

Practical :- 1

Aim:- Socket Programming.

Practical :- 1.1

Aim:- Write TCP and UDP program for one-way communication.

TCP

Server

Practical_1_TCP_Server.java

```
package Practical_1;
import java.io.IOException;
import java.io.OutputStream;
import java.io.PrintStream;
import java.net.ServerSocket;
import java.net.Socket;
public class Practical_1_TCP_Server {
    public static void main(String[] args) throws IOException {
        try (ServerSocket ss = new ServerSocket(777); Socket s = ss.accept()) {
            System.out.println("Connected");
            OutputStream os=s.getOutputStream();
            try (PrintStream pst = new PrintStream(os)) {
                pst.println("Hello Client! I am Server.");
                pst.println("Bye");
            }
        }
    }
}
```

Client

Practical_1_TCP_Client.java

```
package Practical_1;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.Socket;
public class Practical_1_TCP_Client {
    public static void main(String[] args) throws IOException{
        Socket s=new Socket("localhost",777);
        InputStream is=s.getInputStream();
        BufferedReader bfr=new BufferedReader(new InputStreamReader(is));
        String str;
        while ((str=bfr.readLine())!=null) {
            System.out.println("Server: "+str);
        }
        bfr.close();
        is.close();
        s.close();
    }
}
```

```
}  
}
```

Output :

```
run:  
Connected  
BUILD SUCCESSFUL (total time: 1 second)
```

```
run:  
Server: Hello Client! I am Server.  
Server: Bye  
BUILD SUCCESSFUL (total time: 0 seconds)
```

UDP

Server

Practical_1_UDP_Server.java

```
package Practical_1;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
public class Practical_1_UDP_Server {
    public static void main(String[] args) throws SocketException, IOException {
        DatagramSocket s = new DatagramSocket(9876);
        byte[] recive = new byte[1024];
        byte[] send = new byte[1024];
        while (true) {
            DatagramPacket p = new DatagramPacket(recive, recive.length);
            s.receive(p);
            String message = new String(p.getData());
            System.out.println("Recived: " + message);
            InetAddress ip = p.getAddress();
            int port = p.getPort();
            String cap = message.toUpperCase();
            send = cap.getBytes();
            DatagramPacket sendp = new DatagramPacket(send, send.length, ip, port);
            s.send(sendp);
        }
    }
}
```

Client

Practical_1_UDP_Client.java

```
package Practical_1;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.net.UnknownHostException;

public class Practical_1_UDP_Client {
    public static void main(String[] args) throws SocketException, UnknownHostException,
    IOException {
        BufferedReader inp = new BufferedReader(new InputStreamReader(System.in));
```

```
try (DatagramSocket cs = new DatagramSocket()) {  
    InetAddress ip = InetAddress.getByName("localhost");  
    byte[] send = new byte[1024];  
    byte[] receive = new byte[1024];  
    String message = inp.readLine();  
    send = message.getBytes();  
    DatagramPacket p = new DatagramPacket(send, send.length, ip, 9876);  
    cs.send(p);  
    DatagramPacket receivePacket = new DatagramPacket(receive, receive.length);  
    cs.receive(receivePacket);  
    String modifiedSentence = new String(receivePacket.getData());  
    System.out.println("FROM SERVER:" + modifiedSentence);  
}  
}  
}
```

Output :

```
run:  
Hello  
FROM SERVER:HELLO  
BUILD SUCCESSFUL (total time: 5 seconds)
```

```
run:  
Recived: Hello
```

Practical :- 1.2

Aim:- Write TCP and UDP program for CHAT Application.

TCP

Server

Practical_2_TCP_Server.java

```
package Practical_1;
import java.io.*;
import java.net.*;

public class Practical_2_TCP_Server {
    public static void main(String[] args) throws Exception {
        ServerSocket sersock = new ServerSocket(3100);
        System.out.println("Server ready for chatting");
        Socket sock = sersock.accept();
        BufferedReader keyRead = new BufferedReader(new InputStreamReader(System.in));
        OutputStream ostream = sock.getOutputStream();
        PrintWriter pwrite = new PrintWriter(ostream, true);
        InputStream istream = sock.getInputStream();
        BufferedReader receiveRead = new BufferedReader(new InputStreamReader(istream));
        String receiveMessage, sendMessage;

        while (true) {
            if ((receiveMessage = receiveRead.readLine()) != null) {
                System.out.println("Client: "+receiveMessage);
            }
            sendMessage = keyRead.readLine();
            pwrite.println(sendMessage);
            pwrite.flush();
        }
    }
}
```

Client

Practical_2_TCP_Client.java

```
package Practical_1;
import java.io.*;
import java.net.*;

public class Practical_2_TCP_Client {
    public static void main(String[] args) throws Exception {
        Socket sock = new Socket("127.0.0.1", 3100);
        BufferedReader keyRead = new BufferedReader(new InputStreamReader(System.in));
        OutputStream ostream = sock.getOutputStream();
        PrintWriter pwrite = new PrintWriter(ostream, true);
        InputStream istream = sock.getInputStream();
        BufferedReader receiveRead = new BufferedReader(new InputStreamReader(istream));
```

```
System.out.println("Start the Chat, type and press Enter ");
String receiveMessage, sendMessage;
while (true) {
    sendMessage = keyRead.readLine();
    pwrite.println(sendMessage);
    pwrite.flush();
    if ((receiveMessage = receiveRead.readLine()) != null) {
        System.out.println("Server: "+receiveMessage);
    }
}
}
```

Output :

Client Side

```
run:
Start the Chat, type and press Enter
Hi
Server: Hi
I am Client
Server: I am Server
Bye
Server: Bye
```

Server Side

```
run:
Server ready for chatting
Client: Hi
Hi
Client: I am Client
I am Server
Client: Bye
Bye
```

UDP

Server

Practical_2_UDP_Server.java

```
package Practical_1;
import java.io.*;
import java.net.*;
class Practical_2_UDP_Server {
    public static DatagramSocket serversocket;
    public static DatagramPacket dp;
    public static BufferedReader dis;
    public static InetAddress ia;
    public static byte buf[] = new byte[1024];
    public static int cport = 789, sport = 790;

    public static void main(String[] a) throws IOException {
        serversocket = new DatagramSocket(sport);
        dp = new DatagramPacket(buf, buf.length);
        dis = new BufferedReader(new InputStreamReader(System.in));
        ia = InetAddress.getLocalHost();
        System.out.println("Server is Running...");
        while (true) {
            serversocket.receive(dp);
            String str = new String(dp.getData(), 0,
                dp.getLength());
            System.out.println("Client: " + str);
            String str1 = new String(dis.readLine());
            buf = str1.getBytes();
            serversocket.send(new DatagramPacket(buf, str1.length(), ia, cport));
        }
    }
}
```

Client

Practical_2_UDP_Client.java

```
package Practical_1;
import java.io.*;
import java.net.*;
class Practical_2_UDP_Client {
    public static DatagramSocket clientsocket;
    public static DatagramPacket dp;
    public static BufferedReader dis;
    public static InetAddress ia;
    public static byte buf[] = new byte[1024];
    public static int cport = 789, sport = 790;

    public static void main(String[] a) throws IOException {
```



```
clientsocket = new DatagramSocket(cport);
dp = new DatagramPacket(buf, buf.length);
dis = new BufferedReader(new InputStreamReader(System.in));
ia = InetAddress.getLocalHost();
while (true) {
    String str = new String(dis.readLine());
    buf = str.getBytes();

    clientsocket.send(new DatagramPacket(buf,
        str.length(), ia, sport));
    clientsocket.receive(dp);
    String str2 = new String(dp.getData(), 0,
        dp.getLength());
    System.out.println("Server: " + str2);
}
}
```

Output :

Client Side

```
run:
Hello Server
Server: Hello Client
How are You?
Server: I am Fine
```

Server Side

```
run:
Server is Running...
Client: Hello Server
Hello Client
Client: How are You?
I am Fine
```

Practical :- 1.3

Aim:- Write a TCP client and server program to do following:

client> java client localhost/IP Port <enter>

Enter Text: Welcome to RNGPIT

Response from server: TIPGNR ot emocleW

Server

Practical_3_Server.java

```
package Practical_1;
import java.net.*;
import java.io.*;
public class Practical_3_Server {

    public static void main(String args[]) throws Exception {
        ServerSocket ss = new ServerSocket(7777);
        Socket s = ss.accept();
        System.out.println("Connection Established");
        OutputStream obj = s.getOutputStream();
        PrintStream ps = new PrintStream(obj);
        InputStream obj1 = s.getInputStream();
        BufferedReader br = new BufferedReader(new InputStreamReader(obj1));
        String str = br.readLine();
        String newstr = "";
        for (int i = str.length() - 1; i >= 0; i--) {
            char c = str.charAt(i);
            newstr = newstr + c;
        }
        ps.println(newstr);
        ps.close();
        ss.close();
        s.close();
    }
}
```

Client

Practical_3_Client.java

```
package Practical_1;
import java.net.*;
import java.io.*;
public class Practical_3_Client {

    public static void main(String args[]) throws Exception {
        Socket s = new Socket("localhost", 7777);
        BufferedReader kbr = new BufferedReader(new InputStreamReader(System.in));
        InputStream obj = s.getInputStream();
    }
}
```

```
BufferedReader br = new BufferedReader(new InputStreamReader(obj));
OutputStream os = s.getOutputStream();
PrintStream ps = new PrintStream(os);
System.out.println("Enter text");
String str = kbr.readLine();
ps.println(str);
String newStr = br.readLine();
System.out.println("Response from server=" + newStr);
br.close();
s.close();
}
}
```

Output :

```
run:
Connection Established
BUILD SUCCESSFUL (total time: 6 seconds)
```

```
run:
Enter text
Welcome to RNGPIT
Response from server=TIPGNR ot emocleW
BUILD SUCCESSFUL (total time: 4 seconds)
```

Practical :- 1.4

Aim: Write TCP client and server program to get the date and time details from server on the client request.

Server

Practical_4_Server.java

```
package Practical_1;
import java.net.*;
import java.io.*;
import java.util.Date;
public class Practical_4_Server {

    public static void main(String args[]) throws Exception {
        ServerSocket ss = new ServerSocket(7777);
        while (true) {
            System.out.println("Waiting For Connection ...");
            Socket soc = ss.accept();
            DataOutputStream out = new DataOutputStream(soc.getOutputStream());
            out.writeBytes("Server Date " + (new Date()).toString() + "\n");
            out.close();
            soc.close();
        }
    }
}
```

Client

Practical_4_Client.java

```
package Practical_1;
import java.net.*;
import java.io.*;
public class Practical_4_Client {
    public static void main(String args[]) throws Exception {
        Socket s = new Socket("localhost", 7777);
        BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));
        System.out.println(in.readLine());
    }
}
```

Output :

```
run:
Server Date Thu Apr 08 11:13:00 IST 2021
BUILD SUCCESSFUL (total time: 0 seconds)
```

Practical :- 1.5

Aim: Write a client-server program using TCP or UDP where the client sends 10 numbers and server responds with the numbers in sorted order.

Server

Practical_5_Server.java

```
package Practical_1;
import java.net.*;
import java.io.*;
import java.util.*;
public class Practical_5_Server {
    public static void main(String args[]) throws Exception {
        ServerSocket ss = new ServerSocket(7777);
        Socket s = ss.accept();
        System.out.println("connected..... ");
        DataInputStream din = new DataInputStream(s.getInputStream());
        DataOutputStream dout = new DataOutputStream(s.getOutputStream());
        int r, i = 0;
        int n = din.readInt();
        int a[] = new int[n];
        System.out.println("data:");
        int count = 0;
        System.out.println("Receiving Data....");
        for (i = 0; i < n; i++) {
            a[i] = din.readInt();
        }
        System.out.println("Data Received");
        System.out.println("Sorting Data..... ");
        Arrays.sort(a);
        System.out.println("Data Sorted");
        System.out.println("Sending Data..... ");
        for (i = 0; i < n; i++) {
            dout.writeInt(a[i]);
        }
        System.out.println("\nData Sent Successfully");
        s.close();
        ss.close();
    }
}
```

Client

Practical_5_Client.java

```
package Practical_1;
import java.net.*;
import java.io.*;
import java.util.*;
```

```

public class Practical_5_Client {
    public static void main(String[] args) throws Exception {
        Socket s = new Socket("127.0.0.1", 7777);
        if (s.isConnected()) {
            System.out.println("Connected to server");
        }
        System.out.println("Enter size of array:");
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        int a[] = new int[n];
        System.out.println("Enter element to array:");
        DataOutputStream dout = new DataOutputStream(s.getOutputStream());
        dout.writeInt(n);
        for (int i = 0; i < n;
            i++) {
            int r = scanner.nextInt();
            dout.writeInt(r);
        }
        System.out.println(
            "Data Sent");
        DataInputStream din = new DataInputStream(s.getInputStream());
        int r;
        System.out.println("Receiving Sorted Data....");
        for (int i = 0; i < n; i++) {
            r = din.readInt();
            System.out.print(r + " ");
        }
        s.close();
    }
}

```

Output :

```

run:
Connected to server
Enter size of array:
10
Enter element to array:
5 8 1 4 10 9 2 6 3 7
Data Sent
Receiving Sorted Data....
1 2 3 4 5 6 7 8 9 10 BUILD SUCCESSFUL (total time: 37 seconds)

```

```

run:
connected.....
data:
Receiving Data....
Data Received
Sorting Data.....
Data Sorted
Sending Data.....

Data Sent Successfully
BUILD SUCCESSFUL (total time: 39 seconds)

```

Practical :- 1.6

Aim: Write a client-server program using TCP where client sends a string and server checks whether string is palindrome or not and responds with appropriate message.

Server

Practical_6_Server.java

```
package Practical_1;
import java.net.*;
import java.io.*;
public class Practical_6_Server {
    public static void main(String args[]) throws Exception {
        ServerSocket ss = new ServerSocket(7777);
        Socket s = ss.accept();
        System.out.println("Connection Established");
        OutputStream obj = s.getOutputStream();
        PrintStream ps = new PrintStream(obj);
        InputStream obj1 = s.getInputStream();
        BufferedReader br = new BufferedReader(new InputStreamReader(obj1));
        String str = br.readLine();
        String newstr = "";
        for (int i = str.length() - 1; i >= 0; i--) {
            char c = str.charAt(i);
            newstr = newstr + c;
        }
        if (str.equalsIgnoreCase(newstr)) {
            ps.println("string is palindrome ");
        } else {
            ps.println("string is not palindrome ");
        }
        ps.close();
        ss.close();
        s.close();
    }
}
```

Client

Practical_6_Client.java

```
package Practical_1;
import java.net.*;
import java.io.*;
public class Practical_6_Client {
    public static void main(String args[]) throws Exception {
        Socket s = new Socket("localhost", 7777);
        BufferedReader kbr = new BufferedReader(new InputStreamReader(System.in));
```

```
InputStream obj = s.getInputStream();
BufferedReader br = new BufferedReader(new InputStreamReader(obj));
OutputStream os = s.getOutputStream();
PrintStream ps = new PrintStream(os);
System.out.println("Enter text");
String str = kbr.readLine();
ps.println(str);
String newStr = br.readLine();
System.out.println("Response from server=" + newStr);
br.close();
s.close();
}
}
```

Output :

```
run:
Enter text
Vinay
Response from server=string is not palindrome
BUILD SUCCESSFUL (total time: 15 seconds)
```

```
run:
Enter text
Nayan
Response from server=string is palindrome
BUILD SUCCESSFUL (total time: 3 seconds)
```

Practical :- 1.7

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

Server**Practical_7_Server.java**

```

package Practical_1;
import java.io.*;
import java.net.*;
class Practical_7_Server {
    public static void main(String args[]) throws Exception {
        ServerSocket ss = new ServerSocket(7777);
        Socket s = ss.accept();
        System.out.println("connected.....");
        FileInputStream fin = new FileInputStream("F://190843131006//6th
Semester//AJP//Practical//src//Send.txt");
        DataOutputStream dout = new DataOutputStream(s.getOutputStream());
        int r;
        while ((r = fin.read()) != -1) {
            dout.write(r);
        }
        System.out.println("\nFiletransfer Completed");
        s.close();
        ss.close();
    }
}

```

Client**Practical_7_Client.java**

```

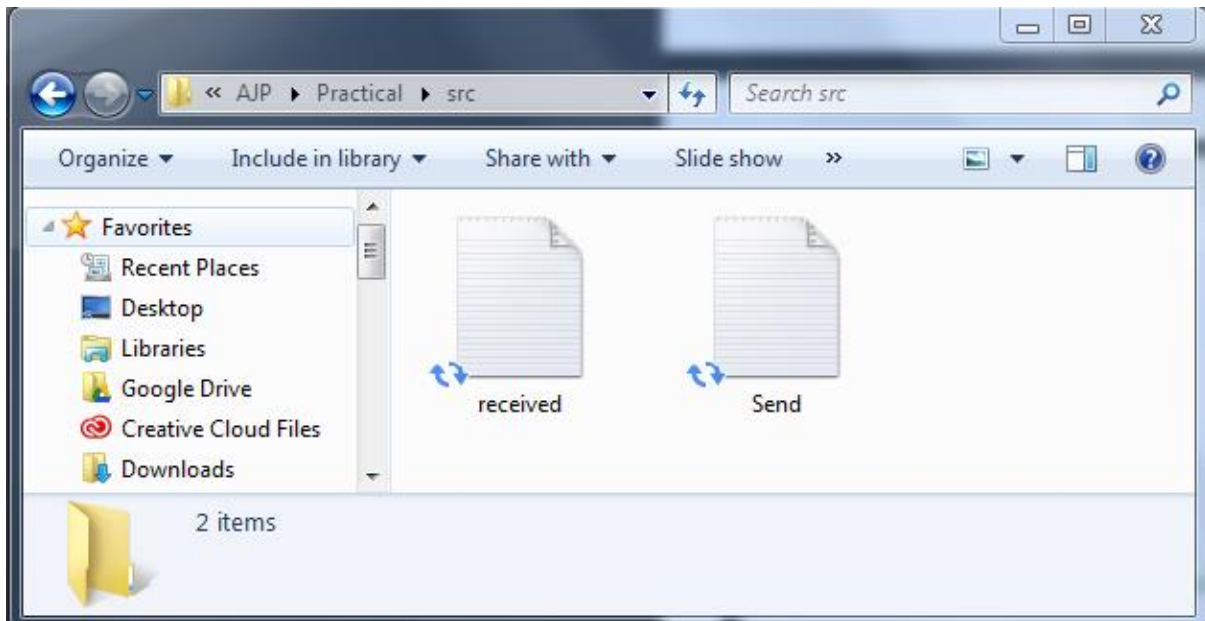
package Practical_1;
import java.io.*;
import java.net.*;
public class Practical_7_Client {
    public static void main(String[] args) throws Exception {
        Socket s = new Socket("localhost", 7777);
        if (s.isConnected()) {
            System.out.println("Connected to server");
        }
        FileOutputStream fout = new FileOutputStream("F://190843131006//6th
Semester//AJP//Practical//src//received.txt");
        DataInputStream din = new DataInputStream(s.getInputStream());
        int r;
        while ((r = din.read()) != -1) {
            fout.write((char) r);
        }
    }
}

```

Output :

```
run:
connected.....

Filetransfer Completed
BUILD SUCCESSFUL (total time: 2 seconds)
```



Practical:-2

Aim:- Write java programs to perform following task using JDBC.

