

07-10-2020

LAB EXERCISES

DONE BY,
K. SASI KIRAN
MCA(R)
2019202049

1:

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author sasi
 */

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

class Auth_Pro implements ActionListener
{
    public JFrame frame;
    public static Connection conn;
    public JTextField txtQuery;
    public JButton btnNewButton, Clear;
    public DefaultTableModel model;
    public JTable table;
```

```

public JComboBox<String> comboBox,QueryBox;
public JScrollPane js;
public String querytype[] = { "All Authors","Authors Book","Book By
Author","Query" };
public static void connection()
{
    String url = "jdbc:postgresql://localhost:5432/testdb";
    String username = "postgres";
    String password = "psql";
    try
    {
        conn= (Connection) DriverManager.getConnection(url,
username, password);
        System.out.println("Connected Successfully!");
    }
    catch (SQLException e)
    {
        System.out.println("Connection Failed! Check output
console");
    }
}
Auth_Pro()
{
    frame = new JFrame();
    frame.getContentPane().setBackground(Color.WHITE);
    frame.setSize(1125, 290);
    frame.setResizable(false);
    frame.setLayout(null);
    Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
    int framewidth = frame.getSize().width;
    int frameheight = frame.getSize().height;
    int framelocationX = (dim.width - framewidth) / 2;
    int framelocationY = (dim.height - frameheight) / 2;
    frame.setLocation(framelocationX, framelocationY);
    frame.getContentPane().setBackground(Color.WHITE);
    frame.setTitle("Query Application");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel lblNewLabel_1 = new JLabel("Query Application");

```

```

        lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD |
Font.ITALIC, 17));
        lblNewLabel_1.setBounds(500, 24, 146, 40);
        frame.getContentPane().add(lblNewLabel_1);

        JLabel lblNewLabel = new JLabel("Choose Query");
        lblNewLabel.setFont(new Font("Bookman Old Style", Font.BOLD, 15));
        lblNewLabel.setBounds(10, 75, 129, 44);
        frame.getContentPane().add(lblNewLabel);

        comboBox = new JComboBox<String>(querytype);
        comboBox.setBackground(Color.WHITE);
        comboBox.setFont(new Font("Bookman Old Style", Font.BOLD, 13));
        comboBox.setBounds(180, 88, 207, 22);
        comboBox.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent evt) {
                combobox_ItemStateChanged(evt);
            }
        });
        frame.getContentPane().add(comboBox);

        JLabel lblNewLabel1 = new JLabel("Choose Author/Book");
        lblNewLabel1.setFont(new Font("Bookman Old Style", Font.BOLD,
15));
        lblNewLabel1.setBounds(10, 130, 170, 22);
        frame.getContentPane().add(lblNewLabel1);

        QueryBox = new JComboBox<String>();
        QueryBox.setBackground(Color.WHITE);
        QueryBox.setFont(new Font("Bookman Old Style", Font.BOLD, 13));
        QueryBox.setBounds(180, 130, 207, 22);
        QueryBox.setEnabled(false);
        frame.getContentPane().add(QueryBox);

        JLabel lblNewLabel2 = new JLabel("Enter Query");
        lblNewLabel2.setFont(new Font("Bookman Old Style", Font.BOLD,
15));
        lblNewLabel2.setBounds(10, 170, 170, 22);
        frame.getContentPane().add(lblNewLabel2);

```

```

txtQuery = new JTextField();
txtQuery.setBackground(Color.WHITE);
txtQuery.setFont(new Font("Bookman Old Style", Font.BOLD, 13));
txtQuery.setBounds(180,170,207,22);
txtQuery.setEnabled(false);
frame.getContentPane().add(txtQuery);

btnNewButton = new JButton("Fetch");
btnNewButton.setBackground(Color.WHITE);
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 15));
btnNewButton.setBounds(10, 210,180, 23);
btnNewButton.addActionListener(this);
frame.getContentPane().add(btnNewButton);

