Ex No : 11 **Data Base Connectivity**

Date :

AIM:

To implement and Execute database Connectivity using Java

PROGRAM:

```
package com.mycompany.databasebasic;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
class UserFrame extends JFrame {
   JLabel head,Author;
   JButton BookDetail;
   JTable table,SelectionTable;
   JComboBox AuthorSelect;
   JScrollPane scroll;
   String AuthorName[] = {};
   UserFrame() {
      Font f = new Font("TimesRoman",Font.BOLD,40);
      head = new JLabel("Online Book Store");
      head.setBounds(100, 30, 400, 50);
      head.setFont(new Font(f.getName(),Font.BOLD,24));
      head.setHorizontalAlignment(SwingConstants.CENTER);
      add(head);
      BookDetail = new JButton("BookDetail");
      BookDetail.setBounds(50, 100, 100, 50);
      add(BookDetail);
      Author = new JLabel("Author");
```

```
        Author.setBounds(80, 300, 100, 50);
        add(Author);
        AuthorSelect = new JComboBox(AuthorName);
        AuthorSelect.setBounds(60, 380, 100, 30);
        add(AuthorSelect);
        DefaultTableModel model = new DefaultTableModel();
        model.addColumn("ID");
        model.addColumn("Title");
        model.addColumn("Author");
        model.addColumn("Price");
        model.addColumn("Quantity");
        table = new JTable(model);
        scroll = new JScrollPane(table);
        scroll.setBounds(200, 100, 500, 200);
        add(scroll);
        SelectionTable = new JTable(model);
        scroll = new JScrollPane(SelectionTable);
        scroll.setBounds(200, 320, 500, 200);
        add(scroll);
        BookDetail.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {
                try {
                    Connection c =
DriverManager.getConnection("jdbc:derby://localhost:1527/ebookshop");
                    Statement st = c.createStatement();
                    ResultSet rs = st.executeQuery("SELECT * FROM BOOKS");
                    DefaultTableModel model = (DefaultTableModel) table.getModel();
                    model.setRowCount(0);
                    while (rs.next()) {
                        model.addRow(new Object[] {
                            rs.getInt(1),
```

```
                    rs.getString(2),
                    rs.getString(3),
                    rs.getDouble(4),
                    rs.getInt(5)
                });
            }
            rs.close();
            st.close();
            c.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
});
    AuthorSelect.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent ae) {
            try {
                Connection c =
DriverManager.getConnection("jdbc:derby://localhost:1527/ebookshop");
                Statement st = c.createStatement();
                ResultSet rs3 = st.executeQuery("SELECT * FROM BOOKS WHERE
Author='" + AuthorSelect.getSelectedItem() + "'");
                DefaultTableModel model = (DefaultTableModel) SelectionTable.getModel();
                model.setRowCount(0);
                while (rs3.next()) {
                    model.addRow(new Object[]{
                        rs3.getInt(1),
                        rs3.getString(2),
                        rs3.getString(3),
                        rs3.getDouble(4),
                        rs3.getInt(5)
                    });
```

```
                }
                rs3.close();
                st.close();
                c.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    });
    setLayout(null);
    setSize(800, 700);
    setTitle("User");
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setVisible(true);
  }
}
class AdminFrame extends JFrame {
  JLabel AdminHead,title1,title2,title3,p1,q1,i1,i2,Author,Quantity,price;
  JComboBox c1,c2,id;
  JTextArea a1,a2,a3,a4,a5,a6,a7;
  JButton UpdatePrice,UpdateQty,Delete,Insert;
  String name[] = {"Item1","Item2","Item3","Item4"};
  AdminFrame() {
    Font f = new Font("TimesRoman",Font.BOLD,40);
    AdminHead = new JLabel("Online Book Store");
