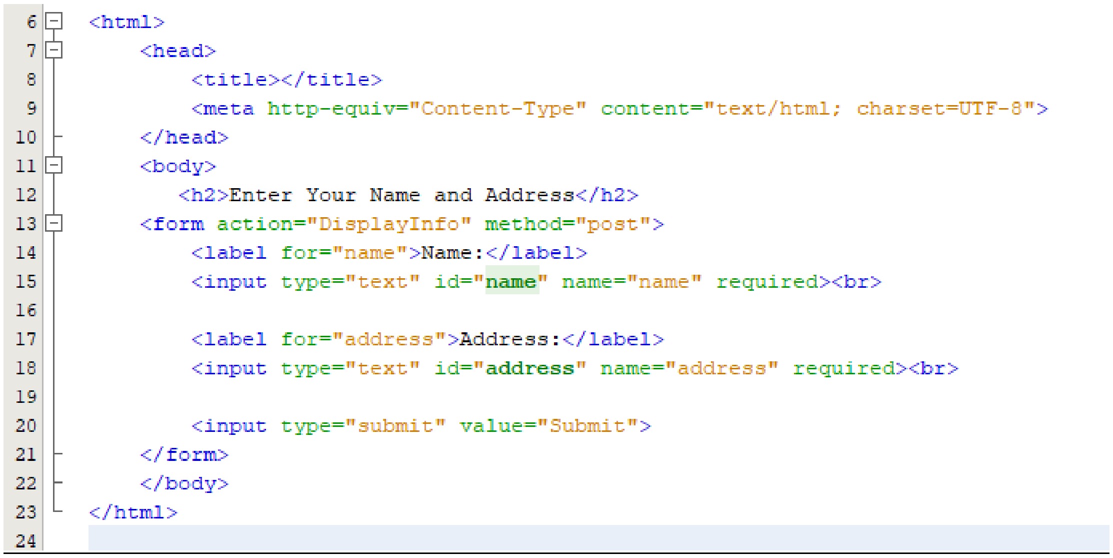
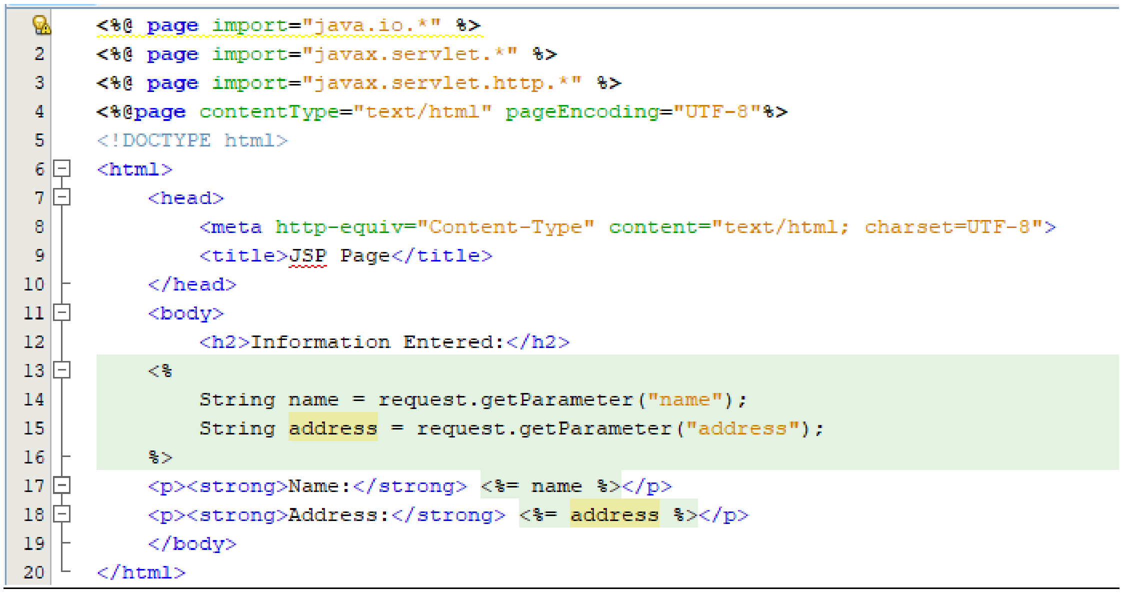
Lab Sheet 6

[1] Write a JSP/Servlet program that takes your name and address from a HTML Form and displays it on a web page.

Source code:

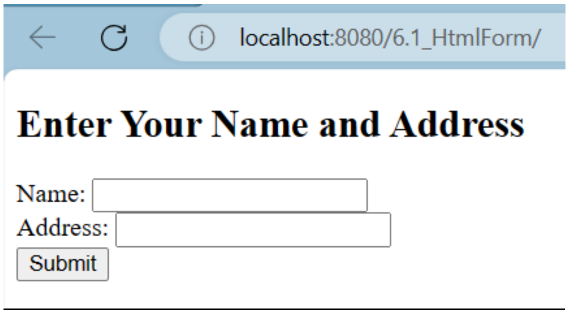


```
<html>
    <head>
        <title></title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
       <h2>Enter Your Name and Address</h2>
    <form action="DisplayInfo" method="post">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required><br>

        <label for="address">Address:</label>
        <input type="text" id="address" name="address" required><br>

        <input type="submit" value="Submit">
    </form>
    </body>
</html>

```



```
<%@ page import="java.io.*" %>
<%@ page import="javax.servlet.*" %>
<%@ page import="javax.servlet.http.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h2>Information Entered:</h2>
    <%
        String name = request.getParameter("name");
        String address = request.getParameter("address");
    %>
    <p><strong>Name:</strong> <%= name %></p>
    <p><strong>Address:</strong> <%= address %></p>
    </body>
</html>
```

Output:



localhost:8080/6.1_HtmlForm/

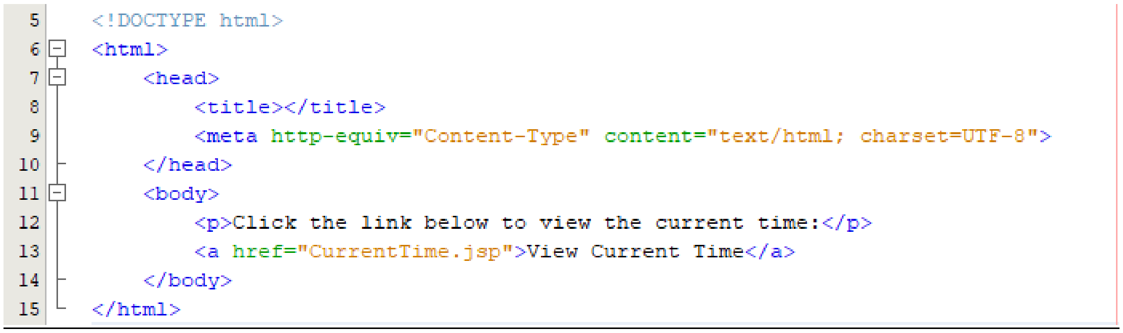
Enter Your Name and Address

Name:

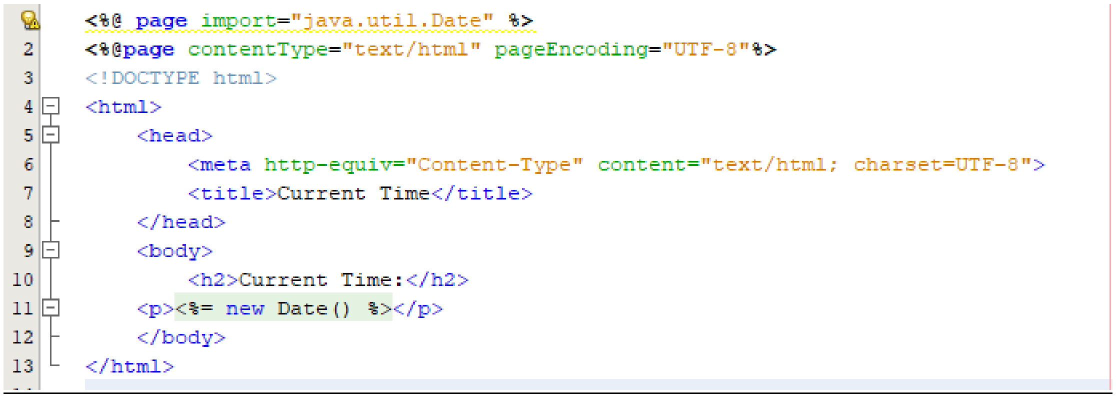
Address:

[2] Write a JSP program that output current time only.