Clear = new JButton("Clear");
Clear.setBackground(Color.WHITE);
Clear.setFont(new Font("Times New Roman", Font.BOLD, 15));
Clear.setBounds(205, 210, 180, 23);
Clear.addActionListener(this);
frame.getContentPane().add(Clear);

table = new JTable();
js = new JScrollPane();
js.setBounds(400, 88, 700, 150);
table.setModel(new javax.swing.table.DefaultTableModel( new Object
[][] {},new String [] {"ID","NAME"}));
js.setViewportViewView(table);
frame.add(js);
}

private void combobox_ItemStateChanged(ItemEvent evt)
{
    model = new DefaultTableModel();
    table.setModel(model);
    model.addColumn("Id");
    model.addColumn("Name");
    Integer data = comboBox.getSelectedIndex();

    if(data.equals(0))
    {
        QueryBox.setEnabled(false);
    }
}

```

```

        btnNewButton.setEnabled(true);
        txtQuery.setEnabled(false);
    }
    else if(data.equals(1) || data.equals(2))
    {
        QueryBox.setEnabled(true);
        btnNewButton.setEnabled(true);
        txtQuery.setEnabled(false);
        try
        {
            Statement st = conn.createStatement();
            ResultSet rs;
            if(data.equals(1))
            {
                rs = st.executeQuery("SELECT author_name FROM
author_details");
            }
            else
            {
                rs = st.executeQuery("SELECT DISTINCT name FROM
book_details");
            }
            QueryBox.removeAllItems();
            while(rs.next())
            {
                QueryBox.addItem(rs.getString(1));
            }
        }
        catch(Exception e)
        {
            System.out.print("Didn't Retrive The Records From Author
Table");
        }
    }
    else
    {
        QueryBox.setEnabled(false);
        btnNewButton.setEnabled(true);
        txtQuery.setEnabled(true);
    }
}

```

```

    }

    @Override
    public void actionPerformed(ActionEvent arg0)
    {
        if (arg0.getSource().equals(btnNewButton))
        {
            Integer data = comboBox.getSelectedIndex();
            if(data.equals(0))
            {
                model = new DefaultTableModel();
                table.setModel(model);
                model.addColumn("Id");
                model.addColumn("Name");
                try
                {
                    Statement st = conn.createStatement();
                    ResultSet rs= st.executeQuery("SELECT * FROM
AUTHOR_DETAILS");

                    QueryBox.removeAllItems();

                    while(rs.next())
                    {
                        model.addRow(new
Object[]{rs.getString("author_id"), rs.getString("author_name")});
                    }
                }
                catch(Exception e)
                {
                    System.out.print("Didn't Retrive The Records From
Author Table");
                }
            }
            else if(data.equals(1))
            {
                String author_name =
QueryBox.getSelectedItem().toString();
                try
                {

```

```

        PreparedStatement st =conn.prepareStatement("select
b.id, b.name, b.year, b.isbn, a.author_name from book_details b,
author_details a where a.author_name=? and b.author_id=a.author_id");
        st.setString(1,author_name);
        ResultSet rs= st.executeQuery();
        model = new DefaultTableModel();
        table.setModel(model);
        model.addColumn("ID");
        model.addColumn("Book Name");
        model.addColumn("Year");
        model.addColumn("ISBN");
        model.addColumn("Author Name");

        while(rs.next())
        {
            model.addRow(new Object[]{
                rs.getString("id"),
                rs.getString("name"),
                rs.getString("year"),
                rs.getString("isbn"),
                rs.getString("author_name"),
            });
        }
    }
    catch(Exception e)
    {
        System.out.print("Didn't Retrive The Records From Book
And Author Table");
    }
}
else if(data.equals(2))
{
    String book_name = QueryBox.getSelectedItem().toString();
    try
    {
        PreparedStatement st =conn.prepareStatement("select
a.author_id, a.author_name from book_details b, author_details a where
b.name=? and b.author_id=a.author_id");
        st.setString(1,book_name);
        ResultSet rs= st.executeQuery();
    }
}

```

```

        model= new DefaultTableModel();
        table.setModel(model);
        model.addColumn("Id");
        model.addColumn("Author Name");

        while(rs.next())
        {
            model.addRow(new Object[]{
                rs.getString("author_id"),
                rs.getString("author_name"),
            });
        }
    }
    catch(Exception e)
    {
        System.out.print("Didn't Retrive The Records From Book
And Author Table");
    }
}
else if(data.equals(3))
{
    try
    {
        String sql =txtQuery.getText();
        Statement st = conn.createStatement();
        ResultSet rs= st.executeQuery(sql);
        ResultSetMetaData rsmd = rs.getMetaData();
        int count = rsmd.getColumnCount();
        model= new DefaultTableModel();
        table.setModel(model);
        String a;
        Object data1[]=new Object[count];