    AdminHead.setBounds(100, 10, 400, 50);
    AdminHead.setFont(new Font(f.getName(),Font.BOLD,24));
    AdminHead.setHorizontalAlignment(SwingConstants.CENTER);
    title1 = new JLabel("Title");
    title1.setBounds(55, 70, 100, 50);
    title2 = new JLabel("Title");
    title2.setBounds(55, 120, 100, 50);
```

```
title3 = new JLabel("Title");
title3.setBounds(320, 230, 100, 50);
p1 = new JLabel("Price");
p1.setBounds(250, 70, 100, 50);
q1 = new JLabel("Quantity");
q1.setBounds(235, 120, 100, 50);
i1 = new JLabel("Id");
i1.setBounds(60, 170, 100, 50);
i2 = new JLabel("Id");
i2.setBounds(80, 230, 100, 50);
Author = new JLabel("Author");
Author.setBounds(80, 280, 100, 50);
price = new JLabel("Price");
price.setBounds(320, 280, 100, 50);
Quantity = new JLabel("Quantity");
Quantity.setBounds(80, 330, 100, 50);
add(AdminHead);add(title1);add(title2);add(title3);add(p1);add(q1);
add(i1);add(i2);add(Author);add(price);add(Quantity);
c1 = new JComboBox(name);
c1.setBounds(110, 84, 100, 30);
c2 = new JComboBox(name);
c2.setBounds(110, 134, 100, 30);
id = new JComboBox(name);
id.setBounds(110, 184, 180, 30);
add(c1);add(c2);add(id);
a1 = new JTextArea(100,100);
a1.setBounds(300, 84, 100, 30);
a2 = new JTextArea(100,100);
a2.setBounds(300, 134, 100, 30);
a3 = new JTextArea(100,100);
a3.setBounds(160, 245, 130, 30);
a4 = new JTextArea(100,100);
```

```
        a4.setBounds(160, 295, 130, 30);
        a5 = new JTextArea(100,100);
        a5.setBounds(160, 345, 130, 30);
        a6 = new JTextArea(100,100);
        a6.setBounds(400, 245, 130, 30);
        a7 = new JTextArea(100,100);
        a7.setBounds(400, 295, 130, 30);
        add(a1);add(a2);add(a3);add(a4);add(a5);add(a6);add(a7);
        UpdatePrice = new JButton("Update Price");
        UpdatePrice.setBounds(420, 84, 150, 30);
        UpdateQty = new JButton("Update Quantity");
        UpdateQty.setBounds(420, 134, 150, 30);
        Delete = new JButton("Delete");
        Delete.setBounds(330, 184, 180, 30);
        Insert = new JButton("Insert");
        Insert.setBounds(320, 350, 210, 30);
        add(UpdatePrice);add(UpdateQty);add(Delete);add(Insert);
        UpdatePrice.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {
                try {
                    Connection c =
DriverManager.getConnection("jdbc:derby://localhost:1527/ebookshop");
                    Statement st = c.createStatement();
                    String author = c2.getSelectedItem().toString();
                    double newPrice = Double.parseDouble(a1.getText()); // Assuming a1 is the
price field
                    st.executeUpdate("UPDATE BOOKS SET Price=" + newPrice + " WHERE
Author='" + author + "'");
                    st.close();
                    c.close();
                } catch (SQLException e) {
                    e.printStackTrace();
```

```
            }
        }
    });
    UpdateQty.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent ae) {
            try {
                Connection c =
DriverManager.getConnection("jdbc:derby://localhost:1527/ebookshop");
                Statement st = c.createStatement();
                String author = c2.getSelectedItem().toString();
                int newQuantity = Integer.parseInt(a2.getText()); // Assuming a2 is the quantity
field
                st.executeUpdate("UPDATE BOOKS SET Quantity=" + newQuantity + "
WHERE Author='" + author + "'");
                st.close();
                c.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    });
    Delete.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent ae) {
            try {
                Connection c =
DriverManager.getConnection("jdbc:derby://localhost:1527/ebookshop");
                Statement st = c.createStatement();
                String author = c2.getSelectedItem().toString();
                st.executeUpdate("DELETE FROM BOOKS WHERE Author='" + author + "'");
                st.close();
                c.close();
```