Source code:

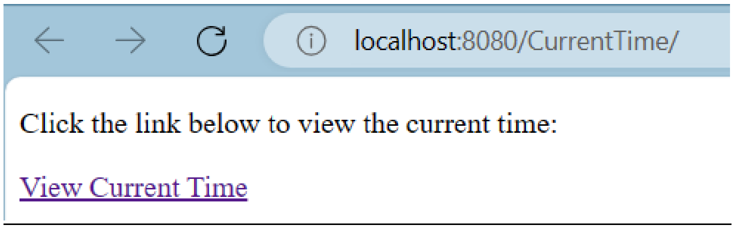


```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <p>Click the link below to view the current time:</p>
        <a href="CurrentTime.jsp">View Current Time</a>
    </body>
</html>
```



```
<%@ page import="java.util.Date" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Current Time</title>
    </head>
    <body>
        <h2>Current Time:</h2>
    <p><%= new Date() %></p>
    </body>
</html>
```

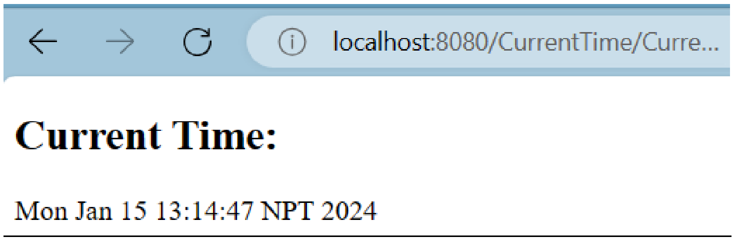
Output:



localhost:8080/CurrentTime/

Click the link below to view the current time:

View Current Time

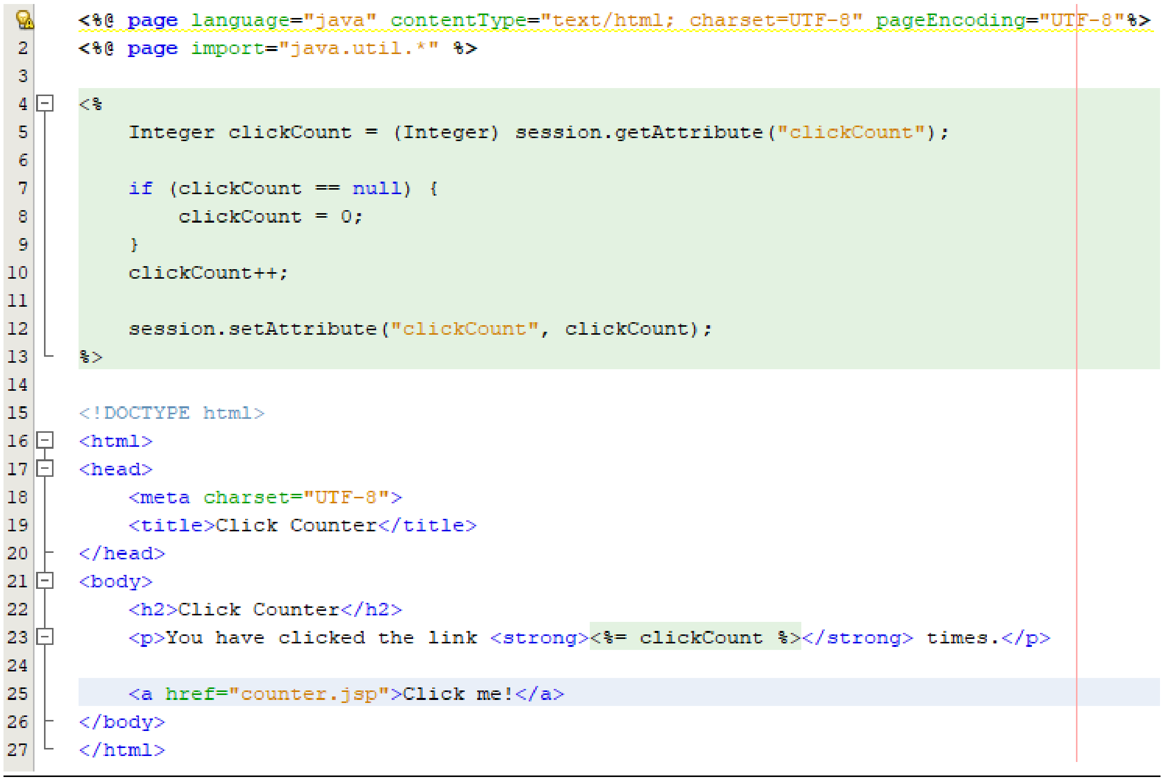


Current Time:

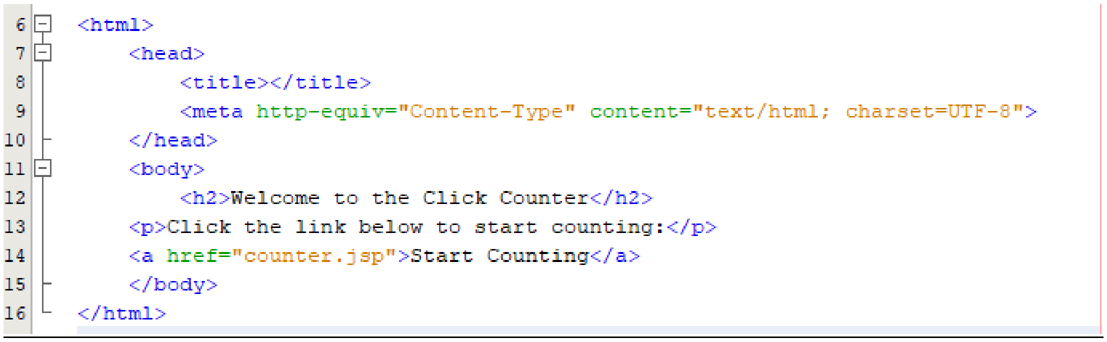
Mon Jan 15 13:14:47 NPT 2024

[3] Write a JSP program that counts the number of times a link is clicked.

Source code:

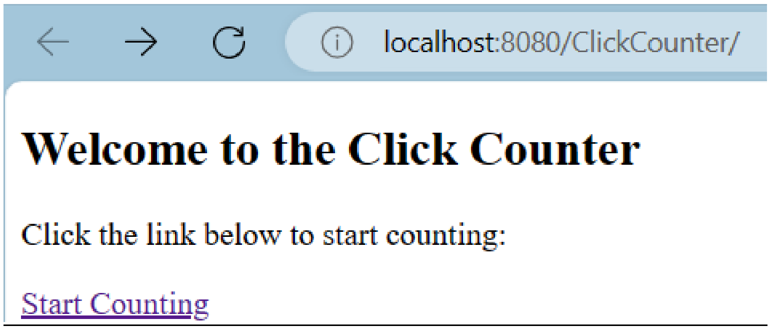


```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.util.*" %>

<%
    Integer clickCount = (Integer) session.getAttribute("clickCount");

    if (clickCount == null) {
        clickCount = 0;
    }
    clickCount++;

    session.setAttribute("clickCount", clickCount);
%>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Click Counter</title>
</head>
<body>
    <h2>Click Counter</h2>
    <p>You have clicked the link <strong><%= clickCount %></strong> times.</p>

    <a href="counter.jsp">Click me!</a>
</body>
</html>
```



```
<html>
    <head>
        <title></title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <h2>Welcome to the Click Counter</h2>
    <p>Click the link below to start counting:</p>
    <a href="counter.jsp">Start Counting</a>
    </body>
</html>
```

Output:

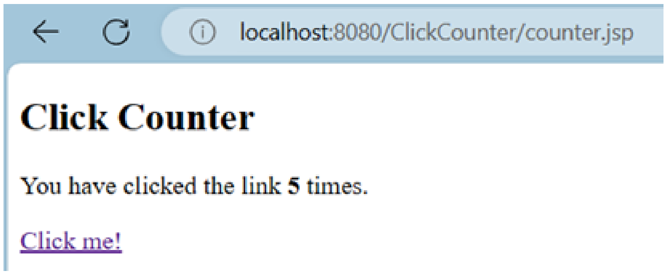


localhost:8080/ClickCounter/

Welcome to the Click Counter

Click the link below to start counting:

