Practical-1

Aim: Create chat application using either TCP or UDP protocol.

Server:

```
import java.io.*; import java.net.*;
public class ServerMulti { static
ServerSocket serverSocket = null; static
Socket client_socket = null;
        static client_Thread Th[] = new client_Thread[10];
public static void main (String args[])
        { int
port_number = 1234;
                System.out.println("Server Started....and Ctrl +c to terminate");
                try{
                serverSocket = new ServerSocket(port_number);
                }catch(IOException e){
} while(true){ try{ client_socket = serverSocket.accept();
                        for(int i = 0; i< 9 ;i++){
                        if(Th[i] == null)
                        {
                                (Th[i] = new client_Thread(client_socket, Th)).start();
                                break;
                                }
                        }
                }catch(IOException e){

                }
                }
        }
}
class client_Thread extends Thread
{
        BufferedReader input = null;
        PrintStream output = null;
        Socket client_socket = null;
        client_Thread Th[];
        public client_Thread(Socket client_socket, client_Thread[] Th){
        this.client_socket = client_socket;
        this.Th = Th;
} public void
run() {
```

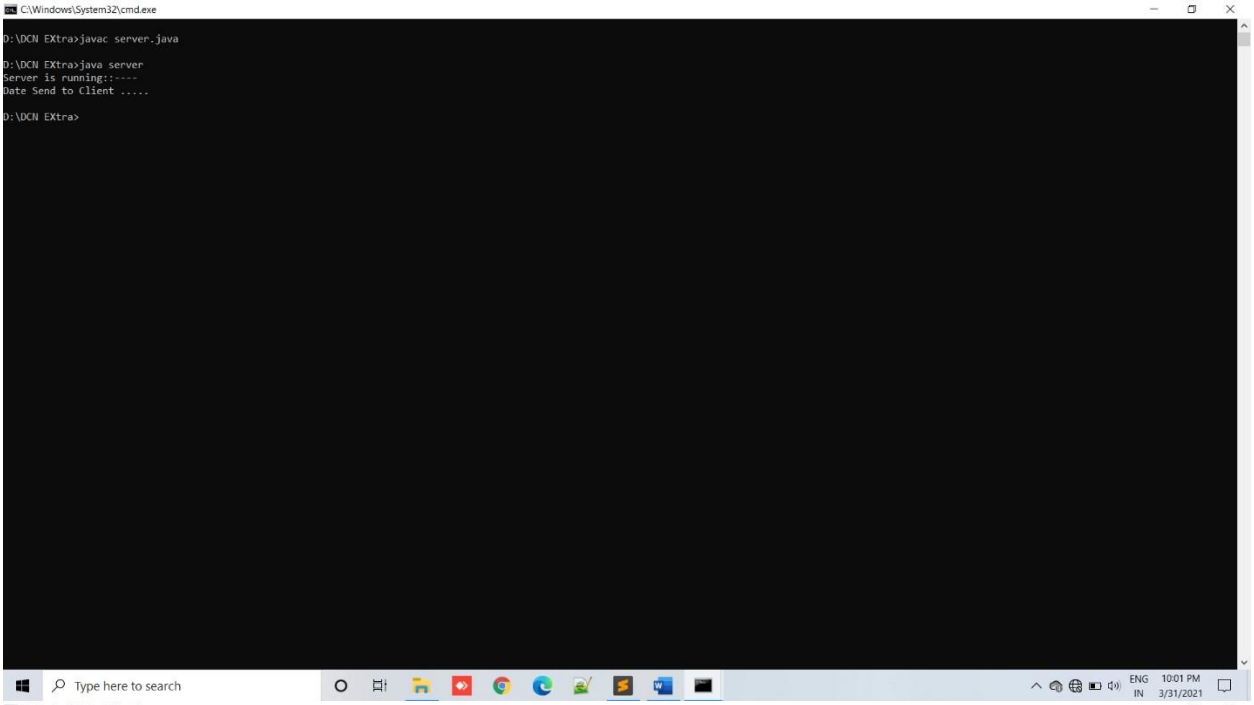
```
        String msg;
        String
        user_name; try{
        input = new BufferedReader(new InputStreamReader(client_socket.getInputStream()));
        output = new PrintStream(client_socket.getOutputStream()); output.println("Enter your
        Name :-"); user_name = input.readLine();
        output.println(user_name + "$$ to leave
        chatroom"); for(int i = 0; i <9 ; i++){ if(Th[i] !=null
        &&Th[i] != this){
        Th[i].output.println("New user added----------"+ user_name);
while(true) { msg = input.readLine();
if(msg.startsWith("$$"))
{ break; } for(int j = 0 ; j
<= 9 ; j++){ if(Th[j] !=
null){
Th[j].output.println("<" + user_name + ">" + msg);
                }
        }
} for(int i1 = 0; i1 <= 9 ; i1++){
if(Th[i1] != null &&Th[i1] != this){
Th[i1].output.println("User Left Room----------" + user_name);
output.println(user_name        +"--------Left        Chatroom------");
output.println("Ctrl + c to exit");
                        }
                }
for(int k = 0 ; k <=9 ; k++){
if(Th[k] == this){ Th[k] =
null; input.close();
output.close();
client_socket.close();
                }
            }
        }
    }
        }catch(IOException e){        }
        }
}
```

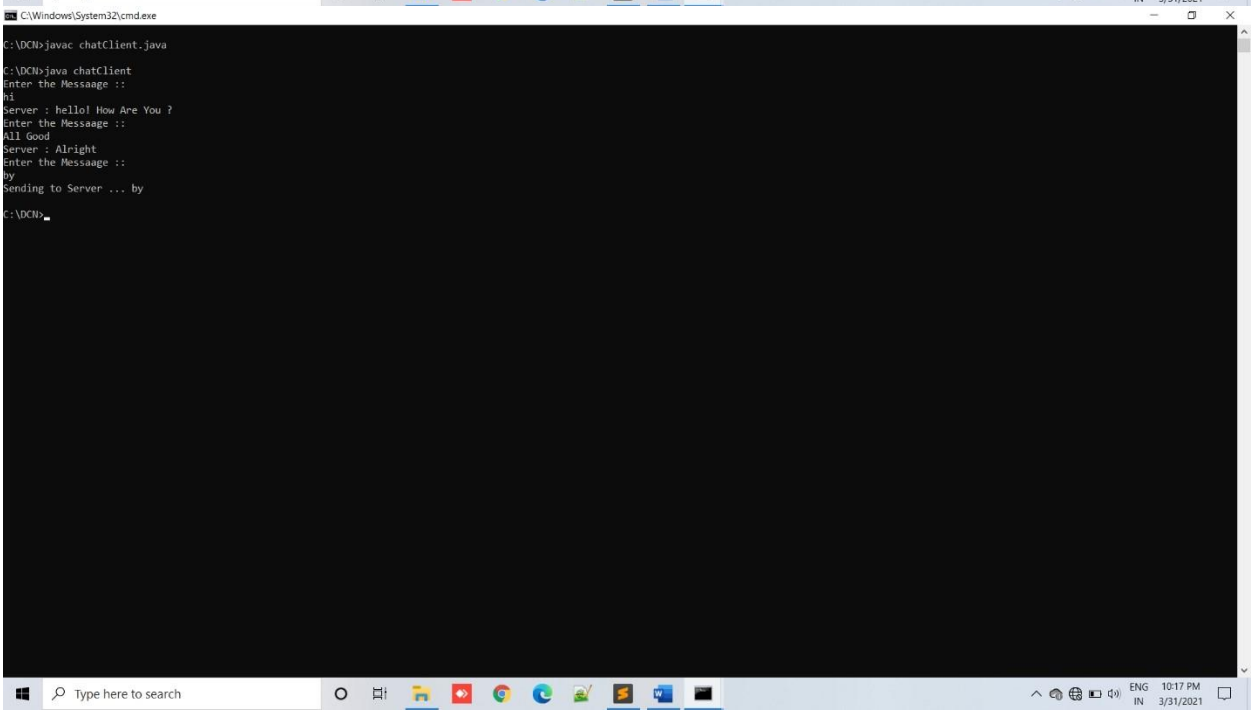
Client: import
java.io.*; import
java.net.*;

```
public class ClientMulti implements Runnable
{
static Socket client_socket = null; static
        PrintStream output = null; static
```

```
        BufferedReader input = null; static
        BufferedReader user_input = null; static
        Boolean Flag = false;
public static void main(String args[])
        { int port_number = 1234;
                String host = "localhost"; try{ client_socket =
        new Socket(host,port_number);
                user_input = new BufferedReader(new InputStreamReader(System.in));
                output = new PrintStream(client_socket.getOutputStream());
input = new BufferedReader(new InputStreamReader(client_socket.getInputStream()));
}catch(UnknownHostException e){
                        System.out.println("Exception error");
                }
                catch(IOException e){ }
        if(client_socket != null){
        try{
        new Thread(new ClientMulti()).start();
while(!Flag)
{ output.println(user_input.readLine());
        }
        output.close();
        input.close();
        client_socket.close();
        }
catch(IOException e){ }
}
} @Override
public void run() {
String msg; try{
while((msg = input.readLine()) != null)
System.out.println(msg);
Flag = true;
}catch(IOException e){ }
        }
}
```

