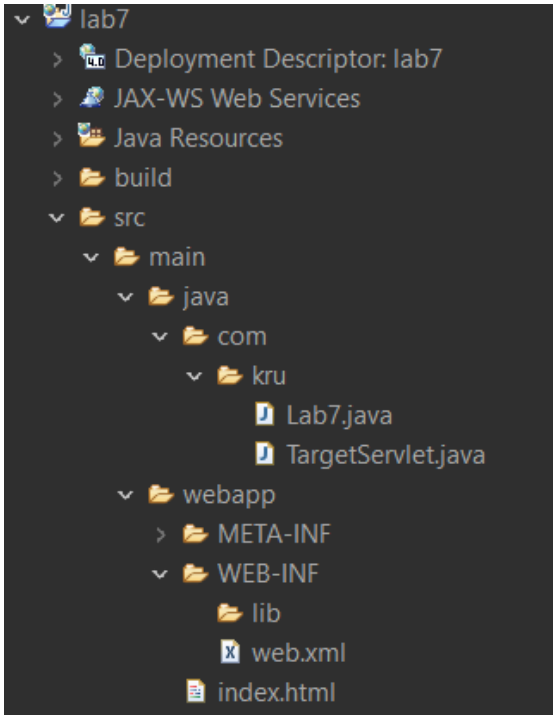
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <display-name>lab7</display-name>
  <servlet>
    <display-name>Lab7</display-name>
    <servlet-name>Lab7</servlet-name>
    <servlet-class>com.kru.Lab7</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Lab7</servlet-name>
    <url-pattern>/Lab7</url-pattern>
  </servlet-mapping>
```

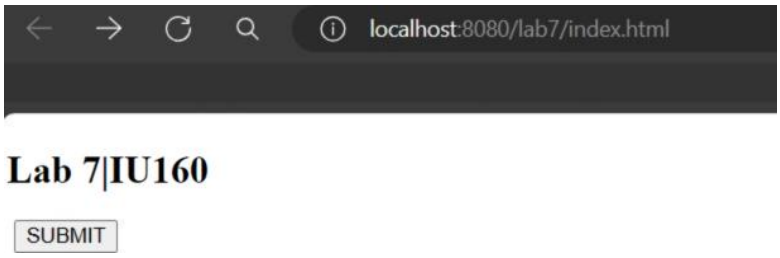
```
  <servlet>
    <description></description>
    <display-name>TargetServlet</display-name>
    <servlet-name>TargetServlet</servlet-name>
    <servlet-class>com.kru.TargetServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>TargetServlet</servlet-name>
    <url-pattern>/TargetServlet</url-pattern>
  </servlet-mapping>
</web-app>
```

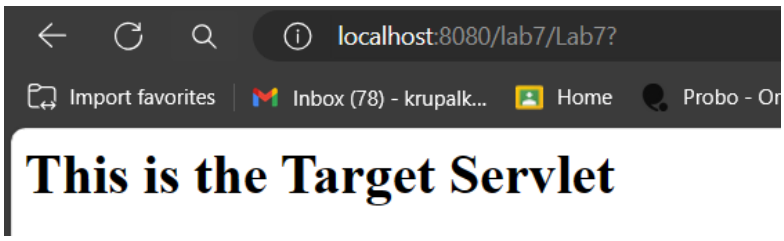
OUTPUT :--

File Structure



Web Output





PRACTICAL 8

AIM : Write down the program for testing the include and forward action tag in jsp.

CODE :--

Index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Lab 8</title>
</head>
<body>
	<h1>Krupal Patel</h1>
	<form action="Lab8.jsp">
		<input type="submit">
	</form>
</body>
</html>
```

Lab8.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
%>
<html>
<head>
   <title>Lab 8</title>
```

