

Practical 1

Aim: Create chat application using TCP protocol.

→ **server.java**

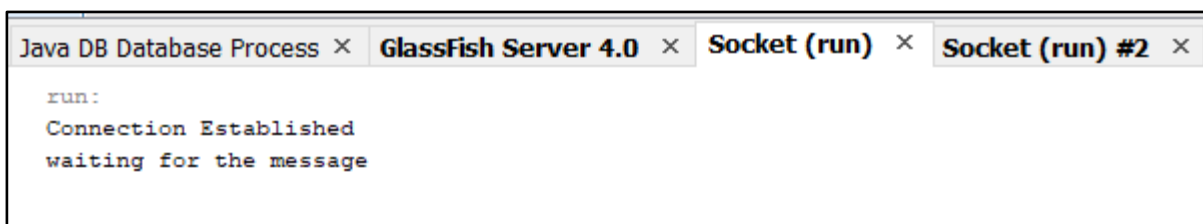
```
import java.io.*;
import java.net.*;

public class server {
    public static void main(String args[]) throws Exception {
        ServerSocket ss = new ServerSocket(1001);
        Socket s = ss.accept();
        System.out.println("Connection Established");
        System.out.println("waiting for the message");
        BufferedReader kb = new BufferedReader(new InputStreamReader(System.in));
        BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));
        PrintStream ps = new PrintStream(s.getOutputStream());
        while (true) {
            String str1, str2;
            while ((str1 = br.readLine()) != null) {
                System.out.println(str1);
                str2 = kb.readLine();
                ps.println(str2 + "\n");
            }
            ps.close();
            br.close();
            kb.close();
            ss.close();
            s.close();
            System.exit(0);
        }
    }
}
```

→client.java

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

public class client {
    public static void main(String args[]) throws Exception {
        System.out.println("write something to start communication");
        Socket s = new Socket("localhost", 1001);
        BufferedReader kb = new BufferedReader(new InputStreamReader(System.in));
        BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));
        DataOutputStream dos = new DataOutputStream(s.getOutputStream());
        String str1, str2;
        while (!(str1 = kb.readLine()).equals("exit")) {
            dos.writeBytes(str1 + "\n");
            str2 = br.readLine();
            System.out.println(str2);
        }
        dos.close();
        br.close();
        kb.close();
        s.close();
    }
}
```

Output:

```
run:
Connection Established
waiting for the message
```

```
Java DB Database Process × GlassFish Server 4.0 × Socket (run) × Socket (run) #2 ×  
run:  
write something to start communication  
Hello Server
```

```
Java DB Database Process × GlassFish Server 4.0 × Socket (run) × Socket (run) #2 ×  
run:  
Connection Established  
waiting for the message  
Hello Server
```

```
Output × HTTP Server Monitor  
Socket (run) × Socket (run) #2 ×  
run:  
Connection Established  
waiting for the message  
Hello Server  
Hello Client
```

```
Output × HTTP Server Monitor  
Socket (run) × Socket (run) #2 ×  
run:  
write something to start communication  
Hello Server  
Hello Client
```

Practical 2

Aim: Implement TCP Server for transferring files using Socket and ServerSocket.

→fileserver.java

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

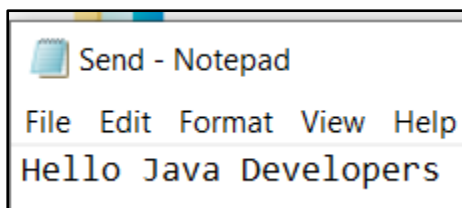
public class fileserver {
    public static void main(String args[]) throws Exception {
        ServerSocket ss = new ServerSocket(8888);
        Socket s = ss.accept();
        File file = new File("E:\\Send.txt");
        FileInputStream fis = new FileInputStream(file);
        BufferedInputStream bis = new BufferedInputStream(fis);
        OutputStream os = s.getOutputStream();
        byte[] contents;
        long fl = file.length();
        long current = 0;
        while (current != fl) {
            int size = 10000;
            if (fl - current >= size) {
                current += size;
            } else {
                size = (int) (fl - current);
                current = fl;
            }
            contents = new byte[size];
            bis.read(contents, 0, size);
            os.write(contents);
            System.out.println("sending file..." + (current * 100) / fl + "% complete!");
        }
        os.flush();
        ss.close();
    }
}
```

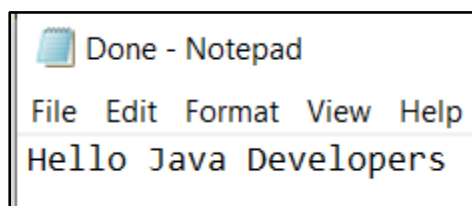
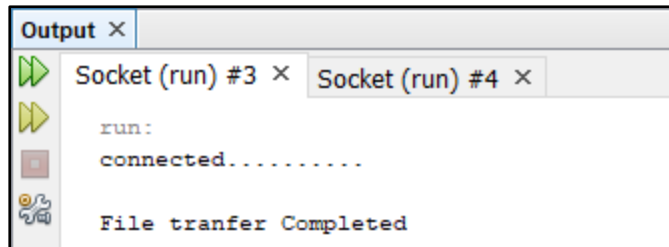
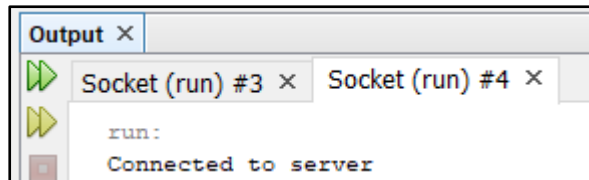
```
s.close();  
System.out.println("file sent successfully!");  
}  
}
```

→fileclient.java

```
import java.io.*;  
import java.net.*;  
  
public class fileclient {  
    public static void main(String args[]) throws Exception {  
        Socket s = new Socket("localhost", 8888);  
        byte[] contents = new byte[10000];  
        FileOutputStream fos = new FileOutputStream("E:\\\\Done.txt");  
        BufferedOutputStream bos = new BufferedOutputStream(fos);  
        InputStream is = s.getInputStream();  
        int byteread = 0;  
        while ((byteread = is.read(contents)) != -1) {  
            bos.write(contents, 0, byteread);  
            bos.flush();  
            s.close();  
            System.out.println("file retrieve successfully!");  
        }  
    }  
}
```

Output:





Practical 3

Aim: Implement any one sorting algorithm using TCP/UDP on server application and give input on client side and client should get sorted output from server side and display sorted input client side.

→ascserver.java

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

class ascserver {
    int a[] = new int[25];

    void sort(int n, int a[]) {
        int i, j, temp;
        for (i = 0; i < n - 1; i++) {
            for (j = 0; j < n - 1; j++) {
                if (a[j] > a[j + 1]) {
                    temp = a[j];
                    a[j] = a[j + 1];
                    a[j + 1] = temp;
                }
            }
        }
    }

    public static void main(String argv[]) throws Exception {
        ServerSocket ss = new ServerSocket(6789);
        while (true) {
            Socket s = ss.accept();
            BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));
            DataOutputStream out = new DataOutputStream(s.getOutputStream());
            ascserver as = new ascserver();
```

```

System.out.println("Waiting for array elements");
int n = br.read();
for (int i = 0; i < n; i++) {
    as.a[i] = br.read();
    System.out.println(as.a[i]);
}
as.sort(n, as.a);
for (int i = 0; i < n; i++) {
    out.write(as.a[i]);
}
System.out.println("array in sorted form is sent");
}
}
}

```

→ascclient.java

```

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

class ascclient {
    public static void main(String argv[]) throws Exception {
        int i, n, x;
        int a[] = new int[25];
        Scanner sc = new Scanner(System.in);
        Socket s = new Socket("localhost", 6789);
        DataOutputStream out = new DataOutputStream(s.getOutputStream());
        BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));
        System.out.println("enter n");
        n = sc.nextInt();
        System.out.println("total no of elements are " + n);
        out.write(n);
        System.out.println("Enter array");
    }
}

```

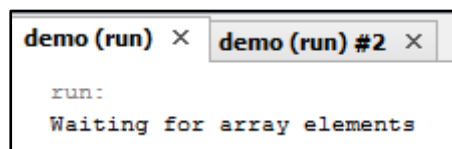


```

for (i = 0; i < n; i++) {
    x = sc.nextInt();
    out.write(x);
}
System.out.println("Sorted Array.");
for (i = 0; i < n; i++) {
    a[i] = br.read();
    System.out.println(a[i]);
}
s.close();
}
}

```

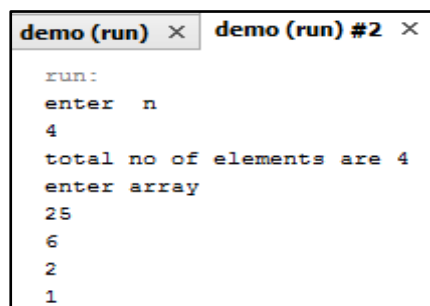
Output:



```

demo (run) × demo (run) #2 ×
run:
Waiting for array elements

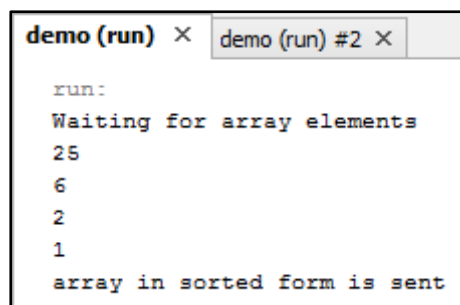
```



```

demo (run) × demo (run) #2 ×
run:
enter n
4
total no of elements are 4
enter array
25
6
2
1

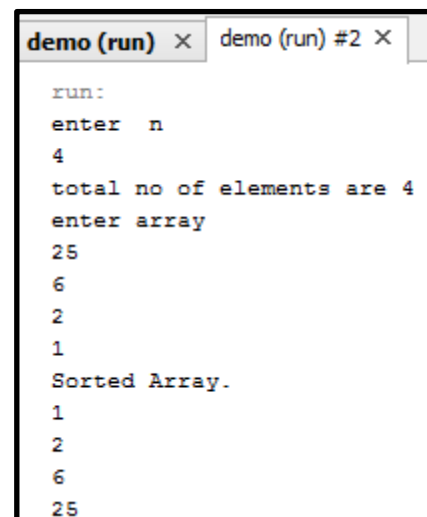
```



```

demo (run) × demo (run) #2 ×
run:
Waiting for array elements
25
6
2
1
array in sorted form is sent

```



```

demo (run) × demo (run) #2 ×
run:
enter n
4
total no of elements are 4
enter array
25
6
2
1
Sorted Array.
1
2
6
25

```

Practical 4

Aim: Write a UDP socket program for Sending DatagramPacket by DatagramSocket.

→**DSender.java**

```
import java.net.*;

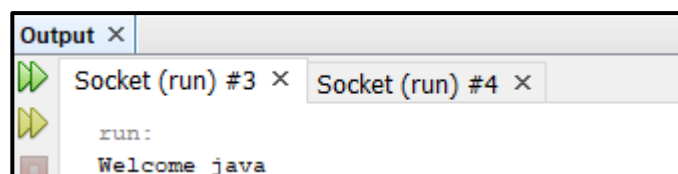
public class DSender
{
    public static void main(String[] args) throws Exception
    {
        DatagramSocket ds = new DatagramSocket();
        String str = "Welcome java";
        InetAddress ip = InetAddress.getByName("127.0.0.1");
        DatagramPacket dp = new DatagramPacket(str.getBytes(), str.length(), ip, 3000); ds.send(dp);
        ds.close();
    }
}
```

→**DReceiver.java**

```
import java.net.*;

public class DReceiver {
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket(3000);
        byte[] buf = new byte[1024];
        DatagramPacket dp = new DatagramPacket(buf, 1024);
        ds.receive(dp);
        String str = new String(dp.getData(), 0, dp.getLength());
        System.out.println(str);
        ds.close();
    }
}
```

Output:



Practical 5

Aim: Create new database, create a new table under that database and use java PreparedStatement to insert records in table.

→ Prepared.java

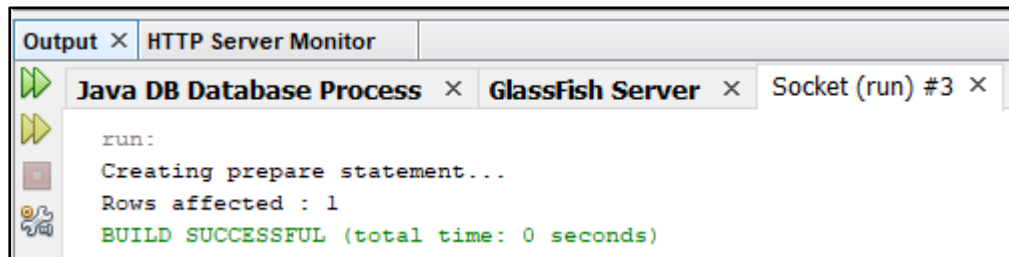
```
import java.sql.*;

public class Prepared {
    public static void main(String[] args)
        throws SQLException, ClassNotFoundException, InstantiationException,
        IllegalAccessException {

        Class.forName("com.mysql.jdbc.Driver");
        Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/studdetail",
            "root", "");
        System.out.println("Creating prepare statement...");
        String sql = "insert into mydetail values(?,?)";
        PreparedStatement stmt = conn.prepareStatement(sql);
        stmt.setInt(1, 103);
        stmt.setString(2, "Niraj");
        int rows = stmt.executeUpdate();
        System.out.println("Rows affected : " + rows);
        stmt.close();
        conn.close();
    }
}
```

Output:

eno	name
101	palak
102	pratik



eno	name
101	palak
102	pratik
103	niraj

Practical 6

Aim: Create new database, create a new table under that database and use java Statement to update records of table.

→ State.java

```
import java.sql.*;
import java.util.*;

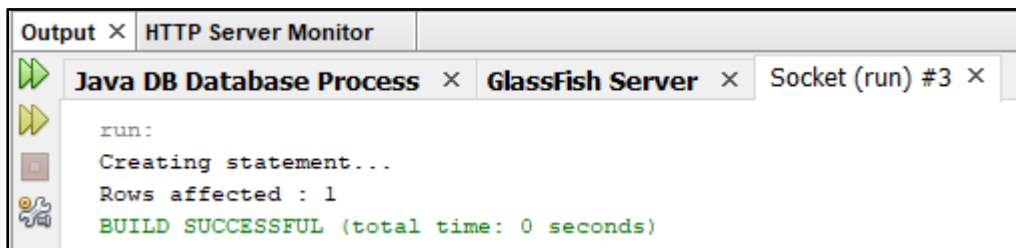
public class State {

    public static void main(String[] args)
        throws SQLException, ClassNotFoundException, InstantiationException,
        IllegalAccessException {

        Class.forName("com.mysql.jdbc.Driver");
        Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studdetail",
            "root", "");
        System.out.println("Creating statement...");
        Statement stmt = conn.createStatement();
        String sql = "update mydetail set name=elon WHERE eno=102";
        int rows = stmt.executeUpdate(sql);
        System.out.println("Rows affected : " + rows);
        stmt.close();
        conn.close();
    }
}
```

Output:

eno	name
101	palak
102	pratik
103	niraj



eno	name
101	palak
102	elon
103	niraj

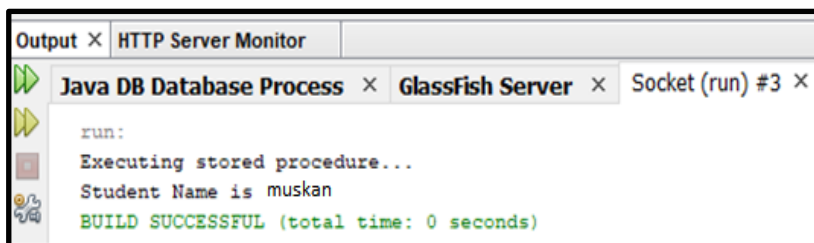
Practical 7

Aim: Create new database, create stored procedure to retrieve student name as output when pass enrollment number as input under that database and use java CallableStatement to retrieve the records.

```
import java.sql.*;

public class Callable {
    public static void main(String[] args)
        throws SQLException, ClassNotFoundException, InstantiationException,
        IllegalAccessException {
        Connection conn =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/studdetail", "root","");
        CallableStatement stmt = conn.prepareCall("{call getStdName (?, ?)}");
        stmt.setInt(1, 105);
        stmt.registerOutParameter(2, java.sql.Types.VARCHAR);
        System.out.println("Executing stored procedure...");
        stmt.execute();
        String stdName = stmt.getString(2);
        System.out.println("Student Name is " + stdName);
    }
}
```

Output:



eno	name
101	palak
101	elon
103	niraj
105	muskan
106	janardhan

Practical 8

Aim:Create servlet file to perform following operation on database insert records, update records, display record, delete record, delete table, delete database.

→index.html

```
<html>
<body>
  <form action="insert">
    <h1>Student Details</h1>
    <table>
      <tr>
        <td>Student Id</td>
        <td><input type="text" name="sid" /></td>
      </tr>
      <tr>
        <td>Student Name</td>
        <td><input type="text" name="sname" /></td>
      </tr>
      <tr>
        <td>Address</td>
        <td><textarea rows="5" cols="20" name="sadd"></textarea></td>
      </tr>
      <tr>
        <td>Contact No</td>
        <td><input type="text" name="scno" /></td>
      </tr>
      <tr>
        <td>Email Id</td>
        <td><input type="text" name="seid" /></td>
      </tr>
      <tr>
        <td><input type="submit" value="Insert" /></td>
      </tr>
    </table>
  </form>
</body>
</html>
```



```

<tr>
    <td><a href="update.html">Update</a></td>
</tr>
<tr>
    <td><a href="display.html">Display</a></td>
</tr>
<tr>
    <td><a href="deleterecord.html">Delete Record</a></td>
</tr>
<tr>
    <td><a href="deletetable.html">Delete Table</a></td>
</tr>
<tr>
    <td><a href="deletedatabase.html">Delete Database</a></td>
</tr>
</table>
</form>
</body>
</html>

```

→insert.java

```

import java.io.*;
import java.sql.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```
@WebServlet(urlPatterns = { "/insert" })
```