Practical:-2.1

Aim:- To create JDBC Connection.

```
package Practical2;

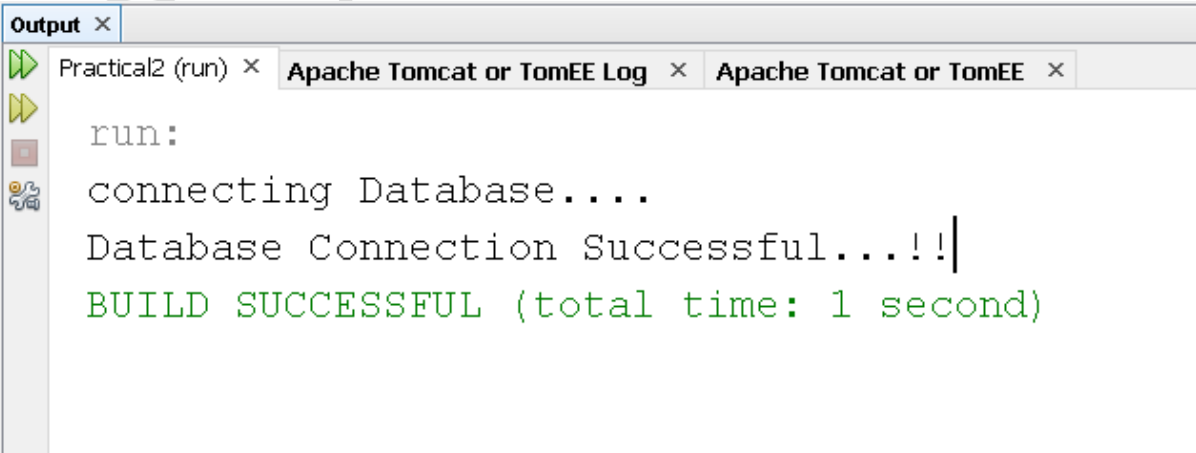
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Practical2_1 {

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

        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("connecting Database....");
        Connection con;
        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/STUDENTS","root","");
        System.out.println("Database Connection Successful...!!");
        con.close();
    }
}
```

Output:-



The screenshot shows an IDE output window with the following content:

```
Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
run:
connecting Database....
Database Connection Successful...!!
BUILD SUCCESSFUL (total time: 1 second)
```

Practical:-2.2

Aim:- Execute and read select queries using JDBC.

```
package Practical2;

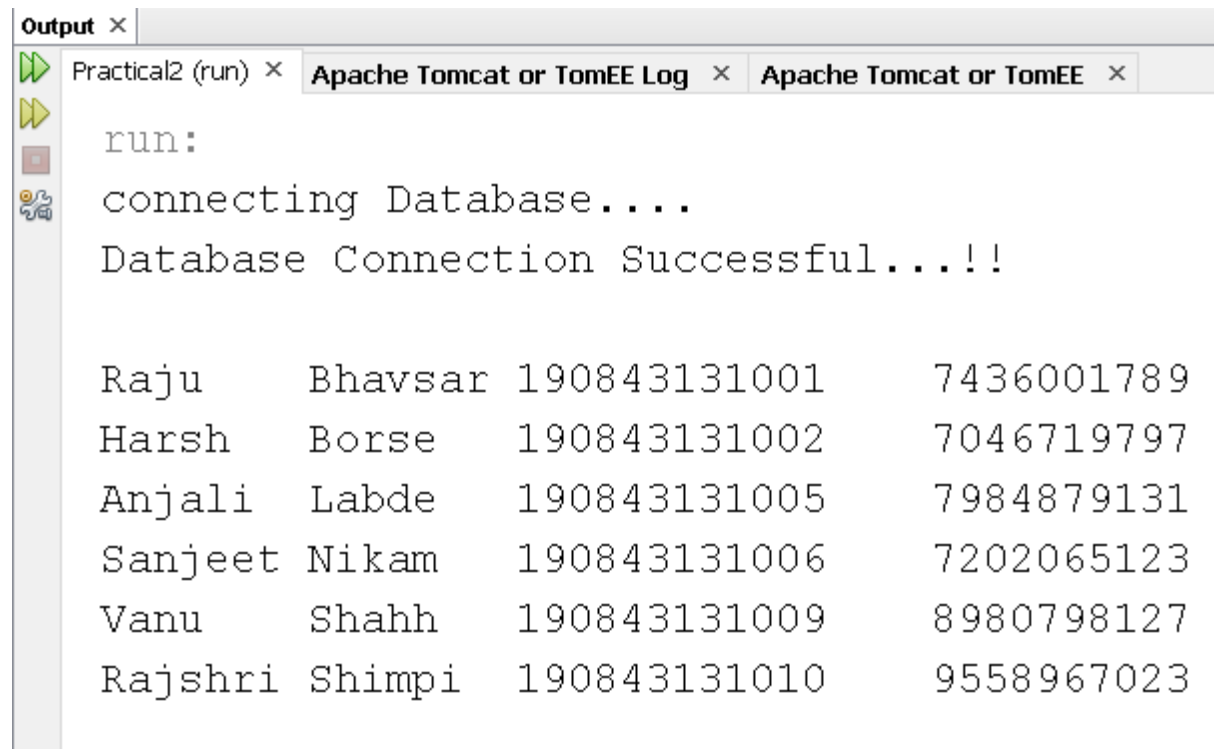
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Practical2_2 {
    public static void main(String[] args) throws ClassNotFoundException, SQLException {

        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("connecting Database....");
        try (Connection con =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/students","root","")) {
            System.out.println("Database Connection Successful...!!");
            try (Statement st = (Statement) con.createStatement()) {
                ResultSet rs;
                rs = st.executeQuery("select * from details");
                while(rs.next())
                {
                    System.out.println();
                    System.out.print(rs.getString(1)+"\t");
                    System.out.print(rs.getString(2)+"\t");
                    System.out.print(rs.getString(3)+"\t");
                    System.out.print(rs.getString(4)+"\t");
                }
            }
        }
    }
}
```

```
}  
}  
}  
}
```

Output:-



```
run:  
connecting Database....  
Database Connection Successful...!!  
  
Raju      Bhavsar 190843131001      7436001789  
Harsh     Borse   190843131002      7046719797  
Anjali    Labde   190843131005      7984879131  
Sanjeet   Nikam   190843131006      7202065123  
Vanu      Shahh   190843131009      8980798127  
Rajshri   Shimpi  190843131010      9558967023
```

Practical:-2.3

Aim:- Update a record in the database using JDBC.

```
package Practical2;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Practical2_3 {

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

        try
        {
            String query;
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("connecting Database....");

            try (Connection con =
                DriverManager.getConnection("jdbc:mysql://localhost:3306/STUDENTS","root","")) {

                System.out.println("Database Connection Successful...!!");

                try (Statement st = (Statement) con.createStatement()) {

                    query = " update details set Contact_no=8155918780 where First_name='Rajshri'";

                    st.executeUpdate(query);

                    System.out.println("1 Record updated succesfully...");

                    ResultSet rs;

                    rs = st.executeQuery("select * from details");

                    while(rs.next())
                    {

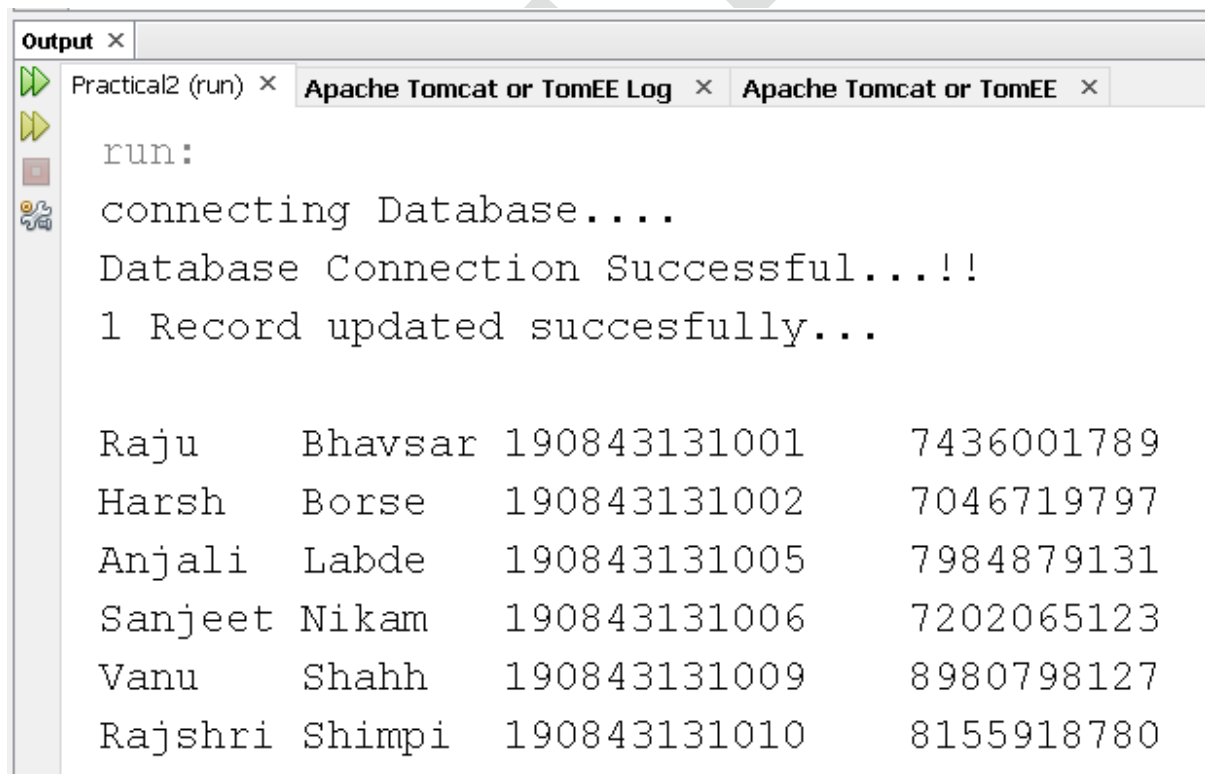
                        System.out.println();
```

```

        System.out.print(rs.getString(1)+"\t");
        System.out.print(rs.getString(2)+"\t");
        System.out.print(rs.getString(3)+"\t");
        System.out.print(rs.getString(4)+"\t");
    }
}
}
} catch(ClassNotFoundException | SQLException e)
{
    System.out.println(e.toString());
}
}
}

```

Output:-



```

Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
run:
connecting Database....
Database Connection Successful....!!
1 Record updated succesfully...

Raju      Bhavsar  190843131001    7436001789
Harsh     Borse    190843131002    7046719797
Anjali    Labde    190843131005    7984879131
Sanjeet   Nikam    190843131006    7202065123
Vanu      Shahh    190843131009    8980798127
Rajshri   Shimpi   190843131010    8155918780

```


Practical:-2.4

Aim:- Execute any type of query in JDBC.

```
package Practical2;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Practical2_4 {

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

        String query;

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

        System.out.println("connecting Database....");

        try (Connection con =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/STUDENTS","root","")) {

            System.out.println("Database Connection Successful...!!");

            try (Statement st = (Statement) con.createStatement()) {

                query = " insert into details values('Bhumi','Mistry',190843131005,7436001234)";

                st.execute(query);

                System.out.println("1 Record inserted succesfully...");

                ResultSet rs;

                rs = st.executeQuery("select * from details");

                while(rs.next())

                {

                    System.out.println();

                    System.out.print(rs.getString(1)+"\t");

                    System.out.print(rs.getString(2)+"\t");

                    System.out.print(rs.getString(3)+"\t");

                    System.out.print(rs.getString(4)+"\t");

                }

            }

        }

    }

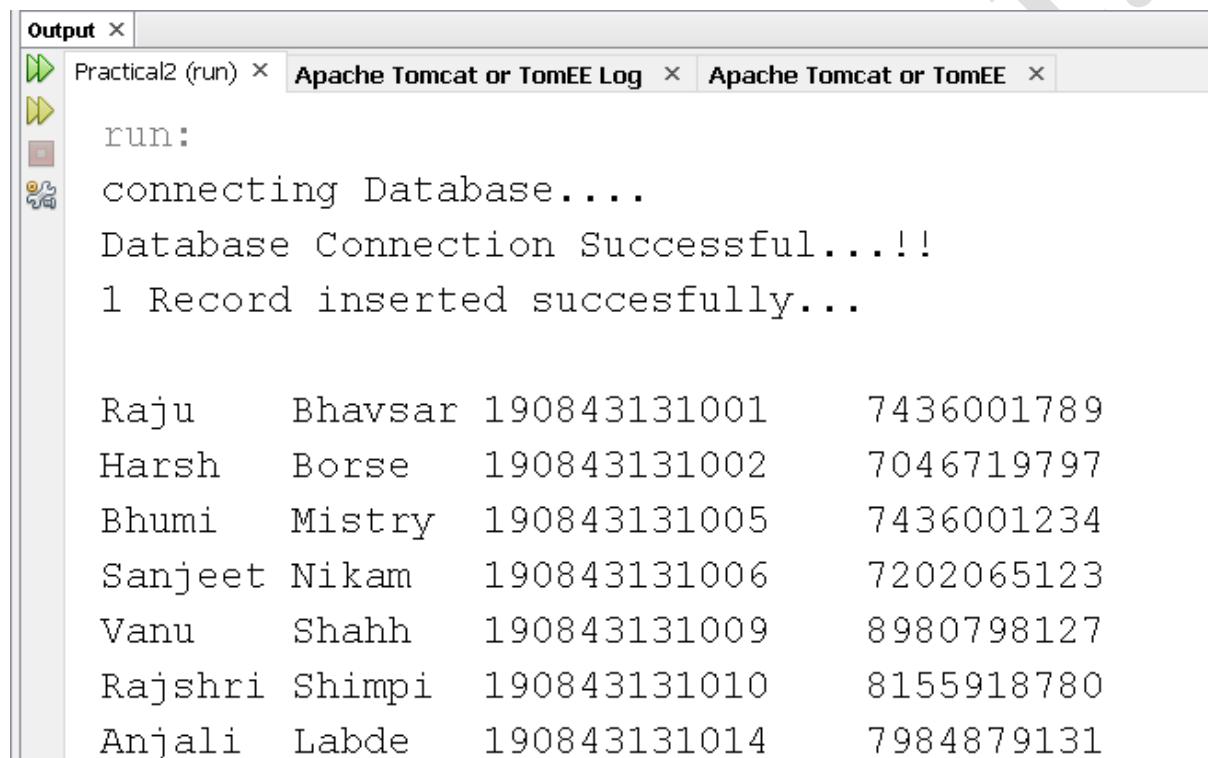
}
```

```

        System.out.print(rs.getString(4)+"\t");
    }
}
}
}
}
}

```

Output:-



```

Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
run:
connecting Database....
Database Connection Successful....!!
1 Record inserted succesfully...

Raju      Bhavsar 190843131001      7436001789
Harsh     Borse   190843131002      7046719797
Bhumi     Mistry  190843131005      7436001234
Sanjeet   Nikam   190843131006      7202065123
Vanu      Shahh   190843131009      8980798127
Rajshri   Shimpi  190843131010      8155918780
Anjali    Labde   190843131014      7984879131

```

Practical:-2.5

Aim:- Use of JDBC prepared statement with ResultSet.

```
package Practical2;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Practical2_5 {

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

        String query;

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

        System.out.println("connecting Database....");

        try (Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/STUDENTS","root","")) {

            System.out.println("Database Connection Successful...!!");

            try (Statement st = (Statement) con.createStatement()) {

                query = " insert into details values(?,?,?,?)";

                PreparedStatement ps=con.prepareStatement(query);

                ps.setString(1,"Kinjal");

                ps.setString(2, "Patel");

                ps.setString(3,"190843131007");

                ps.setString(4,"7046719000");

                int i=ps.executeUpdate();

                System.out.println(" No of Record updated is :"+i);

                ResultSet rs;

                rs = st.executeQuery("select * from details");

                while(rs.next())

                {
```

```

        System.out.println();
        System.out.print(rs.getString(1)+"\t");
        System.out.print(rs.getString(2)+"\t");
        System.out.print(rs.getString(3)+"\t");
        System.out.print(rs.getString(4)+"\t");
    }
}
}
}
}
}

```

Output:-

```

Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
Database Connection Successful...!!
No of Record updated is :1

Raju      Bhavsar 190843131001      7436001789
Harsh     Borse   190843131002      7046719797
Bhumi     Mistry  190843131005      7436001234
Sanjeet   Nikam   190843131006      7202065123
Kinjal    Patel   190843131007      7046719000
Vanu      Shahh   190843131009      8980798127
Rajshri   Shimpi  190843131010      8155918780
Anjali    Labde   190843131014      7984879131

```

Practical:-2.6

Aim:- Get primary key value (auto-generated keys) from inserted queries using JDBC.

```
package Practical2;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Practical2_6 {

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

        PreparedStatement pstmt = null;
        ResultSet rs = null;
        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("connecting Database....");

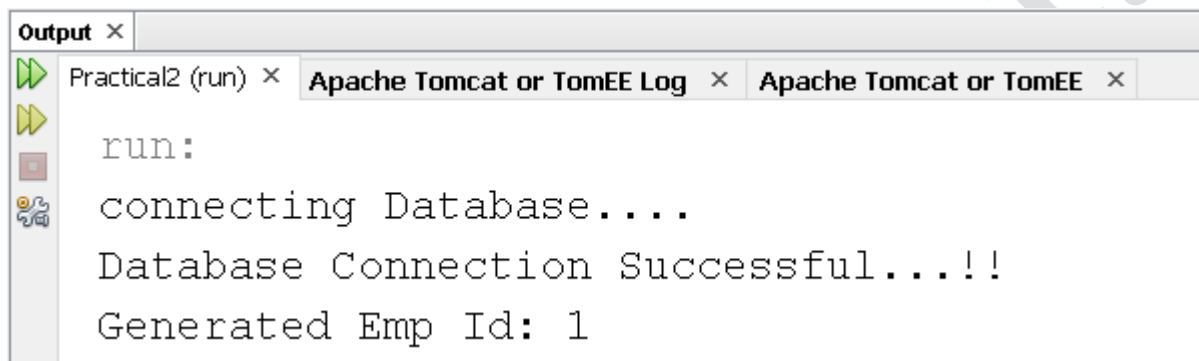
        try (Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/STUDENTS","root","")) {

            System.out.println("Database Connection Successful...!!");

            String query = "insert into emps (name, dept, salary) values (?, ?, ?)";
            pstmt = con.prepareStatement(query, Statement.RETURN_GENERATED_KEYS);
            pstmt.setString(1, "Sanjeet");
            pstmt.setString(2, "CSE Dept");
            pstmt.setInt(3, 90000);
            pstmt.executeUpdate();
            rs = pstmt.getGeneratedKeys();
            while(rs != null && rs.next()){
```

```
        System.out.println("Generated Emp Id: "+rs.getInt(1));  
    }  
  
    }  
}  
}
```