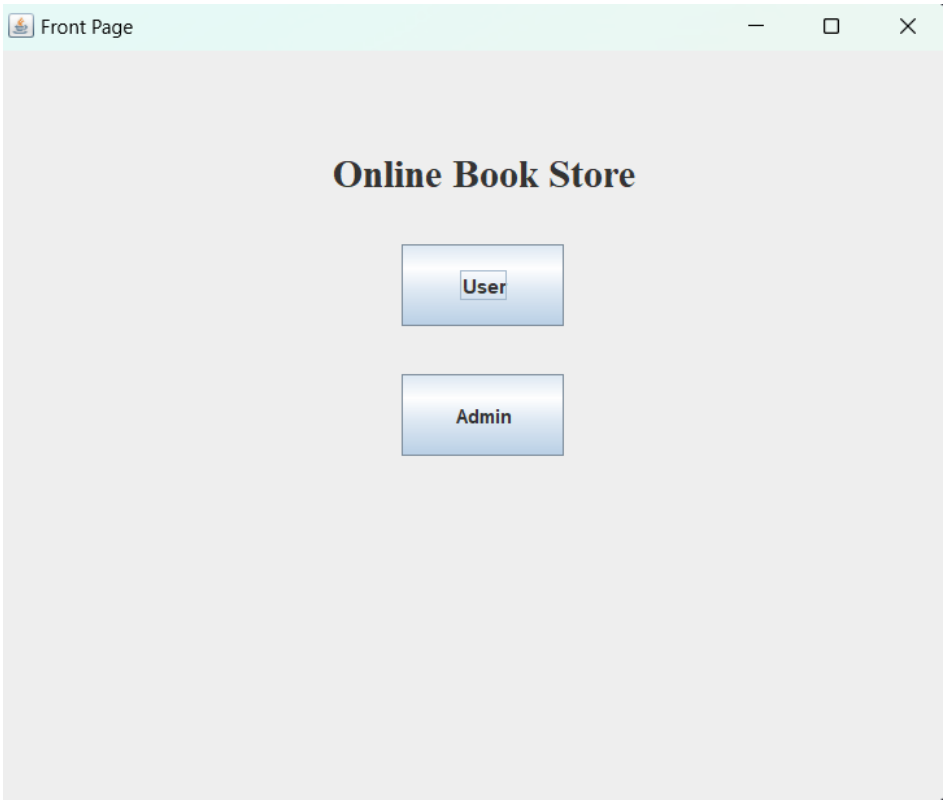
```
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    });
    Insert.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent ae) {
            try {
                Connection c =
DriverManager.getConnection("jdbc:derby://localhost:1527/ebookshop");
                Statement st = c.createStatement();
                String title = a6.getText();
                String author = a4.getText();
                double price = Double.parseDouble(a7.getText());
                int quantity = Integer.parseInt(a5.getText());
                st.executeUpdate("INSERT INTO BOOKS (Title, Author, Price, Quantity)
VALUES ('" +
                        title + "', '" + author + "', " + price + ", " + quantity + ")");
                st.close();
                c.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    });
    setLayout(null);
    setSize(630, 500);
    setTitle("Admin");
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setVisible(true);
  }
}
```