```
</head>
<body>
   <h1>Lab 8|IU182 </h1>
   <jsp:include page="header.jsp" />
   <jsp:forward page="result.jsp" />
</body>
</html>
```

header.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
%>
<html>
<head>
   <title>Header</title>
</head>
<body>
   <h1>This is the header</h1>
</body>
</html>
```

result.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
%>
<html>
<head>
```

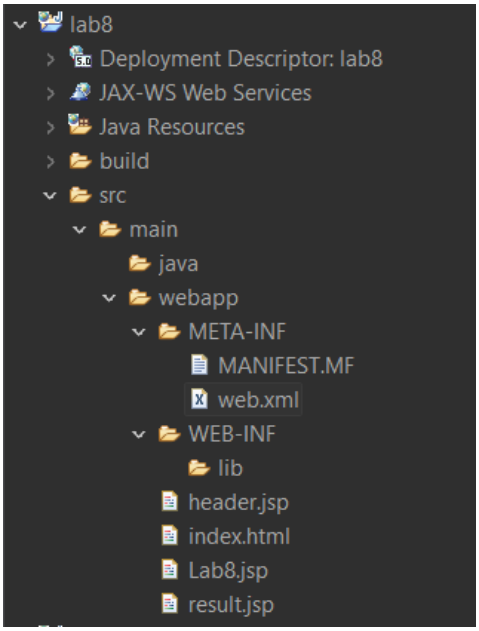
```
    <title>Lab 8</title>
</head>
<body>
    <h1>Lab 8|IU182</h1>
    <p>This is the result page.</p>
</body>
</html>
```

web.xml

```
<web-app>
  <display-name>ajtLab8</display-name>
  <servlet>
    <display-name>Lab8</display-name>
    <servlet-name>Lab8</servlet-name>
    <servlet-class>iu.ved.Lab8</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Lab8</servlet-name>
    <url-pattern>/Lab8</url-pattern>
  </servlet-mapping>
</web-app>
```

OUTPUT :--

File Structure



Web Output

PRACTICAL 9

AIM : Create database of student subject-wise data and retrieve all data using JSP and generate xml structure along with DTD and XML Schema definition Write down a program which demonstrates the core tag of JSTL

CODE :--

Student.jsp :-

```
<%@ page import="java.sql.*" %>
<%@ page contentType="text/xml;charset=UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page trimDirectiveWhitespaces="true" %>
<%@ page import="java.util.List" %>
<%@ page import="java.util.ArrayList" %>
<%
   Connection conn = null;
   try {
      Class.forName("com.mysql.cj.jdbc.Driver");
      conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/students", "root",
"Krupal@123");
   } catch (Exception e) {
      e.printStackTrace();
   }

   List<Integer> studentIds = new ArrayList<>();
   List<String> studentNames = new ArrayList<>();
   try {
      Statement stmt = conn.createStatement();
      ResultSet rs = stmt.executeQuery("SELECT id, name FROM students");
      while (rs.next()) {
         studentIds.add(rs.getInt("id"));
         studentNames.add(rs.getString("name"));
      }
      rs.close();
      stmt.close();
   } catch (Exception e) {
      e.printStackTrace();
   }

   List<String> subjectNames = new ArrayList<>();
```

```
    List<String> subjectGrades = new ArrayList<>();
    try {
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT subject_name, grade FROM subjects");
        while (rs.next()) {
            subjectNames.add(rs.getString("subject_name"));
            subjectGrades.add(rs.getString("grade"));
        }
        rs.close();
        stmt.close();
    } catch (Exception e) {
        e.printStackTrace();
    }

    conn.close();

    request.setAttribute("studentIds", studentIds);
    request.setAttribute("studentNames", studentNames);
    request.setAttribute("subjectNames",  subjectNames);
    request.setAttribute("subjectGrades", subjectGrades);
%>

<?xml version="1.0" encoding="UTF-8"?>
<students xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="students.xsd">
    <c:forEach var="i" begin="0" end="${studentIds.size() - 1}">
        <student>
            <id><c:out value="${studentIds[i]}" /></id>
            <name><c:out value="${studentNames[i]}" /></name>
            <subjects>
                <c:forEach var="subjectIndex" begin="0" end="${subjectNames.size() - 1}">
                    <c:if test="${subjectIndex % studentIds.size() == i}">
                        <subject>
                            <name><c:out value="${subjectNames[subjectIndex]}" /></name>
                            <grade><c:out value="${subjectGrades[subjectIndex]}" /></grade>
                        </subject>
                    </c:if>
                </c:forEach>
            </subjects>
        </student>
    </c:forEach>
</students>
```

Student.dtd

```
<!ELEMENT students (student+)>
<!ELEMENT student (id, name, age, subjects)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT subjects (subject+)>
<!ELEMENT subject (name, grade)>
<!ELEMENT subject_name (#PCDATA)>
<!ELEMENT grade (#PCDATA)>
```

Student.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="students">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="student" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="id" type="xs:int"/>
              <xs:element name="name" type="xs:string"/>
              <xs:element name="subjects">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="subject" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
```

```
                <xs:element name="name" type="xs:string"/>
                <xs:element name="grade" type="xs:string"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:element>
</xs:schema>
```