Output:





Practical-2

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

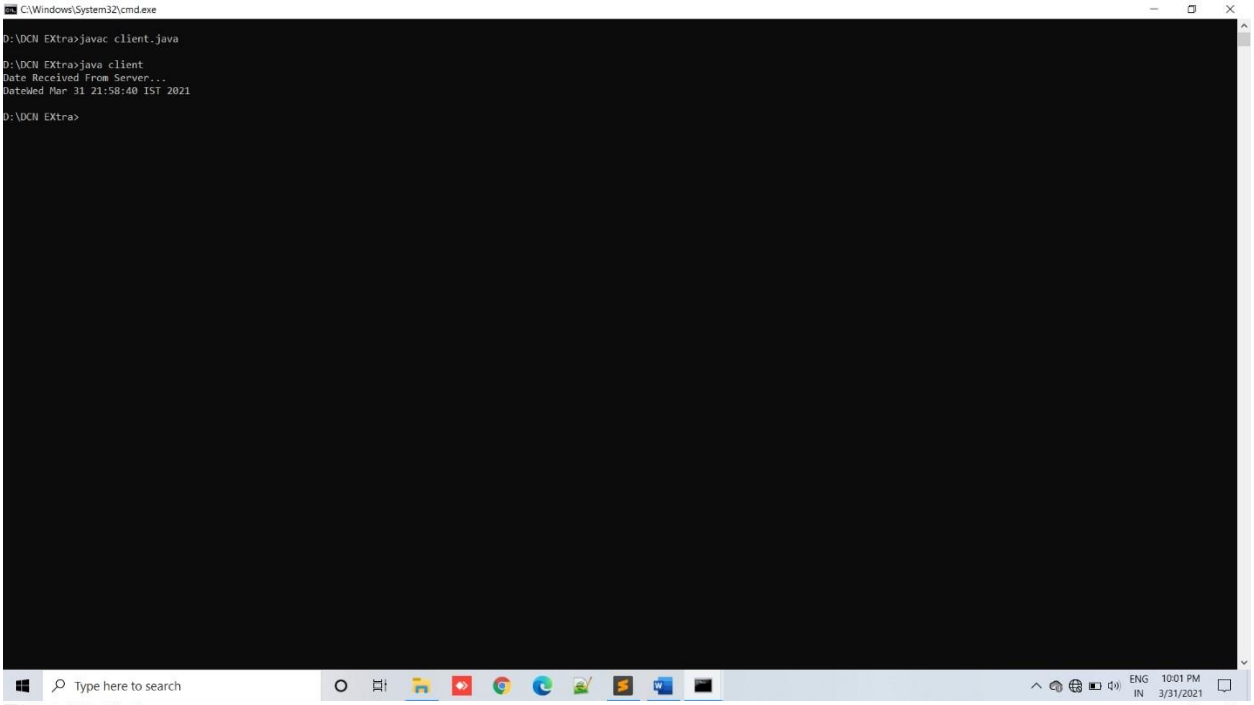
SERVER:

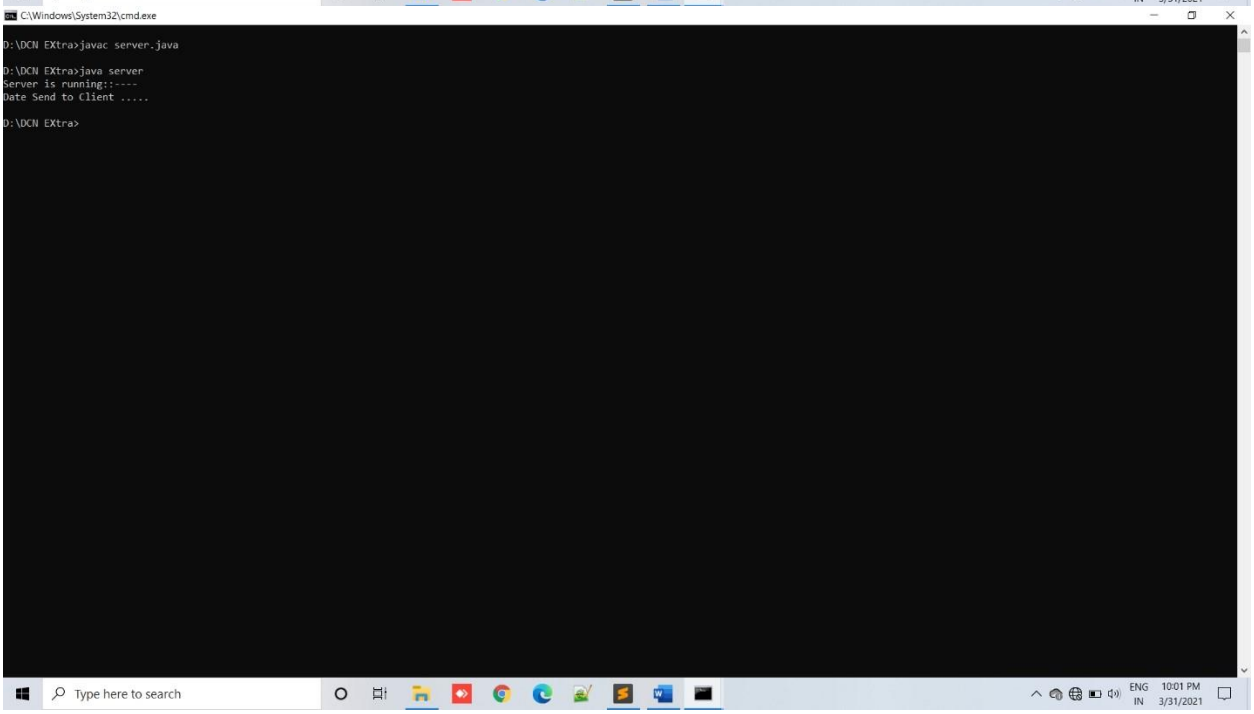
```
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket; import
java.util.Date; import
javax.xml.crypto.Data; public
class server{

        public static void main(String args[]) throws IOException{
System.out.println("Server is running::----");
                ServerSocket ser_socket = new ServerSocket(1233);
                Socket listen_socket = ser_socket.accept();
                DataOutputStream server_out = new
                DataOutputStream(listen_socket.getOutputStream()); server_out.writeBytes("Date" +
                (new Date().toString() + "\n")); server_out.close();
                listen_socket.close();
}

}
```

CLIENT:

```
import java.io.BufferedReader; import
java.io.DataOutputStream; import
java.io.IOException; import
java.io.InputStreamReader; import
java.net.Socket; import
java.net.UnknownHostException; public
class client{

        public static void main(String args[]) throws UnknownHostException, IOException{
Socket cli_socket = new Socket("LocalHost",1233);
                BufferedReader br = new BufferedReader(new
InputStreamReader(cli_socket.getInputStream()));
                System.out.println(br.readLine());

}
}
```