Output:-

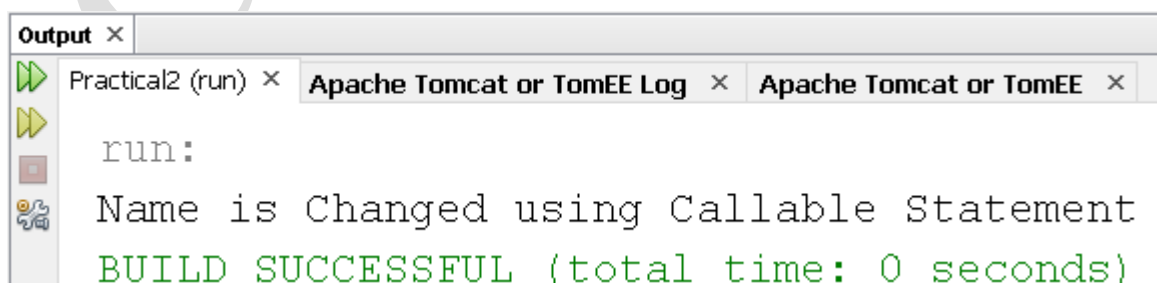


Practical:-2.7

Aim:- To execute stored procedure using Callable Statement statement.

```
package Practical2;
import java.sql.*;
public class Practical2_7 {
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            try (Connection conn =
                DriverManager.getConnection("jdbc:mysql://localhost:3306/STUDENTS","root",""));
                CallableStatement cs = conn.prepareCall("{ call setname(?,?)}") {
                    cs.setInt(1,2);
                    cs.setString(2,"Saurabh");
                    cs.execute();
                    System.out.println("Name is Changed using Callable Statement");
                }
            } catch (ClassNotFoundException | SQLException e){System.out.println(e.toString());}
        }
    }
}
```

Output:-



```
Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
run:
Name is Changed using Callable Statement
BUILD SUCCESSFUL (total time: 0 seconds)
```

Practical:-2.8

Aim:- CallableStatement statement with stored procedure returns OUT parameters.

```
package Practical2;

import java.sql.*;

public class Practical2_8 {

    public static void main(String[] args) {

        try {

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

            try (Connection conn =
                DriverManager.getConnection("jdbc:mysql://localhost:3306/STUDENTS","root","");
                CallableStatement cs = conn.prepareCall("{ call gettitle(?,?)}")) {

                cs.setInt(1,1);

                cs.registerOutParameter(2,Types.VARCHAR);

                cs.execute();

                System.out.println("CallableStatement statement with stored procedure returns OUT
parameter(NAME):- "+cs.getString(2));

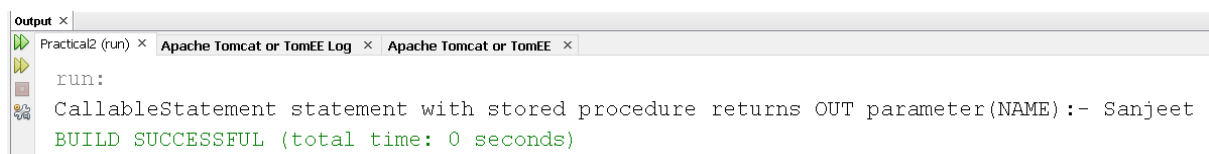
            }

        } catch (ClassNotFoundException | SQLException e){System.out.println(e.toString());}

    }

}
```

Output:-



```
Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
run:
CallableStatement statement with stored procedure returns OUT parameter(NAME):- Sanjeet
BUILD SUCCESSFUL (total time: 0 seconds)
```


Practical:-2.9

Aim:- Callable Statement statement with batch execution.

```
package Practical2;

import java.sql.*;
import java.util.Arrays;

public class Practical2_9 {
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.jdbc.Driver");

            try (Connection conn =
                DriverManager.getConnection("jdbc:mysql://localhost:3306/STUDENTS","root","");
                CallableStatement cs = conn.prepareCall("{call gettitle(?,?)}")) {

                String query1,query2,query3,query4,query5;

                query1="create table student(enr INT PRIMARY KEY, name VARCHAR(20),sem
                INT,branch VARCHAR(10))";

                query2="insert into student values(1906,'Sanjeet',6,'CSE')";
                query3="insert into student values(1910,'Rajeshree',6,'CSE')";
                query4="update student set name='Sanju' where enr=1906";
                query5="delete from student where name='Rajeshree'";

                cs.addBatch(query1);
                cs.addBatch(query2);
                cs.addBatch(query3);
                cs.addBatch(query4);
                cs.addBatch(query5);

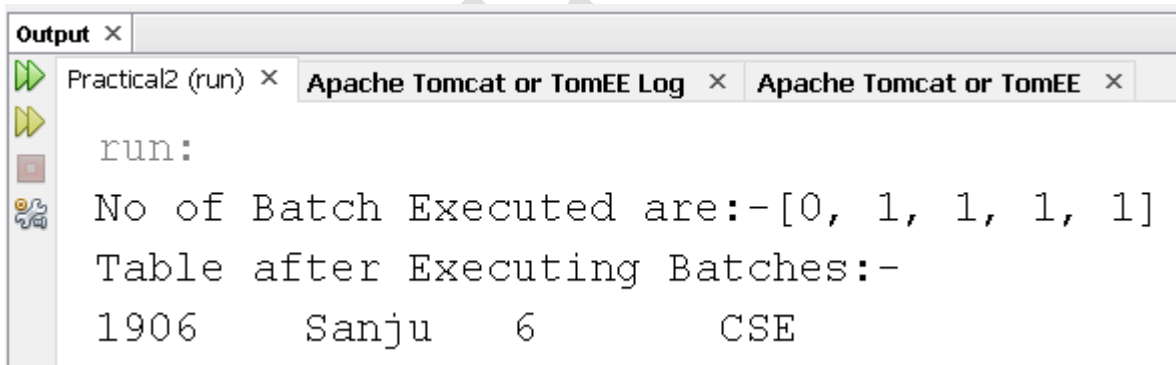
                ResultSet rs;

                int[] i=cs.executeBatch();

                System.out.println("No of Batch Executed are:-"+Arrays.toString(i));

                rs = cs.executeQuery("select * from student");
```

```
System.out.println("Table after Executing Batches:-");
while(rs.next())
{
    System.out.print(rs.getInt(1)+"\t");
    System.out.print(rs.getString(2)+"\t");
    System.out.print(rs.getInt(3)+"\t");
    System.out.print(rs.getString(4)+"\t");
}
System.out.println();
    System.out.println();
}
} catch(ClassNotFoundException | SQLException e){System.out.println(e.toString());}
}
}
```

Output:-

```
Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
run:
No of Batch Executed are:-[0, 1, 1, 1, 1]
Table after Executing Batches:-
1906      Sanju      6          CSE
```

Practical:-2.10

Aim:- To execute SQL function using Callable Statement.

Practical_2_10.java

```
package Practical2;

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Types;

public class Practical2_10 {

    public static void main(String[] args) throws ClassNotFoundException, SQLException {
        try{
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("connecting Database....");

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

            System.out.println("Database Connection Successful...!!");
            CallableStatement cs = con.prepareCall("{ ? = call newFunction(?) }");

            cs.registerOutParameter(1, Types.DATE);

            cs.setString(2,"Sanjeet");

            cs.execute();

            System.out.println("Date of birth: "+cs.getDate(1));

            System.out.println();

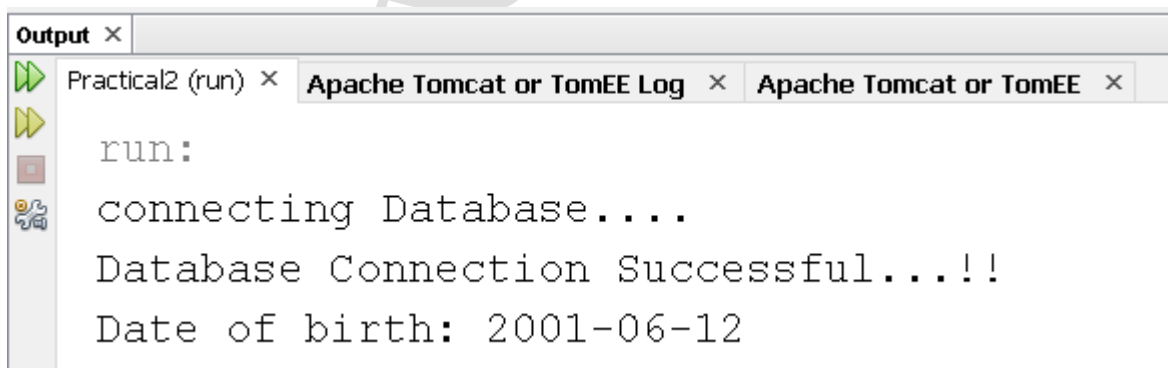
        }catch(ClassNotFoundException | SQLException e)
        {
```

```
        System.out.println(e.toString());  
    }  
}  
}
```

SQL Function: newFunction()

```
CREATE DEFINER=`root` @`localhost` FUNCTION `newFunction`(  
    `f_name` VARCHAR(50)  
)  
RETURNS date  
LANGUAGE SQL  
DETERMINISTIC  
CONTAINS SQL  
SQL SECURITY DEFINER  
COMMENT "  
BEGIN  
  
    DECLARE dateofbirth DATE;  
  
    SELECT dob INTO dateofbirth FROM sample WHERE sname=f_name;  
  
    RETURN dateofbirth;  
  
END
```

Output:-



Practical :- 2.11

Aim:- To execute SQL cursors using Callable Statement.

Practical2_11.java

```
package Practical_2;
import java.sql.CallableStatement;
import java.sql.Connection; import
java.sql.DriverManager; import
java.sql.SQLException; import
java.sql.Types;
public class Practical_2_11 {
    public static void main(String[] args) { try
    {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con
        =DriverManager.getConnection("jdbc:mys
        ql://localhost:3306/ajp", "root", "");
        CallableStatement cst =
        con.prepareCall("{?= call find(?)}")
        ;
        cst.setInt(2, 29);
        cst.registerOutParameter(1, Types.CHAR);
        cst.execute();
        System.out.println("Name: " + cst.getString(1));
    } catch (ClassNotFoundException | SQLException e) {
        System.out.println(e);
    }
    }
}
```

SQL Function: find()

```
CREATE FUNCTION find(nm INT)
RETURNS varchar(30) CHARSET latin1
BEGIN
    DECLARE n varchar(30);
    DECLARE c1 CURSOR FOR SELECT Name FROM student WHERE No=nm;
    OPEN c1;
    FETCH c1 INTO
    n; CLOSE c1;
    RETURN n;
END
```

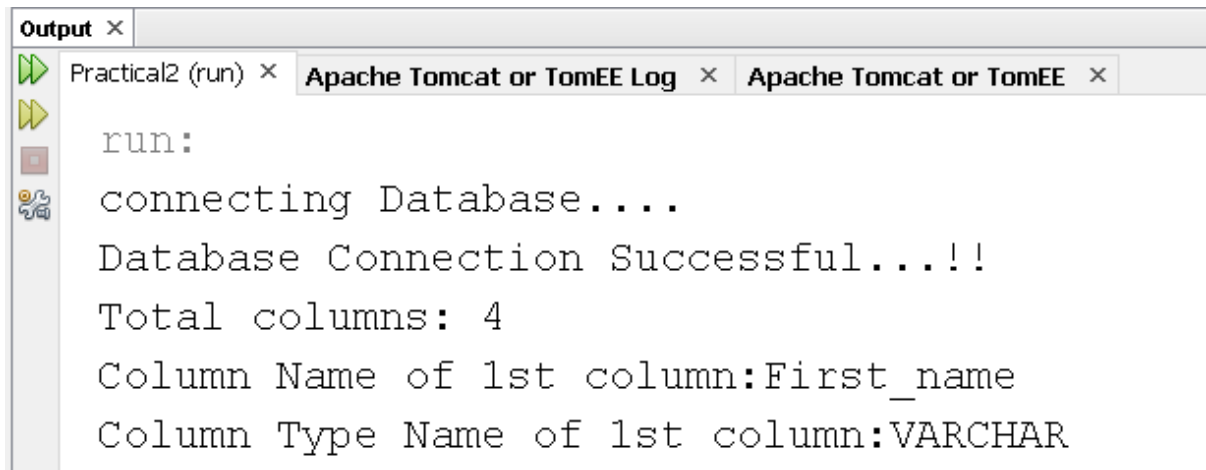
Output:-

```
run:
Name: Bhautik
BUILD SUCCESSFUL (total time: 0 seconds)
```

Practical:-2.12

Aim:- Get column properties from Result Set using Result Set Meta Data.

```
package Practical2;
import java.sql.*;
public class Practical2_12 {
    public static void main(String[] args) throws ClassNotFoundException, SQLException {
        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("connecting Database....");
        try (Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/STUDENTS","root","")) {
            System.out.println("Database Connection Successful...!!");
            try (Statement st = (Statement) con.createStatement()) {
                ResultSet rs = st.executeQuery("SELECT * from details");
                ResultSetMetaData rsmd=rs.getMetaData();
                System.out.println("Total columns: "+rsmd.getColumnCount());
                System.out.println("Column Name of 1st column:"+rsmd.getColumnName(1));
                System.out.println("Column Type Name of 1st column: " +rsmd.getColumnTypeName(1));
                st.close();
                con.close();
            }
        }
        catch(Exception e)
        {System.out.println(e.toString());}
    }
}
```

Output:-

The screenshot shows an IDE output window with the title 'Output'. It contains three tabs: 'Practical2 (run)', 'Apache Tomcat or TomEE Log', and 'Apache Tomcat or TomEE'. The 'Practical2 (run)' tab is active and displays the following text:

```
run:
connecting Database....
Database Connection Successful...!!
Total columns: 4
Column Name of 1st column:First_name
Column Type Name of 1st column:VARCHAR
```

A large, light gray watermark 'CS&E Dep' is visible diagonally across the lower half of the page.

Practical:-2.13

Aim:- Batch update using Statement.

```
package Practical2;

import java.sql.*;
import java.util.Arrays;

public class Practical2_13 {
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            try (Connection conn =
                DriverManager.getConnection("jdbc:mysql://localhost:3306/STUDENTS","root","");
                Statement cs = conn.createStatement()) {
                String query1,query2,query3,query4,query5;
                query1="create table student18(enr INT PRIMARY KEY, name VARCHAR(20),sem
                INT,branch VARCHAR(10))";
                query2="insert into student18 values(06,'Sanjeet',6,'CSE')";
                query3="insert into student18 values(10,'Rajeshree',6,'CSE')";
                query4="update student18 set name='Sanju' where enr=1906";
                query5="delete from student18 where name='Rajeshree'";

                cs.addBatch(query1);
                cs.addBatch(query2);
                cs.addBatch(query3);
                cs.addBatch(query4);
                cs.addBatch(query5);

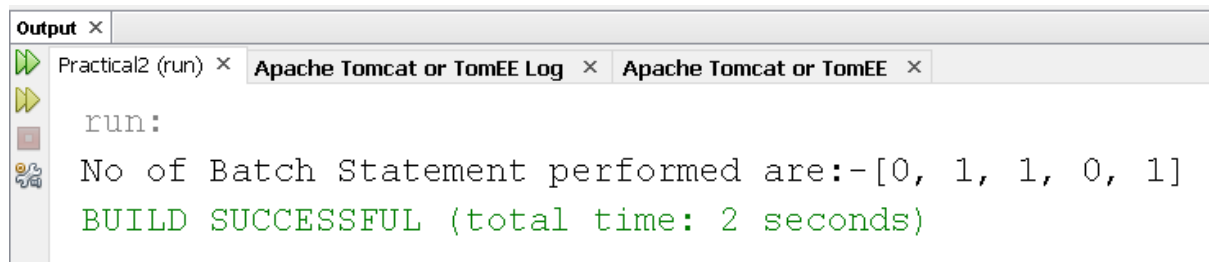
                int[] i=cs.executeBatch();

                System.out.println("No of Batch Statement performed are:-" +Arrays.toString(i));
```



```
    }  
    }catch(ClassNotFoundException | SQLException e){System.out.println(e.toString());}  
    }  
}
```

Output:-



```
Output ×  
Practical2 (run) × Apache Tomcat or TomEE Log × Apache Tomcat or TomEE ×  
run:  
No of Batch Statement performed are:-[0, 1, 1, 0, 1]  
BUILD SUCCESSFUL (total time: 2 seconds)
```

Practical:-2.14

Aim:- Batch update using PreparedStatement.

```
package Practical2;

import java.sql.*;
import java.util.Arrays;

public class Practical2_14 {

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

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

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

        String query= "update details set First_name=? , Last_name=? where Enrollment=?";

        PreparedStatement preparedStatement = null;

        preparedStatement =conn.prepareStatement(query);

        preparedStatement.setString(1, "Sanju");
        preparedStatement.setString(2, "Nick");
        preparedStatement.setString (3,"190843131006" );
        preparedStatement.addBatch();

        preparedStatement.setString(1, "Raju");
        preparedStatement.setString(2, "Bhavs");
        preparedStatement.setString (3,"190843131001" );
        preparedStatement.addBatch();

        preparedStatement.setString(1, "Vanu");
        preparedStatement.setString(2, "Shahh");
        preparedStatement.setString (3,"190843131009" );
        preparedStatement.addBatch();

        preparedStatement.setString(1, "Harshh");
        preparedStatement.setString(2, "Borsee");
        preparedStatement.setString (3,"190843131002" );
```