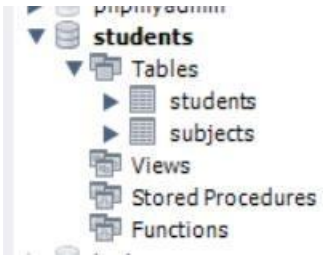
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="https://jakarta.ee/xml/ns/jakartaee" xmlns:jsp="http://java.sun.com/xml/ns/javaee/jsp"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee https://jakarta.ee/xml/ns/jakartaee/web-
app_5_0.xsd" id="WebApp_ID" version="5.0">
  <display-name>jstl prac</display-name>
  <jsp-config>
    <taglib>
      <taglib-uri>http://java.sun.com/jsp/jstl/core</taglib-uri>
      <taglib-location>/WEB-INF/tld/c.tld</taglib-location>
    </taglib>
  </jsp-config>
```

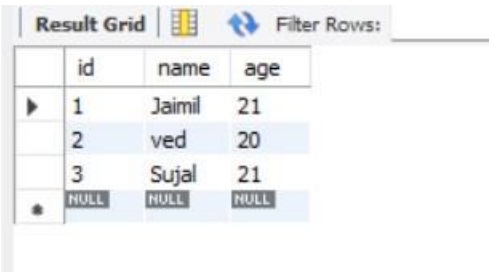
</web-app>

OUTPUT :--

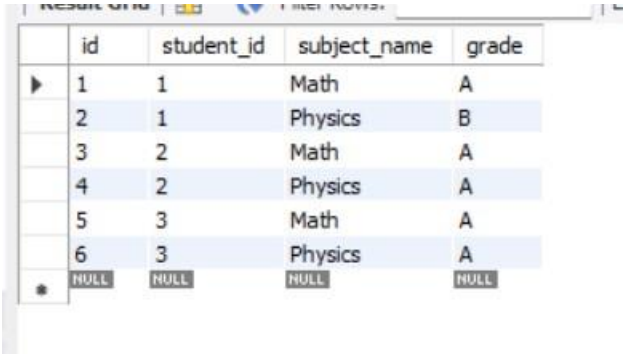
Database:



Student Table:

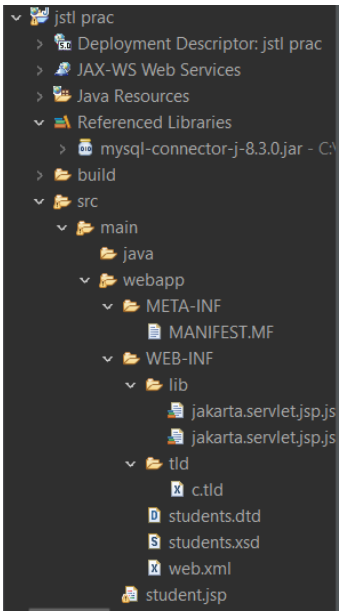


Subject table:

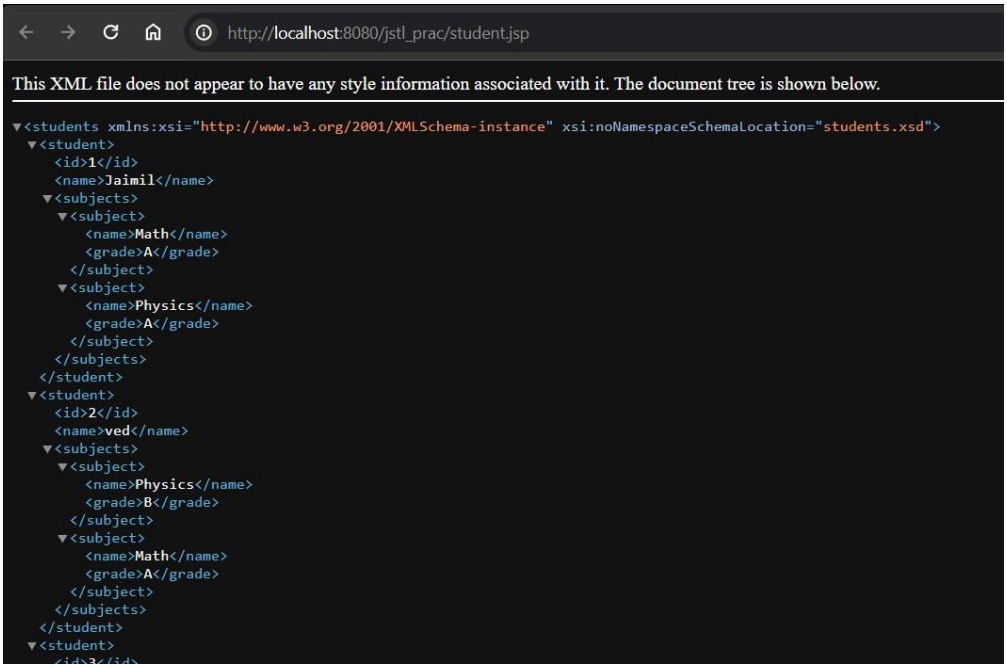


id	student_id	subject_name	grade
1	1	Math	A
2	1	Physics	B
3	2	Math	A
4	2	Physics	A
5	3	Math	A
6	3	Physics	A
NULL	NULL	NULL	NULL

File structure:



Web Output:



PRACTICAL 10

AIM : Implement a chat application using TCP. Write an RMI application where client supplies two numbers & server response by addition. Provide your custom security policy for this application.

Chat Application :

CODE :-

Client.java