Start Counting



localhost:8080/ClickCounter/counter.jsp

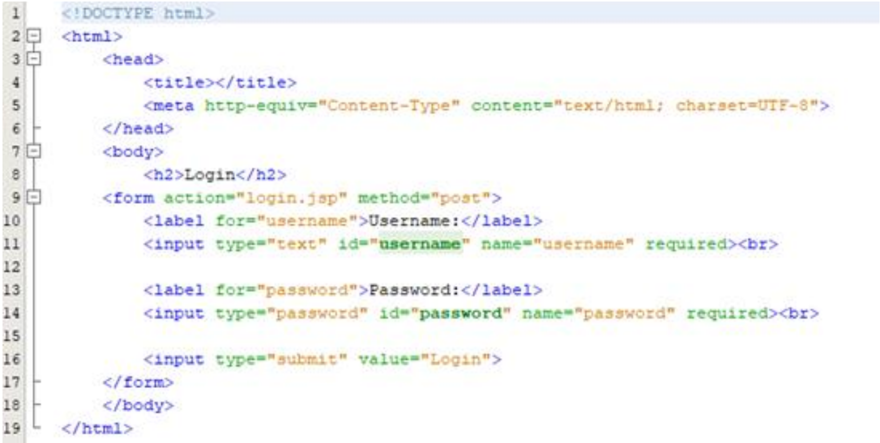
Click Counter

You have clicked the link **5** times.

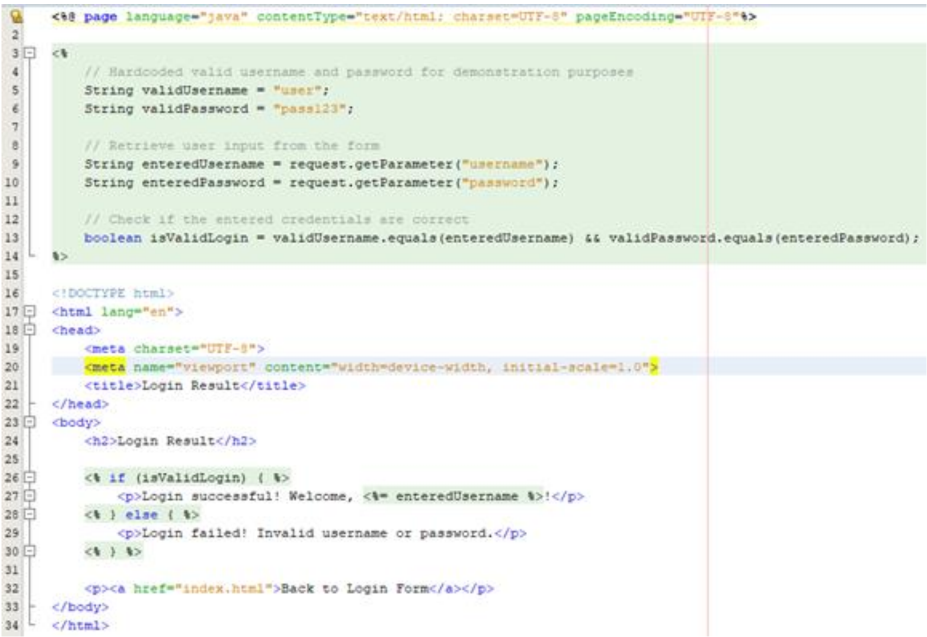
Click me!

[4] Create a login form and check if the user name and password entered by the user are correct.

Source code:

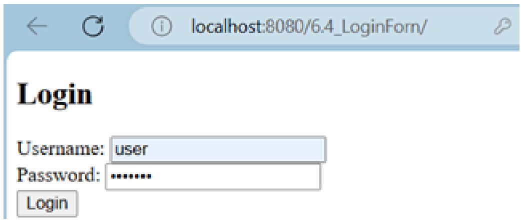


```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <h2>Login</h2>
    <form action="login.jsp" method="post">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required><br>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br>

        <input type="submit" value="Login">
    </form>
    </body>
</html>
```



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

<%
    // Hardcoded valid username and password for demonstration purposes
    String validUsername = "user";
    String validPassword = "pass123";

    // Retrieve user input from the form
    String enteredUsername = request.getParameter("username");
    String enteredPassword = request.getParameter("password");

    // Check if the entered credentials are correct
    boolean isValidLogin = validUsername.equals(enteredUsername) && validPassword.equals(enteredPassword);
%>

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

    <% if (isValidLogin) { %>
        <p>Login successful! Welcome, <%= enteredUsername %>!</p>
    <% } else { %>
        <p>Login failed! Invalid username or password.</p>
    <% } %>

    <p><a href="index.html">Back to Login Form</a></p>
</body>
</html>
```

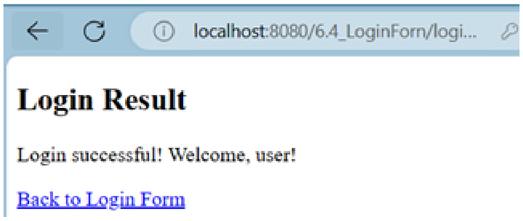
Output:



Login

Username: user

Password: •••••••



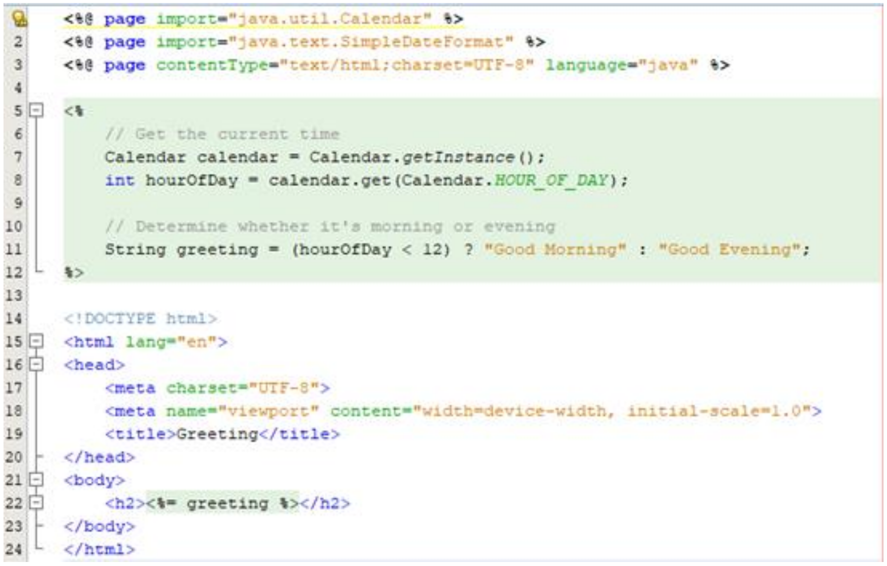
Login Result

Login successful! Welcome, user!

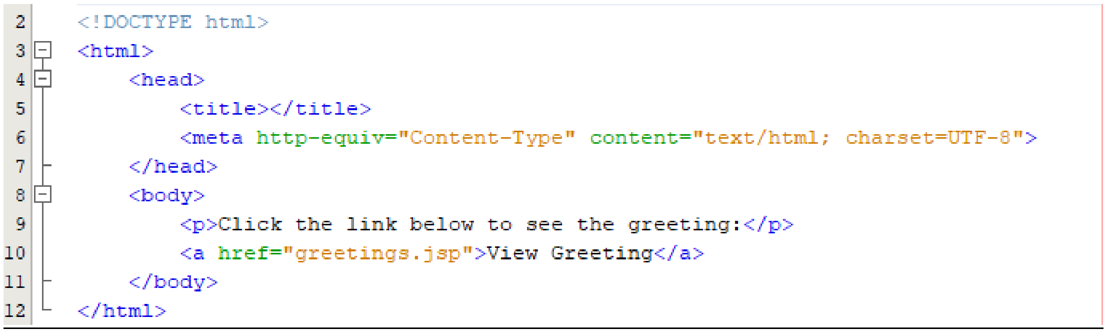
Back to Login Form

[5] Write a JSP program that displays “Good Morning” or “Good Evening” based on the present time.

Source code:

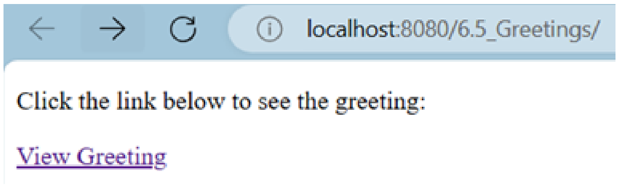