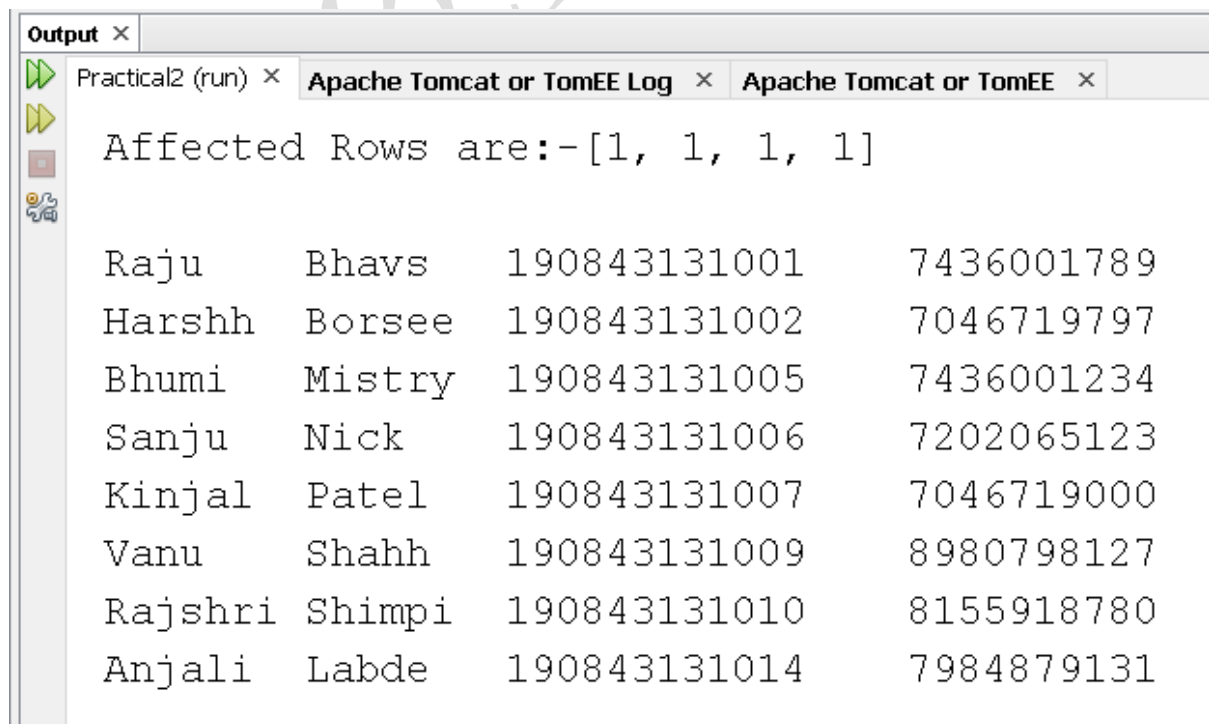
```

preparedStatement.addBatch();
int[] affectedRecords = preparedStatement.executeBatch();
System.out.println("Affected Rows are:-" +Arrays.toString(affectedRecords));
ResultSet rs;

    rs = preparedStatement.executeQuery("select * from details");
    while(rs.next())
    {
        System.out.println();
        System.out.print(rs.getString(1)+"\t");
        System.out.print(rs.getString(2)+"\t");
        System.out.print(rs.getString(3)+"\t");
        System.out.print(rs.getString(4)+"\t");
    }
}
}

```

Output:-



The screenshot shows an IDE output window with the following content:

```

Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
Affected Rows are:-[1, 1, 1, 1]

Raju      Bhavs      190843131001      7436001789
Harshh    Borse     190843131002      7046719797
Bhumi     Mistry     190843131005      7436001234
Sanju     Nick       190843131006      7202065123
Kinjale   Patel      190843131007      7046719000
Vanu      Shahh      190843131009      8980798127
Rajshree  Shimpi     190843131010      8155918780
Anjali    Labde      190843131014      7984879131

```

Practical:-2.15

Aim:- Scrollable result set with read only mode.

```
package Practical2;

import java.sql.*;

public class Practical2_15 {

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

        try (Connection con =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/students","root","")) {

            System.out.println("Database Connection Successful...!!");

            try (Statement st = (Statement)
                con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                    ResultSet.CONCUR_READ_ONLY)) {

                ResultSet rs;

                rs = st.executeQuery("select * from details");

                System.out.println();

                System.out.println("ResultSet Cursor is at before first: "+rs.isBeforeFirst());

                while(rs.next())
                {
                    System.out.print(rs.getString(1)+"\t");
                    System.out.print(rs.getString(2)+"\t");
                    System.out.print(rs.getString(3)+"\t");
                    System.out.print(rs.getString(4)+"\t");

                    System.out.println();

                }

                System.out.println();

                System.out.println("Is After Last: "+rs.isAfterLast());

                while(rs.previous())
                {
```

```

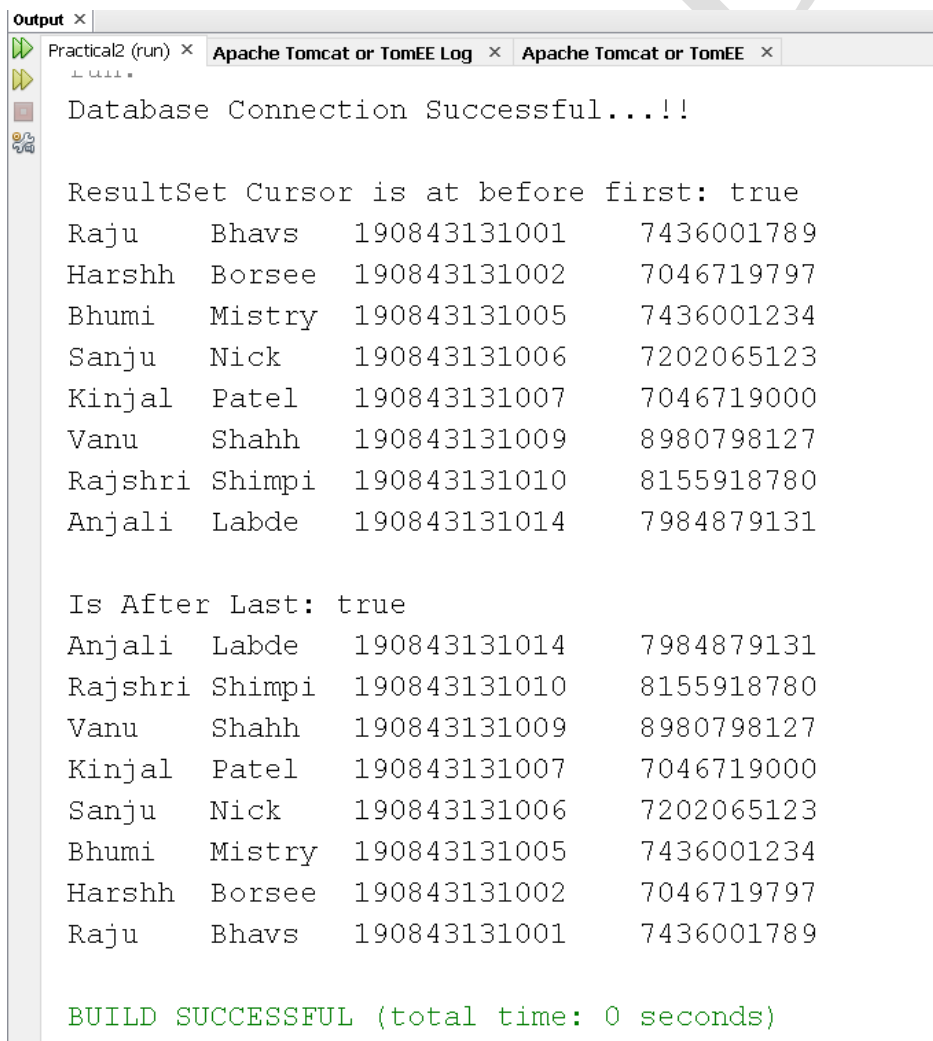
        System.out.print(rs.getString(1)+"\t");
        System.out.print(rs.getString(2)+"\t");
        System.out.print(rs.getString(3)+"\t");
        System.out.print(rs.getString(4)+"\t");

        System.out.println();
    }

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

```

Output:-



```

Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
Database Connection Successful...!!

ResultSet Cursor is at before first: true
Raju      Bhavs      190843131001      7436001789
Harshh    Borsee     190843131002      7046719797
Bhumi     Mistry     190843131005      7436001234
Sanju     Nick       190843131006      7202065123
Kinjal    Patel      190843131007      7046719000
Vanu      Shahh      190843131009      8980798127
Rajshri   Shimpi     190843131010      8155918780
Anjali    Labde      190843131014      7984879131

Is After Last: true
Anjali    Labde      190843131014      7984879131
Rajshri   Shimpi     190843131010      8155918780
Vanu      Shahh      190843131009      8980798127
Kinjal    Patel      190843131007      7046719000
Sanju     Nick       190843131006      7202065123
Bhumi     Mistry     190843131005      7436001234
Harshh    Borsee     190843131002      7046719797
Raju      Bhavs      190843131001      7436001789

BUILD SUCCESSFUL (total time: 0 seconds)

```

Practical:-2.16

Aim:- Updatable result set.

```
package Practical2;

import java.sql.*;

public class Practical2_16 {

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

        try (Connection con =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/students","root","")) {

            System.out.println("Database Connection Successful...!!");

            try (Statement st = (Statement)
                con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE)) {

                ResultSet rs;

                rs = st.executeQuery("select * from details");

                while(rs.next())
                {
                    if("190843131006".equals(rs.getString(3))){
                        rs.updateString(1, "Sanjeet");
                        rs.updateRow();
                        System.out.println("Record updated!!!");
                    }
                }

                System.out.println("After updating Table:-");

                rs = st.executeQuery("select * from details");

                while(rs.next())
                {
                    System.out.println();
                }
            }
        }
    }
}
```

```

        System.out.print(rs.getString(1)+"\t");
        System.out.print(rs.getString(2)+"\t");
        System.out.print(rs.getString(3)+"\t");
        System.out.print(rs.getString(4)+"\t");
    }

}
}
}
}

```

Output:-

```

Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
run:
Database Connection Successful....!!
Record updated!!!
After updating Table:-

Raju      Bhavs      190843131001      7436001789
Harshh    Borse     190843131002      7046719797
Bhumi     Mistry    190843131005      7436001234
Sanjeet   Nick      190843131006      7202065123
Kinjal    Patel     190843131007      7046719000
Vanu      Shahh     190843131009      8980798127
Rajshri   Shimpi    190843131010      8155918780
Anjali    Labde     190843131014      7984879131

```

Practical:-2.17

Aim:- (1) To insert an image (BLOB) into database table. (2) To read an image (BLOB) into database table.

(1)

```
package Practical2;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.sql.*;

public class Practical2_17_1 {

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

        try (Connection con =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/students","root","")) {

            System.out.println("Database Connection Successful...!!");

            try (PreparedStatement ps = (PreparedStatement)con.prepareStatement("insert into
                student_profile values (?,?)")) {

                InputStream is=null;
                ps.setString(1, "190843131006");
                is = new FileInputStream(new File("E:\\Documents\\Degree\\SEM4Result.jpeg"));
                ps.setBinaryStream(2, is);
                int count = ps.executeUpdate();

                System.out.println("Count: "+count);

                System.out.println("Enrollment and Image is Stored Successfully in Database");

            }

        }

    }
}
```

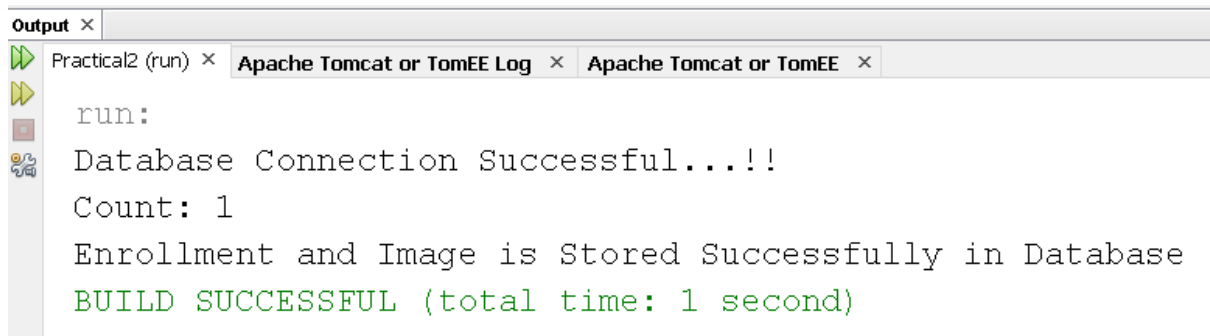


```

    }
}

```

Output:-



```

Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
run:
Database Connection Successful...!!
Count: 1
Enrollment and Image is Stored Successfully in Database
BUILD SUCCESSFUL (total time: 1 second)

```

(2)

```

package Practical2;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.sql.*;

public class Practical2_17_2 {

    public static void main(String[] args) throws ClassNotFoundException, SQLException,
        FileNotFoundException, IOException {

        try (Connection con =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/students","root","")) {

            System.out.println("Database Connection Successful...!!");

            try (Statement st = (Statement)con.createStatement()) {

                InputStream is=null;

                FileOutputStream os = null;

                ResultSet rs= null;

```

```

rs = st.executeQuery("select result from student_profile where enroll=190843131006 ");

File theFile = new File("SEM4Result.jpeg");

os = new FileOutputStream(theFile);

if (rs.next()) {

    is = rs.getBinaryStream("result");

    System.out.println("Reading Result Image from database...");

    byte[] buffer = new byte[1024];
    while (is.read(buffer) > 0) {
        os.write(buffer);
    }

    System.out.println("\nSaved to file: " +
theFile.getAbsolutePath());

    System.out.println("\nCompleted successfully!");

}

}

}

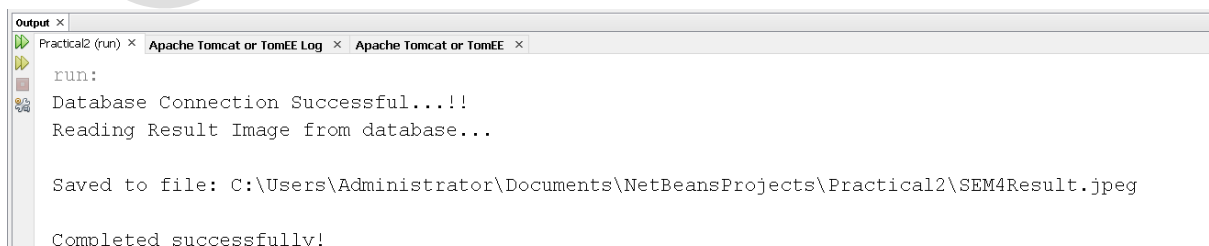
}

}

}

```

Output:-



```

Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
run:
Database Connection Successful...!!
Reading Result Image from database...

Saved to file: C:\Users\Administrator\Documents\NetBeansProjects\Practical2\SEM4Result.jpeg

Completed successfully!

```

Practical:-2.18

Aim:- Demonstrate the use of DatabaseMetaData.

```
package Practical2;

import java.sql.*;

class Practical2_18{

public static void main(String args[]){
try{

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

    try (Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/students","root","")) {
        System.out.println("Database Connection Successful...!!");

        DatabaseMetaData dbmd=con.getMetaData();

        System.out.println("Driver Name: "+dbmd.getDriverName());
        System.out.println("Driver Version: "+dbmd.getDriverVersion());
        System.out.println("UserName: "+dbmd.getUserName());
        System.out.println("Database Product Name: "+dbmd.getDatabaseProductName());
        System.out.println("Database Product Version: "+dbmd.getDatabaseProductVersion());

        String table[]={ "TABLE" };

        ResultSet rs=dbmd.getTables(null,null,null,table);

        System.out.println();
        System.out.println();
        System.out.println("Database Tables are:- ");

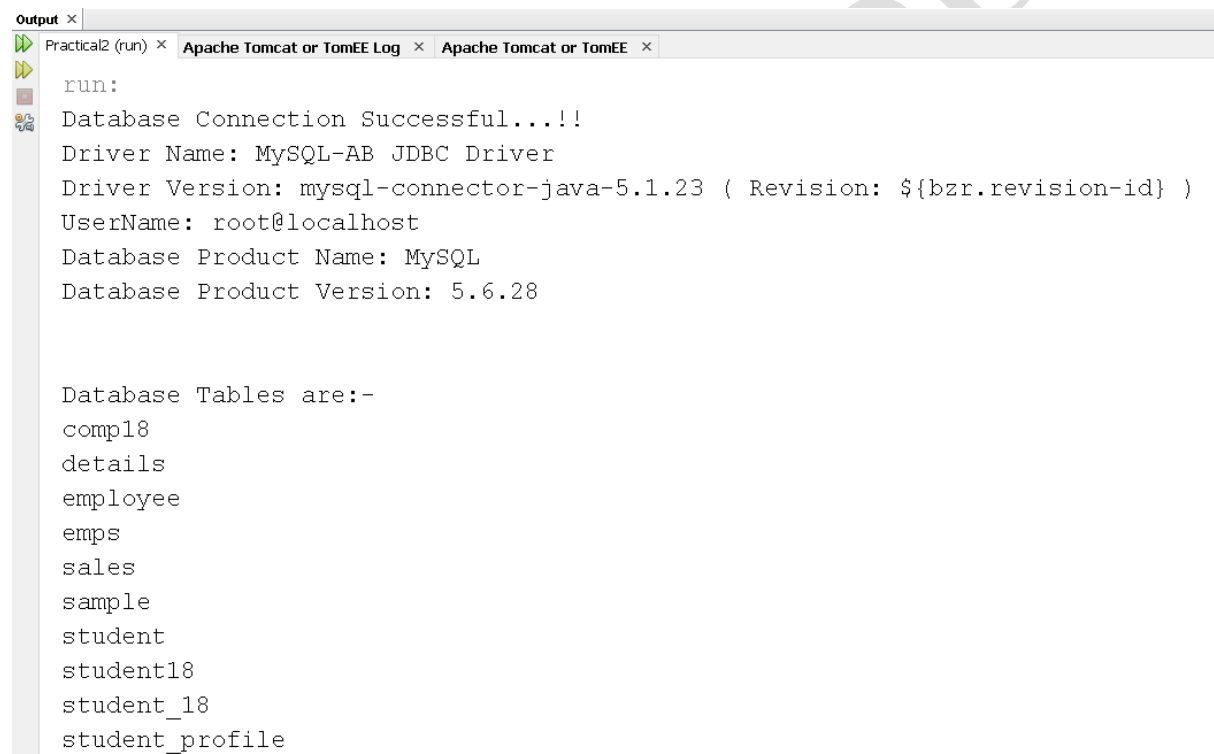
        while(rs.next()){
```

```
        System.out.println(rs.getString(3));
    }

    con.close();
}

}catch(ClassNotFoundException | SQLException e){ System.out.println(e);}
}
}
```

Output:-



```
Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
run:
Database Connection Successful....!!
Driver Name: MySQL-AB JDBC Driver
Driver Version: mysql-connector-java-5.1.23 ( Revision: ${bzd.revision-id} )
UserName: root@localhost
Database Product Name: MySQL
Database Product Version: 5.6.28

Database Tables are:-
comp18
details
employee
emps
sales
sample
student
student18
student_18
student_profile
```

Practical:-2.19

Aim:- To get JDBC Connection object using properties file.