```
package ChatApplication;


import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;


public class Client {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 12345);


            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
            BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));
```

```
            String userInputLine;
            while ((userInputLine = userInput.readLine()) != null) {
                out.println(userInputLine);
                System.out.println("Server: " + in.readLine());
            }

            in.close();
            out.close();
            userInput.close();
            socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

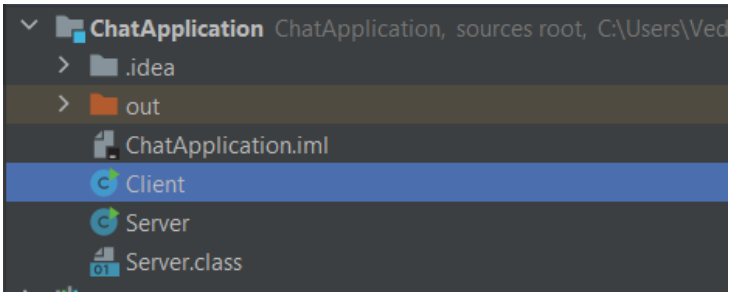
Server.java

```
package ChatApplication;

import java.io.*;
import java.net.*;

public class Server {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(12345);
            System.out.println("Server started. Waiting for clients...");
```

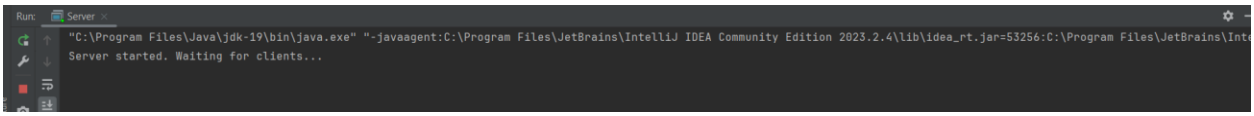
```
        Socket clientSocket = serverSocket.accept();
        System.out.println("Client connected: " + clientSocket);

        BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
        PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

        String inputLine;
        while ((inputLine = in.readLine()) != null) {
            System.out.println("Client: " + inputLine);
            out.println("Server received: " + inputLine);
        }

        in.close();
        out.close();
        clientSocket.close();
        serverSocket.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
  }
}
```

OUTPUT :--

File Structure :



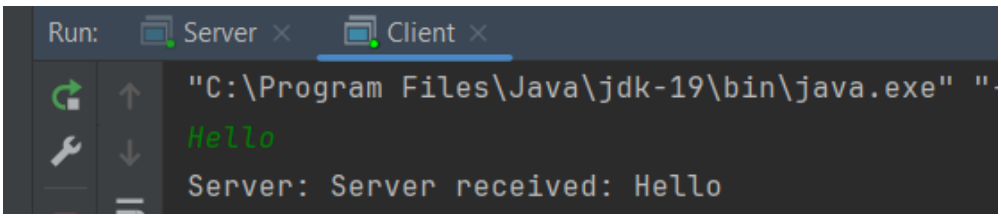
Server Output :




```
Server started. Waiting for clients...
Client connected: Socket[addr=/127.0.0.1,port=53286,localport=12345]
```

