Experiment 5

Connect Swing Form to database using JDBC

Name: Sai Harsha Vardhan AVN

Roll-no: BCSE1823

Batch: ‘A’

Aim: Connect Swing Form to database using JDBC

Theory:

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers:

- JDBC-ODBC Bridge Driver,
- Native Driver,
- Network Protocol Driver, and
- Thin Driver

Before JDBC, ODBC API was the database API to connect and execute the query with the database. But, ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured). That is why Java has defined its own API (JDBC API) that uses JDBC drivers (written in Java language).

We can use JDBC API to handle database using Java program and can perform the following activities:

1. Connect to the database
2. Execute queries and update statements to the database
3. Retrieve the result received from the database.

Swing in Java is a Graphical User Interface (GUI) toolkit that includes the GUI components. Swing provides a rich set of widgets and packages to make sophisticated GUI components for Java applications. Swing is a part of Java Foundation Classes (JFC), which is an API for Java programs that provide GUI.

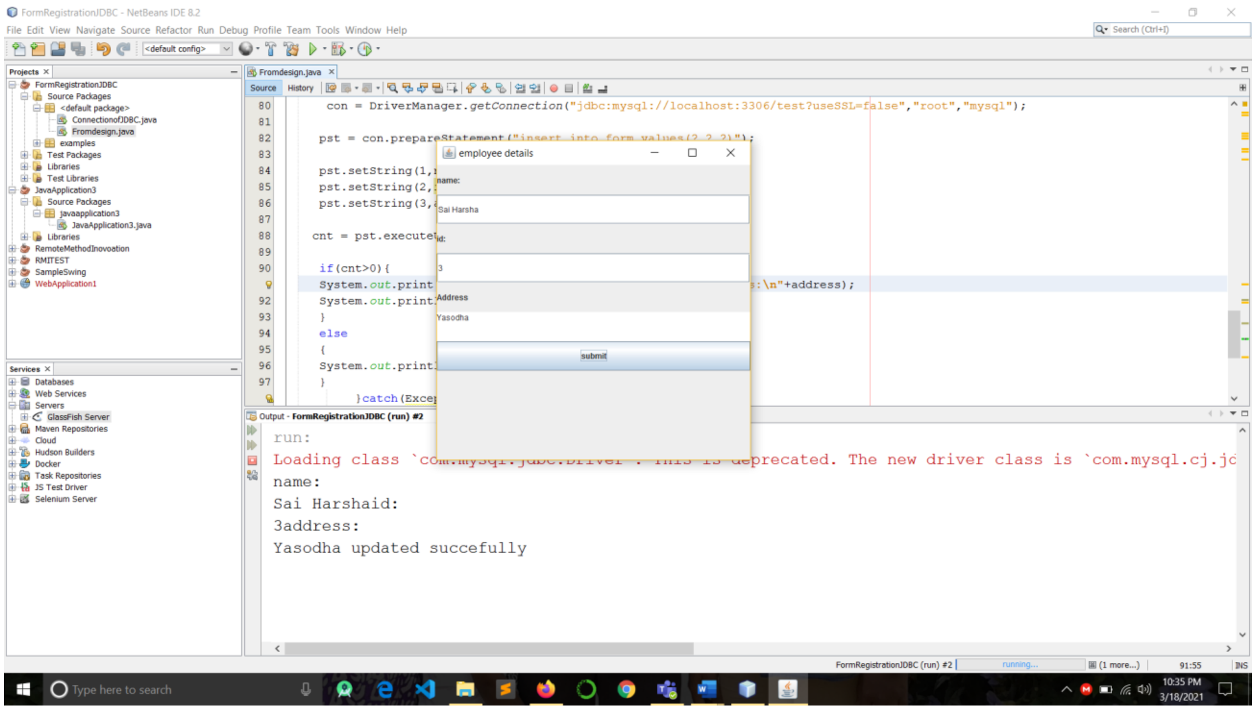
The Java Swing library is built on top of the Java Abstract Widget Toolkit (**AWT**), an older, platform dependent GUI toolkit. You can use the Java GUI programming components like button, textbox, etc. from the library and do not have to create the components from scratch.