Jdbc.properties

```
MYSQLJDBC.driver="com.mysql.jdbc.Driver"
MYSQLJDBC.url="jdbc:mysql://localhost:3306/STUDENTS"
MYSQLJDBC.username="root"
MYSQLJDBC.password=""
```

Practical2_19.java

```
package Practical2;

import java.io.FileInputStream;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class Practical2_19 {

    public static Properties loadPropertiesFile() throws Exception {

        Properties prop = new Properties();
        try (InputStream in = new FileInputStream("jdbc.properties")) {
            prop.load(in);
        }
        return prop;
    }

    public static void main(String[] args) {

        System.out.println("create jdbc connection using properties file");

        Connection con = null;

        try {

            Properties prop = loadPropertiesFile();

            String driverClass = prop.getProperty("MYSQLJDBC.driver");
            String url = prop.getProperty("MYSQLJDBC.url");
            String username = prop.getProperty("MYSQLJDBC.username");
            String password = prop.getProperty("MYSQLJDBC.password");
```

```

        System.out.println("Driver Class:- "+driverClass);
        System.out.println("URL:- "+url);
        System.out.println("Username:- "+username);
        System.out.println("Password:- "+password);
        Class.forName(driverClass);

        con = DriverManager.getConnection(url, username, password);

        if (con != null) {
            System.out.println("connection created successfully using properties file");
        }

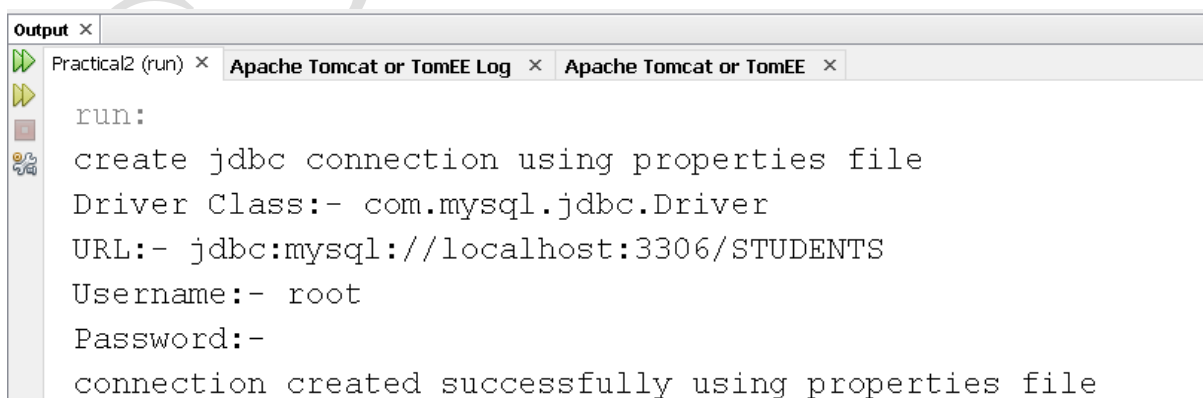
        else {
            System.out.println(" unable to create connection");
        }

    } catch (SQLException e) {
    }
    catch (Exception e) {
    } finally {

        try {
            if (con != null) {
                con.close();
            }
        } catch (SQLException ex) {
        }
    }
}
}

```

Output:-



```

Output x
Practical2 (run) x Apache Tomcat or TomEE Log x Apache Tomcat or TomEE x
run:
create jdbc connection using properties file
Driver Class:- com.mysql.jdbc.Driver
URL:- jdbc:mysql://localhost:3306/STUDENTS
Username:- root
Password:-
connection created successfully using properties file

```

Practical:-2.20

Aim:- To implement commit and rollback.

```
package Practical2;

import java.sql.*;
class Practical2_20{
public static void main(String args[]){

try{
Class.forName("com.mysql.jdbc.Driver");
try (Connection con = DriverManager.getConnection(
    "jdbc:mysql://localhost:3306/STUDENTS","root","")) {
    con.setAutoCommit(false);//bydefault it is true
    Statement stmt=con.createStatement();
    int i=stmt.executeUpdate("insert into sample values(5,'Rajshri','Surat','2000-12-20')");
    con.commit();//commit transaction

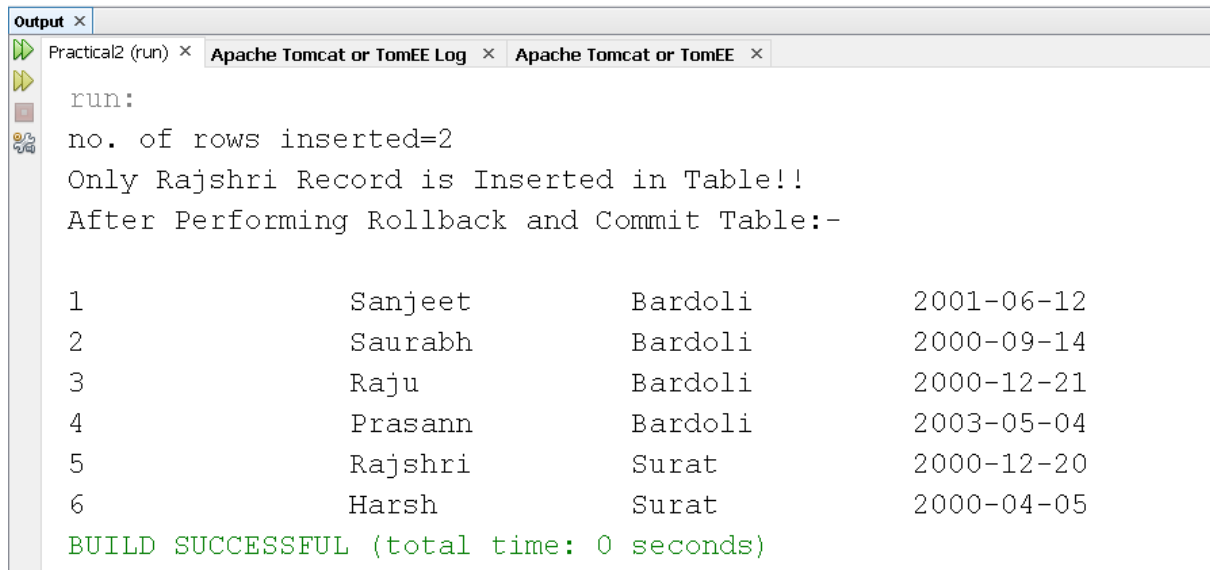
    i+=stmt.executeUpdate("insert into sample values(7,'Anjali','Surat','2000-06-17')");
    System.out.println("no. of rows inserted="+i);
    con.rollback(); //Rollback Transaction

    ResultSet rs;
    rs = stmt.executeQuery("select * from sample");
    System.out.println("Only Rajshri Record is Inserted in Table!!");
    System.out.println("After Performing Rollback and Commit Table:- ");
    while(rs.next())
    {
        System.out.println();
        System.out.print(rs.getString(1)+"\t"+"");
        System.out.print(rs.getString(2)+"\t"+"");
        System.out.print(rs.getString(3)+"\t"+"");
        System.out.print(rs.getDate(4)+"\t"+"");
    }
    System.out.println();
    con.close();

    }

}

catch(ClassNotFoundException | SQLException e)
{
System.out.println(e);
}
}
```

Output:-

```
run:
no. of rows inserted=2
Only Rajshri Record is Inserted in Table!!
After Performing Rollback and Commit Table:-

1          Sanjeet          Bardoli          2001-06-12
2          Saurabh          Bardoli          2000-09-14
3          Raju             Bardoli          2000-12-21
4          Prasann          Bardoli          2003-05-04
5          Rajshri          Surat            2000-12-20
6          Harsh            Surat            2000-04-05
BUILD SUCCESSFUL (total time: 0 seconds)
```


Practical:-3

Aim:- Implement java programs using Servlets.

Practical:-3.1

Aim:- Write a Servlet program to print system date and time.

Practical3_1.java

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

public class Practical3_1 extends HttpServlet{

    @Override

    public void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException{
        PrintWriter pw = response.getWriter();
        Date today = new Date();
        pw.println("<html>"+ "<body><h1>Today Date is:- </h1>");
        pw.println("<b>"+ today+ "</b></body>"+ "</html>");
    }
}
```

Output:-



Practical:-3.2

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.

Index.html

```
<html>

<head>

    <title>Students Details</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

    <center>

        <h1><b>Perform Following Operation</b></h1>

        <table border="">

            <tr><td><a href="insert.html" with="100"
height="100">REGISTRATION</a></td></tr>

            <tr><td><a href="search.html" with="100"
height="100">SEARCH</a></td></tr>

            <tr><td><a href="update.html" with="100"
height="100">UPDATE</a></td></tr>

            <tr><td><a href="delete.html" with="100"
height="100">DELETE</a></td></tr>        </table>

    </center>

</body>

</html>
```

Insert.html

```
<html>

<head>

    <title>Student Details</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

    <center>

        <form action="Servlet1">

            <h1><b>REGISTRATION FORM</b></h1>

            <table border="">

                <tr><td>First Name:</td><td><input type="text" name="fname"
value=""></td></tr>

                <tr><td>Last Name:</td><td><input type="text" name="lname"
value=""></td></tr>

                <tr><td>Semester:</td><td><input type="number" name="sem"
value=""></td></tr>

                <tr><td>Contact No:</td><td><input type="text" name="contact"
value=""></td></tr>

                <tr><td>Enrollment:</td><td><input type="text" name="enroll"
value=""></td></tr>

            </table>

            <table border="">

                <br>

                <tr><td><input type="submit" name="Save" value="Save"></td></tr>

            </table>

        </form>

    </center>

</body>

</html>
```

Search.html

```
<html>

<head>

    <title>Student Details</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

    <center>

        <form action="Servlet4">

            <h1><b>SEARCH RECORD</b></h1>

            <table border="">

                <tr><td>Enter Name You want to search a record :</td><td><input type="text"
name="fname" value=""></td></tr>

            </table>

            <table border="">

                <br>

                <tr><td><input type="Submit" name="Save" value="Search"></td></tr>

            </table>

        </form>

    </center>

</body>

</html>
```

Update.html

```
<html>

<head>

    <title>Student Details</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>
```

```

<body>

<center>

<form action="Servlet2">

    <h1><b>UPDATE RECORD</b></h1>

    <table border="">

        <tr><td>Enter Enrollment You want to update a record :</td><td><input
type="text" name="enroll" value=""></td></tr>

        <tr><td>Enter Name of your Enrollment You want to update a
record :</td><td><input type="text" name="fname" value=""></td></tr>

    </table>

    <table border="">

        <br>

        <tr><td><input type="Submit" name="Save" value="UPDATE"></td></tr>

    </table>

</form>

</center>

</body>

</html>

```

Delete.html

```

<html>

<head>

    <title>Student Details</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

    <center>

    <form action="Servlet3">

        <h1><b>DELETE RECORD</b></h1>

        <table border="">

```

```

        <tr><td>Enter Name You want to delete record :</td><td><input type="text"
name="fname" value=""></td></tr>

        </table>

        <table border="">

            <br>

            <tr><td><input type="Submit" name="Save" value="DELETE"></td></tr>

        </table>

    </form>

</center>

</body>

</html>

```

Servlet1.java

```

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class Servlet1 extends HttpServlet {

    Statement st=null;

    Connection con=null;

    static final String DB_URL="jdbc:mysql://localhost:3306/students";

    static final String USER="root";

    static final String PASS="";

```

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

        String s1,s2,s3,s4,s5,sql;

        s1=request.getParameter("fname");
        s2=request.getParameter("lname");
        s3=request.getParameter("sem");
        s4=request.getParameter("contact");
        s5=request.getParameter("enroll");

        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet Data Insertion</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h3>First Name: " + s1 + "</h3>");
        out.println("<h3>Last Name: " + s2 + "</h3>");
        out.println("<h3>Semester: " + s3 + "</h3>");
        out.println("<h3>Contact: " + s4 + "</h3>");
        out.println("<h3>Enrollment: " + s5 + "</h3>");
        out.println("</body>");
        out.println("</html>");

        Class.forName("com.mysql.jdbc.Driver");
        con = DriverManager.getConnection(DB_URL,USER,PASS);
        st = con.createStatement();
```



```

        sql=" insert into student_18(fname,lname,enroll,contact,sem)
values('"+s1+"','"+s2+"','"+s5+"','"+s4+"', '"+s3+"')";

        st.executeUpdate(sql);

        System.out.println("Record Inserted Sucessfully...");

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

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

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

@Override
public String getServletInfo() {
    return "Short description";
}

}

```

Servlet2.java

```

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;

```

```
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class Servlet2 extends HttpServlet {
    Statement st=null;
    Connection con=null;
    ResultSet rs;
    static final String DB_URL="jdbc:mysql://localhost:3306/students";
    static final String USER="root";
    static final String PASS="";
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String s1,s2,sql;
            s1=request.getParameter("fname");
            s2=request.getParameter("enroll");
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Record Updation</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h3>Name Changed Succesfully </h3>");
            out.println("</body>");
```

```

        out.println("</html>");
Class.forName("com.mysql.jdbc.Driver");
con = DriverManager.getConnection(DB_URL,USER,PASS);
st = con.createStatement();
sql=" update student_18 set fname= '"+s1+"' where enroll='"+s2+"' ";
st.executeUpdate(sql);
System.out.println("Record Updated Sucessfully...");

rs = st.executeQuery("select * from student_18 where fname='"+s1+"'");
while(rs.next())
{
    out.println("<h5>First Name:-"+rs.getString(1)+ "</h5>");
    out.println("<h5>Last Name:-"+rs.getString(2)+ "</h5>");
    out.println("<h5>Enrollment:-"+rs.getString(3)+ "</h5>");
    out.println("<h5>Contact No:-"+rs.getString(4)+ "</h5>");
    out.println("<h5>Sem:-"+rs.getString(5)+ "</h5>");

}
    out.println("</body>");
    out.println("</html>");
} catch(Exception e)
{
    System.out.println(e.toString());
}
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

```

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

Servlet3.java

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.Statement;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class Servlet3 extends HttpServlet {

Statement st=null;

Connection con=null;

static final String DB_URL="jdbc:mysql://localhost:3306/students";

static final String USER="root";

static final String PASS="";

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */

        String s1,sql;

        s1=request.getParameter("fname");
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Record Updation</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h3>Record Deleted of name: " + s1 + "</h3>");
        out.println("</body>");
        out.println("</html>");

        Class.forName("com.mysql.jdbc.Driver");
        con = DriverManager.getConnection(DB_URL,USER,PASS);
        st = con.createStatement();
        sql=" delete from student_18 where fname='"+s1+"' ";
        st.executeUpdate(sql);
        System.out.println("Record Deleted Sucessfully...");

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

```

    }
}

void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

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

@Override
public String getServletInfo() {
    return "Short description";
}
}

```

Servlet4.java

```

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;

```

```
import javax.servlet.http.HttpServletResponse;

public class Servlet4 extends HttpServlet {

    Statement st=null;

    Connection con=null;

    static final String DB_URL="jdbc:mysql://localhost:3306/students";

    static final String USER="root";

    static final String PASS="";

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        try (PrintWriter out = response.getWriter()) {

            String s1,sql;

            s1=request.getParameter("fname");

            out.println("<!DOCTYPE html>");

            out.println("<html>");

            out.println("<head>");

            out.println("<title>Record Updation</title>");

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

            out.println("<body>");

            out.println("<h3>Record Searched Value of name: " + s1 + "</h3>");

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

            con = DriverManager.getConnection(DB_URL,USER,PASS);

            st = con.createStatement();

            System.out.println("Record Search Sucessfully...");

            ResultSet rs;

            rs = st.executeQuery("select * from student_18 where fname='"+s1+"'");

            while(rs.next())
```

```
{
    out.println("<h5>First Name:-"+rs.getString(1)+ "</h5>");
    out.println("<h5>Last Name:-"+rs.getString(2)+ "</h5>");
    out.println("<h5>Enrollment:-"+rs.getString(3)+ "</h5>");
    out.println("<h5>Contact No:-"+rs.getString(4)+ "</h5>");
    out.println("<h5>Sem:-"+rs.getString(5)+ "</h5>");
}

out.println("</body>");
out.println("</html>");
}catch(Exception e)
{
    System.out.println(e.toString());
}
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

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

@Override
public String getServletInfo() {
    return "Short description";
}

}
```


Output:-**Perform Following Operation**

REGISTRATION
SEARCH
UPDATE
DELETE

**➤ Insert Record****REGISTRATION FORM**

First Name:	Sanjeet
Last Name:	Nikam
Semester:	6
Contact No:	7202065123
Enrollment:	190843131006 x

Save



Database Table after Record Registered

Host: 127.0.0.1 Database: students Tables: student_18 Data Query

students.student_18: 3 rows total (approximately)

fname	lname	enroll	contact	sem
Rajeshree	Shimpi	190843131010	123456789	6
Rajjuu	Bhavsar	190843131001	7436001789	6
Sanjeet	Nkam	190843131006	7202065123	6

33 SELECT * FROM 'students'. 'student_18' LIMIT 1000;

➤ Search Record

http://localhost:8080/Servlet1/search.html

SEARCH RECORD

Enter Name You want to search a record :

After Searching Data in Database



Record Searched Value of name: Sanjeet

First Name:-Sanjeet

Last Name:-Nikam

Enrollment:-190843131006

Contact No:-7202065123

Sem:-6



➤ Update Record



UPDATE RECORD

Enter Enrollment You want to update a record :	190843131006
Enter Name of your Enrollment You want to update a record :	Sanjuuu ×

UPDATE



After Updating Record

HeidiSQL interface showing the database 'students' and table 'student_18'. The table contains 3 rows of data:

fname	lname	enroll	contact	sem
Rajeshree	Shimpi	190843131010	123456789	6
Rajjuu	Bhavsar	190843131001	7436001789	6
Sanjuu	Nkam	190843131006	7202065123	6

The SQL query window shows: `SELECT * FROM 'students'. 'student_18' LIMIT 1000;`

➤ Delete Record

Web browser showing the 'DELETE RECORD' form. The form has a text input field labeled 'Enter Name You want to delete record :' with the value 'Sanjuu' entered. Below the input field is a button labeled 'DELETE'.

The browser address bar shows: `http://localhost:8080/Servlet1/delete.html`

The browser address bar shows: `http://localhost:8080/Servlet1/Servlet3`

After Deleting record

The screenshot shows the HeidiSQL interface with the 'student_18' table selected. The table contains two records. The status bar at the bottom indicates the current SQL query is 'SELECT * FROM 'students'. 'student_18' LIMIT 1000;'. The system tray shows the connection is to MySQL 5.6.28 on host 127.0.0.1, with a server uptime of 22:01 h and server time of 10:33 AM on 2/19/2021.

fname	lname	enroll	contact	sem
Rajeshree	Shimpi	190843131010	123456789	6
Rajjuu	Bhavsar	190843131001	7436001789	6

Practical:-3.3

Aim:- Implement Authentication filter using filter API.