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBr
Server started. Waiting for clients...
Client connected: Socket[addr=/127.0.0.1,port=53286,localport=12345]
Client: Hello
```

Client Output :



RMI Application :

CODE :-

Adder.java

```
import java.rmi.*;
public interface Adder extends Remote{
public int add(int x,int y)throws RemoteException;
}
```

AdderRemote.java

```
import java.rmi.*;
import java.rmi.server.*;
public class AdderRemote extends UnicastRemoteObject implements Adder{
AdderRemote()throws RemoteException{
super();
}
public int add(int x,int y){return x+y;}
}
```

MyServer.java

```
import java.rmi.*;
import java.rmi.registry.*;
public class MyServer{
public static void main(String args[]){
try{
Adder stub=new AdderRemote();
```

```
Naming.rebind("rmi://localhost:5000/sonoo",stub);
}catch(Exception e){System.out.println(e);}
}
}
```

MyClient.java

```
import java.rmi.*;
public class MyClient{
public static void main(String args[]){
try{
Adder stub=(Adder)Naming.lookup("rmi://localhost:5000/sonoo");
System.out.println(stub.add(34,4));
}catch(Exception e){}
}
}
```

server.policy

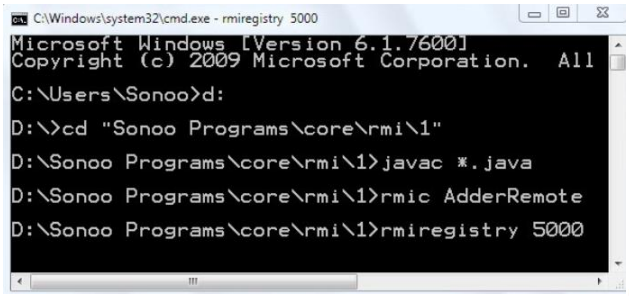
```
grant {
    permission java.security.AllPermission;
};
```

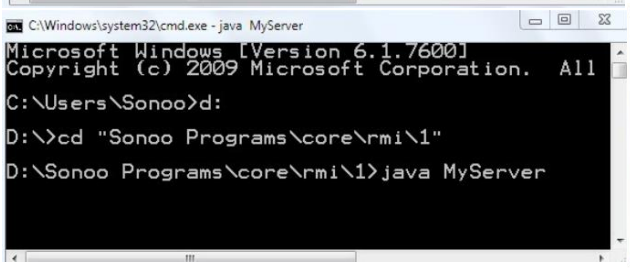
Command for running files with security policy :

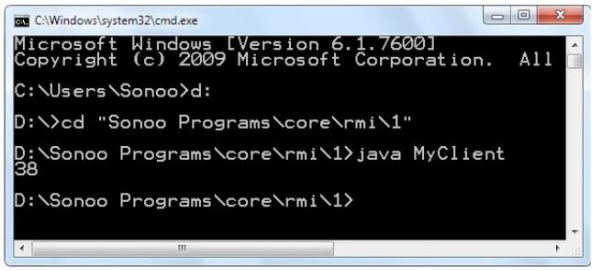
```
java -Djava.security.policy=server.policy Server
```

OUTPUT :--

RMI Output :







PRACTICAL 11

AIM : Write down a program of sending and receiving email using Java Mail API.

CODE :--

SendEmail.java

```
import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;
import javax.activation.*;

public class SendEmail
{
 public static void main(String [] args){
      String to = "veddesai42@gmail.com";
      String from = "jaimilnagar126@gmail.com";
      String host = "localhost";
      Properties properties = System.getProperties();
      properties.setProperty("mail.smtp.host", host);
      Session session = Session.getDefaultInstance(properties);
      try{
         MimeMessage message = new MimeMessage(session);
         message.setFrom(new InternetAddress(from));
         message.addRecipient(Message.RecipientType.TO,new InternetAddress(to));
         message.setSubject("Ping");
```

```
        message.setText("Hello, this is example of sending email  ");
        Transport.send(message);
        System.out.println("message sent successfully....");
      }catch (MessagingException mex) {mex.printStackTrace();}
   }
}
```

OUTPUT :--