```
<%@ page import="java.util.Calendar" %>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%
    // Get the current time
    Calendar calendar = Calendar.getInstance();
    int hourOfDay = calendar.get(Calendar.HOUR_OF_DAY);

    // Determine whether it's morning or evening
    String greeting = (hourOfDay < 12) ? "Good Morning" : "Good Evening";
%>

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



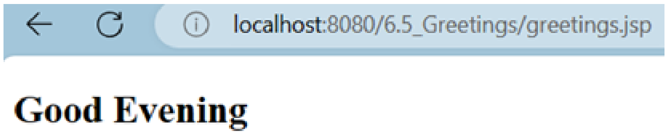
```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <p>Click the link below to see the greeting:</p>
        <a href="greetings.jsp">View Greeting</a>
    </body>
</html>
```

Output:



Click the link below to see the greeting:

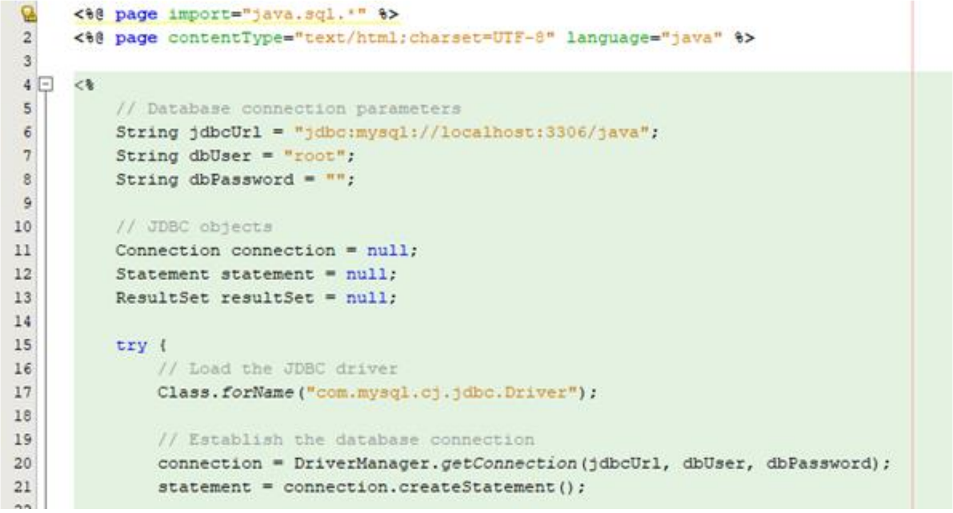
View Greeting



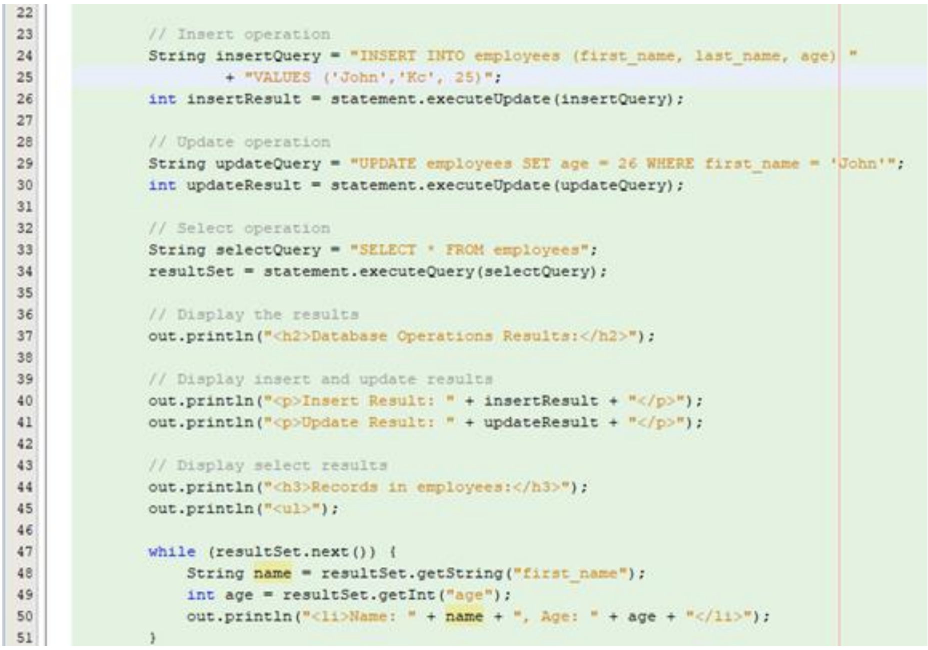
Good Evening

[6] Write a simple JSP program to show the database connectivity. Your program should use insert, delete, select and update operations.

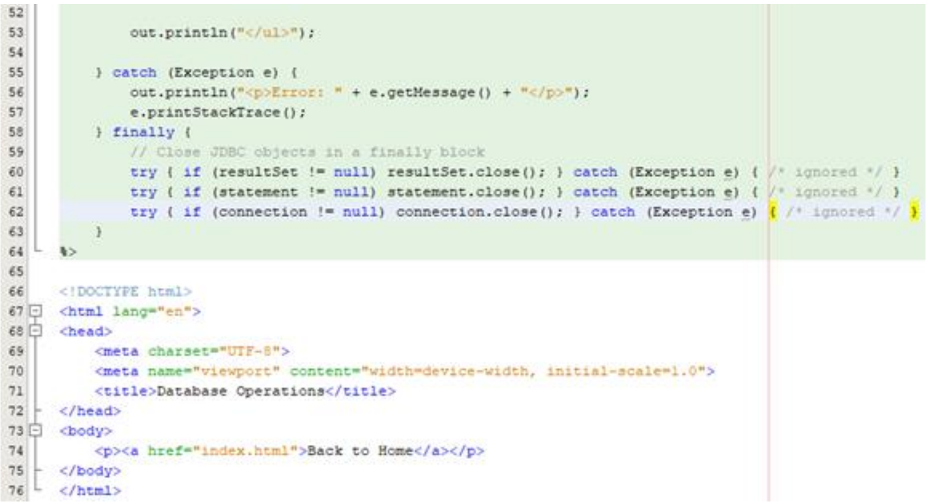
Source code:



```
<%@ page import="java.sql.*" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%
    // Database connection parameters
    String jdbcUrl = "jdbc:mysql://localhost:3306/java";
    String dbUser = "root";
    String dbPassword = "";

    // JDBC objects
    Connection connection = null;
    Statement statement = null;
    ResultSet resultSet = null;

    try {
        // Load the JDBC driver
        Class.forName("com.mysql.cj.jdbc.Driver");

        // Establish the database connection
        connection = DriverManager.getConnection(jdbcUrl, dbUser, dbPassword);
        statement = connection.createStatement();
```