        for(int x=0; x<count;x++){
            a= rsmd洗getColumnName(x+1);
            model.addColumn(a);
        }

        while(rs.next()){
            for(int x=0; x<count;x++){

```



```

        a= rsmd.getColumnName(x+1);
        data1[x]=rs.getString(a);
    }

    model.addRow(data1);
}

}catch(SQLException e){
    System.out.print(e.toString());
}
}

else if(arg0.getSource().equals(Clear))
{
    comboBox.setSelectedIndex(0);
    QueryBox.removeAllItems();
    QueryBox.setEnabled(false);
    txtQuery.setText(null);
    txtQuery.setEnabled(false);
}
}

}

public class Author
{
    public static void main(String[] args)
    {
        Auth_Pro a = new Auth_Pro();
        Auth_Pro.connection();
        a.frame.setVisible(true);
    }
}

/**
 *
 * @author sasi

```

```

*/

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

class Auth_Pro implements ActionListener
{
    public JFrame frame;
    public static Connection conn;
    public JTextField txtQuery;
    public JButton btnNewButton, Clear;
    public DefaultTableModel model;
    public JTable table;
    public JComboBox<String> comboBox, QueryBox;
    public JScrollPane js;
    public String querytype[] = { "All Authors", "Authors Book", "Book By
Author", "Query" };
    public static void connection()
    {
        String url = "jdbc:postgresql://localhost:5432/testdb";
        String username = "postgres";
        String password = "psql";
        try
        {
            conn= (Connection) DriverManager.getConnection(url,
username, password);
            System.out.println("Connected Successfully!");
        }
        catch (SQLException e)
        {
            System.out.println("Connection Failed! Check output
console");
        }
    }
    Auth_Pro()
    {
        frame = new JFrame();
    }
}

```

```

frame.getContentPane().setBackground(Color.WHITE);
frame.setSize(1125, 290);
frame.setResizable(false);
frame.setLayout(null);
Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
int framewidth = frame.getSize().width;
int frameheight = frame.getSize().height;
int framelocationX = (dim.width - framewidth) / 2;
int framelocationY = (dim.height - frameheight) / 2;
frame.setLocation(framelocationX, framelocationY);
frame.getContentPane().setBackground(Color.WHITE);
frame.setTitle("Query Application");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.getContentPane().setLayout(null);

JLabel lblNewLabel_1 = new JLabel("Query Application");
lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD |
Font.ITALIC, 17));
lblNewLabel_1.setBounds(500, 24, 146, 40);
frame.getContentPane().add(lblNewLabel_1);

JLabel lblNewLabel = new JLabel("Choose Query");
lblNewLabel.setFont(new Font("Bookman Old Style", Font.BOLD, 15));
lblNewLabel.setBounds(10, 75, 129, 44);
frame.getContentPane().add(lblNewLabel);

comboBox = new JComboBox<String>(querytype);
comboBox.setBackground(Color.WHITE);
comboBox.setFont(new Font("Bookman Old Style", Font.BOLD, 13));
comboBox.setBounds(180, 88, 207, 22);
comboBox.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        combobox_ItemStateChanged(evt);
    }
});
frame.getContentPane().add(comboBox);

JLabel lblNewLabel1 = new JLabel("Choose Author/Book");
lblNewLabel1.setFont(new Font("Bookman Old Style", Font.BOLD,
15));

```

```

lblNewLabel1.setBounds(10, 130, 170, 22);
frame.getContentPane().add(lblNewLabel1);

QueryBox = new JComboBox<String>();
QueryBox.setBackground(Color.WHITE);
QueryBox.setFont(new Font("Bookman Old Style", Font.BOLD, 13));
QueryBox.setBounds(180, 130, 207, 22);
QueryBox.setEnabled(false);
frame.getContentPane().add(QueryBox);