Output:





Practical-3

Aim: Create Servlet file which contains following functions:

1.Connect 2. Create Database 3. Create Table 4. Insert Records into respective table 5. Update records of particular table of database 6. Delete Records from table. 7. Delete table and also database.

index.html Code:

```
<!DOCTYPE html>
<html>
<head>
<title>Java</title>
<link rel="stylesheet" type="text/css" href="main.css"> </head>
<body bgcolor="#222222">
<h1 align="center">Welcome To Student Portal</h1>
<div class="main">
<input type="button" class="button" name="insert_user" onclick="window.location.href='page.html'" value="Insert Student"><br>
<input type="button" class="button green" name="delete_user" onclick="window.location.href='delete.html'" value="Delete Student"><br>
<input type="button" class="button grey" name="update_user" onclick="window.location.href='update.html'" value="Update Student"><br>
<input type="button" class="button blue" name="select_user" onclick="window.location.href='select.html'" value="Select Student">
</div>
</body>
</html>
```

---delete.html

Code:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Servlet</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" type="text/css" href="main.css">

    </head>
    <body bgcolor="#222222">
<form action="WebServlet" method="post" style="text-align:center;">
<h2 style="color:white;font-family: Times; ">Student Information</h2>
<input class="side" type="text" pattern="[0-9]*" required="" name="s_enrollno" minlength="12" maxlength="12" placeholder="Enrollment No" style=" width: 370px;"><br>

<input class="b1" type="submit" value="Delete Data" name="s1">

</form>
    </body>
</html>
```

---page.html

Code:

```
<!DOCTYPE html>
<html>
<head>
        <title>Servlet</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
        <link rel="stylesheet" type="text/css" href="main.css">

</head>
<body bgcolor="#222222">
<form action="WebServlet" method="post" style="text-align:center;">



<h1 style="color:white;font-family: Times; ">Student Information</h1>


<input class="side" type="text" pattern="[0-9]*" required="" name="s_enrollno" placeholder="Enrollment No"
minlength="12" maxlength="12" maxlength="12"><br>
<input class="side a" type="text" pattern="[A-Za-z]+" required="" name="s_firstname" placeholder="First Name">

<input class="side b" type="text" pattern="[A-Za-z]+" name="s_lastname" placeholder="Last Name"
required><br>

<input class="side" type="text" pattern="[A-Za-z]+" name="s_branch" placeholder="Branch" required><br>

<input class="side" type="text" pattern="[0-9]*" required="" name="s_mobileno" placeholder="Mobile No"
minlength="10" maxlength="10"><br>

<input class="b1" type="submit" value="Insert Data" name="s1" >

</form>

</body>
</html>
```

----------select.html

Code:

```
<!DOCTYPE html>
<html>
   <head>
      <title>Servlet</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <link rel="stylesheet" type="text/css" href="main.css">
      </head>
   <body bgcolor="#222222">
<form action="WebServlet" method="post" style="text-align:center;">
<h1 style="color:white;font-family: Times; ">Student Information</h1>
<input class="side" type="text" pattern="[0-9]*" required="" name="s_enrollno" placeholder="Enrollment No"
minlength="12" maxlength="12" style=" width: 370px;"><br>
<input class="b1" type="submit" value="View Data" name="s1">

</form>
   </body>
</html>
```

----------update.html

Code:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Servlet</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" type="text/css" href="main.css">
       </head>
    <body bgcolor="#222222">
<form action="WebServlet" action="newupdate.html" method="post" style="text-align:center;">

<h1 style="color:white;font-family: Times; ">Student Information</h1>
<h2 style="color:white;">Insert Enrollment No To Change Details Of Student</h2>
<input class="side" type="text" pattern="[0-9]*" required="" name="s_enrollno" placeholder="Enrollment No"
minlength="12" maxlength="12"><br> <h2 style="color:white;"">Insert New Data</h2>
<input class="side a" type="text" pattern="[A-Za-z]+" required="" name="s_firstname" placeholder="First Name">

<input class="side b" type="text" pattern="[A-Za-z]+" name="s_lastname" placeholder="Last Name"
required><br>

<input class="side" type="text" pattern="[A-Za-z]+" name="s_branch" placeholder="Branch" required><br>

<input class="side" type="text" pattern="[0-9]*" required="" name="s_mobileno" placeholder="Mobile No"
minlength="10" maxlength="10" ><br>
<input class="b1" type="submit" value="Update Data" name="s1" >

</form>
    </body>
</html>
```

---main.css

Code:

---body

```
{
 margin:0px;
 padding:0px;
}
.main
{
    text-align: center;
}
.button { background-color: #f44336; /*
    Green */ color: white; padding: 15px
    32px; text-align: center; text-
    decoration: none; font-size: 30px;
    margin: 8px 2px; cursor: pointer;
    width: 400px; border: 0px solid black;
    transition: width 0.5s,background-color
    0.5s;
}
.button:hover { width:
99%; background-color:
```

```
white; color: black; }
input
{ margin: 50px;
} h1 {
color:white;
}
.green { background-color:
#4CAF50;
}
.grey { background-color:
#555555;
}
.blue { background-color:
#008CBA;
}
.b1
{ width: 395px; height: 40px;
   margin:10px; color: white;
   background-color: #008CBA;
   border: 0px solid grey;
   cursor: pointer;
}
.side { padding:
        10px; margin:
        10px; width:
        370px;
        }
        .a
        { width:
         160px; }
        .b
        { width:
         160px
        }
```

WebServlet.java Code:

```
import java.io.*; import java.sql.Connection; import
java.sql.DriverManager; import java.sql.PreparedStatement;
import java.sql.ResultSet; import java.sql.SQLException; import
java.sql.Statement;



import javax.servlet.*; import

javax.servlet.http.*; public class WebServlet

extends HttpServlet {


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

```
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

    try
        {
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("loaded driver succesfull");
            Connection cn =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/mysql","root","birju"); Statement
            st=cn.createStatement(); st.executeUpdate("create database student"); st.executeUpdate("use
            student");
            st.execute("create table student2 (s_enrollno BIGINT,s_firstname varchar(20),s_lastname
varchar(20),s_branch varchar(10),s_mobileno BIGINT)");

            ResultSet rs ;


           String s2 = request.getParameter("s1");

           Long eno = Long.valueOf(request.getParameter("s_enrollno"));

           String fname =request.getParameter("s_firstname");
           String lname = request.getParameter("s_lastname");
           String branch =request.getParameter("s_branch");
           String mobile=request.getParameter("s_mobileno");

           PreparedStatement sr;

           Statement stmt = cn.createStatement();
            ResultSet res;

            // For Insert Data......
            switch (s2) {
            //For Delete Data......
               case "Insert Data":
                  String str = "insert into student2
                  values(?,?,?,?,?)"; sr = cn.prepareStatement(str);
                  sr.setLong(1,eno); sr.setString(2,fname);
                  sr.setString(3,lname); sr.setString(4,branch);
                  sr.setString(5,mobile); int i = sr.executeUpdate();
                  out.println("Student Data inserted");
                  cn.close();
                  break;
               case "Delete Data":
                  String str1 = "select s_enrollno from student2 where s_enrollno = "+eno+" ";
                  rs =stmt.executeQuery(str1);
                  if(rs.next())
                  { str1 = "delete from student2 where s_enrollno = ?";
                     sr = cn.prepareStatement(str1);
                     sr.setLong(1,eno);
                     sr.executeUpdate();
                     out.println("Student Data Deleted");
                  }
                  else{
```