```

public class insert extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            try {
                out.println("Record Inserted Successfully<br>");
                int sid = Integer.parseInt(request.getParameter("sid"));
                String sname = request.getParameter("sname");
                String sadd = request.getParameter("sadd");
                String scno = request.getParameter("scno");
                String seid = request.getParameter("seid");
                out.println("<a href='update.html' >Update</a><br>");
                out.println("<a href='display.html' >Display</a></br>");
                out.println("<a href='deleterecord.html' >Delete record</a><br>");
                out.println("<a href='deletetable.html' >Delete table</a><br>");
                out.println("<a href='deletedatabase.html' >Delete database</a><br>");
                Class.forName("com.mysql.jdbc.Driver");
                Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/stud",
                    "root", "");
                Statement stmt = con.createStatement();
                String query = "insert into std values('" + sid + "','" + sname + "','" + sadd + "','" + scno +
                    "','" + seid + "')";
                stmt.executeUpdate(query);
                stmt.close();
                con.close();

            } catch (SQLException e) {
                System.err.println("Exception:" + e.getMessage());
            } catch (ClassNotFoundException ex) {
                Logger.getLogger(insert.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
}

```

```

    }
}
}

```

→update.html

```

<html>
<body>
    <div>
        <h1 align="centre">update</h1>
        <form action="update">
            <table>
                <tr>
                    <td>Student Id</td>
                    <td><input type="text" name="sid" /></td>
                </tr>
                <tr>
                    <td>Student Name</td>
                    <td><input type="text" name="sname" /></td>
                </tr>
                <tr>
                    <td>Address</td>
                    <td><textarea rows="5" cols="20" name="sadd"></textarea></td>
                </tr>
                <tr>
                    <td>Contact No</td>
                    <td><input type="text" name="scno" /></td>
                </tr>
                <tr>
                    <td>Email Id</td>
                    <td><input type="text" name="seid" /></td>
                </tr>
                <tr>
                    <td><input type="submit" value="update" /></td>
                </tr>
            </table>
        </form>
    </div>

```

```

        </table>
    </form>
</div>
</body>
</html>

```

→update.java

```

import java.io.*;
import java.sql.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(urlPatterns = { "/update" })
public class update extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            try {
                int id = Integer.parseInt(request.getParameter("sid"));
                String name = request.getParameter("sname");
                String add = request.getParameter("sadd");
                int cno = Integer.parseInt(request.getParameter("scno"));
                String eid = request.getParameter("seid");
                out.println("<a href='index.html' >Insert</a><br>");
                out.println("<a href='update.html' >Update</a><br>");
            }
        }
    }
}

```

```

out.println("<a href='display.html' >Display</a></br>");
out.println("<a href='deleterecord.html' >Delete record</a></br>");
out.println("<a href='deletetable.html' >Delete table</a></br>");
out.println("<a href='deletedatabase.html' >Delete database</a></br>");
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/stud",
    "root", "");
Statement stmt = con.createStatement();
String query = "UPDATE std SET sname=" + name + ",sadd=" + add + ",scno=" + cno
    + ",seid=" + eid
    + " WHERE sid=" + id + " ";
stmt.executeUpdate(query);
stmt.close();
con.close();
out.println("Records are updated....<br>");
} catch (SQLException e) {
    System.err.println("Exception:" + e.getMessage());
} catch (ClassNotFoundException ex) {
    Logger.getLogger(insert.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
}

```

→display.html

```

<html>
<body>
<div>
    <form action="display">
        To display table click on the button <input type="submit" value="Display records">
    </form>
</div>
</body>
</html>

```

→display.java

```

import java.io.*;
import java.sql.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(urlPatterns = { "/display" })
public class display extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/stud",
                "root", "");
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<body>");
            // out.println(request.getContextPath());
            Statement stmt = null;
            ResultSet rs = null;
            stmt = con.createStatement();
            String q = "select * from std";
            rs = stmt.executeQuery(q);
            out.println("<table border='1'>");
            out.println("<tr>");
            out.println("<th>Student No</th>");
            out.println("<th>Student Name</th>");

```

```

out.println("<th>Student Address</th>");
out.println("<th>Student ContactNo</th>");
out.println("<th>Student Email</th>");
out.println("</tr>");
while (rs.next()) {
    int id = rs.getInt(1);
    String name = rs.getString(2);
    String add = rs.getString(3);
    String cno = rs.getString(4);
    String eid = rs.getString(5);
    out.println("<tr>");
    out.println("<td>" + id + "</td>");
    out.println("<td>" + name + "</td>");
    out.println("<td>" + add + "</td>");
    out.println("<td>" + cno + "</td>");
    out.println("<td>" + eid + "</td>");
    out.println("</tr>");
}
out.println("</table>");
out.println("<a href='index.html' >Insert</a></br>");
out.println("<a href='update.html' >Update</a></br>");
out.println("<a href='deleterecord.html' >Delete record</a></br>");
out.println("<a href='deletetable.html' >Delete table</a></br>");
out.println("<a href='deletedatabase.html' >Delete database</a></br>");
out.println("</body>");
out.println("</html>");
rs.close();
stmt.close();
} catch (SQLException e) {
    System.err.println("Exception:" + e.getMessage());
} catch (ClassNotFoundException ex) {
    Logger.getLogger(insert.class.getName()).log(Level.SEVERE, null, ex);
}

```

```

    }
}

```

→deleterecord.html

```

<html>
<body>
    <div>
        <h1 align="centre">DELETE</h1>
        <form action="deleterecord">
            Student id: <input type="text" name="sid"><br><br>
            <input type="submit" value="Delete Record">
        </form>
    </div>
</body>
</html>

```

→deleterecord.java

```

import java.io.*;
import java.sql.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(urlPatterns = { "/deleterecord" })
public class deleterecord extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

```



```

try (PrintWriter out = response.getWriter()) {
    try {
        int id = Integer.parseInt(request.getParameter("sid"));
        out.println(id + " Record is deleted...<br>");
        out.println("<a href='index.html' >Insert</a><br>");
        out.println("<a href='update.html' >Update</a><br>");
        out.println("<a href='display.html' >Display</a><br>");
        out.println("<a href='deleterecord.html' >Delete record</a><br>");
        out.println("<a href='deletetable.html' >Delete table</a><br>");
        out.println("<a href='deletedatabase.html' >Delete database</a><br>");
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/stud",
            "root", "");
        Statement stmt = con.createStatement();
        String query = "delete from std WHERE sid=" + id + "";
        stmt.executeUpdate(query);
        stmt.close();
        con.close();
    } catch (SQLException e) {
        System.err.println("Exception:" + e.getMessage());
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(insert.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```

→deletetable.html

```

<html>
<body>
    <div>
        <form action="deletetable">
            To delete table click on button <input type="submit" value="Delete table">
        </form>
    </div>

```

```
</body>
```

```
</html>
```

→deletetable.java

```
import java.io.*;
import java.sql.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(urlPatterns = { "/deletetable" })
public class deletetable extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            try {
                out.println("Table is deleted...<br>");
                out.println("<a href='index.html' >Insert</a><br>");
                out.println("<a href='update.html' >Update</a><br>");
                out.println("<a href='display.html' >Display</a><br>");
                out.println("<a href='deleterecord.html' >Delete record</a><br>");
                out.println("<a href='deletetable.html' >Delete table</a><br>");
                out.println("<a href='deletedatabase.html' >Delete database</a><br>");
                Class.forName("com.mysql.jdbc.Driver");
                Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/stud",
                    "root", "");
                Statement stmt = conn.createStatement();
```

```

        String query = "drop table std";
        stmt.executeUpdate(query);
        stmt.close();
        conn.close();
    }

    catch (SQLException e) {
        System.err.println("Exception:" + e.getMessage());
    }
} catch (ClassNotFoundException ex) {
    Logger.getLogger(insert.class.getName()).log(Level.SEVERE, null, ex);
}

}
}

```

→deletedatabase.html

```

<html>
<body>
    <div>
        <form action="deletedatabase">
            To delete database click on button <input type="submit" value="Delete databse">
        </form>
    </div>
</body>
</html>

```

→deletedatabase.java

```

import java.io.*;
import java.sql.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.*;

```

```

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

@WebServlet(urlPatterns = { "/deletedatabase" })
public class deletedatabase extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            try {
                out.println("Database is deleted...<br>");
                out.println("<a href='index.html' >Insert</a><br>");
                out.println("<a href='update.html' >Update</a><br>");
                out.println("<a href='display.html' >Display</a><br>");
                out.println("<a href='deleterecord.html' >Delete record</a><br>");
                out.println("<a href='deletetable.html' >Delete table</a><br>");
                out.println("<a href='deletedatabase.html' >Delete database</a><br>");
                Class.forName("com.mysql.jdbc.Driver");
                Connection conn = DriverManager.getConnection(
                    "jdbc:mysql://localhost:3306/stud?zeroDateTimeBehavior=convertToNull [root on
                    Default schema]",
                    "root", "");
                Statement stmt = conn.createStatement();
                String query = "DROP database stud";
                stmt.executeUpdate(query);
                stmt.close();
                conn.close();
            }

            catch (SQLException e) {