```
Message sent successfully.
```

RecieveEmail.java

```
import java.io.IOException;
import java.util.Properties;
import javax.mail.Folder;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.NoSuchProviderException;
import javax.mail.Session;
import com.sun.mail.pop3.POP3Store;
public class ReceiveMail{
 public static void receiveEmail(String pop3Host, String storeType,
  String user, String password) {
  try {
   //1) get the session object
   Properties properties = new Properties();
   properties.put("mail.pop3.host", pop3Host);
   Session emailSession = Session.getDefaultInstance(properties);

   //2) create the POP3 store object and connect with the pop server
   POP3Store emailStore = (POP3Store) emailSession.getStore(storeType);
   emailStore.connect(user, password);

   //3) create the folder object and open it
   Folder emailFolder = emailStore.getFolder("INBOX");
   emailFolder.open(Folder.READ_ONLY);
```

```
  //4) retrieve the messages from the folder in an array and print it
  Message[] messages = emailFolder.getMessages();
  for (int i = 0; i < messages.length; i++) {
   Message message = messages[i];
   System.out.println("---------------------------------");
   System.out.println("Email Number " + (i + 1));
   System.out.println("Subject: " + message.getSubject());
   System.out.println("From: " + message.getFrom()[0]);
   System.out.println("Text: " + message.getContent().toString());
  }

  //5) close the store and folder objects
  emailFolder.close(false);
  emailStore.close();

 } catch (NoSuchProviderException e) {e.printStackTrace();}
 catch (MessagingException e) {e.printStackTrace();}
 catch (IOException e) {e.printStackTrace();}
}

public static void main(String[] args) {

 String host = "mail.javatpoint.com";//change accordingly
 String mailStoreType = "pop3";
 String username= "veddesai42@gmail.com";
 String password= "xxxxx";//change accordingly
```

```
  receiveEmail(host, mailStoreType, username, password);

 }
}
```

OUTPUT :--

```
---------------------------------
Email Number 1
Subject: Subject of the email
From: sender@example.com
Text: Content of the email
```

Lab – 12

AIM: Implement simple framework using hibernate and studies its architecture.

Employee.java:

```
package entities;
public class Employee {

	private int empId;
	private String empName;

	public Employee(int empId, String empName)
	{
		super();
		this.empId = empId;
		this.empName = empName;
	}
	public int getEmpId() { return empId; }
	public void setEmpId(int empId) { this.empId = empId; }
	public String getEmpName() { return empName; }
	public void setEmpName(String empName)
	{
		this.empName = empName;
	}
}
```

Employee.hbm.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
<class name="entities.Employee" table="employee">
<id name="empId"></id>
<property name="empName"></property>
</class>
</hibernate-mapping>
```

hibernate.cfg.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="hibernate.hbm2ddl.auto">create</property>
<property name="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</property>
<property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/demo</property>
<property name="hibernate.connection.username">root</property>
<property name="hibernate.connection.password">root</property>
<mapping resource="resources/Employee.hbm.xml"></mapping>
</session-factory>
</hibernate-configuration>
```

SessionFactoryProvider.java:

```
package utilities;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class SessionFactoryProvider {
	public static SessionFactory provideSessionFactory()
	{
		Configuration config = new Configuration();
		config.configure("resources/hibernate.cfg.xml");
		return config.buildSessionFactory();
	}
}
```

Create.java:

```
package dao;
import entities.Employee;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import utilities.SessionFactoryProvider;

public class Create {
	public static void main(String[] args)
	{
		try {
			SessionFactory sessionFactory
				= SessionFactoryProvider
					.provideSessionFactory();
```

```
                Session session = sessionFactory.openSession();
Transaction t = session.beginTransaction();

                Employee emp = new Employee(101, "John");
                session.save(emp);
                t.commit();

                sessionFactory.close();
            }
            catch (Exception e) {
                System.out.println(e);
            }
        }
}
```

Output:

```
mysql> use demo;
Database changed
mysql> select * from employee;
+-------+---------+
| empId | empName |
+-------+---------+
|   101 | John    |
+-------+---------+
1 row in set (0.01 sec)

mysql>
```

Lab-13

AIM: Use Hibernate Query Language to insert, update and delete records in the database.

SessionFactoryProvider.java :

```
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class SessionFactoryProvider {
	public static SessionFactory provideSessionFactory()
	{
		Configuration config=new Configuration();
		config.configure();
		return config.buildSessionFactory();
	}
}
```

Student.java :

```
import javax.persistence.*;