```
                out.println("Student Data Not Valid");
            }       cn.close();
            break;
        case "Update Data":
            String str3 = "select * from student2 where s_enrollno = "+eno+"";
            rs =stmt.executeQuery(str3);
            if(rs.next())
            { str3= "update student2 set s_enrollno=? , s_firstname=?,s_lastname=?,s_branch=?,s_mobileno=?
where s_enrollno = "+eno+"";
                sr=cn.prepareStatement(str3);
                sr.setLong(1,eno);
                sr.setString(2,fname);
                sr.setString(3,lname);
                sr.setString(4,branch);
                sr.setString(5,mobile);
                sr.executeUpdate();

                out.println("Update Student2 Data
            Succesfull"); cn.close(); } else{
                out.println("Student Data Not Valid");
            } break; case "View Data": res = stmt.executeQuery("select * from student2
        where s_enrollno="+eno+""); if(res.next()){
                Long enu= res.getLong("s_enrollno");
                String fname1= res.getString("s_firstname");
                String lname1= res.getString("s_lastname");
                String branch1= res.getString("s_branch"); String mb1=
                res.getString("s_mobileno"); out.println("<html><body>");
                out.println("Enrollment No:- "); out.println(enu);out.println("<br>");
                out.println("First Name:- ");
                out.println(fname1);out.println("<br>"); out.println("Last Name:- ");
                out.println(lname1);out.println("<br>"); out.println("Branch:- ");
                out.println(branch1); out.println("<br>");
        out.println("Mobile No:- ");out.println(mb1);out.println("<br>");
        out.println("</html></body>");
      } else{
        out.println("Student Data is Not Valid");
        }    break;
      }
    }catch(ClassNotFoundException | SQLException e   )
    { System.out.println(e); }}}
```

----------web.xml

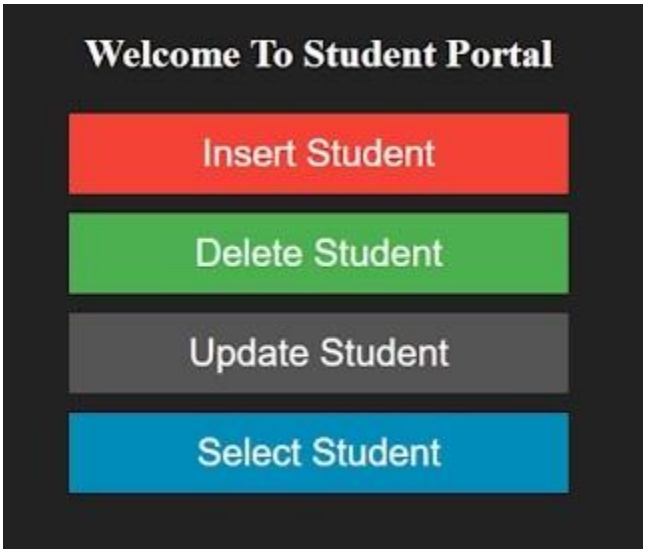
Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee">
   <servlet>
      <servlet-name>WebServlet</servlet-name>
      <servlet-class>WebServlet</servlet-class>
   </servlet>
   <welecome-file-list>
      <welcome-file>MainPage.html</welcome-file>
        <welcome-file>index.html</welcome-file>
```

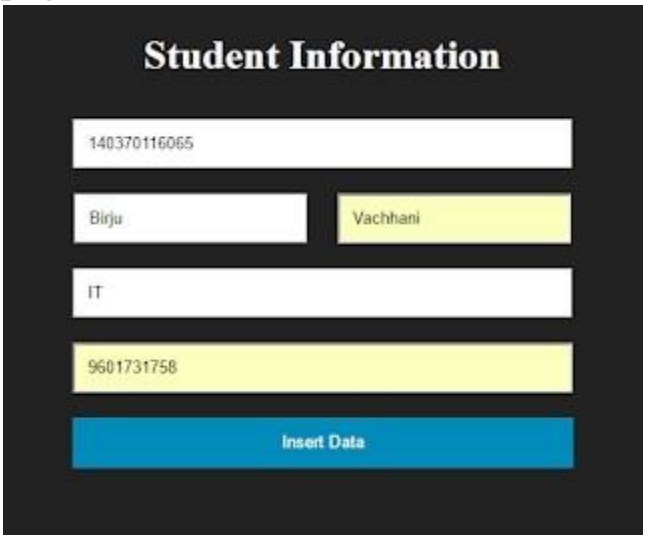
```
    </welecome-file-list>

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

OUTPUT :

Index.html



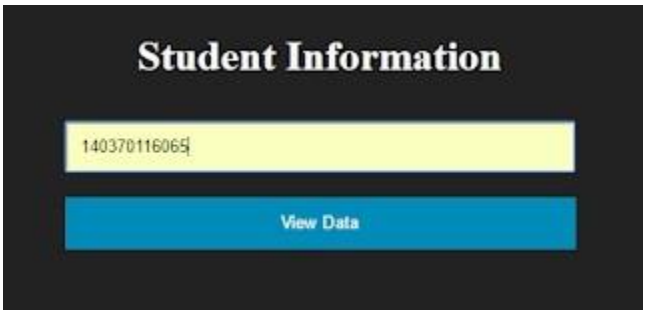
page.html



Result :

Student Data inserted

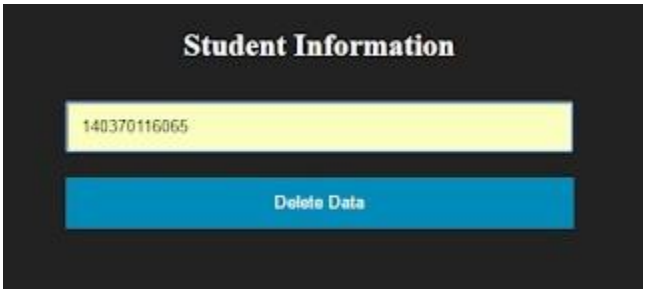
select.html



Result :

Enrollment No:- 140370116065
First Name:- Birju
Last Name:- Vachhani
Branch:- IT
Mobile No:- 9726337873

delete.html



Result :

Student Data Deleted

update.html

Result :

Update Student2 Data Succesfull

Practical-4

Aim: User can create a new database and also create new table under that database. Once database has been created then user can perform database operation by calling above functions. Use following Java Statement interface to implement program:

- **Statement 2. Prepared statement 3. Callable statement**

SqlCon.java Code:

```
import java.sql.*; import java.util.Scanner; public class SqlCon

{ public static void main(String[] args)throws Exception

  {

    System.out.println("Enter database name to create new database:");

    Scanner scan=new Scanner(System.in);

    String dbname=scan.next();

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

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

    Statement stmt=con.createStatement(); stmt.executeUpdate("create database "+dbname);

    System.out.println("Database Created"); stmt.execute("use "+dbname);

    System.out.println("Enter table name to create table:");

    String tbname=scan.next();

    stmt.executeUpdate("create table "+tbname+" (id int,name char(20),branch char(10),address
char(20))");
    System.out.println("Table Created with (id,nam,branch,address) fields");

    while(true)

    {

      System.out.println("Enter your choice:\n1.Insert\t2.Delete\t3.select\t4.Exit");

      int choice=scan.nextInt(); switch(choice)

      { case 1:
```

```
{
    System.out.println("Enter (id,name,branch,address) of student:");
    int id=scan.nextInt();
    String name=scan.next();
    String branch=scan.next();
    String add=scan.next();
    PreparedStatement sr;
    String str = "insert into "+tbname+" values(?,?,?,?)";
    sr = con.prepareStatement(str);
    sr.setInt(1,id);
    sr.setString(2,name);
    sr.setString(3,branch);
    sr.setString(4,add);
    sr.executeUpdate();
    System.out.println("Data inserted");
break; } case 2:
{
    System.out.println("Enter id of student:");
    int id=scan.nextInt();
    ResultSet rs=stmt.executeQuery("select * from "+tbname+" where id="+id);
    if(rs.next())
    { stmt.executeUpdate("delete from "+tbname+" where id="+id);
        System.out.println("Data deleted");
    }
    else
    {
```