```

```

        System.err.println("Exception:" + e.getMessage());
    }
} catch (ClassNotFoundException ex) {
    Logger.getLogger(insert.class.getName()).log(Level.SEVERE, null, ex);
}

}
}

```

→web.xml

```

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

```

```
<servlet-name>update</servlet-name>
<url-pattern>/update</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>display</servlet-name>
  <servlet-class>display</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>display</servlet-name>
  <url-pattern>/display</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>deleterecord</servlet-name>
  <servlet-class>deleterecord</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>deleterecord</servlet-name>
  <url-pattern>/deleterecord</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>deletetable</servlet-name>
  <servlet-class>deletetable</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>deletetable</servlet-name>
  <url-pattern>/deletetable</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>deletedatabase</servlet-name>
  <servlet-class>deletedatabase</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>deletedatabase</servlet-name>
```

```

<url-pattern>/deletedatabase</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
</welcome-file-list>
</web-app>

```

Output:

localhost / localhost / stud / std | x

localhost:8080/Ser

Import favorites | Gmail | YouTube

Student Details

Student Id

Student Name

Address

Contact No

Email Id

[Update](#)

[Display](#)

[Delete Record](#)

[Delete Table](#)

[Delete Database](#)

sid	sname	sadd	scno	seid
101	Jay	Surat	2147483647	123@gmail.com

localhost / localhost / stud / std | X

localhost:8080/ServletA

Import favorites | Gmail | YouTube

update

Student Id

Student Name

Address

Contact No

Email Id

localhost / localhost / stud / std | X

localhost:8080/Se

Import favorites | Gmail | You

Record Inserted Successfully

[Update](#)

[Display](#)

[Delete record](#)

[Delete table](#)

[Delete database](#)

sid	sname	sadd	scno	seid
101	Jay	Surat	2147483647	123@gmail.com
102	kunj	surat	78544585	kunj@yahoo.com

localhost / localhost / stud / std | X

localhost:8080/Servlet

Import favorites | Gmail | YouTube

[Insert](#)

[Update](#)

[Display](#)

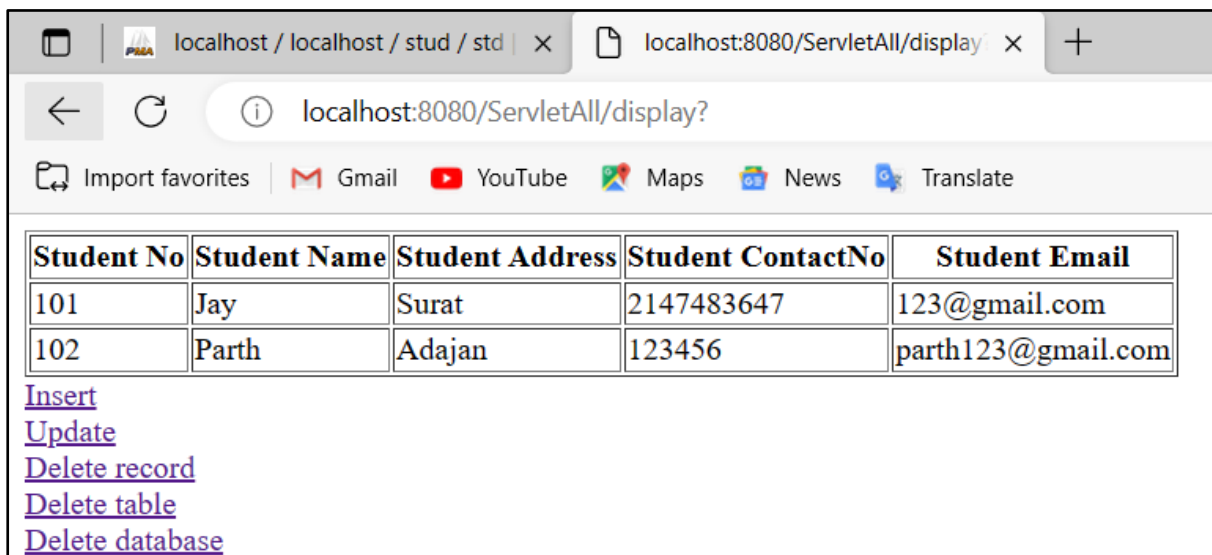
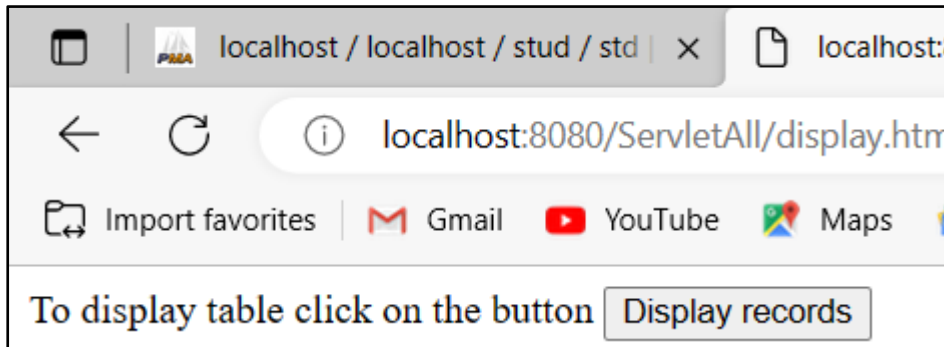
[Delete record](#)

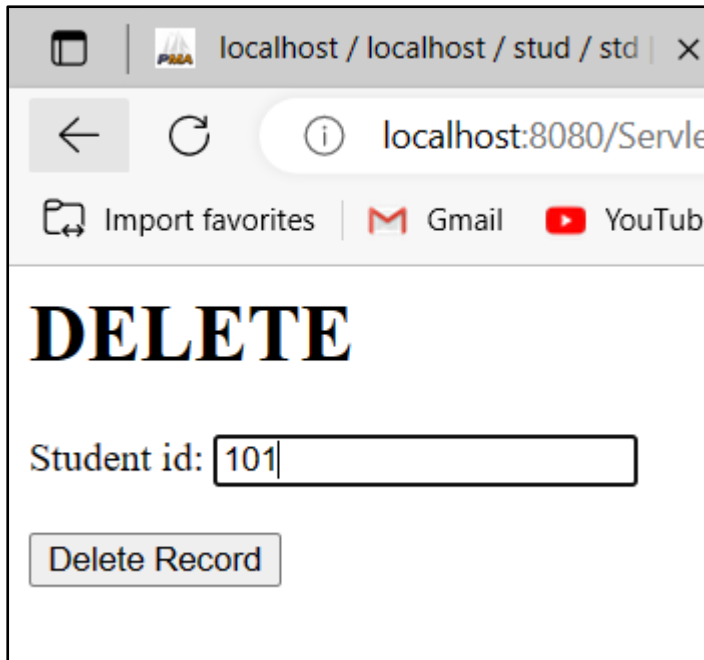
[Delete table](#)

[Delete database](#)

Records are updated....

sid	sname	sadd	scno	seid
101	Jay	Surat	2147483647	123@gmail.com
102	Parth	Adajan	123456	parth123@gmail.com





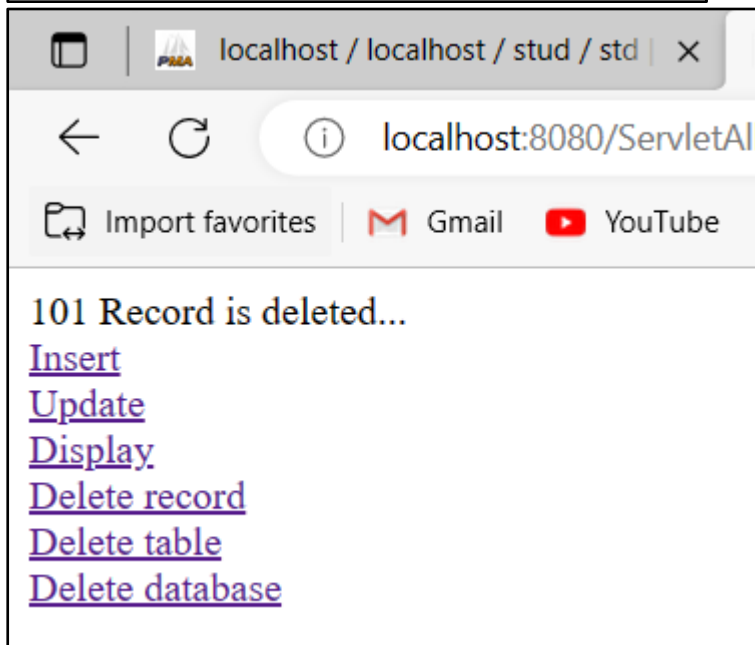
localhost / localhost / stud / std | X

localhost:8080/Servlet

Import favorites | Gmail | YouTube

DELETE

Student id:



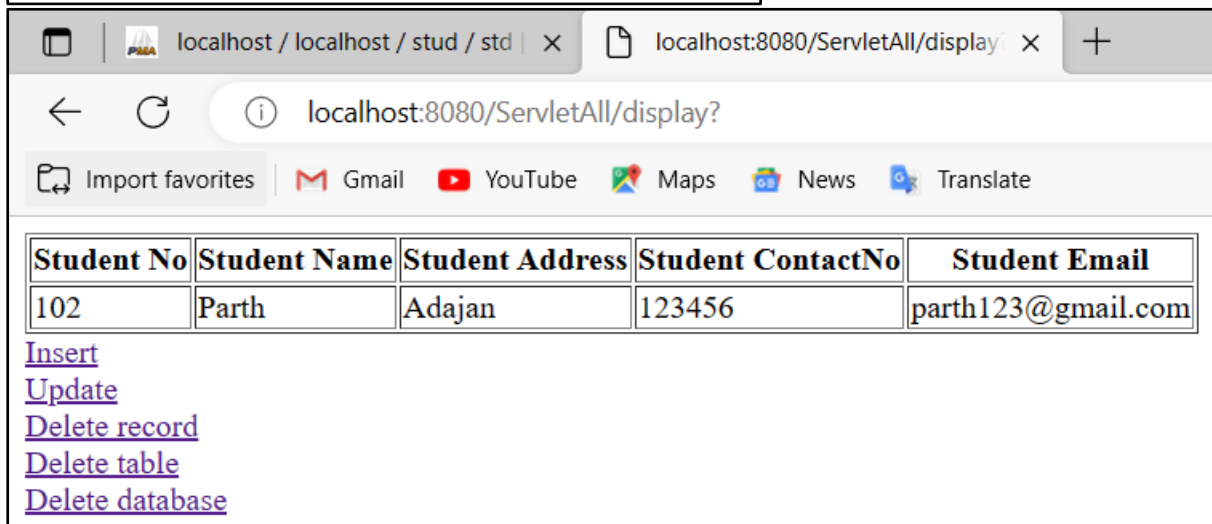
localhost / localhost / stud / std | X

localhost:8080/ServletAll

Import favorites | Gmail | YouTube

101 Record is deleted...

[Insert](#)
[Update](#)
[Display](#)
[Delete record](#)
[Delete table](#)
[Delete database](#)



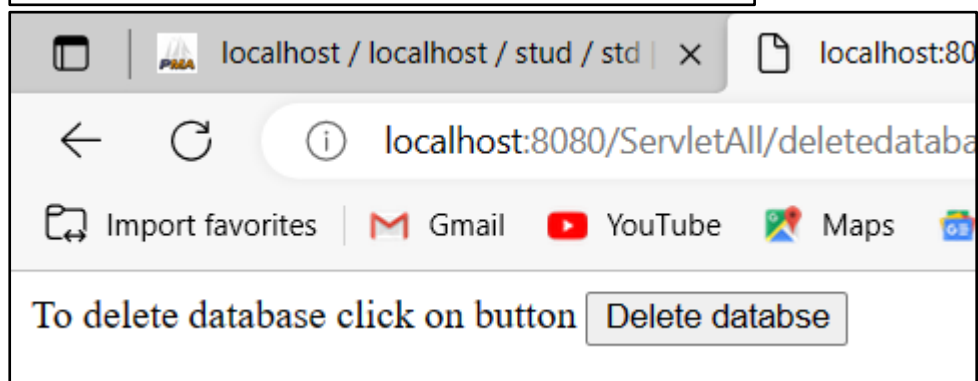
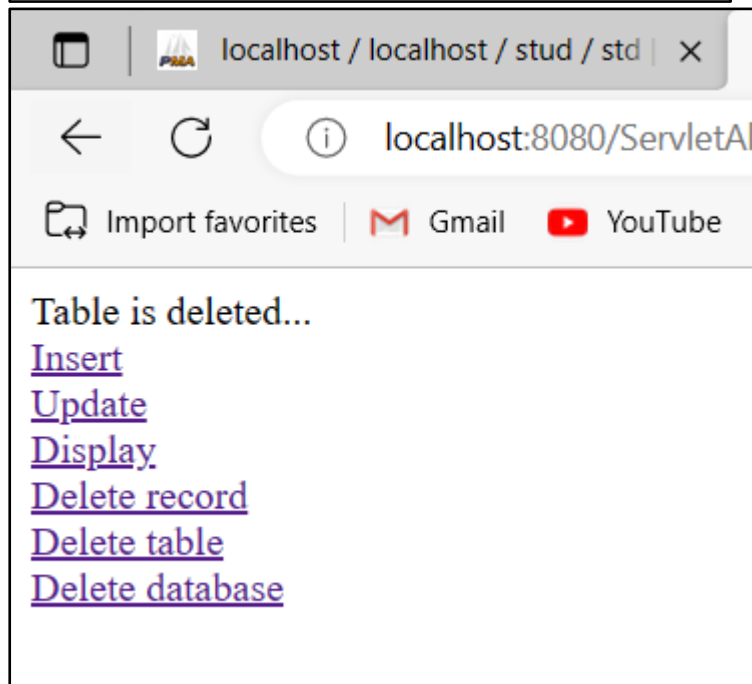
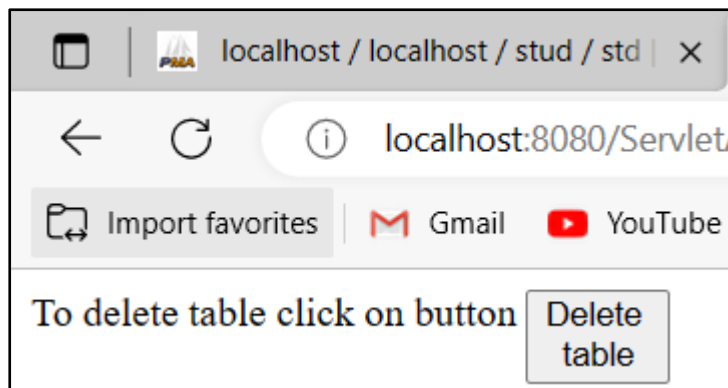
localhost / localhost / stud / std | X | localhost:8080/ServletAll/display | X

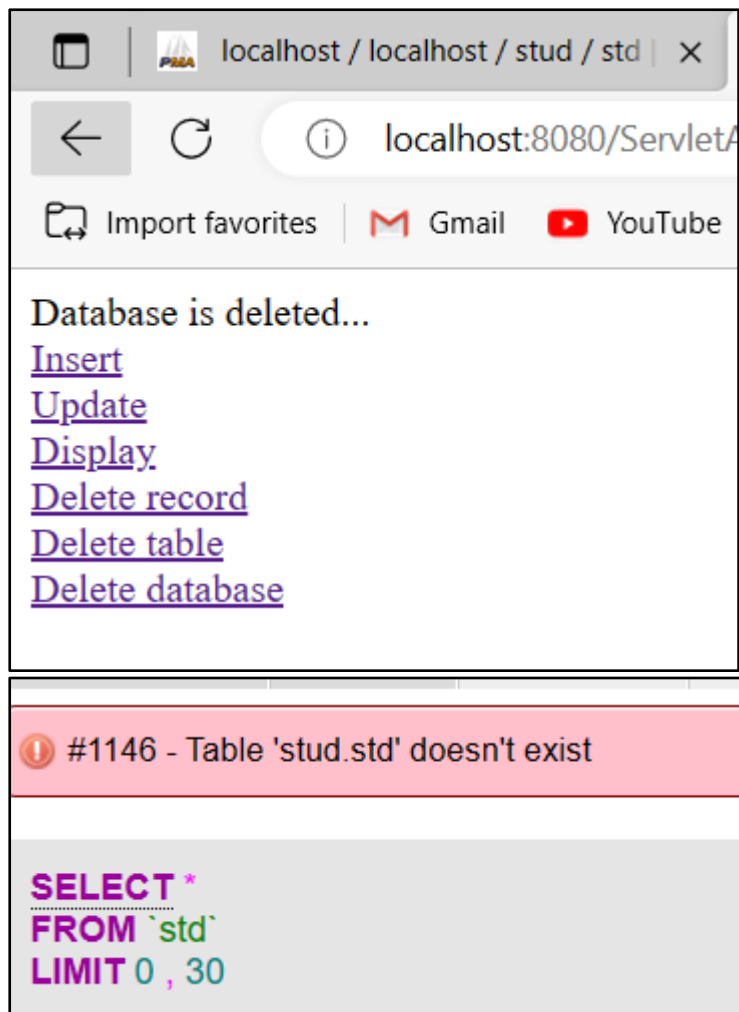
localhost:8080/ServletAll/display?

Import favorites | Gmail | YouTube | Maps | News | Translate

Student No	Student Name	Student Address	Student ContactNo	Student Email
102	Parth	Adajan	123456	parth123@gmail.com

[Insert](#)
[Update](#)
[Delete record](#)
[Delete table](#)
[Delete database](#)





Practical 9

Aim: Implement authentication filter using filters API.