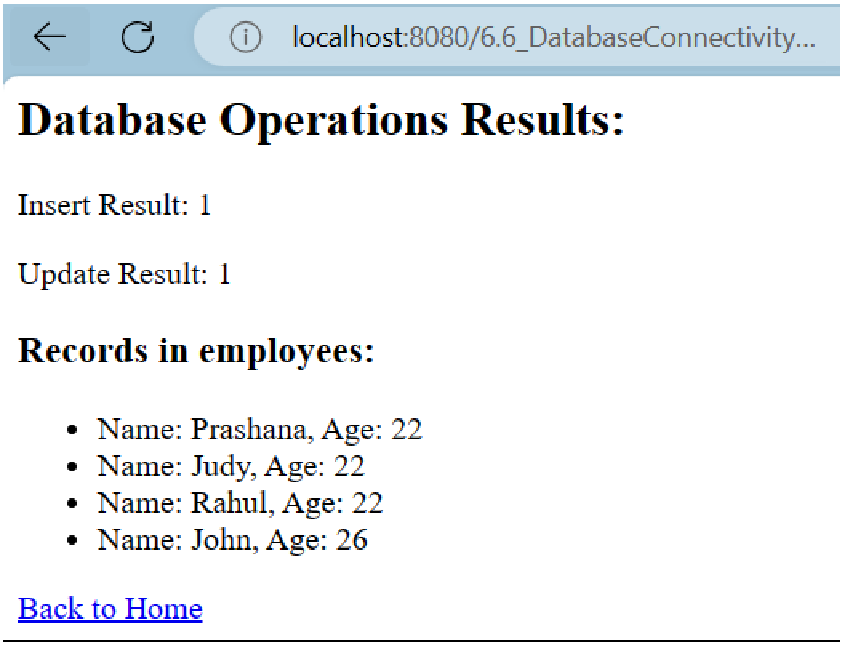


```

        // Insert operation
        String insertQuery = "INSERT INTO employees (first_name, last_name, age) "
                + "VALUES ('John','Kc', 25)";
        int insertResult = statement.executeUpdate(insertQuery);

        // Update operation
        String updateQuery = "UPDATE employees SET age = 26 WHERE first_name = 'John'";
        int updateResult = statement.executeUpdate(updateQuery);

        // Select operation
        String selectQuery = "SELECT * FROM employees";
        resultSet = statement.executeQuery(selectQuery);

        // Display the results
        out.println("<h2>Database Operations Results:</h2>");

        // Display insert and update results
        out.println("<p>Insert Result: " + insertResult + "</p>");
        out.println("<p>Update Result: " + updateResult + "</p>");

        // Display select results
        out.println("<h3>Records in employees:</h3>");
        out.println("<ul>");

        while (resultSet.next()) {
            String name = resultSet.getString("first_name");
            int age = resultSet.getInt("age");
            out.println("<li>Name: " + name + ", Age: " + age + "</li>");
        }
```



```

        out.println("</ul>");

    } catch (Exception e) {
        out.println("<p>Error: " + e.getMessage() + "</p>");
        e.printStackTrace();
    } finally {
        // Close JDBC objects in a finally block
        try { if (resultSet != null) resultSet.close(); } catch (Exception e) { /* ignored */ }
        try { if (statement != null) statement.close(); } catch (Exception e) { /* ignored */ }
        try { if (connection != null) connection.close(); } catch (Exception e) { /* ignored */ }
    }
%>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Database Operations</title>
</head>
<body>
    <p><a href="index.html">Back to Home</a></p>
</body>
</html>
```

Output:



Database Operations Results:

Insert Result: 1

Update Result: 1

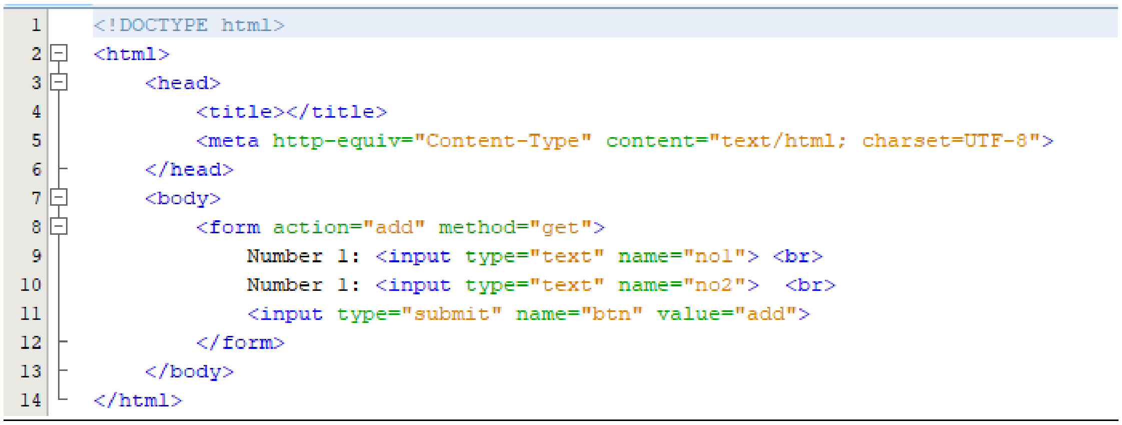
Records in employees:

- Name: Prashana, Age: 22
- Name: Judy, Age: 22
- Name: Rahul, Age: 22
- Name: John, Age: 26

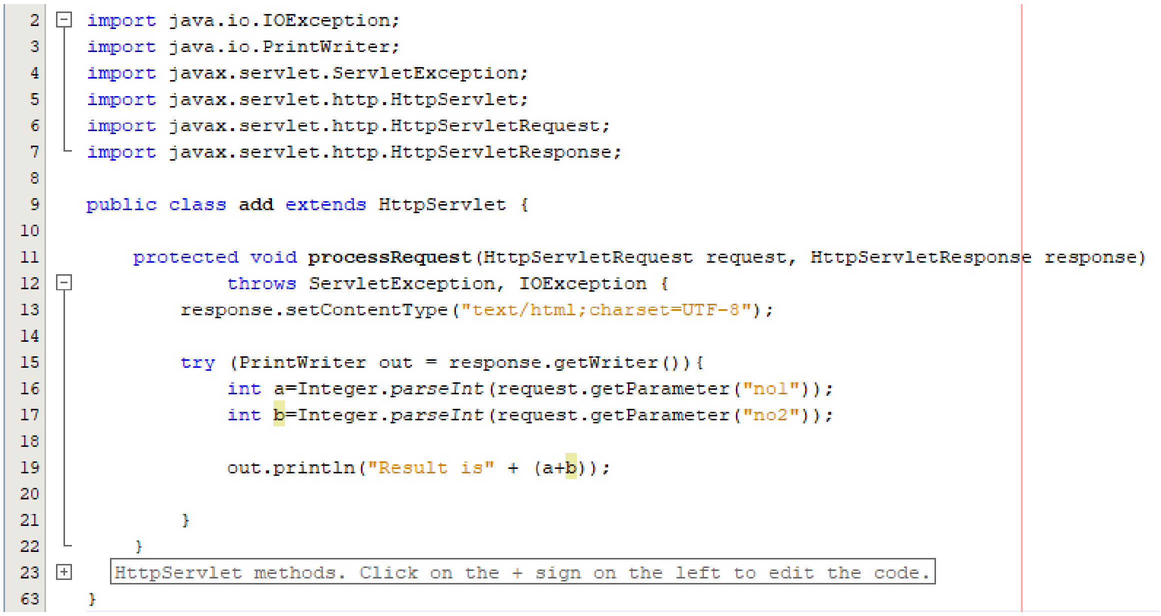
Back to Home

[7] Write a simple program to demonstrate the implementation of servlet.

Source code:

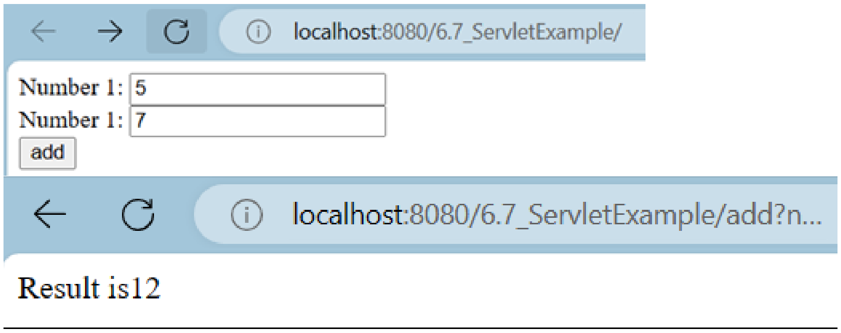


```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <form action="add" method="get">
            Number 1: <input type="text" name="no1"> <br>
            Number 1: <input type="text" name="no2">  <br>
            <input type="submit" name="btn" value="add">
        </form>
    </body>
</html>
```



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

public class add extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        try (PrintWriter out = response.getWriter()){
            int a=Integer.parseInt(request.getParameter("no1"));
            int b=Integer.parseInt(request.getParameter("no2"));

            out.println("Result is" + (a+b));