Index.html

```
<form action="AdminServlet">  
Name:<input type="text" name="name"/><br/>  
Password:<input type="password" name="password"/><br/>  
  
<input type="submit" value="login">  
  
</form>
```

MyFilter.java

```
import java.io.IOException;  
import java.io.PrintWriter;  
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 out=resp.getWriter();  
        String password=req.getParameter("password");  
        if(password.equals("admin")){  
            chain.doFilter(req, resp);//sends request to next resource  
        }  
        else{
```

```
        out.print("username or password error!");  
        RequestDispatcher rd=req.getRequestDispatcher("index.html");  
        rd.include(req, resp);  
    }  
}  
  
@Override  
    public void destroy() {}  
}
```

AdminServlet.java

```
import java.io.IOException;  
import java.io.PrintWriter;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.*;  
  
public class AdminServlet extends HttpServlet {  
    @Override  
    public void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html");  
        try (PrintWriter out = response.getWriter()) {  
            out.print("welcome ADMIN");  
        }  
    }  
}
```

Web.xml

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

<web-app >
  <servlet>
    <servlet-name>AdminServlet</servlet-name>
    <servlet-class>AdminServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>AdminServlet</servlet-name>
    <url-pattern>/AdminServlet</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <filter>
    <filter-name>f1</filter-name>
    <filter-class>MyFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>f1</filter-name>
    <url-pattern>/AdminServlet</url-pattern>
  </filter-mapping>
</web-app>
```


Output:-

➤ Login Page for Admin



➤ After Validating Admin.



➤ If not Admin



A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/v3.3/AdminServlet?name=Admin&password=123`. The page content displays the message "username or password error!". Below this message are two input fields: "Name:" and "Password:". A "login" button is positioned below the "Password:" field.



Practical:-3.4

Aim:- Apply XSLT (Style) to generated xml document and print your result.

Practical3_4.java

```
package com.javacodegeeks;
import org.w3c.dom.Document;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import javax.xml.transform.stream.StreamSource;
import java.io.File;

public class Practical3_4 {

    private static Document document;

    public static void main(String[] args) throws Exception {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

        File xml = new
File("C:\\Users\\Administrator\\Documents\\NetBeansProjects\\Practical3_4\\persons.xml");

        File xsl = new
File("C:\\Users\\Administrator\\Documents\\NetBeansProjects\\Practical3_4\\persons.xsl");

        DocumentBuilder builder = factory.newDocumentBuilder();
        document = builder.parse(xml);
```

```
// Use a Transformer for output
```

```
TransformerFactory transformerFactory = TransformerFactory.newInstance();
```

```
StreamSource style = new StreamSource(xsl);
```

```
Transformer transformer = transformerFactory.newTransformer(style);
```

```
DOMSource source = new DOMSource(document);
```

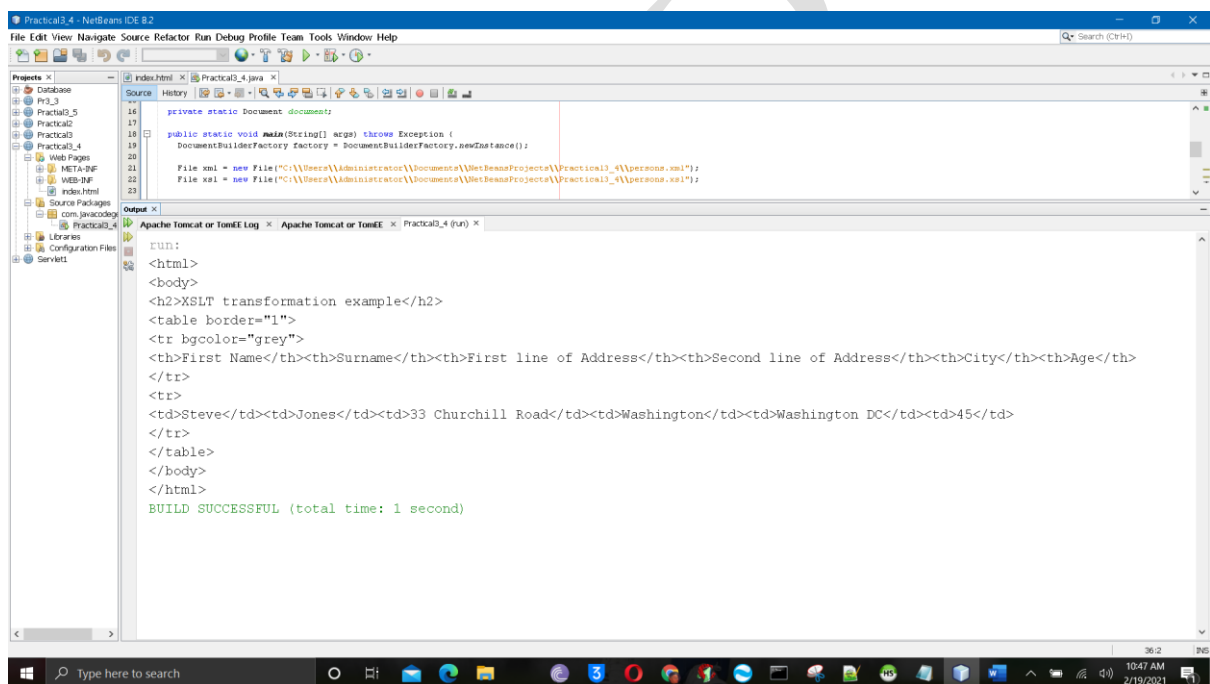
```
StreamResult result = new StreamResult(System.out);
```

```
transformer.transform(source, result);
```

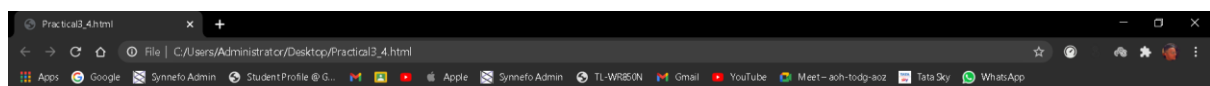
```
}
```

```
}
```

Output:-



➤ After Running above HTML code



XSLT transformation example

First Name	Surname	First line of Address	Second line of Address	City	Age
Steve	Jones	33 Churchill Road	Washington	Washington DC	45

Practical:-3.5

Aim:- Design a form to input details of an employee and submit the data to a servlet. Write code for servlet that will save the entered details as a new record in database table Employee with fields (EmpId, EName, Email, Age).

Index.html

```
<html>
  <head>
    <title>Student Details</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <center>
      <form action="Servlet2">
        <h1><b>REGISTRATION FORM</b></h1>
        <table border="">
          <tr><td>Employee ID:</td><td><input type="text" name="eid"
value=""></td></tr>
          <tr><td>Employee Name:</td><td><input type="text" name="ename"
value=""></td></tr>
          <tr><td>Email:</td><td><input type="text" name="email" value=""></td></tr>
          <tr><td>Age:</td><td><input type="number" name="age" value=""></td></tr>
        </table>
        <table border="">
          <br>
          <tr><td><input type="submit" name="Save" value="Save"></td></tr>
        </table>
      </form>
    </center>
```

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

Servlet2.java

```
import java.io.IOException;  
import java.io.PrintWriter;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.Statement;  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
public class Servlet2 extends HttpServlet {  
    Statement st=null;  
    Connection con=null;  
    static final String DB_URL="jdbc:mysql://localhost:3306/students";  
    static final String USER="root";  
    static final String PASS="";  
  
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html;charset=UTF-8");  
        try (PrintWriter out = response.getWriter()) {  
            /* TODO output your page here. You may use following sample code. */  
  
            String s1,s2,s3,s4,s5,sql;
```

```

s1=request.getParameter("eid");
s2=request.getParameter("ename");
s3=request.getParameter("email");
s4=request.getParameter("age");
out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<title>Servlet Registration</title>");
out.println("</head>");
out.println("<body>");
out.println("<h3>Employee Id: " + s1 + "</h3>");
out.println("<h3>Employee Name: " + s2 + "</h3>");
out.println("<h3>Email: " + s3 + "</h3>");
out.println("<h3>Age: " + s4 + "</h3>");
out.println("</body>");
out.println("</html>");

Class.forName("com.mysql.jdbc.Driver");
con = DriverManager.getConnection(DB_URL,USER,PASS);
st = con.createStatement();

sql=" insert into employee(e_id,e_name,email,age)
values('"+s1+"','"+s2+"','"+s3+"','"+s4+"')";
st.executeUpdate(sql);
System.out.println("Record Inserted Sucessfully...");

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

@Override

```

```

protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

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

@Override
public String getServletInfo() {
    return "Short description";
}
}

```

Output:-



REGISTRATION FORM

Employee ID:	<input type="text" value="5123"/>
Employee Name:	<input type="text" value="Sanjeet"/>
Email:	<input type="text" value="nicksanjeet23@gmail.com"/>
Age:	<input type="text" value="19"/> x



➤ Record you Registered



Employee Id: 5123

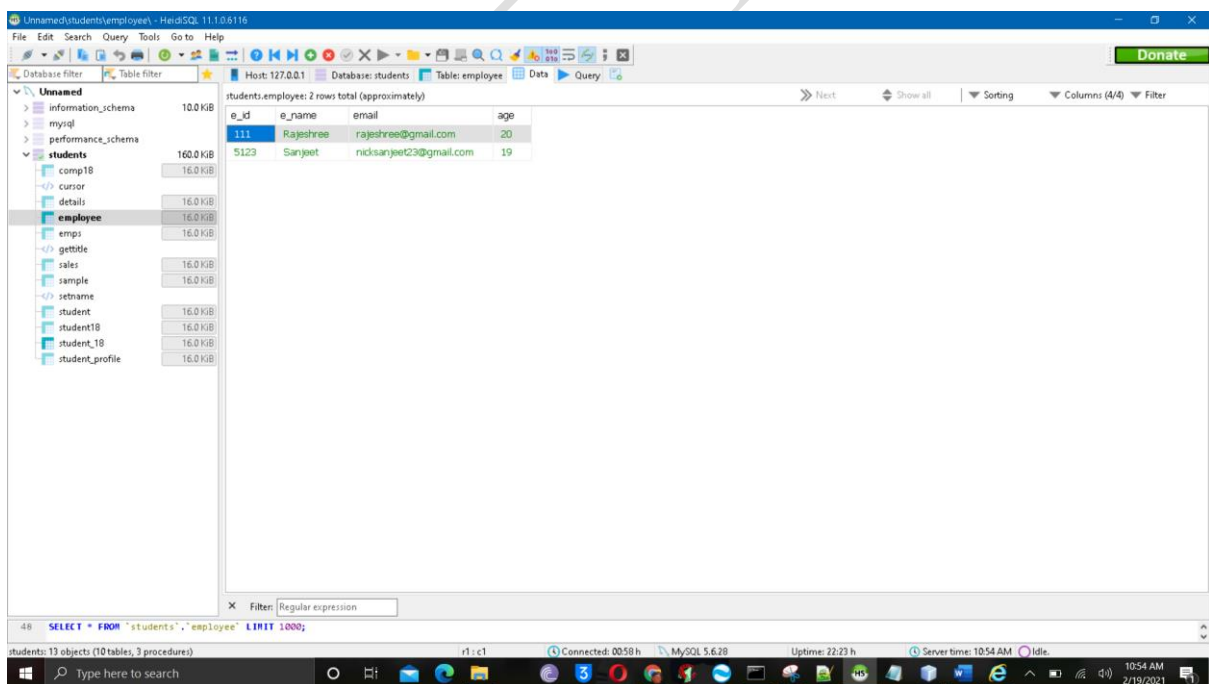
Employee Name: Sanjeet

Email: nicksanjeet23@gmail.com

Age: 19



➤ After Registering Employee Record is inserted into Database



Practical:-4

Aim:- Implement program using JSP

Practical:-4.1

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

Index.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Cookie using JSP</title>
</head>
<body>
<center>
    <h3>Enter First and Last Name to Store in Cookie</h3>
    <form action="newjsp.jsp" method="GET">
        First Name: <input type="text" name="fname"><br><br />
        Last Name: <input type="text" name="lname" /><br><br>
        <input type="submit" value="Submit" />
    </form>
</center>
</body>
</html>
```

Newjsp.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">
<%
    Cookie fname = new Cookie("fname",request.getParameter("fname"));
```

```
Cookie lname = new Cookie("lname",request.getParameter("lname"));
fname.setMaxAge(60*60*10);
lname.setMaxAge(60*60*10);
response.addCookie( fname );
response.addCookie( lname );

%>

<html>

<head><title>Cookie using JSP</title></head>

<body>

    <h3>Value of Cookie with JSP </h3>

    <b>First Name:</b><%= request.getParameter("fname")%><br>
    <b>Last Name:</b> <%= request.getParameter("lname")%>

</body>

</html>
```

Output

- **Entering values for Storing into the Cookies.**

Enter First and Last Name to Store in Cookie

First Name:

Last Name:

http://localhost:8080/Prac4_1/newjsp.jsp

Type here to search

10:34 PM 4/8/2021

- **After Submitting First and Last Name.**



Value of Cookie with JSP

First Name: ABC
Last Name: XYZ



Practical:-4.2

**Aim:- Implement the shopping cart for users for the online shopping.
Apply the concept of session.**

Index.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

  <head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

    <title>Shopping Cart - Login</title>

  </head>

  <body background="img/bg1.jpg"><center>

    <div class="container">

      <div class="headbanner">

        <h1>

          <center>

            [My Shopping Cart]

          </center>

        </h1>

      </div>

      <div class="mycontent">

        <div class="space"><span><a class="formtext">Login</a></span></div>

        <div class="formcontent">

          <form action="loginval" method="post">

            <table border="2px">

              <tr>

                <td class="formtext">Username :</td>

                <td><input id="name" name="uname" type="text" size="30"
/></td><td><a>[Any name]</a></td>

              </tr>
```

```

        <tr>

            <td class="formtext">Password :</td>

            <td><input id="pas" name="pass" type="password" size="30"
/></td><td><a>[Pass = 1234]</a></td>

        </tr>

        <tr>

            <td colspan="3"><center> <input type="submit"
value="Submit"/></td></center>

        </tr>

    </table>

</form>

</div>

</div>

</div>

</center>

</body>

</html>

```

Shop.jsp

```

<% @page import="java.util.ArrayList"%>

<% @ page import="classes.Item" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

    <head>

        <%

            String user = (String) session.getAttribute("user");

            if (user == null) {

                response.sendRedirect("index.jsp");

            }

        %>

        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

```

```

<title>Shopping Cart - Shop</title>
</head>
<body background="img/bg3.jpg"><center>
<div class="container">
  <form action="requesthandle" method="post">
    <div class="headbanner">
      <h1>
        <center>
          [My Shopping Cart]
        </center>
      </h1>
    </div>
    <div class="mycontent">
      <div class="cartof">
        <center><a>Cart Of [<% out.print(session.getAttribute("user"));%>]<input
name="logout" type="submit" value="Logout"></input></a></center>
      </div>
      <div class="cartcontent">
        <div class="myitems">
          <table width="600px" cellpadding="0" cellspacing="0" border="2px">
            <tr>
              <th>#id</th>
              <th>Item</th>
              <th>Price</th>
              <th>Action</th>
            </tr>
            <%if (session.getAttribute("itemlist") != null) {
              ArrayList mycart = (ArrayList) session.getAttribute("itemlist");
              for (int i = 0; i < mycart.size(); i++) {
                Item it = (Item) mycart.get(i);
                %>

```



```

        <tr>
            <td align="center"><% out.print(i);%></td>
            <td align="center"><% out.print(it.name);%></td>
            <td align="center"><% out.print(it.price);%></td>
            <td align="center"><input name="del" type="submit" value="Delete"
onclick="this.value=<% out.print(i);%>"></input></td>
        </tr>
    <% }
    }
    %>
</table>
</div>
<div class="total">
    <a>My Total : $[<% out.print(session.getAttribute("total"));%>]</a><br />
    <a>Total Qty: [<% ArrayList il = (ArrayList)
session.getAttribute("itemlist");
out.print(il.size());%>]</a><br />
    <input name="chkout" type="submit" value="Checkout" />
</div>
</div>
<div class="items">
    <table width="900px" border="2px">
        <tr class="border_bottom">
            <td>#1</td>
            <td>Sunglass</td>
            <td>Ray-Ban, Dark Purple Sunglass with the Casing</td>
            <td>$34</td>
            <td></td>
            <td><input name="addtocart1" type="submit" value="Add to Cart"
/></td>
        </tr>

```

```
<tr class="border_bottom">
  <td>#2</td>
  <td>Wrist Watch</td>
  <td>Quartz, Men's wrist watch, Black</td>
  <td>$66</td>
  <td></td>
  <td><input name="addtocart2" type="submit" value="Add to Cart"
/></td>
</tr>
<tr class="border_bottom">
  <td>#3</td>
  <td>Camera</td>
  <td>Lumix, 16x Digital Camera</td>
  <td>$167</td>
  <td></td>
  <td><input name="addtocart3" type="submit" value="Add to Cart"
/></td>
</tr>
<tr class="border_bottom">
  <td>#4</td>
  <td>Shoes</td>
  <td>Bettans, 60 Leather Shoes, Brown</td>
  <td>$23</td>
  <td></td>
  <td><input name="addtocart4" type="submit" value="Add to Cart"
/></td>
</tr>
</table>
</div>
</div>
</form>
```

```
</div>
</center> </body>
</html>
```

Error.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Shopping Cart - Login</title>
</head>
  <body background="img/bg1.jpg">
    <center>
      <form action="index.jsp" method="post">
<div class="container">
  <div class="headbanner">
    <h1><center>
      [My Shopping Cart]
</center></h1>
    </div>
    <div class="mycontent">
      <h3 align="center">Oops! Error<br />Your password is incorrect, Try
Again!<br /><input type="submit" value="Back" /></h3>
    </div>
  </div>
</form>
</center>
  </body>
</html>
```

Checkout.jsp

```

<% @page import="java.util.ArrayList"%>
<% @ page import="classes.Item" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Shopping Cart - Check out</title>
  </head>
  <body background="img/bg1.jpg"><center>
    <form action="purchase" method="post">
      <%
        ArrayList it_list = (ArrayList) session.getAttribute("itemlist");
      %>
      <div class="container">
        <div class="headbanner">
          <h1><center>
            [My Shopping Cart]
          </center></h1>
        </div>
        <div class="mycontent">
          <a>Checkout My Cart</a><br />
          <table width="500px" border="2px">
            <%
              for (int i = 0; i < it_list.size(); i++) {
                classes.Item itm = (Item) it_list.get(i);
              %>
            <tr>
              <td><%out.print(itm.name);%></td>
              <td><%out.print(itm.price);%></td>

```