JLabel lblNewLabel2 = new JLabel("Enter Query");
lblNewLabel2.setFont(new Font("Bookman Old Style", Font.BOLD,
15));

lblNewLabel2.setBounds(10, 170, 170, 22);
frame.getContentPane().add(lblNewLabel2);

txtQuery = new JTextField();
txtQuery.setBackground(Color.WHITE);
txtQuery.setFont(new Font("Bookman Old Style", Font.BOLD, 13));
txtQuery.setBounds(180, 170, 207, 22);
txtQuery.setEnabled(false);
frame.getContentPane().add(txtQuery);

btnNewButton = new JButton("Fetch");
btnNewButton.setBackground(Color.WHITE);
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 15));
btnNewButton.setBounds(10, 210, 180, 23);
btnNewButton.addActionListener(this);
frame.getContentPane().add(btnNewButton);

Clear = new JButton("Clear");
Clear.setBackground(Color.WHITE);
Clear.setFont(new Font("Times New Roman", Font.BOLD, 15));
Clear.setBounds(205, 210, 180, 23);
Clear.addActionListener(this);
frame.getContentPane().add(Clear);

table = new JTable();
js = new JScrollPane();
js.setBounds(400, 88, 700, 150);

```

```

        table.setModel(new javax.swing.table.DefaultTableModel( new Object
[[[] {}],new String [] {"ID","NAME"}));
        js.setViewportView(table);
        frame.add(js);
    }
    private void combobox_ItemStateChanged(ItemEvent evt)
    {
        model = new DefaultTableModel();
        table.setModel(model);
        model.addColumn("Id");
        model.addColumn("Name");
        Integer data = comboBox.getSelectedIndex();

        if(data.equals(0))
        {
            QueryBox.setEnabled(false);
            btnNewButton.setEnabled(true);
            txtQuery.setEnabled(false);
        }
        else if(data.equals(1) || data.equals(2))
        {
            QueryBox.setEnabled(true);
            btnNewButton.setEnabled(true);
            txtQuery.setEnabled(false);
            try
            {
                Statement st = conn.createStatement();
                ResultSet rs;
                if(data.equals(1))
                {
                    rs = st.executeQuery("SELECT author_name FROM
author_details");
                }
                else
                {
                    rs = st.executeQuery("SELECT DISTINCT name FROM
book_details");
                }
                QueryBox.removeAllItems();
                while(rs.next())

```

```

        {
            QueryBox.addItem(rs.getString(1));
        }
    }
    catch(Exception e)
    {
        System.out.print("Didn't Retrive The Records From Author
Table");
    }
}
else
{
    QueryBox.setEnabled(false);
    btnNewButton.setEnabled(true);
    txtQuery.setEnabled(true);
}
}
@Override
public void actionPerformed(ActionEvent arg0)
{
    if (arg0.getSource().equals(btnNewButton))
    {
        Integer data = comboBox.getSelectedIndex();
        if(data.equals(0))
        {
            model = new DefaultTableModel();
            table.setModel(model);
            model.addColumn("Id");
            model.addColumn("Name");
            try
            {
                Statement st = conn.createStatement();
                ResultSet rs= st.executeQuery("SELECT * FROM
AUTHOR_DETAILS");

                QueryBox.removeAllItems();

                while(rs.next())
                {
                    model.addRow(new
Object[]{rs.getString("author_id"), rs.getString("author_name")});

```

```

        }
    }
    catch(Exception e)
    {
        System.out.print("Didn't Retrive The Records From
Author Table");
    }
}
else if(data.equals(1))
{
    String author_name =
QueryBox.getSelectedItem().toString();
    try
    {
        PreparedStatement st =conn.prepareStatement("select
b.id, b.name, b.year, b.isbn, a.author_name from book_details b,
author_details a where a.author_name=? and b.author_id=a.author_id");
        st.setString(1,author_name);
        ResultSet rs= st.executeQuery();
        model = new DefaultTableModel();
        table.setModel(model);
        model.addColumn("ID");
        model.addColumn("Book Name");
        model.addColumn("Year");
        model.addColumn("ISBN");
        model.addColumn("Author Name");

        while(rs.next())
        {
            model.addRow(new Object[]{
                rs.getString("id"),
                rs.getString("name"),
                rs.getString("year"),
                rs.getString("isbn"),
                rs.getString("author_name"),
            });
        }
    }
    catch(Exception e)
    {

```