→index.html

```
<html>
<body>
  <form action="first">
    Username:<input type="text" name="user" /><br />
    Password:<input type="text" name="pass" /><br />
    <input type="submit" value="submit" />
  </form>
</body>
</html>
```

→myfilter.java

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

public class myfilter implements Filter {
    private Object filterConfig;

    public void init(FilterConfig fc) throws ServletException {
    }

    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
        throws IOException, ServletException {
        PrintWriter out = response.getWriter();
        String pass = request.getParameter("pass");
        if (pass.equals("1234")) {
            chain.doFilter(request, response);
        } else {
            out.println("You have entered a wrong password");
        }
    }

    public void destroy() {
    }

    public myfilter() {
        this.filterConfig = null;
    }
}
```

→first.java

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class first extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<body>");
            PrintWriter o = response.getWriter();
            String user = request.getParameter("user");
            out.println("Welcome " + user);
            out.println("</body>");
            out.println("</html>");
        }
    }
}

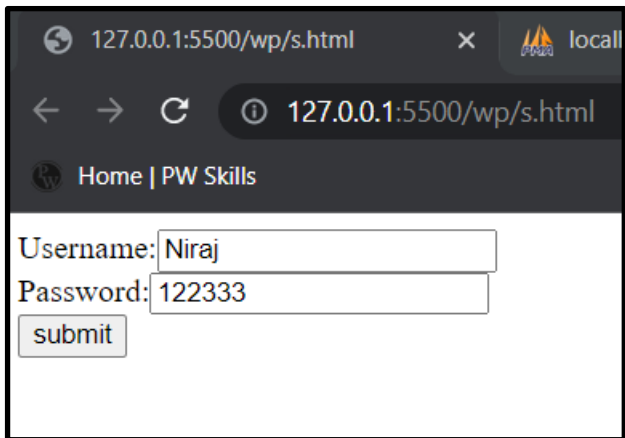
```

→web.xml

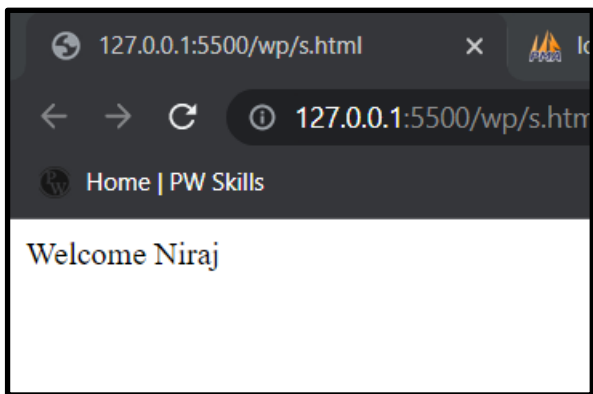
```

<?xml version="1.0" encoding="UTF-8"?>
<web-app>
    <filter>
        <filter-name>myfilter</filter-name>
        <filter-class>myfilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>myfilter</filter-name>
        <servlet-name>first</servlet-name>
        <url-pattern>/first</url-pattern>
    </filter-mapping>
    <servlet>
        <servlet-name>first</servlet-name>
        <servlet-class>first</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>first</servlet-name>
        <url-pattern>/first</url-pattern>
    </servlet-mapping>
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
    </welcome-file-list>
</web-app>

```

Output:

A screenshot of a web browser window. The address bar shows '127.0.0.1:5500/wp/s.html'. The page has a dark header with 'Home | PW Skills'. Below the header is a login form with two input fields: 'Username:' containing 'Niraj' and 'Password:' containing '122333'. A 'submit' button is located below the password field.



A screenshot of a web browser window. The address bar shows '127.0.0.1:5500/wp/s.html'. The page has a dark header with 'Home | PW Skills'. Below the header, the text 'Welcome Niraj' is displayed.

Practical 10

Aim: Implement student registration form with enrollment number, first name, last name, semester, contact number. Store the details in database. Also implement search, delete and modify facility for student records.

→Form.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1" %>
```

```
<html>
```

```
<body>
```

```
<form action="Insert.jsp">
```

```
<table border="1px solid black">
```

```
<tr>
```

```
<td>ENROLLMENT_NUMBER: </td>
```

```
<td><input type="text" name="en"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>FIRSTNAME: </td>
```

```
<td><input type="text" name="fn"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>LASTNAME: </td>
```

```
<td><input type="text" name="ln"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>SEMESTER: </td>
```

```
<td><input type="text" name="se"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>CONTACT_NUMBER: </td>
```

```
<td><input type="text" name="no"></td>
```

```
</tr>
```

```
<tr>
```



```

        <td><input type="submit" value="Save"></td>
    </tr>
</table>
</form>
<a href="Search.jsp">Search</a>
</body>
</html>

```

→Insert.jsp

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-
8859-1" import="java.sql.*" %>

```

```

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Insert title here</title>
</head>
<body>
    <% String en=request.getParameter("en"); String fn=request.getParameter("fn"); String
        ln=request.getParameter("ln"); String se=request.getParameter("se"); String
        no=request.getParameter("no");
        Class.forName("com.mysql.jdbc.Driver"); Connection
        c=DriverManager.getConnection("jdbc:mysql://localhost:3306/studreg","root","");
Statement
        s2=c.createStatement(); s2.executeUpdate("insert into

studrecord(Eno,Fname,Lname,Sem,Contact)values('"+en+"','"+fn+"','"+ln+"','"+se+"','"+no+"')");
s2.close();
        c.close(); response.sendRedirect("Form.jsp"); %>
</body>
</html>

```

→Edit.jsp

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-
8859-1" import="java.sql.*" %>

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Insert title here</title>
</head>
<body>
    <% int id=Integer.parseInt(request.getParameter("y")); int i=0;
Class.forName("com.mysql.jdbc.Driver");
    Connection
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/studreg","root",""); Statement
    st=c.createStatement(); ResultSet rs=st.executeQuery("select * from studrecord where
Eno="+ id);
%>
<%
while(rs.next())
{
String en = rs.getString(" Eno"); String fn=rs.getString("Fname"); String ln=rs.getString("Lname");
String
    se=rs.getString("Sem"); String no=rs.getString("Contact"); %>
    <form action="Update.jsp">
        <table border="1px solid black">
            <tr>
                <td><input type="hidden" name="id1" value='<%=id%>'></td>
            </tr>
            <tr>
                <td>ENROLLMENT_NUMBER: </td>
                <td><input type="text" name="en" value="<%=en%>"></td>
            </tr>
            <tr>

```

```

        <td>FIRSTNAME: </td>
        <td><input type="text" name="fn" value="<%=fn%>"></td>
    </tr>
    <tr>
        <td>LASTNAME: </td>
        <td><input type="text" name="ln" value="<%=ln%>"></td>
    </tr>
    <tr>
        <td>SEMESTER: </td>
        <td><input type="text" name="se" value="<%=se%>"></td>
    </tr>
    <tr>
        <td>CONTACT_NUMBER: </td>
        <td><input type="text" name="no" value="<%=no%>"></td>
    </tr>
    <tr>
        <td><input type="submit" value="Update"></td>
    </tr>
</table>
</form>
<% } c.close(); st.close(); %>
</body>
</html>

```

→Update.jsp

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-
8859-1" import="java.sql.*" %>

```

```

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Insert title here</title>
</head>

```

```

<body>
    <% int id=Integer.parseInt( request.getParameter("id1")); String
en=request.getParameter("en"); String
    fn=request.getParameter("fn"); String ln=request.getParameter("ln"); String
se=request.getParameter("se");
    String no=request.getParameter("no"); Class.forName("com.mysql.jdbc.Driver");
Connection
    c=DriverManager.getConnection("jdbc:mysql://localhost:3306/studreg","root","");
Statement
    st=c.createStatement(); st.executeUpdate("update studrecord set Eno='"+en+"",
Fname='"+fn+"",
    ,Lname='"+ln+"",Sem='"+se+"", Contact='"+no+" where Eno='"+id+"");
st.close();
c.close();
response.sendRedirect(" Search.jsp"); %>
</body>
</html>

```

→Serach.jsp

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-
8859-1" import="java.sql.*" %>

```

```

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Insert title here</title>
</head>
<body>
    <% Class.forName("com.mysql.jdbc.Driver"); Connection
    c=DriverManager.getConnection("jdbc:mysql://localhost:3306/studreg","root","");
Statement
    st=c.createStatement(); ResultSet rs=st.executeQuery("select * from studrecord"); %>

```

```

<table border="1px solid black">
  <tr>
    <th>ENROLLMENT_NUMBER</th>
    <th>FIRSTNAME</th>
    <th>LASTNAME</th>
    <th>SEMESTER</th>
    <th>CONTACT_NUMBER</th>
    <th>Action</th>
    <th>Action</th>
  </tr>
  <% while(rs.next()) { String en=rs.getString("Eno"); String fn=rs.getString("Fname");
String
                                ln=rs.getString("Lname");          String          se=rs.getString("Sem");          String
no=rs.getString("Contact"); %>
    <tr>
      <td>
        <% out.println(en);%>
      </td>
      <td>
        <% out.println(fn);%>
      </td>
      <td>
        <% out.println(ln);%>
      </td>
      <td>
        <% out.println(se);%>
      </td>
      <td>
        <% out.println(no);%>
      </td>
      <td><a href="Delete.jsp?x=<%=en%>">Delete</a></td>
      <td><a href="Edit.jsp?y=<%=en%>">Edit</a></td>
    </tr>

```

```
<% } %>
```

```
</body>
```

```
</html>
```

→Delete.jsp

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1" import="java.sql.*" %>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