        }
    }
    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

Output:



localhost:8080/6.7_ServletExample/

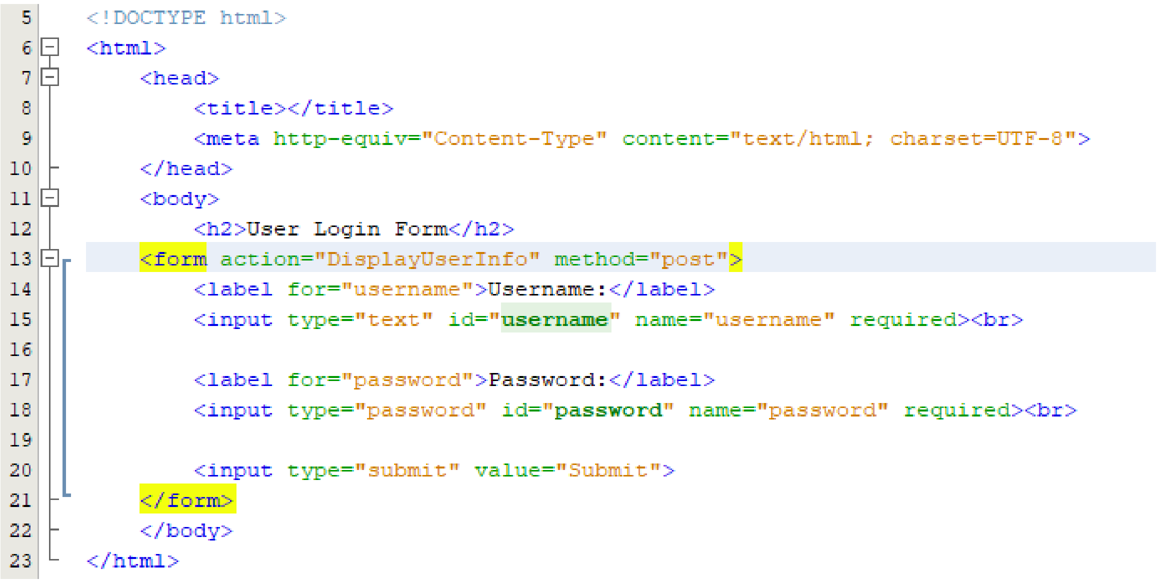
Number 1: 5
Number 1: 7

localhost:8080/6.7_ServletExample/add?n...

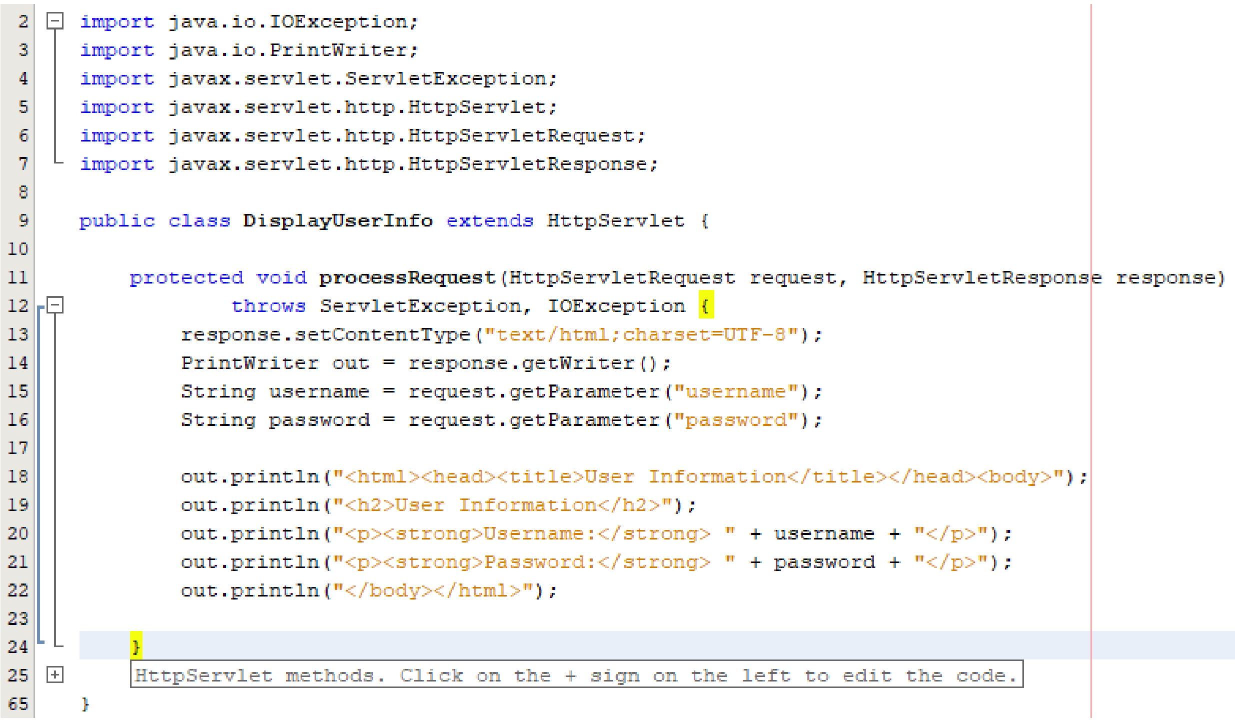
Result is12

[8] Design a simple HTML form with two text fields for user name and password and display the content of these text boxes in another page using servlet.

Source code:

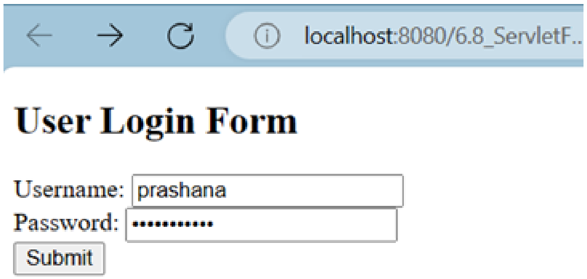


```html
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <h2>User Login Form</h2>
    <form action="DisplayUserInfo" method="post">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required><br>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br>

        <input type="submit" value="Submit">
    </form>
    </body>
</html>
```



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

public class DisplayUserInfo extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        out.println("<html><head><title>User Information</title></head><body>");
        out.println("<h2>User Information</h2>");
        out.println("<p><strong>Username:</strong> " + username + "</p>");
        out.println("<p><strong>Password:</strong> " + password + "</p>");
        out.println("</body></html>");

    }
    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

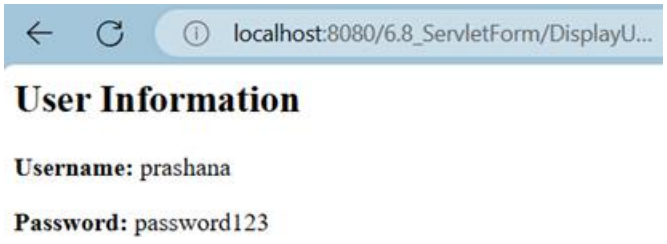
Output:



User Login Form

Username: prashana

Password: •••••••••••



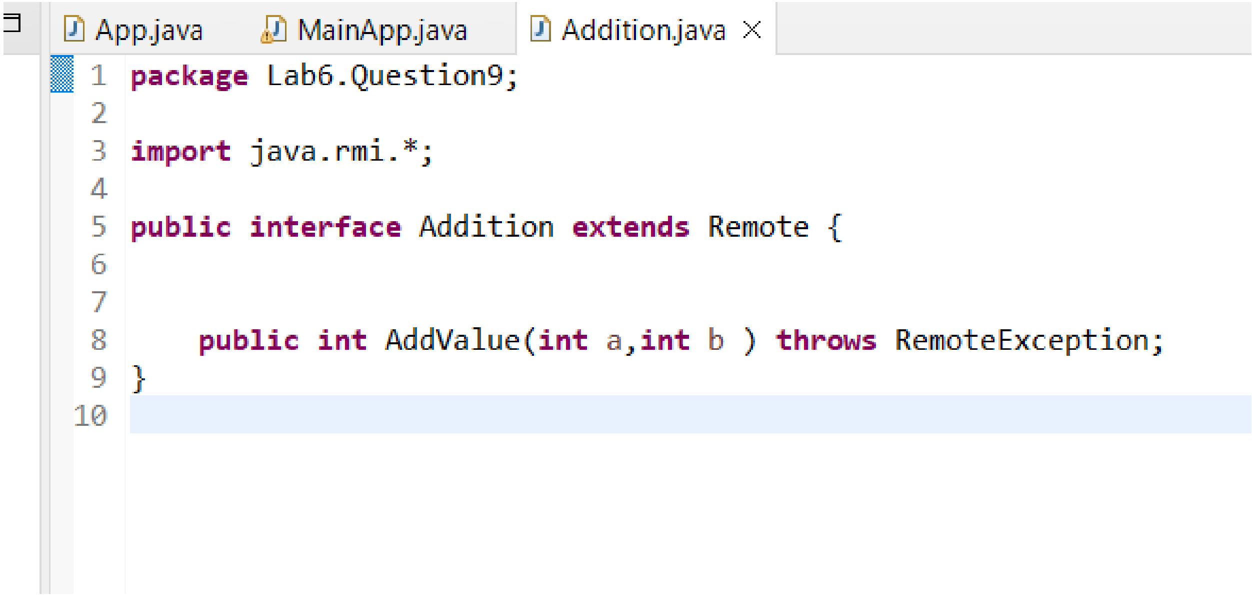
User Information

Username: prashana

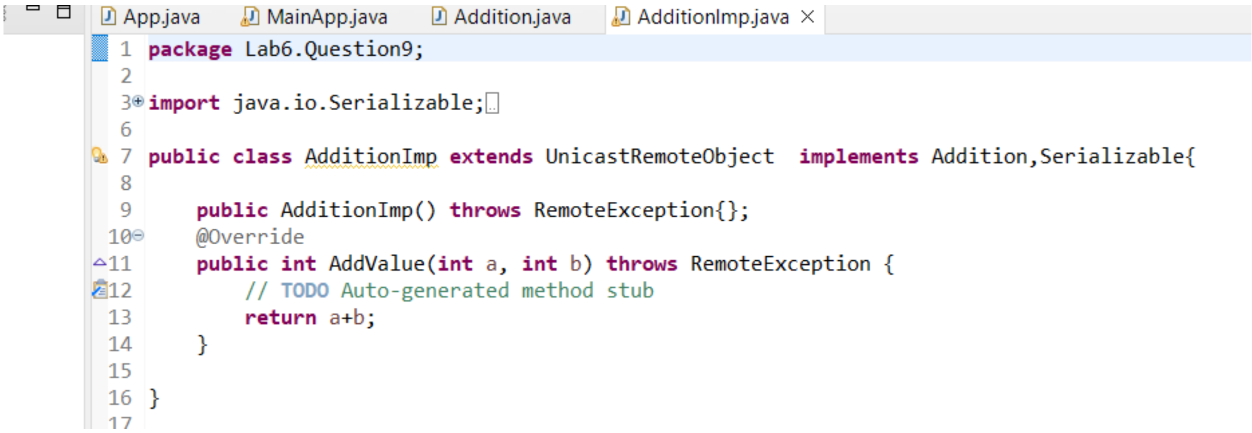
Password: password123

[9] WAP to create client and server using RMI. [Make necessary assumptions in case needed]

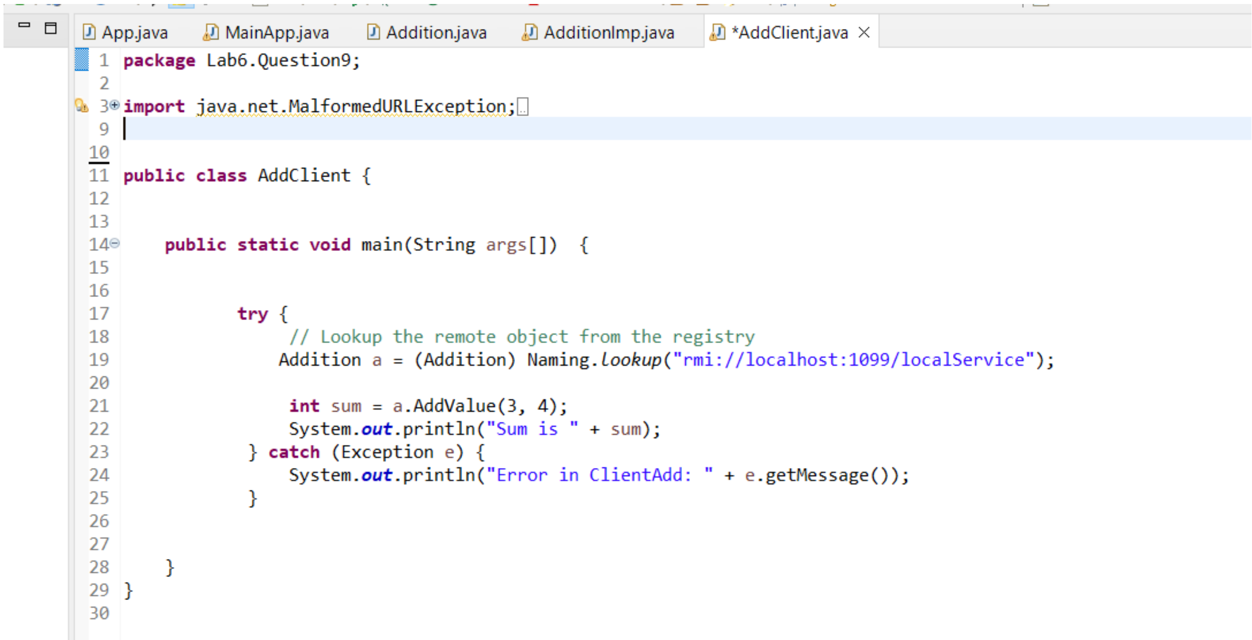
Source code:



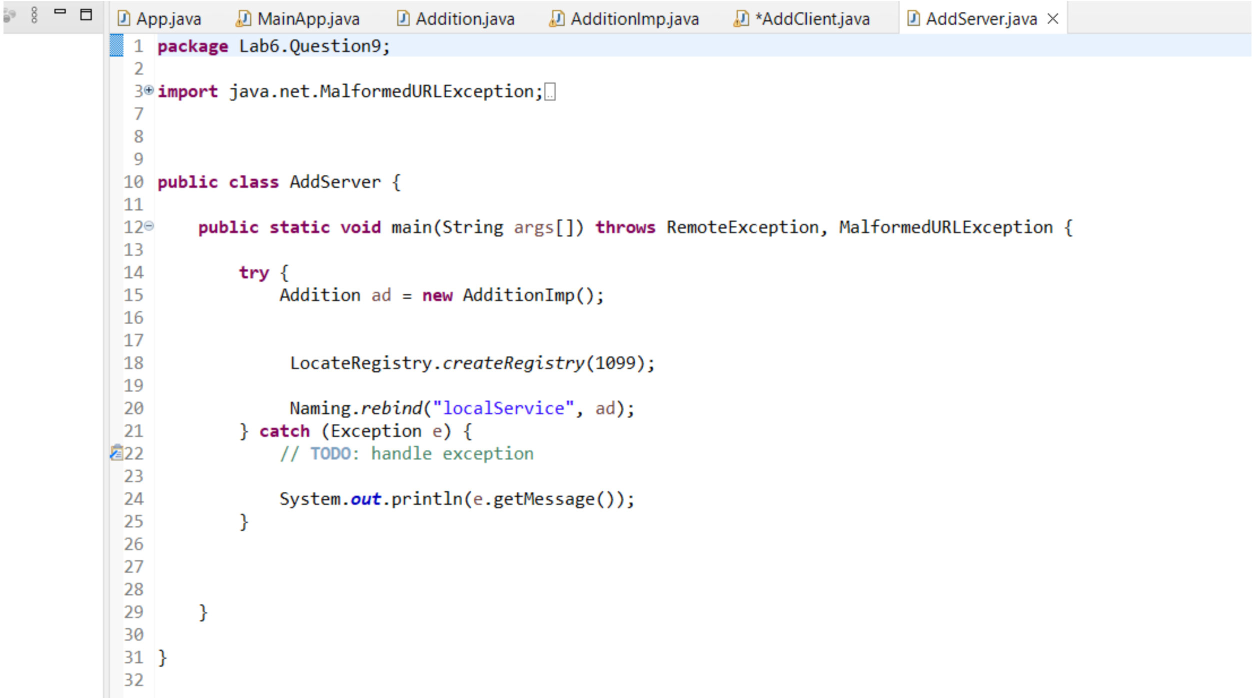
```java
package Lab6.Question9;

import java.rmi.*;

public interface Addition extends Remote {


    public int AddValue(int a,int b ) throws RemoteException;
}

```



```java
package Lab6.Question9;

import java.io.Serializable;

public class AdditionImp extends UnicastRemoteObject  implements Addition,Serializable{

    public AdditionImp() throws RemoteException{};
    @Override
    public int AddValue(int a, int b) throws RemoteException {
        // TODO Auto-generated method stub
        return a+b;
    }

}
```