```

        System.out.print("Didn't Retrive The Records From Book
And Author Table");
    }
}
else if(data.equals(2))
{
    String book_name = QueryBox.getSelectedItem().toString();
    try
    {
        PreparedStatement st =conn.prepareStatement("select
a.author_id, a.author_name from book_details b, author_details a where
b.name=? and b.author_id=a.author_id");
        st.setString(1,book_name);
        ResultSet rs= st.executeQuery();
        model= new DefaultTableModel();
        table.setModel(model);
        model.addColumn("Id");
        model.addColumn("Author Name");

        while(rs.next())
        {
            model.addRow(new Object[]{
                rs.getString("author_id"),
                rs.getString("author_name"),
            });
        }
    }
    catch(Exception e)
    {
        System.out.print("Didn't Retrive The Records From Book
And Author Table");
    }
}
else if(data.equals(3))
{
    try
    {
        String sql =txtQuery.getText();
        Statement st = conn.createStatement();
        ResultSet rs= st.executeQuery(sql);
    }
}

```



```

        ResultSetMetaData rsmd = rs.getMetaData();
        int count = rsmd.getColumnCount();
        model= new DefaultTableModel();
        table.setModel(model);
        String a;
        Object data1[]=new Object[count];

        for(int x=0; x<count;x++){
            a= rsmd.getColumnName(x+1);
            model.addColumn(a);
        }

        while(rs.next()){
            for(int x=0; x<count;x++){
                a= rsmd.getColumnName(x+1);
                data1[x]=rs.getString(a);
            }

            model.addRow(data1);
        }

        }catch(SQLException e){
            System.out.print(e.toString());
        }
    }
}

else if(arg0.getSource().equals(Clear))
{
    comboBox.setSelectedIndex(0);
    QueryBox.removeAllItems();
    QueryBox.setEnabled(false);
    txtQuery.setText(null);
    txtQuery.setEnabled(false);
}