```

        </tr>
        <% } %>
    <tr>
        <td>My
Total</td><td>${<% out.print(session.getAttribute("total")); %>}</td>
    </tr>
    <tr>
        <td><input type="submit" value="Purchase" /></td>
    </tr>
    <tr>
        <td></td></tr>
    </table>
</div>
</div>
</form>
</center>
</body>
</html>

```

Success.jsp

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>Shopping Cart - Success</title>
    </head>
    <body background="img/bg1.jpg"><center>
        <% if(session.getAttribute("purch")!="true"){response.sendRedirect("index.jsp");
        } %>
        <form action="shop.jsp" method="post">
            <div class="container">

```

```

<div class="headbanner">
    <h1><center>
        [My Shopping Cart]
    </center></h1>
</div>
<div class="mycontent">
    <h3 align="center">Purchase has been succeeded! Thank You.<br /><input
type="submit" value="Ok" /></h3>
</div>
</div>
</form>
</center></body>
</html>

```

RequestHandle.java

```

import classes.Item;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class requesthandle extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        response.setContentType("text/html;charset=UTF-8");
        HttpSession mysession = request.getSession();

```

```
ArrayList mycart = (ArrayList) mysession.getAttribute("itemlist");
int value = (Integer) mysession.getAttribute("total");
String i1 = request.getParameter("addtocart1");
String i2 = request.getParameter("addtocart2");
String i3 = request.getParameter("addtocart3");
String i4 = request.getParameter("addtocart4");
String chk = request.getParameter("chkout");
String logout = request.getParameter("logout");
String pressdel = request.getParameter("del");
if (i1 != null) {
    Item myitem = new Item("#1", "Sunglass", 34);
    value = value + 34;
    mycart.add(myitem);
    mysession.setAttribute("itemlist", mycart);
    mysession.setAttribute("total", value);
    response.sendRedirect("shop.jsp");
} else if (i2 != null) {
    Item myitem = new Item("#2", "Wrist Watch", 66);
    value = value + 66;
    mycart.add(myitem);
    mysession.setAttribute("itemlist", mycart);
    mysession.setAttribute("total", value);
    response.sendRedirect("shop.jsp");
} else if (i3 != null) {
    Item myitem = new Item("#3", "Camera", 167);
    value = value + 167;
    mycart.add(myitem);
    mysession.setAttribute("itemlist", mycart);
    mysession.setAttribute("total", value);
    response.sendRedirect("shop.jsp");
}
```

```

    } else if (i4 != null) {
        Item myitem = new Item("#4", "Shoes", 23);
        value = value + 23;
        mycart.add(myitem);
        mysession.setAttribute("itemlist", mycart);
        mysession.setAttribute("total", value);
        response.sendRedirect("shop.jsp");
    } else if (chk != null) {
        mysession.setAttribute("chk", chk);
        response.sendRedirect("checkout.jsp");
    } else if (logout != null) {
        mysession.invalidate();
        response.sendRedirect("index.jsp");
    } else if (pressdel != null) {
        Item item_to_Delete = (Item) mycart.get(Integer.parseInt(pressdel));
        value = value - item_to_Delete.price;
        mysession.setAttribute("total", value);
        mycart.remove(Integer.parseInt(pressdel));
        mysession.setAttribute("tod", pressdel);
        response.sendRedirect("shop.jsp");
    }
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

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

```



```

        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    } // </editor-fold>
}

```

Loginval.java

```

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import javax.jms.Session;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import javax.xml.ws.Dispatch;

public class loginval extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String username = (String) request.getParameter("uname");
        String password = (String) request.getParameter("pass");
        if (password.equals("1234")) {
            ArrayList cart = new ArrayList();

            int totalcost = 0;

            HttpSession mysession = request.getSession();
            mysession.setAttribute("user", username);
            mysession.setAttribute("itemlist", cart);

```

```

        mysession.setAttribute("total", totalcost);
        response.sendRedirect("shop.jsp");
    }else{
        response.sendRedirect("error.jsp");
    }
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

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

@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

Addtocart.java

```

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

```

```

import javax.servlet.http.HttpSession;

public class addtocart extends HttpServlet {

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

    }

    @Override

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

        processRequest(request, response);

    }

    @Override

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

        processRequest(request, response);

    }

    @Override

    public String getServletInfo() {

        return "Short description";

    } // </editor-fold>
}

```

Purchase.java

```

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class purchase extends HttpServlet {

```

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    ArrayList newlist = new ArrayList();
    int newval = 0;
    HttpSession mysession = request.getSession();
    mysession.setAttribute("purch", "true");
    mysession.setAttribute("itemlist", newlist);
    mysession.setAttribute("total", newval);
    response.sendRedirect("success.jsp");
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

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

@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

Item.java

```

package classes;

public class Item {

```

```
public String id;  
public String name;  
public int price;  
public Item(String a, String b, int c) {  
    this.id = a;  
    this.name = b;  
    this.price = c;  
}  
}
```

Shopping.png



Camera.jpg



Shoes.jpg



Watch.jpg



Sunglass.jpgPaywith.png

Output

- Login Page



[My Shopping Cart]

Login

Username :	<input type="text" value="User1"/>	[Any name]
Password :	<input type="password" value="...."/>	[Pass = 1234]
<input type="button" value="Submit"/>		

- After entering Wrong Password



[My Shopping Cart]

Oops! Error
Your password is incorrect, Try Again!

- Shopping Page for User1 where User1 can Add the Product to Cart as well as Delete the Product from the Cart and Also Checkout with Selected Product and Can also Logout.



[My Shopping Cart]

Cart Of [User1]

#id	Item	Price	Action
0	Sunglass	34	<input type="button" value="Delete"/>
1	Wrist Watch	66	<input type="button" value="Delete"/>
2	Camera	167	<input type="button" value="Delete"/>
3	Shoes	23	<input type="button" value="Delete"/>

My Total : \$[290]
 Total Qty: [4]


#1	Sunglass	Ray-Ban, Dark Purple Sunglass with the Casing	\$34		<input type="button" value="Add to Cart"/>
#2	Wrist Watch	Quartz, Men's wrist watch, Black	\$66		<input type="button" value="Add to Cart"/>
#3	Camera	Lumix, 16x Digital Camera	\$167		<input type="button" value="Add to Cart"/>
#4	Shoes	Bettans, 60 Leather Shoes, Brown	\$23		<input type="button" value="Add to Cart"/>

- Checkout Page after Adding Product in to the Cart.

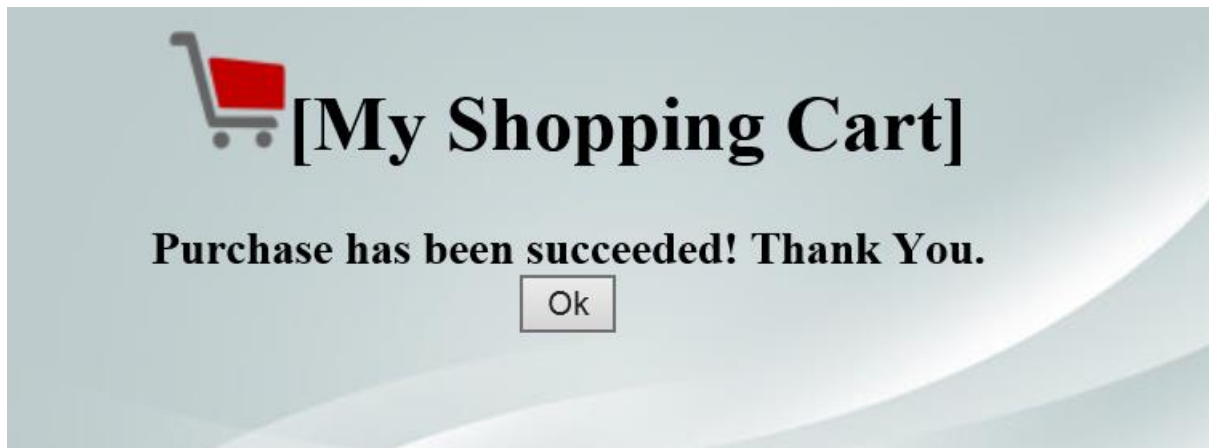


[My Shopping Cart]

Checkout My Cart

Sunglass	34
Wrist Watch	66
Camera	167
Shoes	23
My Total	\$[290]
<input type="button" value="Purchase"/>	
    	

- After checking out, Success Message is Displayed



Practical:-4.3

Aim:- Write a web application which takes id, name, mobile no, semester, marks, percentage pass to servlet. Servlet forward to model class having method `getid()`, `getname()`, `getmobno()`, `getsem()`, `getmarks()` and `getPercentage()`. Display all the information in .jsp page.

Index.html

```
<html>
  <head>
    <title>Student Details</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <center>
      <form action="demo">
        <h1><b>REGISTRATION FORM</b></h1>
        <table border="">
          <tr><td>ID:</td><td><input type="text" name="id" value=""></td></tr>
          <tr><td>Name:</td><td><input type="text" name="name" value=""></td></tr>
          <tr><td>Mobile No:</td><td><input type="text" name="mobile" value=""></td></tr>
          <tr><td>Semester:</td><td><input type="text" name="sem" value=""></td></tr>
          <tr><td>Marks:</td><td><input type="text" name="mark" value=""></td></tr>
          <tr><td>Percentage:</td><td><input type="text" name="per" value=""></td></tr>
        </table><br>
        <tr><td><input type="submit" name="Save" value="Save"></td></tr>
      </form>
    </center>
  </body>
```

</html>

Demo.java

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

public class demo extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String destination = "newjsp.jsp";

            String s1,s2,s3,s4,s5,s6;
            s1=request.getParameter("id");
            s2=request.getParameter("name");
            s3=request.getParameter("mobile");
            s4=request.getParameter("sem");
            s5=request.getParameter("mark");
            s6=request.getParameter("per");

            Person p=new Person();
            p.setID(s1);
            p.setNAME(s2);
            p.setMOBILE(s3);
            p.setSEM(s4);
            p.setMARK(s5);
```

```

        p.setPER(s6);

        request.setAttribute("person", p);
        RequestDispatcher rd = request.getRequestDispatcher(destination);
        rd.forward(request,response);
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet demo</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet demo at " + request.getContextPath() + "</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

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

@Override
public String getServletInfo() {
    return "Short description";
}

```

```
}
```

Person.java

```
public class Person {
```

```
    public String id,name,mobile,sem,mark,per;
```

```
    public void Person()
```

```
    {
```

```
    }
```

```
    public void setID(String id)
```

```
    {
```

```
        this.id=id;
```

```
    }
```

```
    public String getID()
```

```
    {
```

```
        return id;
```

```
    }
```

```
    public void setName(String name)
```

```
    {
```

```
        this.name=name;
```

```
    }
```

```
    public String getName()
```

```
    {
```

```
        return name;
```

```
    }
```

```
    public void setMOBILE(String mobile)
```

```
    {
```

```
        this.mobile=mobile;
```

```
    }
```

```
public String getMOBILE()
{
    return mobile;
}
public void setSEM(String sem)
{
    this.sem=sem;
}
public String getSEM()
{
    return sem;
}
public void setMARK(String mark)
{
    this.mark=mark;
}
public String getMARK()
{
    return mark;
}
public void setPER(String per)
{
    this.per=per;
}
public String getPER()
{
    return per;
}
}
```

Newjsp.jsp

```
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h3>Details are:-</h3>
    <table border='5'>
      <tr><td>ID is:</td><td>${person.ID}</td></tr>
      <tr><td>Name is:</td><td>${person.NAME}</td></tr>
      <tr><td>Mobile No is:</td><td>${person.MOBILE}</td></tr>
      <tr><td>Semester is:</td><td>${person.SEM}</td></tr>
      <tr><td>Mark is:</td><td>${person.MARK}</td></tr>
      <tr><td>Percentage is:</td><td>${person.PER}</td></tr>
    </table>
  </body>
</html>
```

Output



REGISTRATION FORM

ID:	123
Name:	ABC
Mobile No:	1234567890
Semester:	6
Marks:	99
Percentage:	99

Save

- After Submitting Details are:-



Details are:-

ID is:	123
Name is:	ABC
Mobile No is:	1234567890
Semester is:	6
Mark is:	99
Percentage is:	99

Practical:-4.4

Aim:- Write a student bean class with property student_id, name, semester, address and percentage. Write jsp page to set and display all property.

Index.html

```
<html>
  <head><title>Student Registration using Beans</title></head>
  <body>
    <h3>Enter Student's Details </h3>
    <form action="print.jsp">

Student ID: <input type="text" name="sid"/><br><br>
Student Name: <input type="text" name="sname"/><br><br>
Student Semester: <input type="text" name="sem"/><br><br>
Student Address: <input type="text" name="add"/><br><br>
Student Percentage: <input type="text" name="per"/><br><br>
    <input type="submit" value="Submit"/>

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

Print.jsp

```
<jsp:useBean id="st" class="beans.student">
<jsp:setProperty name="st" property="*" />
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```



```

<title>JSP Page</title>

</head>

<body>

    <h2>Details using Bean are:-</h2>

    <table border='5'>

        <tr><td>ID is:</td><td><jsp:getProperty property="sid" name="st"/></td></tr>

        <tr><td>NAmE is:</td><td><jsp:getProperty property="sname"
name="st"/></td></tr>

        <tr><td>Semester is:</td><td><jsp:getProperty property="sem"
name="st"/></td></tr>

        <tr><td>Address is:</td><td><jsp:getProperty property="add"
name="st"/></td></tr>

        <tr><td>Percentage is:</td><td><jsp:getProperty property="per"
name="st"/></td></tr>

    </table>

</body>

</html>

</jsp:useBean>

```

Student.java

```

package beans;

public class student
{
    public String sid,sname,sem,add,per;

    public void setSid(String sid)
    {
        this.sid=sid;
    }

    public String getSid()
    {
        return sid;
    }
}

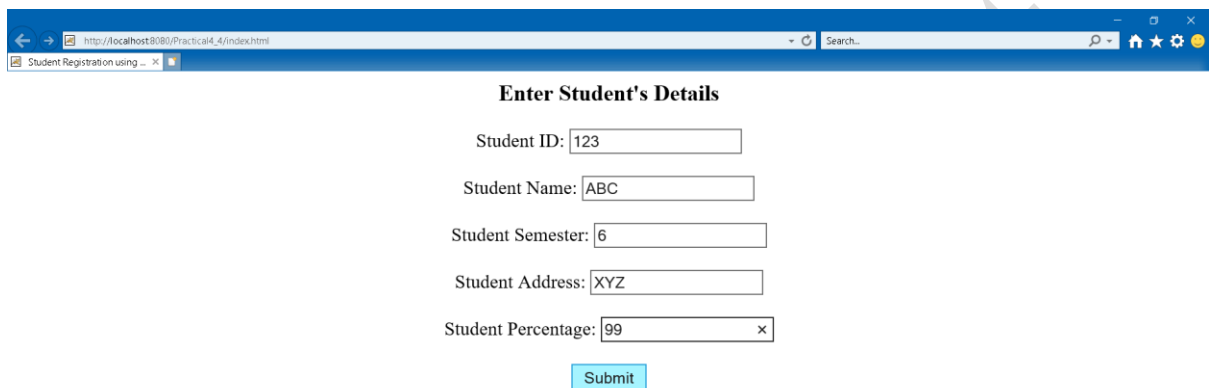
```

```
}  
  
public void setName(String sname)  
{  
    this.sname=sname;  
}  
  
public String getSname()  
{  
    return sname;  
}  
  
public void setSem(String sem)  
{  
    this.sem=sem;  
}  
  
public String getSem()  
{  
    return sem;  
}  
  
    public void setAdd(String add)  
    {  
        this.add=add;  
    }  
  
    public String getAdd()  
    {  
        return add;  
    }  
  
    public void setPer(String per)  
    {  
        this.per=per;  
    }  
  
    public String getPer()
```

```
{  
    return per;  
}  
}
```

Output

- **Entering values**

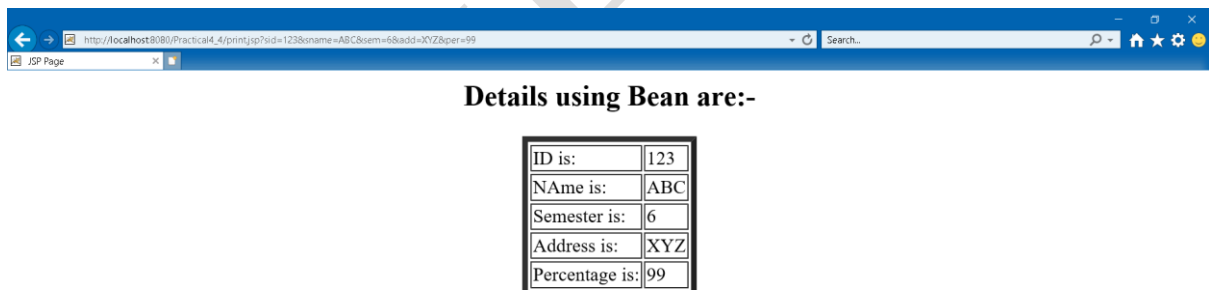


The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/Practical4_4/index.html`. The page title is "Student Registration using ...". The form is titled "Enter Student's Details" and contains the following fields:

- Student ID:
- Student Name:
- Student Semester:
- Student Address:
- Student Percentage:

A blue "Submit" button is located below the fields.

- **After inserting Values.**



The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/Practical4_4/print.jsp?id=123&name=ABC&sem=6&add=XYZ&per=99`. The page title is "JSP Page". The content is titled "Details using Bean are:-" and displays the following details in a table:

ID is:	123
NAme is:	ABC
Semester is:	6
Address is:	XYZ
Percentage is:	99

Practical:-4.5

Aim:- Write a java bean named “student” having roll no and name having getter & setter methods. Write a JSP page to set the roll number and name for a student object and then print them by reading from object.

Index.html

```
<html>
  <head><title>Student Registration using Beans</title></head>
  <body><center>
    <h3>Click Submit to See Details </h3>
    <form action="print.jsp">
      <input type="submit" value="Submit"/>
    </form>
  </center></body>
</html>
```

Print.jsp

```
<jsp:useBean id="st" class="beans.Student">
<jsp:setProperty name="st" property="*" />
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body><center>
    <jsp:setProperty name = "st" property = "sid" value = "190843131006"/>
    <jsp:setProperty name = "st" property = "sname" value = "Sanjeet"/>
```

```
<h2>Details using Bean are:-</h2>

<table border='5'>

    <tr><td>ID is:</td><td><jsp:getProperty property="sid" name="st"/></td></tr>

    <tr><td>Name is:</td><td><jsp:getProperty property="sname"
name="st"/></td></tr>

</table>

</center></body>

</html>

</jsp:useBean>
```

Student.java

```
package beans;

public class Student
{
    public String sid,sname;
    public void setSid(String sid)
    {
        this.sid=sid;
    }
    public String getSid()
    {
        return sid;
    }
    public void setName(String sname)
    {
        this.sname=sname;
    }
    public String getSname()
    {
        return sname;
    }
}
```

```
}  
}
```

Output



Click Submit to See Details

Submit

- After Submitting Details are :-



Details using Bean are:-

ID is:	190843131006
Name is:	Sanjeet

Practical:-4.6

Aim:- Write a JSP program using JSTL SQL taglib to display student details in tabular form by iterating through the database table student.