```
            System.out.println("Data with id "+id+" doesn't exsist");
        }
    break; }
    case 3:
    {
        System.out.println("Enter id of student:");
        int id=scan.nextInt();
        stmt.execute("CREATE PROCEDURE "+dbname+".selector (IN idr INT) "+
        "BEGIN "+
        "SELECT * "+
        "FROM "+tbname+" "+
        "WHERE id = idr; "+
        "END");
        CallableStatement cs = con.prepareCall("{call "+dbname+".selector(?)}"); cs.setInt(1, id);
        System.out.println("procedure call sucess");
        cs.execute();
        ResultSet rs=cs.getResultSet();
        if(rs.next())
        {
            System.out.println("id\tName\tBranch\tAddress");

System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3)+"\t"+rs.getString(4));
        }
    break; }
    case 4:
    { break;
    }
```

```
            default:
            {
                System.out.println("Invalid option");
            } }
        if(choice==4
        )
        { break;
        }
    }
    con.close();
  }
}
```

OUTPUT :

C:\Windows\system32\cmd.exe

```
K:\Subjects\Advanced JAVA\Practicals\8>java SqlCon
Enter database name to create new database:
stdb
Database Created
Enter table name to create table:
stinfo
Table Created with (id,nam,branch,address) fields
Enter your choice:
1.Insert        2.Delete        3.select        4.Exit
1
Enter (id,name,branch,address) of student:
65
birju
IT
junagadh
Data inserted
Enter your choice:
1.Insert        2.Delete        3.select        4.Exit
3
Enter id of student:
65
procedure call sucess
id      Name    Branch  Address
65      birju   IT      junagadh
Enter your choice:
1.Insert        2.Delete        3.select        4.Exit
2
Enter id of student:
65
Data deleted
Enter your choice:
1.Insert        2.Delete        3.select        4.Exit
4

K:\Subjects\Advanced JAVA\Practicals\8>
```

Practical-5

Aim: Create Servlet file and study web descriptor file.

Servlet file

Sun Microsystem defines a unique directory structure that must be followed to create a servlet application. The web.xml (deployement descriptor) file is kept under WEB-INF folder.

Creating a Servlet

There are three different ways to create a servlet:

1) By implementing Servlet interface
2) By extending GenericServlet class
3) By extending HttpServlet class

But mostly a servlet is created by extending HttpServlet abstract class. As discussed earlier HttpServlet gives the definition of service() method of the Servlet interface. The servlet class that we will create should not override service() method. Our servlet class will override only doGet() or doPost() method.

When a request comes in for the servlet, the Web Container calls the servlet's service() method and depending on the type of request the service() method calls either the doGet() or doPost() method.

Web descriptor file

Deployment Descriptor(DD) is an XML document that is used by Web Container to run Servlets and JSP pages. DD is used for several important purposes such as:

- Mapping URL to Servlet class.
- Initializing parameters.
- Defining Error page.
- Security roles.
- Declaring tag libraries. **Web.xml:**

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"xmlns="http://java.sun.com/xml/ns/javaee"xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
<display-name>servletjdbc</display-name>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
<welcome-file>index.htm</welcome-file>
<welcome-file>index.jsp</welcome-file>
<welcome-file>default.html</welcome-file>
<welcome-file>default.htm</welcome-file>
<welcome-file>default.jsp</welcome-file>
</welcome-file-list>
<servlet>
```

```
<servlet-name>Firstservlet</servlet-name>
<servlet-class>stddatabase.Firstservlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Firstservlet</servlet-name>
<url-pattern>/Secondservlet</url-pattern>
</servlet-mapping>
<filter>
<filter-name>Authfilter</filter-name>
<filter-class>stddatabase.Authfilter</filter-class>
</filter>
<filter-mapping>
<filter-name>Authfilter</filter-name>
<url-pattern>/Firstservlet</url-pattern>
</filter-mapping>

</web-app>
```

OUTPUT :

Here the Servlet class and name is WebServlet and and <url-pattern> tag map the /Servlet URL in Servlet and fetch result about URL.<url-pattern> parent tag is <servlet-mapping> and always start with <web-app> and end with</web-app>.

Practical-6

Aim: Create login form and perform state management using Cookies, HttpSession.

JSP:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>SERVLETS</title>
</head>
<%
String uname=" ";
String pname=" ";
Cookie[] cookieArray = request.getCookies();
if(cookieArray != null)
{ for(Cookie cookie: cookieArray){
        if(cookie.getName().equals("Usercookie")){ uname
        = cookie.getValue();
                }
                if(cookie.getName().equals("Passcookie")){
                        pname = cookie.getValue();
                }

        }
}


%>
<body bgcolor="sky blue">
<form name="myform" method="get" action="FirstServlet">
<table align="center"></table>
<tr>
<td><h1>Enter Your Details to Login</h1></td>
<b>Username:</b></td><input type="text" name="username"value='<%=uname%>'/>
<b>Password:</b></td><input type="password"
name="password"value='<%=pname%>'/><br>
<input type="submit" value="Login"><br>
```

```
</table>
</form>
</body>
</html>
```

Servlet 1:

```
importjava.io.IOException;
importjava.io.PrintWriter;

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


/**
* Servlet implementation class FirstServlet */
@WebServlet("/FirstServlet")
public class FirstServlet extends HttpServlet { private
        static final long serialVersionUID = 1L;

    /**
* @see HttpServlet#HttpServlet()
     */ public
FirstServlet() {
super();
        // TODO Auto-generated constructor stub
    }

        /**
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */ protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws
ServletException, IOException {
                // TODO Auto-generated method stub
                //response.getWriter().append("Served at: ").append(request.getContextPath());
                response.setContentType("text/html;charset=UTF-8");

                PrintWriterpw= response.getWriter();
                pw.println("You have just logged in..!!!<br>");

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

```
String Password=request.getParameter("password");
HttpSession session = request.getSession();
session.setAttribute("Username", Username);
pw.println("Hello" +Username+",<br>");
String sessionId = session.getId();

Cookie ucookie=new Cookie("Usercookie",Username);
Cookie pcookie=new Cookie("Passcookie",Password);

response.addCookie(ucookie);

response.addCookie(pcookie); pw.println("Your cookies has

been added!!!<br><br><br>");

pw.println("<html><body bgcolor='pink'>");
pw.println("<b> Session Id is :-</b>" + sessionId);
pw.println("<br>Your name has been sent to the session!!");
pw.println("<a href='SecondServlet'>Click here</a>");
pw.println("<br><br>Want to redirect to the home page??");
pw.println("<a href='index.jsp'>click here</a>");
pw.println("</html></body>");

    }
```

} **Servlet**

2:

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


/**
* Servlet implementation class SecondServlet
*/
@WebServlet("/SecondServlet") public class
SecondServlet extends HttpServlet { private static
final long serialVersionUID = 1L;

    /**
```

```
* @see HttpServlet#HttpServlet()
    */
public SecondServlet() {
super();
       // TODO Auto-generated constructor stub
    }

          /**
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
           */ protected void doGet(HttpServletRequest request, HttpServletResponse response)
          throws
ServletException, IOException {
                    // TODO Auto-generated method stub
                    //response.getWriter().append("Served at:
                    ").append(request.getContextPath());
                    response.setContentType("text/html;charset=UTF-8"); java.io.PrintWriterpw =
                    response.getWriter(); HttpSession session = request.getSession();
                    pw.println("Welcome"+session.getAttribute("Username"));
pw.println("<html><body bgcolor='yellow'>");
                    pw.println("<br/>");
                    pw.println("<br><br>Want to redirect to the home page??");
                    pw.println("<br><br>Want to redirect to the home page??");
                    pw.println("<a href='index.jsp'>click here</a>");
                    pw.println("</html></body>");