@Entity
public class Student {
	@Id
	private int id;
	private String name;
	private int std;

	public Student() {
	}
	public Student(int id, String name, int std) {
```

```
            this.id = id;
            this.name = name;
        this.std = std;
        }
        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 int getStd() {
            return std;
        }
        public void setStd(int std) {
            this.std = std;
        }
}
```

hibernate.cfg.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
		"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
		"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
	<session-factory>
		<property name="hibernate.hbm2ddl.auto">create</property>
		<property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
		<property name="hibernate.connection.password">root</property>
		<property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/student_info</property>
		<property name="hibernate.connection.username">root</property>
		<property
name="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</property>
		<mapping class="beans.Student"></mapping>
	</session-factory>
</hibernate-configuration>
```

Create.java :

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import utilities.SessionFactoryProvider;
public class Create {
	public static void main(String[] args)
	{
		SessionFactory
sessionFactory=SessionFactoryProvider.provideSessionFactory();
```

```
            Session session=sessionFactory.openSession();
            Transaction t=session.beginTransaction();

            Student s=new Student(101,"John",10);
            session.save(s);
            t.commit();
            sessionFactory.close();
        }
}
```

Update.java :

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import utilities.SessionFactoryProvider;

public class Update {
        public static void main(String[] args)
        {
            SessionFactory
sessionFactory=SessionFactoryProvider.provideSessionFactory();
            Session session=sessionFactory.openSession();
            Transaction t=session.beginTransaction();

            Student s=session.get(Student.class, 101);
            s.setStd(11);
            session.save(s);
            t.commit();
            sessionFactory.close();
        }
```

```
}
```

Delete.java :

```
package crudOperations;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import beans.Student;
import utilities.SessionFactoryProvider;

public class Delete {
	public static void main(String[] args)
	{
		SessionFactory
sessionFactory=SessionFactoryProvider.provideSessionFactory();
		Session session=sessionFactory.openSession();
		Transaction t=session.beginTransaction();

		Student s=session.get(Student.class, 101);
		session.delete(s);
		t.commit();

		sessionFactory.close();
	}
}
```

output:

Create.java:

```
mysql> use student_info;
Database changed
mysql> select * from student;
+-----+------+-----+
| id  | name | std |
+-----+------+-----+
| 101 | John |  10 |
+-----+------+-----+
1 row in set (0.11 sec)
```

Update.java:

```
mysql> select * from student;
+-----+------+-----+
| id  | name | std |
+-----+------+-----+
| 101 | John |  11 |
+-----+------+-----+
1 row in set (0.00 sec)
```

Delete.java:

```
mysql> select * from student;
Empty set (0.00 sec)
```

Lab – 14

AIM: Demonstrate CRUD operation using DAO and Spring Framework API.

Books.java :

```
package com.javatpoint.model;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table
public class Books
{
@Id
@Column
private int bookid;
@Column
private String bookname;
@Column
private String author;
@Column
private int price;
public int getBookid()
{
return bookid;
}
public void setBookid(int bookid)
{
this.bookid = bookid;
```

```
}
public String getBookname()
{
return bookname;
}
public void setBookname(String bookname)
{
this.bookname = bookname;
}
public String getAuthor()
{
return author;
}
public void setAuthor(String author)
{
this.author = author;
}
public int getPrice()
{
return price;
}
public void setPrice(int price)
{
this.price = price;
}
}
```

BooksController.java :

```
package com.javatpoint.controller;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;
import com.javatpoint.model.Books;
import com.javatpoint.service.BooksService;

@RestController
public class BooksController
{
@Autowired
BooksService booksService;
@GetMapping("/book")
private List<Books> getAllBooks()
{
return booksService.getAllBooks();
}

@GetMapping("/book/{bookid}")
private Books getBooks(@PathVariable("bookid") int bookid)
{
return booksService.getBooksById(bookid);
}
@DeleteMapping("/book/{bookid}")
private void deleteBook(@PathVariable("bookid") int bookid)
{
booksService.delete(bookid);
```

```
}
@PostMapping("/books")
private int saveBook(@RequestBody Books books)
{
booksService.saveOrUpdate(books);
return books.getBookid();
}
@PutMapping("/books")
private Books update(@RequestBody Books books)
{
booksService.saveOrUpdate(books);
return books;
}
}
```