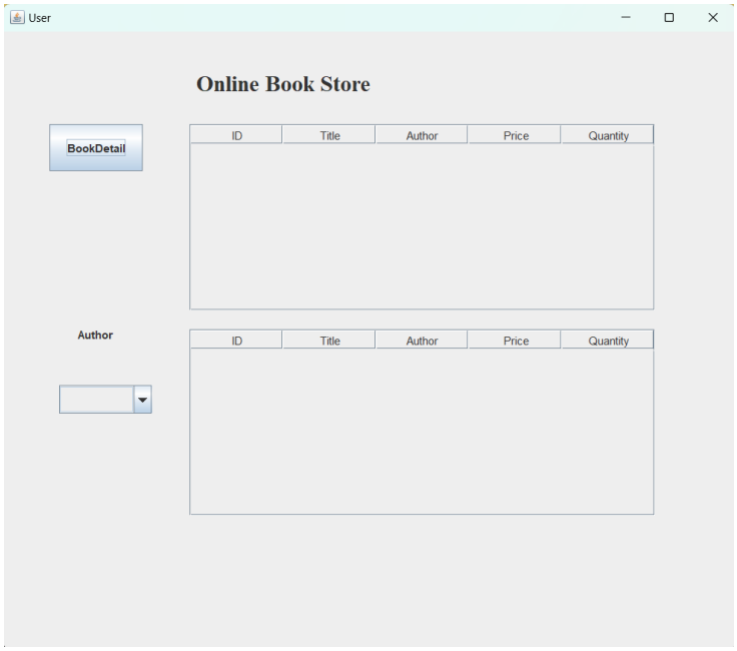
NAME: NAVEENKUMAR K
ROLLNO:22CSEB47

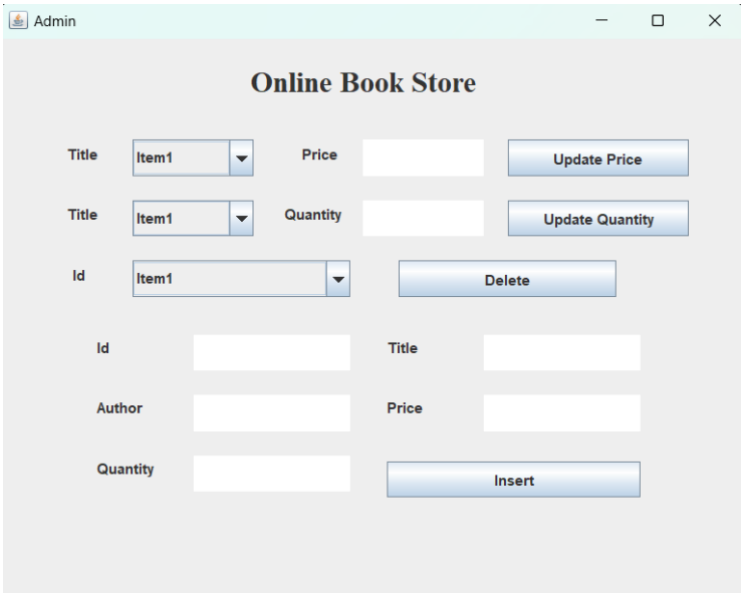
```
class page extends JFrame implements ActionListener {
    JLabel l1;
    JButton User,Admin;
    page(){
        Font f = new Font("TimesRoman",Font.BOLD,40);
        l1 = new JLabel("Online Book Store");
        l1.setBounds(100, 50, 400, 50);
        User = new JButton("User");
        User.setBounds(250, 120, 100, 50);
        Admin = new JButton("Admin");
        Admin.setBounds(250, 200, 100, 50);
        l1.setFont(new Font(f.getName(),Font.BOLD,24));
        l1.setHorizontalAlignment(SwingConstants.CENTER);
        setLayout(null);
        add(l1);
        add(User);
        add(Admin);
        User.addActionListener(this);
        Admin.addActionListener(this);
        setSize(600,500);
        setTitle("Front Page");
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    @Override
    public void actionPerformed(ActionEvent ae){
        if(ae.getSource() == User){
            new UserFrame();
        }
        if(ae.getSource() == Admin){
            new AdminFrame();
        }
```

```
    }
}
public class DataBaseBasic {
    public static void main(String[] args) throws IOException, SQLException {
        new page();
    }
}
```

OUTPUT:







Code/output(15)	
Quiz(5)	
Record(5)	
Total(25)	
Initial	

RESULT

Thus to implement and Execute database Connectivity using Java have been executed successfully.

NAME: NAVEENKUMAR K
ROLLNO:22CSEB47

NAME: NAVEENKUMAR K
ROLLNO:22CSEB47