          }



}
```

Output:

localhost:8080/Ks/FirstServlet?username=+pooja&password=pojja

You have just logged in..!!!
Hello pooja,
Your cookies has been added!!!

Session Id is :-F66FF8070220A93D2A94C7329AEF5EC8
Your name has been sent to the session!! Click here

Want to redirect to the home page?? click here

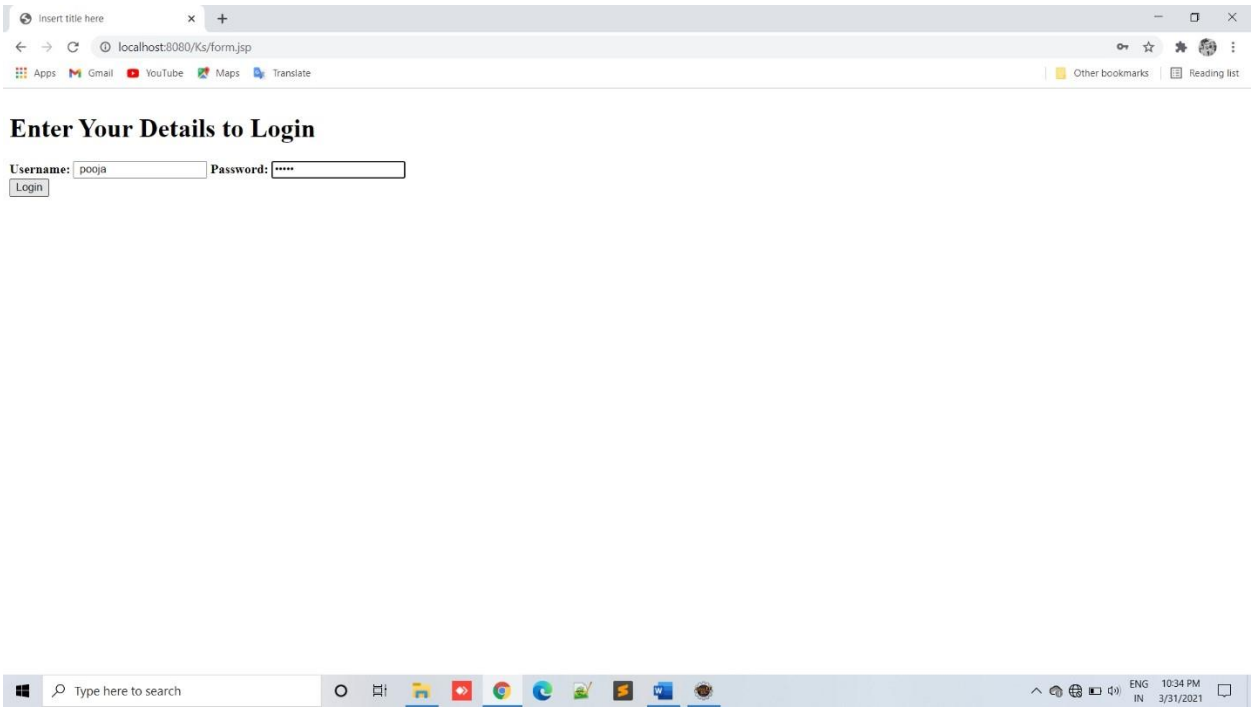
localhost:8080/Ks/SecondServlet

Welcome pooja

Want to redirect to the home page??

Want to redirect to the home page?? click here

localhost:8080/Ks/form.jsp


Insert title here
localhost:8080/Ks/form.jsp
Apps
Gmail
YouTube
Maps
Translate
Other bookmarks
Reading list
Enter Your Details to Login
Username: pooja
Password:
Login
Type here to search
ENG
IN
10:34 PM
3/31/2021

Practical-7

Aim: Implement Authentication filter using filter API.

index.html Code:

```
<html>
<head>
<title>Filter API</title>
<style>
.b1
{ width: 370px; height: 42px;
    margin:10px; color: white;
    background-color: #008CBA;
    border: 0px solid grey;
    cursor: pointer;
}
.side
{ padding:
 10px; margin:
 20px; width:
 370px;
}
.b1:hover
{ background-color:
 #4CAF50;
}
</style>
</head>
<body bgcolor="#222222">
    <div class="box">
        <form action="filtering" style="text-align:center;">
   <h1 style="color:white;font-family: Times;margin-top:160px;">Login Portal</h1>
   <input class="side" type="text" required="" name="name" placeholder="UserName"><br>
   <input class="side" type="password" required="" name="password" placeholder="Password"
style=margin-top:-5px;"><br>
   <input class="b1" type="submit" value="Login">
  </form>
</div>
</body>

</html>
```

MyFilter.java Code:

---import

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

```
public class MyFilter implements Filter
{
   @Override
   public void init(FilterConfig arg0) throws ServletException {}
   @Override
   public void doFilter(ServletRequest req, ServletResponse resp,FilterChain chain) throws IOException,
ServletException
   {
      PrintWriter pw=resp.getWriter(); String
      pwd=req.getParameter("password");
      if(pwd.equals("birju"))
      { chain.doFilter(req, resp);
     } else
      { pw.print("Invalid Password");
         RequestDispatcher redi=req.getRequestDispatcher("index.html");
         redi.include(req, resp);
}
   } @Override public
   void destroy() {}

}
```

----------ServletFilter
Code:

----------import
java.io.*;

```
import javax.servlet.http.HttpServlet; import
javax.servlet.ServletException; import
javax.servlet.http.HttpServletRequest; import
javax.servlet.http.HttpServletResponse;

public class ServletFilter extends HttpServlet
{
@Override
public void doGet(HttpServletRequest res,HttpServletResponse resp) throws
IOException,ServletException
{ resp.setContentType("text/html;charset=UTF-8");
      PrintWriter out = resp.getWriter(); String str = res.getParameter("name");
out.print("<h1>Hello "+str+"</h1><h2>You have sucessfully logged in....</h2>"); }
}
```

----------web.xml
Code:

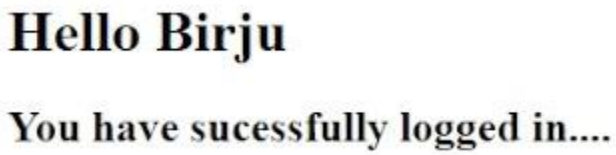
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
```

```
    <servlet>
    <servlet-name>ServletFilter</servlet-name>
    <servlet-class>ServletFilter</servlet-class>
    </servlet>
<welecome-file-list>
        <welcome-file>index.html</welcome-file>
    </welecome-file-list>
  <servlet-mapping>
    <servlet-name>ServletFilter</servlet-name>
    <url-pattern>/filtering</url-pattern>
  </servlet-mapping>

<filter>
  <filter-name>f1</filter-name>
  <filter-class>MyFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>f1</filter-name>
 <url-pattern>/filtering</url-pattern> </filter-mapping>


</web-app>
```

OUTPUT :

Index.html

Result :



Practical-8

Aim: Implement a JSP program to accept username and password from HTML and display it on another page.

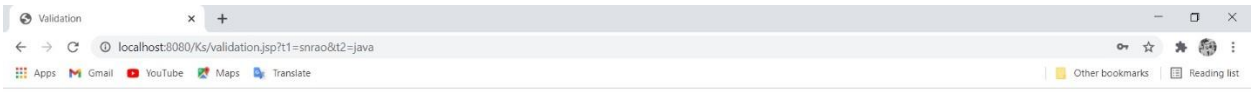
Userpass.html

```
<html>
<body>
<form method="get" action="http://localhost:8888/india/Validation.jsp">
<h3>
   Enter User Name <input type="text" name="t1"> <br>
Enter Password      <input type="password" name="t2"> <br>
   <input type="submit" value="Please Validate">
   <input type="reset" value="Clear Please">
</h3>
</body>
</html>
```

Validation.jsp

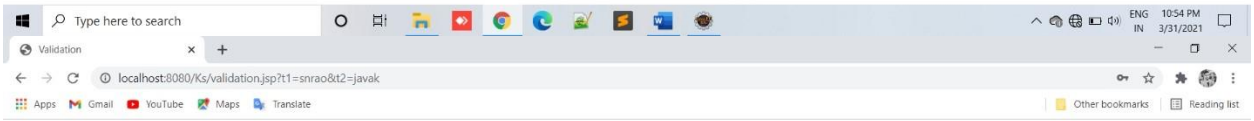
```
<html>
<body>
<h2 align="center"> Validating User Name and Password </h2>

<%
  String str1=request.getParameter("t1");
  String str2=request.getParameter("t2");

  if(str1.equalsIgnoreCase("snrao") && str2.equals("java"))
  { out.println("<h3>Thankyou, you are
    VALID</h3>");
  }
  else
  { out.println("<h3>Sorry, you are
  INVALID</h3>"); }
%>
</body>
</html>
```