Index.html

```
<html>
  <head>
    <title>Student Details</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <center>
      <form action="newjsp.jsp">
        <h2><b>Click Submit Button to View Table Information using JSTL Tag</b></h2>
        <input type="submit" name="Save" value="Save">
      </form>
    </center>
  </body>
</html>
```

Newjsp.jsp

```
<% @ page import="java.io.*,java.util.*,java.sql.*"%>
<% @ page import="javax.servlet.http.*,javax.servlet.*" %>
<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<% @ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<html>
<head>
<title>sql:query Tag</title>
</head>
```

```

<body>
<sql:setDataSource var="db" driver="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/STUDENTS"
    user="root" password="" />
<sql:query dataSource="${db}" var="rs">
SELECT * from details;
</sql:query>
<table border="2" width="100%">
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Enrollment No</th>
<th>Contact No</th>
</tr>
<c:forEach var="table" items="${rs.rows}">
<tr>
<td><c:out value="${table.First_name}"/></td>
<td><c:out value="${table.Last_name}"/></td>
<td><c:out value="${table.Enrollment}"/></td>
<td><c:out value="${table.Contact_no}"/></td>
</tr>
</c:forEach>
</table>
</body>
</html>

```

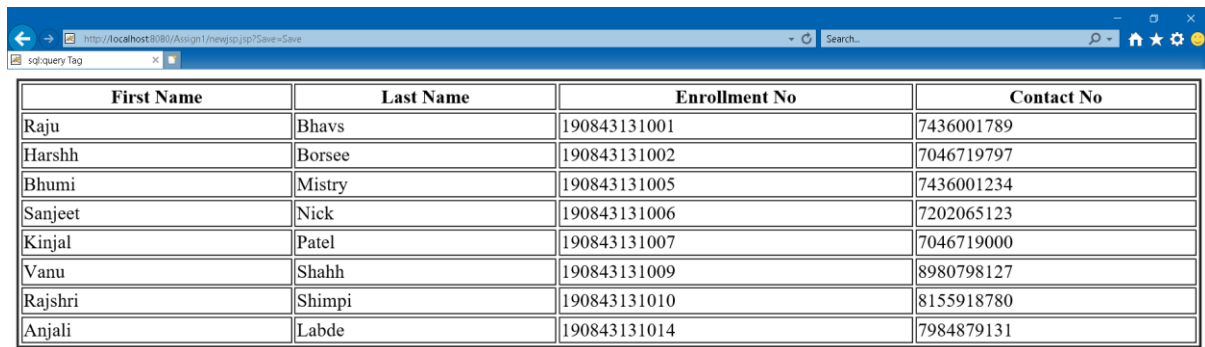
Output



Click Submit Button to View Table Information using JSTL Tag

Save

- Values of Table using JSTL Tag are:-



First Name	Last Name	Enrollment No	Contact No
Raju	Bhavs	190843131001	7436001789
Harshh	Borsee	190843131002	7046719797
Bhumi	Mistry	190843131005	7436001234
Sanjeet	Nick	190843131006	7202065123
Kinjal	Patel	190843131007	7046719000
Vanu	Shahh	190843131009	8980798127
Rajshri	Shimpi	190843131010	8155918780
Anjali	Labde	190843131014	7984879131

Practical:-5

Aim:- Implement using JSF(JavaServer Faces).

Practical:-5.1

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

index.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:f="http://xmlns.jcp.org/jsf/core">

    <h:head>

        <title>Facelet Title</title>

    </h:head>

    <h:body>

        <h:form>

            <h:outputLabel for="username">User Name</h:outputLabel>

            <h:inputText id="username" value="#{user.name}" required="true"
requiredMessage="Enter Username First...!!" >

                <f:validateRequired/>

                <f:validateLength minimum="5" maximum="20" />

                <f:validateRegex pattern="^[a-zA-Z]+(.)?[\s]*$" />

            </h:inputText><br></br>

            <h:commandButton id="submit-button" value="Submit" action="response.xhtml"/>

        </h:form>

    </h:body>

</html>
```

response.xhtml

```

<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

    xmlns:h="http://xmlns.jcp.org/jsf/html">

    <h:head>

        <title>Welcome Page</title>

    </h:head>

    <h:body>

        <h2>Hello, <h:outputText value="#{user.name}"></h:outputText></h2>

    </h:body>

</html>

```

User.java

```

import com.sun.istack.internal.NotNull;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
@ManagedBean
@RequestScoped
public class User {
    String name;
    public String getName()
    {
        return name;
    }
    public void setName(String value)
    {
        this.name=value;
    }
}

```

Output:-



Enter Username

User Name:

Submit

- Enter Username First...!!



Enter Username

User Name:

Submit

- j_idt5:username: Validation Error: Length is less than allowable minimum of '5'



Enter Username

User Name:

Submit



Hello, Sanjeet

Practical-6.1

Aim:- Write program to get all students data from Database using Hibernate. Write necessary xml files.

StoreData.java:-

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import java.util.*;
import org.hibernate.Query;

public class StoreData {
    public static void main(String[] args) {
        //creating configuration object
        Configuration cfg=new Configuration();
        cfg.configure("hibernate.cfg.xml");//populates the data of the configuration file
        //creating session factory object
        SessionFactory factory=cfg.buildSessionFactory();
        //creating session object
        Session session=factory.openSession();
        //creating transaction object
        String hql = "FROM Employee";
        Query query = session.createQuery(hql);
        List results = query.list();
        //Employee e1=(Employee)results.get(0);
        Iterator it = results.iterator();
        System.out.println("id\tfirstname\tlastname");
        System.out.println("=====");
        while(it.hasNext()){
            Employee e1=(Employee) it.next();
            System.out.print(e1.getId()+"\t");
```

```
System.out.print(e1.getFirstname()+"\t");
System.out.print("\t"+e1.getLastname());
System.out.println("");
    }
    Transaction t=session.beginTransaction();
    session.close();
}
}
```

Employee.java

```
public class Employee {
    private int id;
    private String firstname;
    private String lastname;
    int getId(){
        return id;
    }
    String getFirstname(){
        return firstname;
    }
    String getLastname(){
        return lastname;
    }
    void setId(int id){
        this.id=id;
    }
    void setFirstname(String firstname){
        this.firstname=firstname;
    }
    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 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
  <class name="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
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:3308/employee?zeroDateTimeBeha
vior=convertToNull</property>
    <property name="hibernate.connection.username">root</property>
    <mapping resource="employee.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```


Output:-

```
C:\Program Files\Java\jdk1.8.0_241\bin\janak\sem6\hib>javac StoreData.java

C:\Program Files\Java\jdk1.8.0_241\bin\janak\sem6\hib>java StoreData
log4j:WARN No appenders could be found for logger (org.hibernate.cfg.Environment).
log4j:WARN Please initialize the log4j system properly.
id      firstname      lastname
=====
1       emp1             emp
0       sanjeet          nikam
2       raj              bhavasr

C:\Program Files\Java\jdk1.8.0_241\bin\janak\sem6\hib>
```

Practical-6.2

Aim:- Write Hibernate application to store customer records and retrieve the customer record including name, contactnumber, address.

Customer.java

```
public class Customer {  
    private int id;  
    private String firstName;  
    private String lastName;  
    private String c_number;  
    private String address;  
    public Customer() { }  
    public Customer(String fname, String lname, String c_number,String address) {  
        this.firstName = fname;  
        this.lastName = lname;  
        this.c_number = c_number;  
        this.address = address;  
    }  
    public int getId() {  
        return id;  
    }  
    public void setId( int id ) {  
        this.id = id;  
    }  
    public String getFirstName() {  
        return firstName;  
    }  
    public void setFirstName( String first_name ) {  
        this.firstName = first_name;  
    }  
}
```

```

public String getLastName() {
    return lastName;
}

public void setLastName( String last_name ) {
    this.lastName = last_name;
}

public String getc_number() {
    return c_number;
}

public void setc_number( String c_number ) {
    this.c_number = c_number;
}

public String getAddress() {
    return address;
}

public void setAddress( String address ) {
    this.address = address;
}
}

```

Customers.hbm.xml

```

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

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

    <class name = "Customer" table = "customer">
        <meta attribute = "class-description">
            This class contains the students detail.
        </meta>
        <id name = "id" type = "int" column = "id">
            <generator class="native"/>
        </id>
        <property name = "firstName" column = "first_name" type = "string"/>
    </class>

```

```

    <property name = "lastName" column = "last_name" type = "string"/>
    <property name = "c_number" column = "c_number" type = "string"/>
    <property name = "address" column = "address" type = "string"/>
</class>
</hibernate-mapping>

```

ManageCustomer.java

```

import java.util.List;
import java.util.Date;
import java.util.Iterator;
import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class ManageCustomer {
    private static SessionFactory factory;
    public static void main(String[] args) {
        try {
            factory = new Configuration().configure().buildSessionFactory();
        } catch (HibernateException ex) {
            System.err.println("Failed to create sessionFactory object." + ex);
            throw new ExceptionInInitializerError(ex);
        }
        ManageCustomer ME = new ManageCustomer();
        /* Add few Customer records in database */
        Integer stdID1 = ME.addCustomer("Arjav","Desai","4564787098","surat");
        Integer stdID2 = ME.addCustomer("Tulsi", "Patel", "4564787098","navsari");
        Integer stdID3 = ME.addCustomer("Parva", "Gurav", "4564787098","bardoli");
        Integer stdID4 = ME.addCustomer("Darshit", "Desai", "4564787098","navsari");
        Integer stdID5 = ME.addCustomer("Tejas", "Patel", "4564787098","surat");
        /* List down all the Customer */
        ME.listCustomer();
    }
}

```

```

    /* Update Customer's records */
//    ME.updateCustomer(stdID1, 105);

    /* Delete an Customer from the database */
    ME.deleteCustomer(stdID2);

    /* List down new list of the Customer */
    ME.listCustomer(); }

/* Method to CREATE an Customer in the database */
public Integer addCustomer(String fname, String lname, String c_number,String address){
    Session session = factory.openSession();
    Transaction tx = null;
    Integer CustomerID = null;
    try {
        tx = session.beginTransaction();
        Customer Customer = new Customer(fname, lname, c_number, address);
        CustomerID = (Integer) session.save(Customer);
        tx.commit();
    } catch (HibernateException e) {
        if (tx!=null) tx.rollback();
        e.printStackTrace();
    } finally {
        session.close();
    }
    return CustomerID;
}

/* Method to READ all the Customer */
public void listCustomer( ){
    Session session = factory.openSession();
    Transaction tx = null;
    try {
        tx = session.beginTransaction();
        List Customer = session.createQuery("FROM Customer").list();
        for (Iterator iterator = Customer.iterator(); iterator.hasNext();){

```

```
Customer Customer1 = (Customer) iterator.next();
System.out.print("First Name: " + Customer1.getFirstName());
System.out.print(" Last Name: " + Customer1.getLastName());
System.out.println(" Contact Number: " + Customer1.getc_number());
System.out.println(" Address:- " + Customer1.getAddress());
}
tx.commit();
} catch (HibernateException e) {
    if (tx!=null) tx.rollback();
    e.printStackTrace();
} finally {
    session.close();
}
}

public void deleteCustomer(Integer CustomerID){
    Session session = factory.openSession();
    Transaction tx = null;
    try {
        tx = session.beginTransaction();
        Customer Customer = (Customer)session.get(Customer.class, CustomerID);
        session.delete(Customer);
        tx.commit();
    } catch (HibernateException e) {
        if (tx!=null) tx.rollback();
        e.printStackTrace();
    } finally {
        session.close();
    }
}
}
```

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/customers</property>

        <property name="hibernate.connection.username">root</property>

        <property name="hibernate.connection.password">india</property>

        <mapping resource="Customers.hbm.xml"/>

    </session-factory>

</hibernate-configuration>
```

Output:-

```
Output x
Java DB Database Process x Practical6_2 (run) x
INFO: HH0000400: Using dialect: org.hibernate.dialect.MySQLDialect
May 09, 2011 8:34:01 PM org.hibernate.engine.transaction.internal.TransactionFactoryInitiator initiateService
INFO: HH0000399: Using default transaction strategy (direct JDBC transactions)
May 09, 2011 8:34:01 PM org.hibernate.hql.internal.ast.ASTQueryTranslatorFactory <init>
INFO: HH0000397: Using ASTQueryTranslatorFactory
First Name: Arjav Last Name: Desai Contact Number: 4564787098
Address:- surat
First Name: Tulsi Last Name: Patel Contact Number: 4564787098
Address:- navsari
First Name: Parva Last Name: Gurav Contact Number: 4564787098
Address:- bardoli
First Name: Darshit Last Name: Desai Contact Number: 4564787098
Address:- navsari
First Name: Tejas Last Name: Patel Contact Number: 4564787098
Address:- surat
First Name: Arjav Last Name: Desai Contact Number: 4564787098
Address:- surat
First Name: Parva Last Name: Gurav Contact Number: 4564787098
Address:- bardoli
First Name: Darshit Last Name: Desai Contact Number: 4564787098
Address:- navsari
First Name: Tejas Last Name: Patel Contact Number: 4564787098
Address:- surat
```

Result Grid					
	id	first_name	last_name	c_number	address
	16	Arjav	Desai	4564787...	surat
	18	Parva	Gurav	4564787...	bardoli
	19	Darshit	Desai	4564787...	navsari
	20	Tejas	Patel	4564787...	surat
	NULL	NULL	NULL	NULL	NULL

Practical-7

Aim:- Write an Application to keep Record and Retrieve Record of Student. The Record includes Student id, Enrollment no, Semester, SPI. Use MVC architecture.

Practical-7

Aim:- Write an Application to keep Record and Retrieve Record of Student. The Record includes Student id, Enrollment no, Semester, SPI. Use MVC architecture.

Index.html:-

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:f="http://java.sun.com/jsf/core">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Index</title>
</head>
<h:body>
<f:view>
<h3>Enter Details</h3>
<h:form>
<table> <tr>
<td>ID:</td><td><h:inputText value="#{bean.id}"/></td>
</tr>
<tr>
<td>Enrollment Number:</td><td><h:inputText value="#{bean.en}"/></td>
</tr>
<tr>
<td>Semester:</td><td><h:inputText value="#{bean.sem}"/></td>
</tr>
<tr>
<td>SPI:</td><td><h:inputText value="#{bean.spi}"/></td>
</tr> </table>
<h:commandButton value="Submit" action="#{bean.submit}"/>
</h:form>
<br></br>
<h:outputText value="#{bean.data}" escape="false" />
</f:view>
</h:body>
</html>
```

Bean.java:-

```

package jsfpackage;
import java.io.Serializable;
import javax.faces.bean.ManagedBean;
import java.sql.*;
@ManagedBean
public class bean implements Serializable {
private static final long serialVersionUID = 6529685098267757690L;
private int id;

private String en;
private int sem;
private float spi;
private String data = "";
public String submit() throws SQLException, ClassNotFoundException{
Class.forName("com.mysql.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/test?characterEncoding=utf8&u
seSSL=false&useUnicode=true","root","root");
PreparedStatement st = con.prepareStatement("insert into student values(?,?,?,?)");
st.setInt(1, getId());
st.setString(2, getEn());
st.setInt(3, getSem());
st.setFloat(4, getSpi());
st.execute();
Statement stt = con.createStatement();
ResultSet rs = stt.executeQuery("select * from student");
this.data = "<table style=\"width:50%\" bgcolor=\"cyan\" border=\"1px\"><tr
bgcolor=\"teal\"><th>Id</th><th>Enrollment</th><th>Semester</th><th>SPI</th></tr>";
while(rs.next()) {
int i = rs.getInt(1);
String en = rs.getString(2);
int sm = rs.getInt(3);
float sp = rs.getFloat(4);

this.data += "<tr><td>"+i+"</td><td>"+en+"</td><td>"+sm+"</td><td>"+sp+"</td></tr>";
}
this.data += "</table>";
con.close();
return "index.xhtml";
}
public String getData() {

```

```

return this.data;
}
public int getId() {
return id;
}
public void setId(int id) {
this.id = id;
}
public String getEn() {
return en;
}
public void setEn(String en) {
this.en = en;
}
public int getSem() {
return sem;
}
public void setSem(int sem) {
this.sem = sem;
}
}
public float getSpi() {
return spi;
}
public void setSpi(float spi) {
this.spi = spi;
}
}

```

Web.xml:-

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
;http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">
<display-name>P11 JSF</display-name>
<welcome-file-list>
<welcome-file>index.xhtml</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>Faces Servlet</servlet-name>
<servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
<servlet-name>Faces Servlet</servlet-name>
<url-pattern>/faces/*</url-pattern>
</servlet-mapping>
<context-param>
<description>State saving method: 'client' or 'server' (=default). See JSF Specification
2.5.2</description>
<param-name>javax.faces.STATE_SAVING_METHOD</param-name>
<param-value>client</param-value>
</context-param>
<context-param>
<param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
<param-value>resources.application</param-value>
</context-param>
<listener>
<listener-class>com.sun.faces.config.ConfigureListener</listener-class>
</listener>
</web-app>

```

Faces-config.xml:-

```

<?xml version="1.0" encoding="UTF-8"?>
<faces-config
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-facesconfig_2_2.xsd"
version="2.2">

<navigation-rule>
<from-view-id>/index.xhtml</from-view-id>
<navigation-case>
<from-outcome>index</from-outcome>
<to-view-id>/index.xhtml</to-view-id>
</navigation-case>
</navigation-rule>

```

```
<managed-bean>
<managed-bean-name>bean</managed-bean-name>
<managed-bean-class>jsfpackage.bean</managed-bean-class>
<managed-bean-scope>session</managed-bean-scope>
</managed-bean>
</faces-config>
```

Output:-



Enter details to add Record:

ID:	<input type="text" value="1"/>
Enrollment Number:	<input type="text" value="cse.180840131008"/>
Current Sem:	<input type="text" value="6"/>
SPI (Eg:8.65):	<input type="text" value="9.2"/>
<input type="button" value="Submit"/>	

Stored Records:

Id	Enrollment	Semester	SPI
1	cse.180840131008	6	9.2