Code:

```
import com.mysql.cj.protocol.Resultset;

import java.awt.Component;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.*;

import java.sql.*;

import static jdk.nashorn.internal.runtime.Debug.id;

public class Fromdesign {

    public static void main(String arrgs[])

{

JFrame f=new JFrame("employee details");

JLabel l1=new JLabel("name:");

f.add(l1);

final JTextField t1=new JTextField(30);

f.add(t1);

JLabel l2=new JLabel("id:");

f.add(l2);

final JTextField t2=new JTextField(30);

f.add(t2);

JLabel l3=new JLabel("Address");

f.add(l3);
```

```
final JTextArea area=new JTextArea();

f.add(area);

/*JLabel l4=new JLabel("gender");

f.add(l4);

JRadioButton r1=new JRadioButton("male");

JRadioButton r2=new JRadioButton("female");

ButtonGroup bg=new ButtonGroup();

bg.add(r1);

bg.add(r2);

f.add(r1);

f.add(r2);*/

JButton fb = new JButton("submit");

f.add(fb);

f.setVisible(true);

f.setSize(500,500);

f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

f.setLayout(new GridLayout(10,2));

   fb.addActionListener(new ActionListener(){

    public void actionPerformed(ActionEvent e){

         Connection con;

      PreparedStatement pst;

     String name = t1.getText() , address =  area.getText();

     String id =  t2.getText();
     int cnt =0;

        try{

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

    con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test?useSSL=false","root","mysql");

   pst = con.prepareStatement("insert into form values(?,?,?)");

   pst.setString(1,name);

   pst.setString(2,id);

   pst.setString(3,address);

  cnt = pst.executeUpdate();

   if(cnt>0){

   System.out.print("name:\n"+name+"\n"+"id:\n"+id+"\n"+"address:\n"+address);

   System.out.println("updated succefully");

   }

   else

   {

   System.out.println("updation failed");

   }

        }catch(Exception ex){

                 System.out.print(ex);

           }

        }

    });

 } }
```

Output:

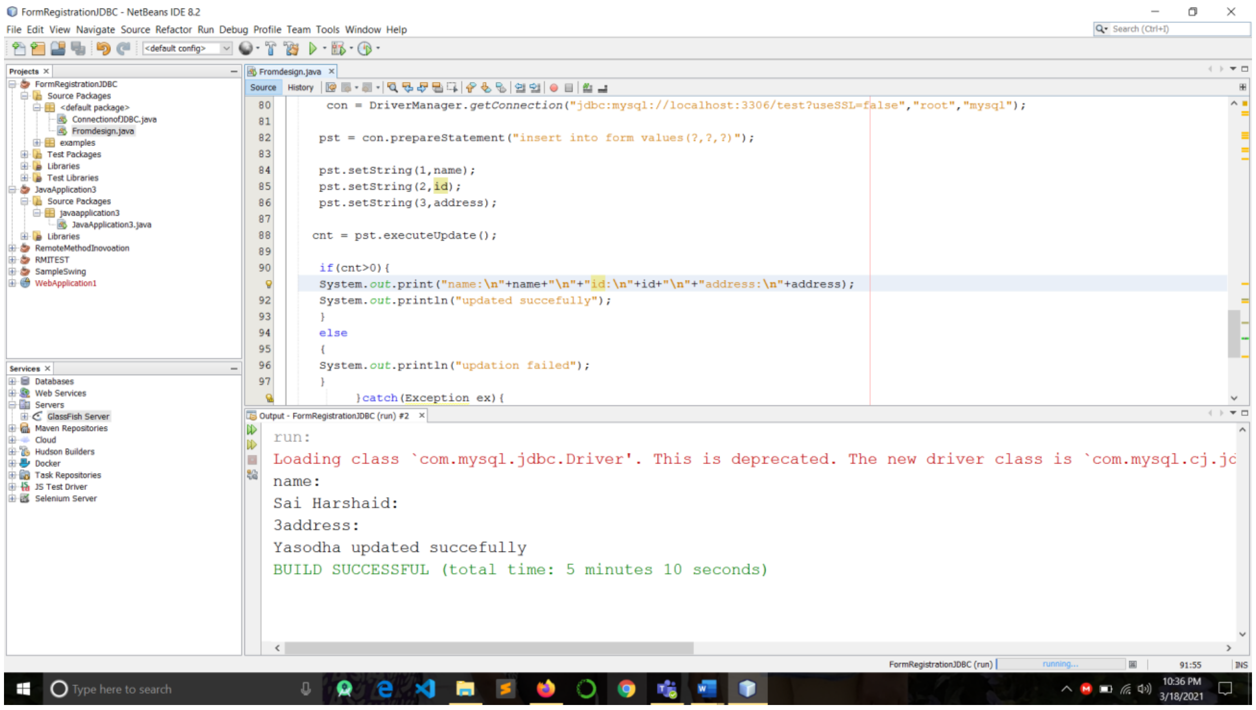


FormRegistrationJDBC - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Fromdesign.java
Source History
employee details
submit
Output - FormRegistrationJDBC (run) #2

```
 con = DriverManager.getConnection("jdbc:mysql://localhost:3306/test?useSSL=false","root","mysql");

pst = con.prepareStatement("insert into form values(?,?,?)");

pst.setString(1,
pst.setString(2,
pst.setString(3,

cnt = pst.execute

if(cnt>0){
System.out.print                                              :\n"+address);
System.out.print
}
else
{
System.out.print
}
    }catch(Exce
```

name:
Sai Harsha
id:
3
Address
Yasodha

```
run:
Loading class `co                        deprecated. The new driver class is `com.mysql.cj.jc
name:
Sai Harshaid:
3address:
Yasodha updated succefully
```