BooksService.java :

```
package com.javatpoint.service;
import java.util.ArrayList;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.javatpoint.model.Books;
import com.javatpoint.repository.BooksRepository;
@Service
public class BooksService
{
@Autowired
BooksRepository booksRepository;

public List<Books> getAllBooks()
```

```
{
List<Books> books = new ArrayList<Books>();
booksRepository.findAll().forEach(books1 -> books.add(books1));
return books;
}
public Books getBooksById(int id)
{
return booksRepository.findById(id).get();
}

public void saveOrUpdate(Books books)
{
booksRepository.save(books);
}

public void delete(int id)
{
booksRepository.deleteById(id);
}
public void update(Books books, int bookid)
{
booksRepository.save(books);
}
}
```

BooksRepository.java :

```
package com.javatpoint.repository;
import org.springframework.data.repository.CrudRepository;
import com.javatpoint.model.Books;
public interface BooksRepository extends CrudRepository<Books, Integer>
```

```
{
}
```

application.properties :

```
spring.datasource.url=jdbc:h2:mem:books_data
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true
```

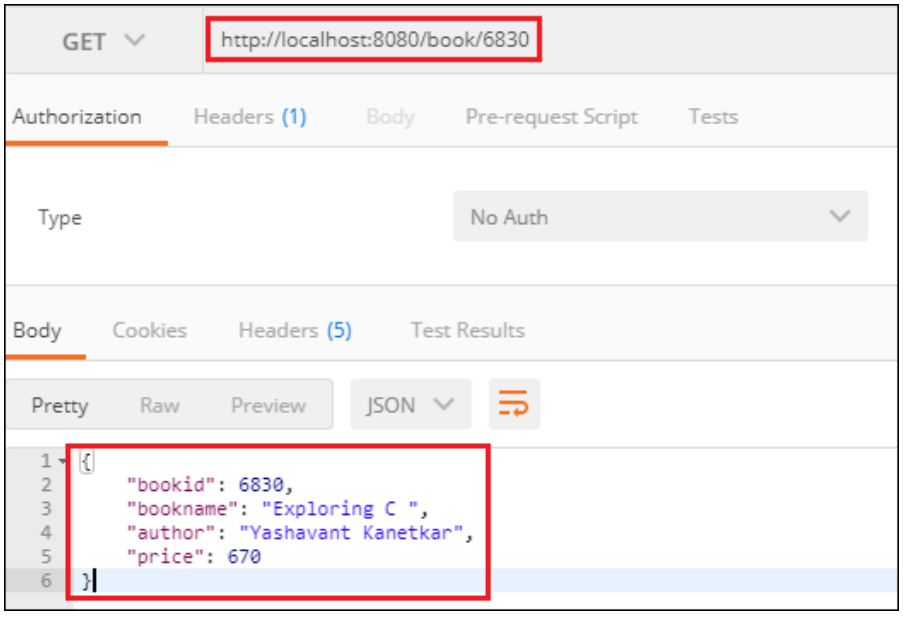
SpringBootCrudOperationApplication.java :

```
package com.javatpoint;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class SpringBootCrudOperationApplication
{
public static void main(String[] args)
{
SpringApplication.run(SpringBootCrudOperationApplication.class, args);
}
}
```

JSON :

```
{
   "bookid": "5433",
   "bookname": "Core and Advance Java",
   "author": "R. Nageswara Rao",
   "price": "800"
```

}
{
"bookid": "0982",
"bookname": "Programming with Java",
"author": "E. Balagurusamy",
"price": "350"
}
{
"bookid": "6321",
"bookname": "Data Structures and Algorithms in Java",
"author": "Robert Lafore",
"price": "590"
}
{
"bookid": "6830",
"bookname": "Yashavant Kanetkar",
"author": "Exploring C",
"price": "670"
}



DELETE 5433 :

SELECT * FROM BOOKS;

BOOKID	AUTHOR	BOOKNAME	PRICE
982	E. Balagurusamy	Programming with Java	350
6321	Robert Lafore	Data Structures and Algorithms in Java	590
6830	Yashavant Kanetkar	Exploring C	670

(3 rows, 23 ms)

UPDATE 6321 :

{

"bookid": "6321",

"bookname": "Data Structures and Algorithms in Java",

"author": "Robert Lafore",

"price": "500"

}

SELECT * FROM BOOKS;

BOOKID	AUTHOR	BOOKNAME	PRICE
982	E. Balagurusamy	Programming with Java	350
6321	Robert Lafore	Data Structures and Algorithms in Java	500
6830	Yashavant Kanetkar	Exploring C	390

(3 rows, 20 ms)