```
</head>
```

```
<body>
```

```
<% int id=Integer.parseInt(request.getParameter("x"));
```

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

```
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/studreg","root","");
```

```
Statement
```

```
st=c.createStatement(); st.executeUpdate("delete from studrecord where eno="+id);
```

```
st.close();
```

```
c.close();
```

```
response.sendRedirect(" Search.jsp"); %>
```

```
</body>
```

```
</html>
```

Output:

localhost:8080/JSPRegistration/Form.js

Import favorites | Gmail | YouTube | Maps

ENROLLMENT_NUMBER:	501
FIRSTNAME:	yash
LASTNAME:	lathiya
SEMESTER:	6
CONTACT_NUMBER:	426557656

Save

[Search](#)

Eno	Fname	Lname	Sem	Contact
501	yash	lathiya	6	426557656
506	vivek	gabani	6	36837908

ENROLLMENT_NUMBER	FIRSTNAME	LASTNAME	SEMESTER	CONTACT_NUMBER	Action	Action
501	yash	lathiya	6	426557656	Delete	Edit
506	vivek	gabani	6	36837908	Delete	Edit

localhost / localhost / studreg / s

localhost:8080/JSPRegistration/Edit.jsp?y

Import favorites | Gmail | YouTube | Maps | N

ENROLLMENT_NUMBER:	501
FIRSTNAME:	keyur
LASTNAME:	lathiya
SEMESTER:	6
CONTACT_NUMBER:	426557656
<input type="button" value="Update"/>	

localhost / localhost / studreg / s | 3160707.pdf

localhost:8080/JSPRegistration/Search.jsp

Import favorites | Gmail | YouTube | Maps | News | Translate

ENROLLMENT_NUMBER	FIRSTNAME	LASTNAME	SEMESTER	CONTACT_NUMBER	Action	Action
501	keyur	lathiya	6	426557656	Delete	Edit
506	vivek	gabani	6	36837908	Delete	Edit

Practical 11

Aim: Implement cookies to store firstname and lastname using Java server pages.

→Main.jsp

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1" %>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

```
</head>
```

```
<body>
```

```
<% Cookie firstName=new Cookie("first_name", request.getParameter("first_name"));
```

```
Cookie lastName=new
```

```
Cookie("last_name", request.getParameter("last_name")); %>
```

```
<html>
```

```
<head>
```

```
<title>Setting Cookies</title>
```

```
</head>
```

```
<body>
```

```
<center>
```

```
<h1>Setting Cookies</h1>
```

```
</center>
```

```
<ul>
```

```
<li>
```

```
<p><b>First Name:</b>
```

```
<%= request.getParameter("first_name")%>
```

```
</p>
```

```
</li>
```

```
<li>
```

```
<p><b>Last Name:</b>
```

```
<%= request.getParameter("last_name")%>
```

```
</p>
```

```
</li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

```
</body>
```

```
</html>
```

→Hello.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1" %>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

```
<title>Insert title here</title>
```

```
</head>
```

```
<body>
```

```
<form action="Main.jsp" method="GET">
```

```
First Name: <input type="text" name="first_name">
```

```
<br />
```

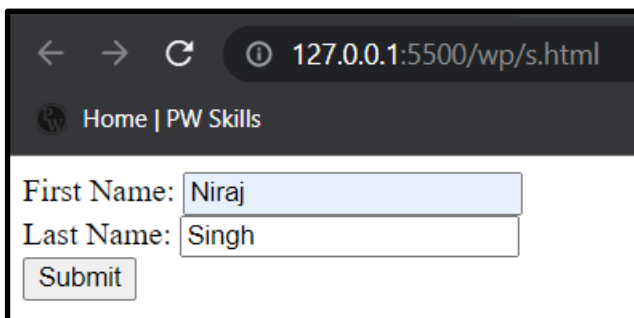
```
Last Name: <input type="text" name="last_name" />
```

```
<br />
```

```
<input type="submit" value="Submit" />
```

```
</form>
```

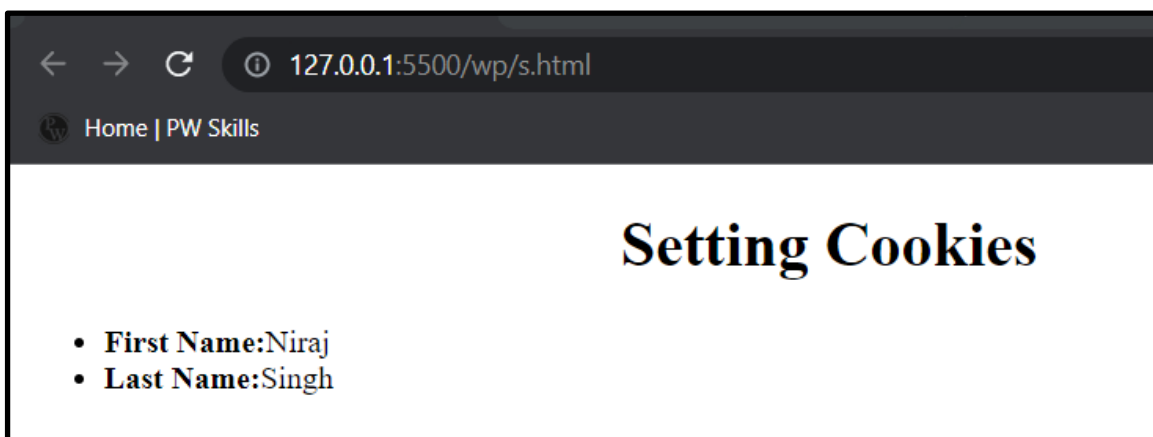
```
</body>
```

Output

First Name: Niraj

Last Name: Singh

Submit



Setting Cookies

- **First Name:**Niraj
- **Last Name:**Singh

Practical 12

Aim: Design a web page that takes the Username from user and if it is a valid username prints “Welcome Username”.

→index.html

```
<html>
<body>
  <form action="loginPage">
    User Name:<input type="text" name="uname" /><br /><br>
    Password:<input type="password" name="upass" /><br /><br>
    <input type="submit" value="SUBMIT" />
  </form>
</body>
</html>
```

→Login.java

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

@WebServlet("/loginPage")
public class Login extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public Login() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter pwriter = response.getWriter();
        String name = request.getParameter("uname");
        String pass = request.getParameter("upass");
        if (name.equals("Admin") && pass.equals("root")) {
            RequestDispatcher dis = request.getRequestDispatcher("welcome");
            dis.forward(request, response);
        } else {
            pwriter.print("Username or password is incorrect!");
            RequestDispatcher dis = request.getRequestDispatcher("index.html");
            dis.include(request, response);
        }
    }
}
```

→Welcome.java

```

import java.io.IOException;
import java.io.PrintWriter;
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("/Welcome")
public class Welcome extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public Welcome() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter pw = response.getWriter();
        String name = request.getParameter("uname");
        pw.print("Hello " + name + "!<br>");
        pw.print(" Welcome to this site!");
    }
}

```

→web.xml

```

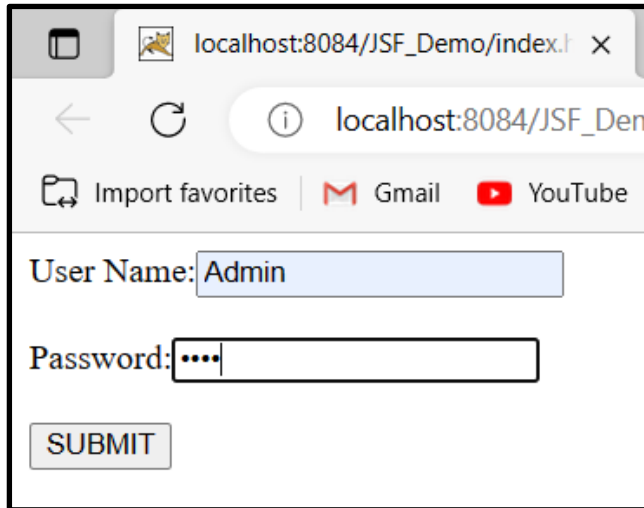
<web-app>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
</welcome-file-list>
<servlet>
<servlet-name>Login</servlet-name>
<servlet-class>Login</servlet-class>
</servlet>
<servlet>
<servlet-name>Welcome</servlet-name>
<servlet-class>Welcome</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Login</servlet-name>
<url-pattern>/loginPage</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>Welcome</servlet-name>
<url-pattern>/welcome</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>index.html</welcome-file>

```

</welcome-file-list>

</web-app>

Output:



localhost:8084/JSF_Demo/index.html

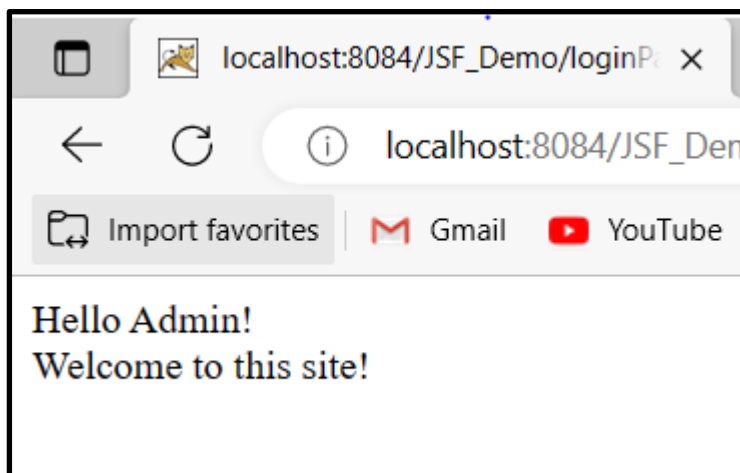
localhost:8084/JSF_Demo

Import favorites | Gmail | YouTube

User Name: Admin

Password:

SUBMIT



localhost:8084/JSF_Demo/loginPage.html

localhost:8084/JSF_Demo

Import favorites | Gmail | YouTube

Hello Admin!
Welcome to this site!

Practical 13

Aim: Study and implement hibernate.

Step 1: Create a database containing a table with a primary key.

Step 2: Create a web application and open hibernate.cfg.xml. In the multi-view XML editor, expand the Configuration Properties node under Optional Properties. Set the values as shown here.

→**hibernate.cfg.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD
3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
    <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/studdetail?zeroDateTimeBehavior
=convertToNull</property>
    <property name="hibernate.connection.username">root</property>
    <mapping resource="newpkg/Students.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```

Step 3: Add hibernate library function.

Step 4: Create a new package in source package name as newpkg and create a HibernateUtil file save as HibernateUtil.java on this package. It will automatically generate the following code.

→**HibernateUtil.java**