Output:



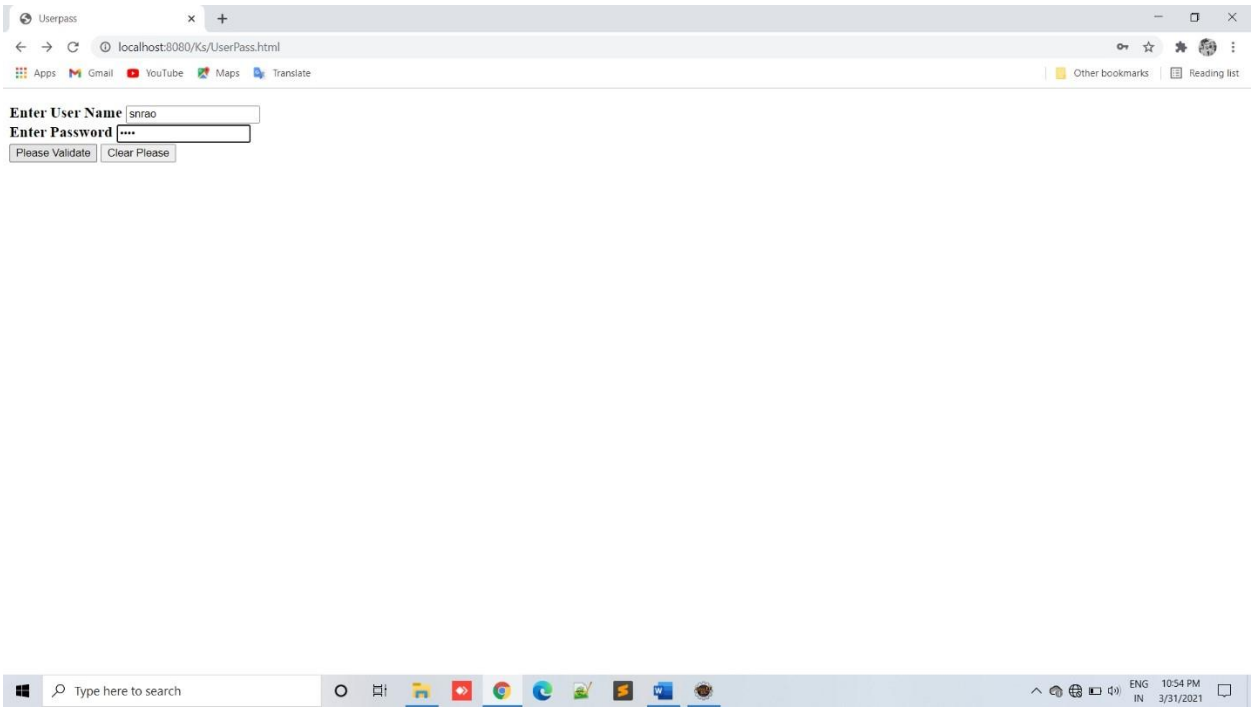
Validating User Name and Password

Thankyou, you are VALID



Validating User Name and Password

Sorry, you are INVALID


Userpass
localhost:8080/Ks/UserPass.html
Apps
Gmail
YouTube
Maps
Translate
Other bookmarks
Reading list
Enter User Name
snrao
Enter Password
Please Validate
Clear Please
Type here to search
ENG
IN
10:54 PM
3/31/2021

Practical-9

Aim: Study and Implement Hibernate.

Hibernate is a high-performance Object/Relational persistence and query service, which is licensed under the open source GNU Lesser General Public License (LGPL) and is free to download. Hibernate not only takes care of the mapping from Java classes to database tables (and from Java data types to SQL data types), but also provides data query and retrieval facilities. This tutorial will teach you how to use Hibernate to develop your database based web applications in simple and easy steps.

Employee.java package com.javatpoint.mypackage;	
	public class Employee {
	private int id;
	private String firstName,lastName;
	public int getId() {
	return id;
	}
	public void setId(int id) {
	this.id = id;
	}
	public String getFirstName() {
	return firstName;
	}
	public void setFirstName(String firstName) {
	this.firstName = firstName;
	}
	public String getLastName() {
	return lastName;
	}
	public void setLastName(String lastName) {
	this.lastName = lastName;
	}
	}

employee.hbm.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 5.3//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-5.3.dtd">
```

```
<hibernate-mapping>
  <class name="com.javatpoint.mypackage.Employee" table="emp1000">
    <id name="id">
     <generator class="assigned"></generator>
    </id>

    <property name="firstName"></property>
    <property name="lastName"></property>

  </class>

 </hibernate-mapping>
```

hibernate.cfg.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD 5.3//EN"
        "http://hibernate.sourceforge.net/hibernate-configuration-5.3.dtd">

<hibernate-configuration>

    <session-factory>
        <property name="hbm2ddl.auto">update</property>
        <property name="dialect">org.hibernate.dialect.Oracle9Dialect</property>
        <property name="connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
        <property name="connection.username">system</property>
        <property name="connection.password">oracle</property>
        <property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>
    <mapping resource="employee.hbm.xml"/>
    </session-factory>

</hibernate-configuration>
```

Output:

D:\hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration SYSTEM "http://hibernate.sourceforge.net/hibernate-configuration-5.3.dtd" PUBLIC "-//Hibernate/Hibernate Configuration DTD 5.3//EN">
- <hibernate-configuration>
  - <session-factory>
      <property name="hbm2ddl.auto">update</property>
      <property name="dialect">org.hibernate.dialect.Oracle9Dialect</property>
      <property name="connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
      <property name="connection.username">system</property>
      <property name="connection.password">oracle</property>
      <property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>
      <mapping resource="employee.hbm.xml"/>
    </session-factory>
</hibernate-configuration>
```

D:\employee.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping SYSTEM "http://hibernate.sourceforge.net/hibernate-mapping-5.3.dtd" PUBLIC "-//Hibernate/Hibernate Mapping DTD 5.3//EN">
- <hibernate-mapping>
  - <class table="emp1000" name="com.javatpoint.mypackage.Employee">
    - <id name="id">
        <generator class="assigned"/>
      </id>
      <property name="firstName"/>
      <property name="lastName"/>
    </class>
</hibernate-mapping>
```

Practical-10

Aim: Study and Implement MVC using Spring Framework.

MVC (Model View Controller)

Model view controller is a software architecture design pattern. It provides solution to layer an application by separating three concerns business, presentation and control flow. Model contains business logic, controller takes care of the interaction between view and model. Controller gets input from view and coverts it in preferable format for the model and passes to it. Then gets the response and forwards to view. View contains the presentation part of the application.