}

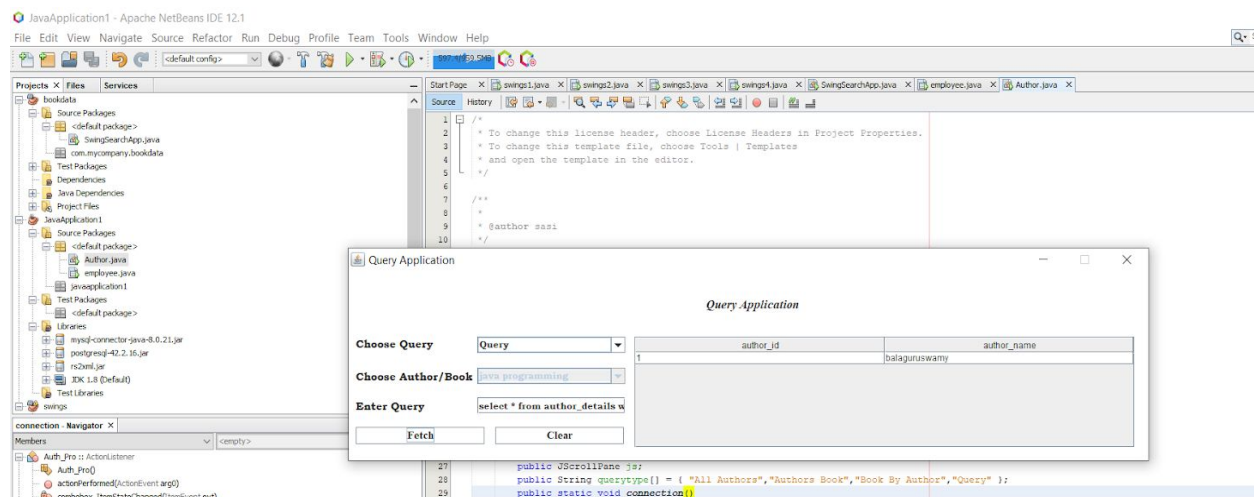
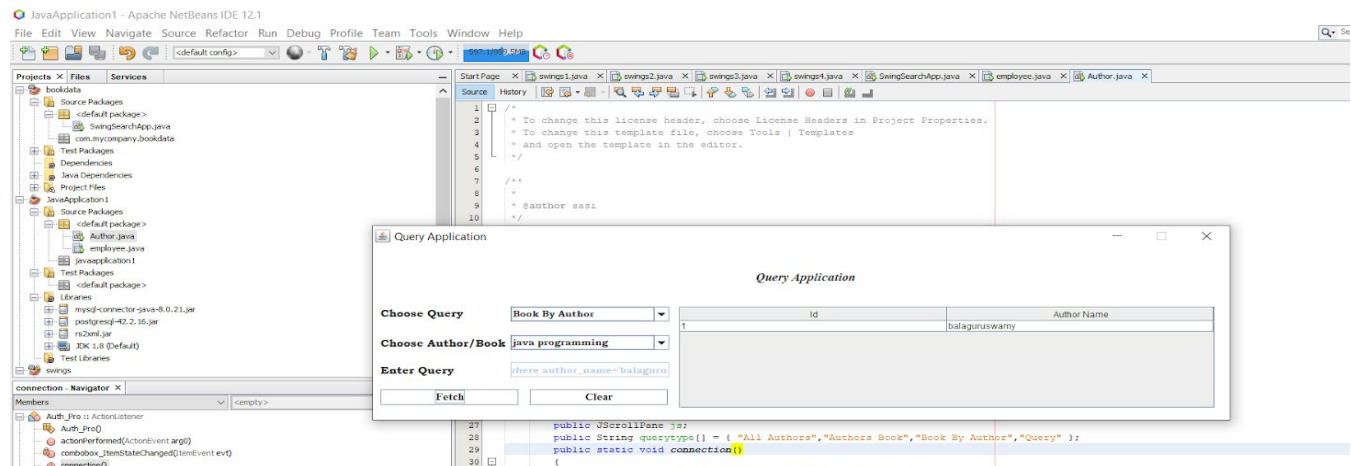
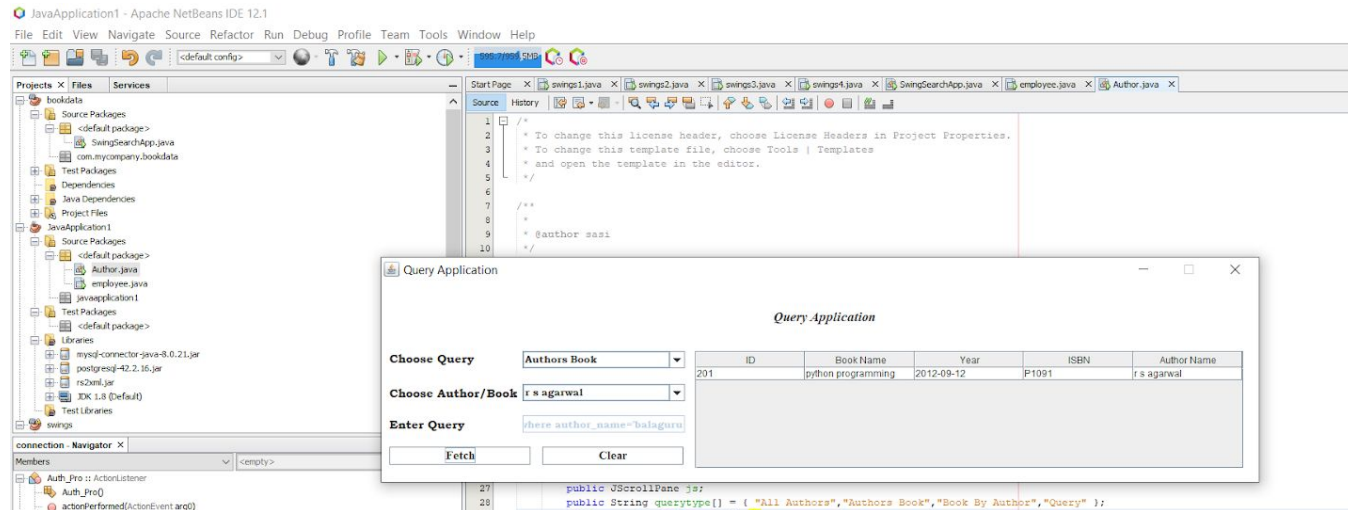
}