```
package newpkg;
import org.hibernate.cfg.AnnotationConfiguration;
import org.hibernate.SessionFactory;

public class HibernateUtil {

    private static final SessionFactory sessionFactory;

    static {
        try {

            sessionFactory = new AnnotationConfiguration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            // Log the exception.
            System.err.println("Initial SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }
}
```

```

    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}

```

Step 5: Right-click the newpkg node in the Projects window and choose New → Other to open the new file wizard. Select Hibernate Reverse Engineering Wizard in the Hibernate category. Select the required table from Available Tables and click Add to add the table to Selected Tables. It will generate following file.

Step 6: Right-click the newpkg node in the Projects window and choose New → Other to open the new file wizard. Select Hibernate Mapping Files and POJOs from a Database in the Hibernate category. It will create Mydetail.hbm.xml file which contains hibernate mapping file. It will also create Mydetail.java file which contains following code.

→Students.java

```

package newpkg;

public class Mydetail implements java.io.Serializable {

    private Integer eno;
    private String name;

    public Mydetail() {
    }

    public Mydetail(String name) {
        this.name = name;
    }

    public Integer getEno() {
        return this.eno;
    }

    public void setEno(Integer eno) {
        this.eno = eno;
    }

    public String getName() {
        return this.name;
    }

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

```

→Students.hbm.xml

```

<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<!-- Generated 27 May, 2023 4:33:24 PM by Hibernate Tools 3.6.0 -->
<hibernate-mapping>
    <class name="newpkg.Mydetail" table="mydetail" catalog="studdetail">
        <id name="eno" type="java.lang.Integer">
            <column name="eno" />
            <generator class="identity" />
        </id>
        <property name="name" type="string">
            <column name="name" length="20" not-null="true" />
        </property>
    </class>
</hibernate-mapping>

```

Step 7: Finally create a java file which contains main() to run this program.

→ mainHiber.java:

```

package newpkg;
package newpkg;

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

public class mainHiber {
    public static void main(String[] args) {
        Session s = HibernateUtil.getSessionFactory().getCurrentSession();
        Transaction t = s.beginTransaction();
        Mydetail s1 = new Mydetail();
        s1.setEno(101);
        s1.setName("Payal");
        s.save(s1);
        t.commit();
    }
}

```

Output:

eno	name
101	Payal

Practical 14

Aim: Study and implement MVC using spring framework.

→ web.xml

```
<web-app version="3.0"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_3_0.xsd">
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/dispatcher-servlet.xml</param-value>
</context-param>
<listener>
<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
<servlet>
<servlet-name>dispatcher</servlet-name>
<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
<load-on-startup>2</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>dispatcher</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
<session-config>
<session-timeout>
30
</session-timeout>
</session-config>
<welcome-file-list>
<welcome-file></welcome-file>
</welcome-file-list>
</web-app>
```

→ dispatcher-servlet.xml

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

<http://www.springframework.org/schema/mvc>
[http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd">](http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd)

```
<context:component-scan base-package="com.outbottle" />
<mvc:annotation-driven />
<bean id="jspViewResolver"
      class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <property name="viewClass"
    value="org.springframework.web.servlet.view.JstlView" />
  <property name="prefix" value="/WEB-INF/jsp/" />
  <property name="suffix" value=".jsp" />
</bean>
</beans>
```

→ DefaultController.java

```
package com.outbottle.hellospring.controllers;
```

```
import com.outbottle.hellospring.entities.Person;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
```

```
@Controller
```

```
public class DefaultController {
    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String index(ModelMap map) {
        map.addAttribute("hello", "Hello Spring from Netbeans!!");
        return "index";
    }

    @RequestMapping(value = "/viewdemo", method = RequestMethod.GET)
    public String demo(ModelMap map) {
        Person person = new Person();
        person.setName("Ashwin");
        person.setAge(24);
        map.put("personObject", person);
        map.addAttribute("helloAgain", "Hello (Again) Spring from Netbeans!!");
        return "demo";
    }
}
```

```
@RequestMapping(value = "/person/{name}", method = RequestMethod.GET)
public String demo(@PathVariable(value = "name") String name, ModelMap map) {
    Person person = new Person();
    person.setName(name);
    person.setAge(24);
    map.put("personObject", person);
}
```

```

        map.addAttribute("helloAgain", "The name passed in is the name returned.");
        return "demo";
    }

    @RequestMapping(value = "/person/{name}/{age}", method = RequestMethod.GET)
    public String demo(@PathVariable(value = "name") String name, @PathVariable(value = "age")
Integer age, ModelMap map) {
        Person person = new Person();
        person.setName(name);
        person.setAge(age);
        map.put("personObject", person);
        map.addAttribute("helloAgain", "The name passed in along with the age.");
        return "demo";
    }

    @RequestMapping(value = "/paramdemo", method = RequestMethod.GET)
    public String paramDemo(ModelMap map) {
        map.addAttribute("id", "Not passed In");
        map.addAttribute("other", "Not Passed In");
        return "paramdemo";
    }

    @RequestMapping(value = "/paramdemo1", method = RequestMethod.GET)
    public String paramDemo1(@RequestParam(value = "id", required = true) Long id,
        @RequestParam(value = "other", required = true) String other, ModelMap map) {
        map.addAttribute("id", id);
        map.addAttribute("other", other);
        return "paramdemo";
    }

    @RequestMapping(value = "/paramdemo2", method = RequestMethod.GET)
    public String paramDemo2(@RequestParam(value = "id", required = true) Long id,
        @RequestParam(value = "other", required = false) String other, ModelMap map) {
        map.addAttribute("id", id);
        if (other != null)
            map.addAttribute("other", other);
        else
            map.addAttribute("other", "Not passed in");
        return "paramdemo";
    }
}

```

→ index.jsp

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

<body>
  <h1>Hello Spring</h1>
  <p>
    This was passed in from the controller thus showing that
    the controller was accessed before the page was rendered.
    This is MVC (Model View Controller) in action.
  </p>
  <p>
    Simple values can be rendered as so \${ } i.e. here's the value from the controller:
    <blockquote>${hello}</blockquote>
  </p>
</body>

</html>
```

→ Person.java

```
package com.outbottle.hellospring.entities;
```

```
public class Person {
  private String name;
  private int age;

  public int getAge() {
    return age;
  }

  public void setAge(int age) {
    this.age = age;
  }

  public String getName() {
    return name;
  }

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

→ demo.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<body>
  <p>normal parameter:
  <blockquote>${helloAgain}</blockquote>
  </p>
  <h3>Person</h3>
  <p>Name: ${personObject.name}</p>
  <p>Age: ${personObject.age}</p>
  <p>
    <a href="<%=request.getContextPath()%>/person/Rohit">Show Rohit</a>
  </p>
  <p>
    <a href="<%=request.getContextPath()%>/person/Ashish/28">Show Ashish age 28</a>
  </p>
  <p>
    <a href="<%=request.getContextPath()%>/viewdemo">Show Default</a>
  </p>
</body>

</html>

```

Output:**→ index.jsp****Hello Spring**

This was passed in from the controller thus showing that the controller was accessed before the page was rendered. This is MVC (Model View Controller) in action.

Simple values can be rendered as so \${ } i.e. here's the value from the controller:

Hello Spring from Netbeans!!

[Demo](#)

On clicking the Demo link:

→ demo.jsp

normal parameter:

Hello (Again) Spring from Netbeans!!

Person

Name: Ashwin

Age: 24

[Show Rohit](#)

[Show Ashish age 28](#)

[Show Default](#)

→ Person.java

normal parameter:

The name passed in is the name returned.

Person

Name: Rohit

Age: 24

[Show Rohit](#)

[Show Ashish age 28](#)

[Show Default](#)