Implementation

Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://java.sun.com/xml/ns/javaee"
   xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
   id="WebApp_ID" version="2.5">
   <display-name>Spring Hello World</display-name>
   <welcome-file-list>
      <welcome-file>/</welcome-file>
   </welcome-file-list>

   <servlet>
      <servlet-name>springDispatcher</servlet-name>
      <servlet-class>
      org.springframework.web.servlet.DispatcherServlet </servlet-
      class>
     <init-param>
         <param-name>contextConfigLocation</param-name>
         <param-value>/WEB-INF/config/spring-context.xml</param-value>
      </init-param>
      <load-on-startup>1</load-on-startup>
```

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

Spring Configuration

```
<?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:mvc="http://www.springframework.org/schema/mvc"
        xmlns:context="http://www.springframework.org/schema/context"
        xsi:schemaLocation="
      http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd
      http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
      http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.0.xsd">

        <context:component-scan base-package="com.javapapers.spring.mvc" />
        <mvc:annotation-driven />

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

Controller ackage

```
com.javapapers.spring.mvc;

import org.springframework.stereotype.Controller; import
org.springframework.ui.Model; import
org.springframework.web.bind.annotation.RequestMapping; import
org.springframework.web.bind.annotation.RequestMethod; import
org.springframework.web.bind.annotation.RequestParam;

@Controller public class
HelloWorldController {

        @RequestMapping("/")
        public String hello() {
        return "hello";
```

```
	}
	@RequestMapping(value = "/hi", method = RequestMethod.GET)
	public String hi(@RequestParam("name") String name, Model model) {
	String message = "Hi " + name + "!"; model.addAttribute("message",
	message);
		return "hi";
	}

}
```

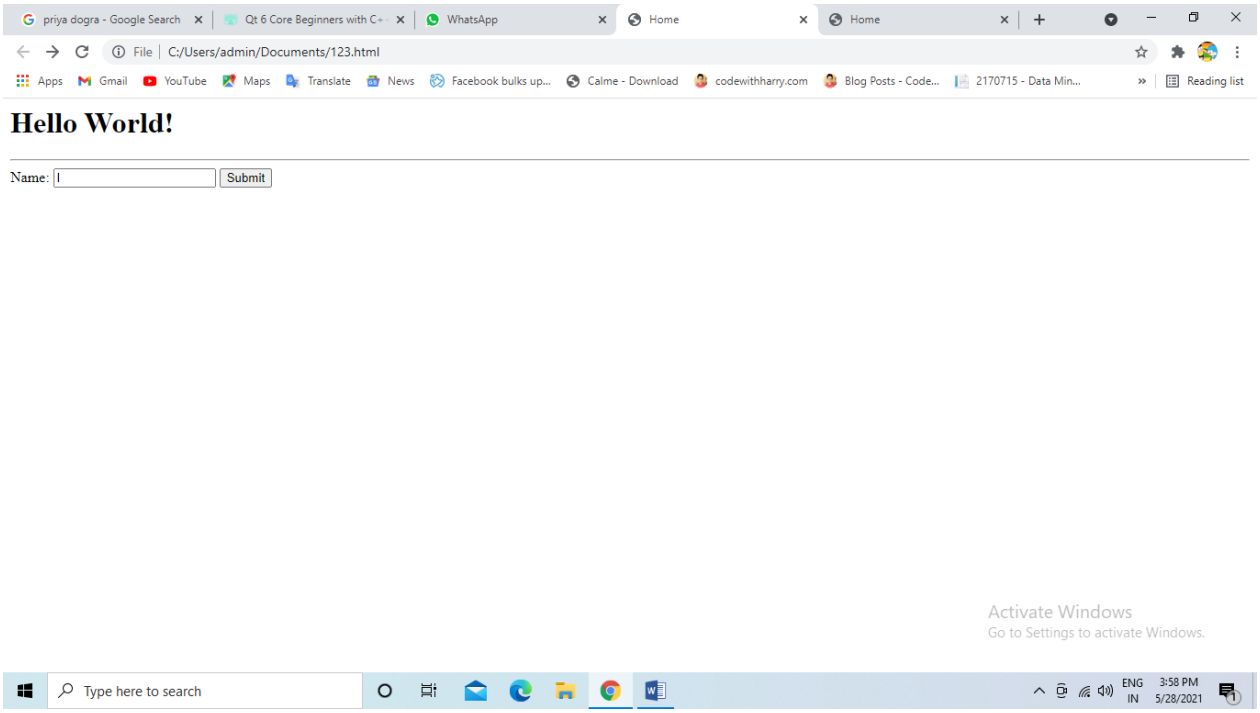
View – Hello World

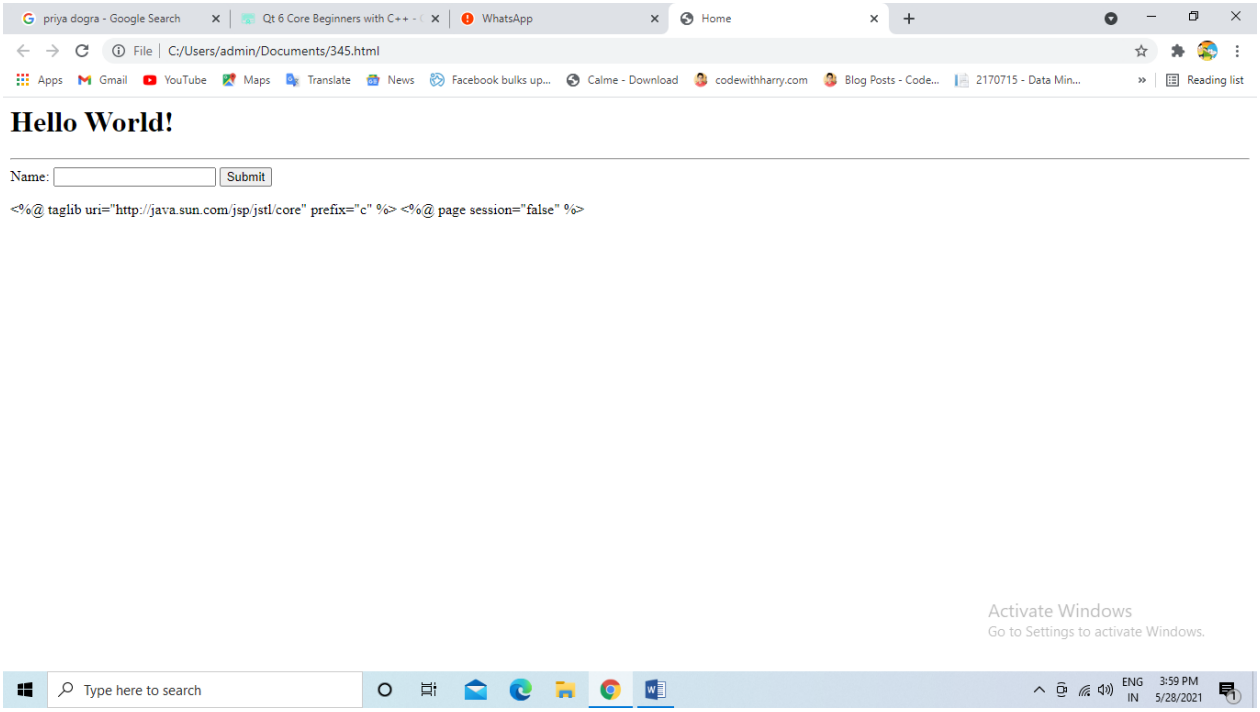
```
<html>
<head>
<title>Home</title>
</head>
<body>
	<h1>Hello World!</h1>

<hr/>

	<form action="hi">
		Name: <input type="text" name="name">	<input type="submit"
value="Submit">
	</form>

</body>
</html>
```

Use key "message" in model to get the value and print it.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page session="false" %>
<html>
   <head>
      <title>Result</title>
   </head>
   <body>
      <h1><c:out value="${message}"></c:out></h1>
   </body>
</html>
```

Output:


priya dogra - Google Search
Qt 6 Core Beginners with C++
WhatsApp
Home
Home
File | C:/Users/admin/Documents/123.html
Apps Gmail YouTube Maps Translate News Facebook bulks up... Calme - Download codewithharry.com Blog Posts - Code... 2170715 - Data Min... Reading list
Hello World!
Name:
Submit
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG IN 3:58 PM 5/28/2021


priya dogra - Google Search
Qt 6 Core Beginners with C++
WhatsApp
Home
File | C:/Users/admin/Documents/345.html
Apps Gmail YouTube Maps Translate News Facebook bulks up... Calme - Download codewithharry.com Blog Posts - Code... 2170715 - Data Min... Reading list
Hello World!
Name:
Submit
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %> <%@ page session="false" %>
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG IN 3:59 PM 5/28/2021

CONCLUSION :

Spring is a versatile framework that allows building MVC applications. Building a simple application with Spring is quick and transparent. The application can also be integrated with a database easily using JPA.