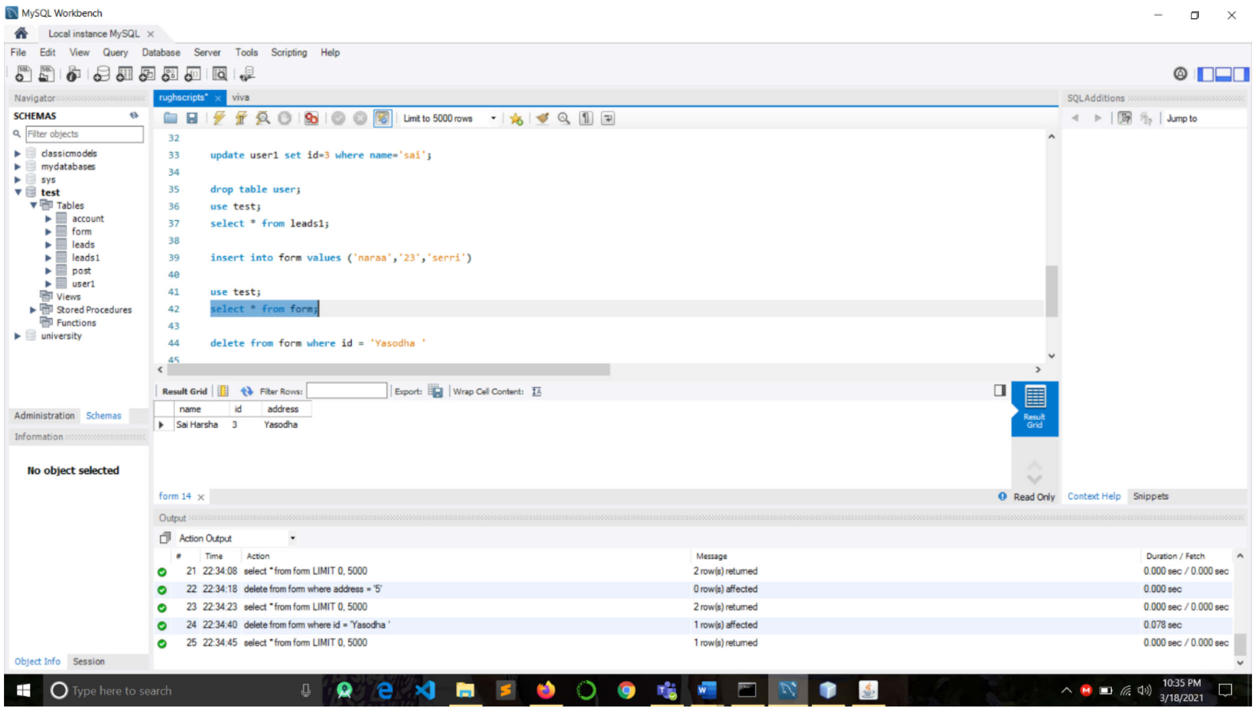


FormRegistrationJDBC - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Fromdesign.java
Source History
Output - FormRegistrationJDBC (run) #2

```
 con = DriverManager.getConnection("jdbc:mysql://localhost:3306/test?useSSL=false","root","mysql");

pst = con.prepareStatement("insert into form values(?,?,?)");

pst.setString(1,name);
pst.setString(2,id);
pst.setString(3,address);

cnt = pst.executeUpdate();

if(cnt>0){
System.out.print("name:\n"+name+"\n"+"id:\n"+id+"\n"+"address:\n"+address);
System.out.println("updated succefully");
}
else
{
System.out.println("updation failed");
}
    }catch(Exception ex){
```

```
run:
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jc
name:
Sai Harshaid:
3address:
Yasodha updated succefully
BUILD SUCCESSFUL (total time: 5 minutes 10 seconds)
```

The entries from the registration form are successfully updated in the data base.



```
update user1 set id=3 where name='sai';

drop table user;
use test;
select * from leads1;

insert into form values ('naraa','23','serri')

use test;
select * from form;

delete from form where id = 'Yasodha '
```

name	id	address
Sai Harsha	3	Yasodha

```
#   Time      Action                                          Message             Duration / Fetch
21  22:34:08  select * from form LIMIT 0, 5000                2 row(s) returned   0.000 sec / 0.000 sec
22  22:34:18  delete from form where address = '5'            0 row(s) affected   0.000 sec
23  22:34:23  select * from form LIMIT 0, 5000                2 row(s) returned   0.000 sec / 0.000 sec
24  22:34:40  delete from form where id = 'Yasodha '          1 row(s) affected   0.078 sec
25  22:34:45  select * from form LIMIT 0, 5000                1 row(s) returned   0.000 sec / 0.000 sec
```

<u>Conclusion</u>: Connect Swing Form to database using JDBC