Ex. No.	**:**	**Date**	**:**
Roll No.	**:**	**Reg. No.**	**:**

Invoking Servlet from HTML

Aim:

Program in Java using Servlets to invoke servlets from HTML forms.

Procedure:

1. The ServletRequest class includes methods that allow you to read the names and values of parameters that are included in a client request.
2. Create a Web page is defined in PostParameters.htm and a servlet is defined in PostParametersServlet.java.
3. Define a table that contains two labels and two text fields.
4. One of the labels is Employee and the other is Phone.
5. Create a submit button.
6. Set the action parameter of the form tag specifies a URL.
7. Define a service( ) method to process client requests.
8. Use the getParameterNames( ) method returns an enumeration of the parameter names. These are processed in a loop.
9. Display the parameter name and value are output to the client.
10. Obtain the parameter value via the getParameter( ) method.
11. Compile the servlet and perform these steps to test this example:
    1. Start Tomcat (if it is not already running).
    2. Display the Web page in a browser.
    3. Enter an employee name and phone number in the text fields.
    4. Submit the Web page.
12. After following these steps, the browser will display a response that is dynamically generated by the servlet.

Ex. No. : **Date :**

Roll No. : **Reg. No. :**

Servlet – Displaying the Student Details

Aim:

Programs in Java to create three-tier applications using Servlets for displaying student details by assuming that student information is available in a database which has been stored in a database server.

Procedure:

1. The HttpServletRequest class includes methods that allow you to read the names and values of parameters that are included in a client request.
2. Create a Web page is defined in Student.htm and a servlet is defined in Student.java.
3. Define a table that contains one label and one text fields.
4. Set the labels as Roll No..
5. Create a submit button.
6. Set the action parameter of the form tag specifies a URL.
7. Define a doPost( ) method to process client requests.
8. Obtain the parameter value via the getParameter( ) method.
9. Establish a connection with a database, load the database specific driver by calling the forName() method of the Class class:
   Class.forName(“com.mysql.jdbc.Driver”)
10. Use the getConnection() method from the DriverManager class to establish a connection with a database.:
    <protocol>:<subprotocol>:<subname>
11. Use the PreparedStatement object to execute parameterized queries.
12. Create the PreparedStatement object using the prepareStatement() method of the Connection object.
    stat = con.prepareStatement(“Select * from student where rollno = ?”) ;
13. Set the value of ‘?’ parameter, by calling an appropriate setXXX() method, where XXX is the datatype of the parameter. For example:
    stat.setString(1, mrollno) ;
    ResultSet result = stat.executeQuery() ;
14. Display the details as output to the client.

15. Compile the servlet and perform these steps to test this example:
    1. Start Tomcat (if it is not already running).
    2. Display the Web page in a browser.
    3. Enter an Roll No. in the text field.
    4. Submit the Web page.
16. After following these steps, the browser will display a response that is dynamically generated by the servlet.