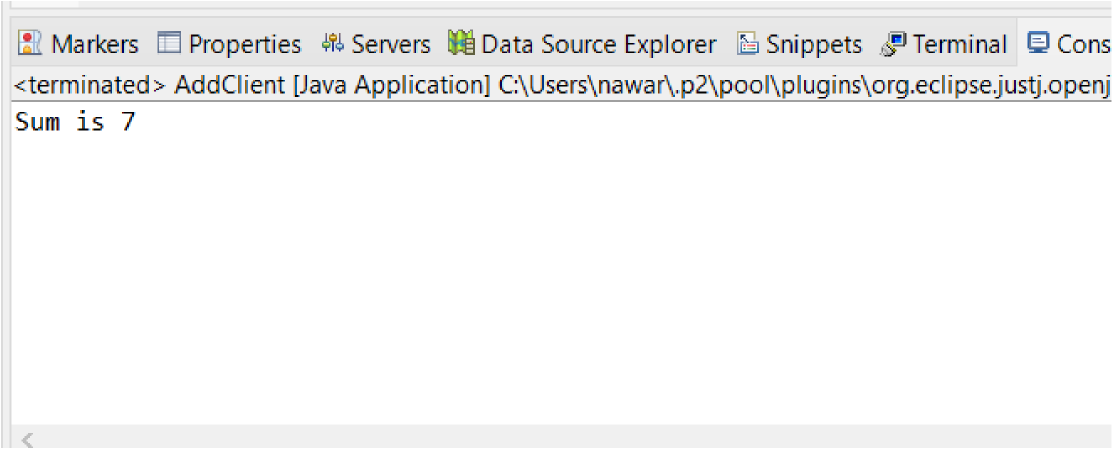


```java
package Lab6.Question9;

import java.net.MalformedURLException;


public class AddClient {


    public static void main(String args[])  {


           try {
                // Lookup the remote object from the registry
               Addition a = (Addition) Naming.lookup("rmi://localhost:1099/localService");

                int sum = a.AddValue(3, 4);
                System.out.println("Sum is " + sum);
            } catch (Exception e) {
                System.out.println("Error in ClientAdd: " + e.getMessage());
            }


    }
}

```



```java
package Lab6.Question9;

import java.net.MalformedURLException;



public class AddServer {

    public static void main(String args[]) throws RemoteException, MalformedURLException {

        try {
            Addition ad = new AdditionImp();


             LocateRegistry.createRegistry(1099);

             Naming.rebind("localService", ad);
        } catch (Exception e) {
            // TODO: handle exception

            System.out.println(e.getMessage());
        }



    }

}

```

Output:



```
<terminated> AddClient [Java Application] C:\Users\nawar\.p2\pool\plugins\org.eclipse.justj.openj
Sum is 7
```

[10] Write a simple CORBA program to implement client and server architecture. [Make necessary assumptions].

Source code:

```
package Lab6.RMI;

import java.rmi.*;

public interface Addition extends Remote {


	public int AddValue(int a,int b ) throws RemoteException;
	public int SubValue(int a,int b ) throws RemoteException;
}



package Lab6.RMI;

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

public class AdditionImp extends UnicastRemoteObject  implements
Addition,Serializable{

	public AdditionImp() throws RemoteException{};
	@Override
	public int AddValue(int a, int b) throws RemoteException {
		// TODO Auto-generated method stub
		return a+b;
	}
	@Override
	public int SubValue(int a, int b) throws RemoteException {
```

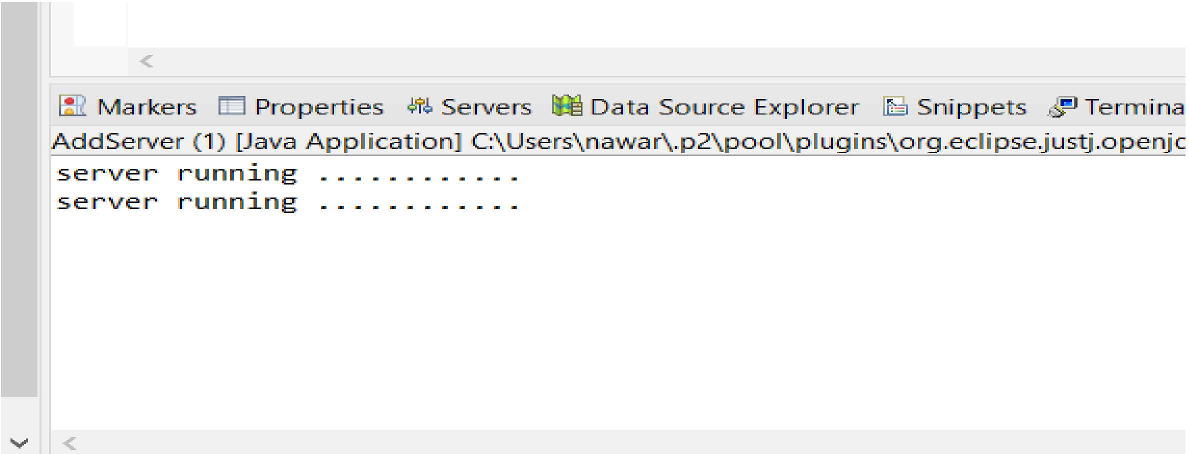
```
             // TODO Auto-generated method stub
             return a-b;
       }

}
```

```
package Lab6.RMI;

import java.net.MalformedURLException;
import java.rmi.*;
import java.rmi.registry.LocateRegistry;

public class AddServer {
       public static void main(String args[]) throws RemoteException,
MalformedURLException {

             try {
                    System.out.println("server running ............");
                    Addition ad = new AdditionImp();
              LocateRegistry.createRegistry(1099);
               Naming.rebind("localService", ad);
               System.out.println("server running ............");
             } catch (Exception e) {
                    // TODO: handle exception
                    System.out.println(e.getMessage());
             }

       }
```

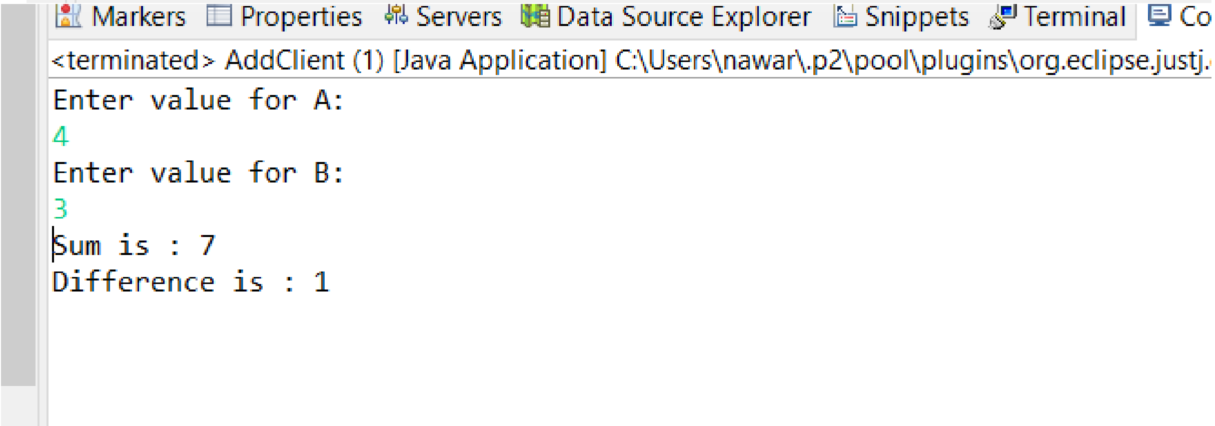
```
package Lab6.RMI;

import java.net.MalformedURLException;

import java.rmi.Naming;
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.util.Scanner;


public class AddClient {

       public static void main(String args[])  {
                             try {
                  // Lookup the remote object from the registry
                      Addition calc = (Addition)
Naming.lookup("rmi://localhost:1099/localService");
                      System.out.println("Enter value for A:");
                      int a,b;
```

```
           a=new Scanner(System.in).nextInt();

           System.out.println("Enter value for B:");
           b=new Scanner(System.in).nextInt();
                   int sum = calc.AddValue(a, b);
                int sub = calc.SubValue(a, b);
         System.out.println("Sum is : " + sum);
         System.out.println("Difference is : " + sub);
       } catch (Exception e) {
         System.out.println("Error in ClientAdd: " + e.getMessage());
       }
          }
}
```

Output:



```
AddServer (1) [Java Application] C:\Users\nawar\.p2\pool\plugins\org.eclipse.justj.openjc
server running ............
server running ............
```



```
<terminated> AddClient (1) [Java Application] C:\Users\nawar\.p2\pool\plugins\org.eclipse.justj.
Enter value for A:
4
Enter value for B:
3
Sum is : 7
Difference is : 1
```