public class Author
{
    public static void main(String[] args)

```

```
{  
    Auth_Pro a = new Auth_Pro();  
    Auth_Pro.connection();  
    a.frame.setVisible(true);  
}  
}
```

OUTPUT:



2:

AUTHORS.JAVA

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author sasi
 */

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.sql.*;
import java.awt.event.ActionEvent;

class Auth implements ActionListener
{
    public JFrame frmAuthorDatabase;
    public JTextField id_txt;
    public JTextField nm_txt;
    public Connection conn;
    public JButton add,update,clear,exit;
    public JLabel error;
    public String id,name;
    Auth()
    {
        frmAuthorDatabase = new JFrame();
        frmAuthorDatabase.setTitle("Author Database");
        frmAuthorDatabase.getContentPane().setBackground(Color.WHITE);
        frmAuthorDatabase.setBounds(100, 100, 341, 283);
        frmAuthorDatabase.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frmAuthorDatabase.getContentPane().setLayout(null);
    }
}
```

```
JLabel database_lbl = new JLabel("Author Database");
database_lbl.setFont(new Font("Times New Roman", Font.BOLD, 25));
database_lbl.setBounds(74, 11, 202, 30);
frmAuthorDatabase.getContentPane().add(database_lbl);

JLabel auth_id = new JLabel("Author Id");
auth_id.setFont(new Font("Bookman Old Style", Font.BOLD, 15));
auth_id.setBounds(10, 60, 89, 24);
frmAuthorDatabase.getContentPane().add(auth_id);

JLabel authnm_lnl = new JLabel("Author Name");
authnm_lnl.setFont(new Font("Bookman Old Style", Font.BOLD, 15));
authnm_lnl.setBounds(10, 95, 103, 24);
frmAuthorDatabase.getContentPane().add(authnm_lnl);

id_txt = new JTextField();
id_txt.setFont(new Font("Bookman Old Style", Font.BOLD, 14));
id_txt.setBounds(143, 60, 173, 24);
frmAuthorDatabase.getContentPane().add(id_txt);
id_txt.setColumns(10);

nm_txt = new JTextField();
nm_txt.setFont(new Font("Bookman Old Style", Font.BOLD, 14));
nm_txt.setBounds(143, 95, 173, 24);
frmAuthorDatabase.getContentPane().add(nm_txt);
nm_txt.setColumns(10);

add = new JButton("Add");
add.addActionListener(this);
add.setBackground(Color.WHITE);
add.setFont(new Font("Algerian", Font.PLAIN, 17));
add.setBounds(10, 130, 143, 30);
frmAuthorDatabase.getContentPane().add(add);

update = new JButton("Update");
update.addActionListener(this);
update.setBackground(Color.WHITE);
update.setFont(new Font("Algerian", Font.PLAIN, 17));
update.setBounds(173, 130, 143, 30);
```

```

frmAuthorDatabase.getContentPane().add(update);

clear = new JButton("Clear");
clear.addActionListener(this);
clear.setBackground(Color.WHITE);
clear.setFont(new Font("Algerian", Font.PLAIN, 17));
clear.setBounds(10, 171, 143, 30);
frmAuthorDatabase.getContentPane().add(clear);

exit = new JButton("Exit");
exit.addActionListener(this);
exit.setBackground(Color.WHITE);
exit.setFont(new Font("Algerian", Font.PLAIN, 17));
exit.setBounds(173, 171, 143, 30);
frmAuthorDatabase.getContentPane().add(exit);

error = new JLabel("Welcome");
error.setForeground(Color.RED);
error.setFont(new Font("Bookman Old Style", Font.PLAIN, 15));
error.setBounds(124, 209, 110, 24);
frmAuthorDatabase.getContentPane().add(error);
frmAuthorDatabase.setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e)
{
    id = id_txt.getText().toUpperCase();
    name = nm_txt.getText().toUpperCase();
    if(e.getSource().equals(add))
    {
        conn = Connections.getConnections();
        if(conn != null)
        {
            System.out.print("Connected");
        }
        try
        {
            Statement st = conn.createStatement();
            st.execute("INSERT INTO AUTHOR_DETAILS
(AUTHOR_ID,AUTHOR_NAME) VALUES ('" + id + "','" + name + "')");

```

```

        error.setText("Add Successfully");
        id_txt.setText(null);
        nm_txt.setText(null);
    }
    catch(Exception ex)
    {
        error.setText("Not Inserted");
    }
}
else if(e.getSource().equals(update))
{
    conn = Connections.getConnections();
    if(conn != null)
    {
        System.out.print("Connected");
    }
    try
    {
        Statement st = conn.createStatement();
        st.execute("UPDATE AUTHOR_DETAILS SET AUTHOR_NAME=' " +
name + "' WHERE AUTHOR_ID=' " + id + "'");
        error.setText("Update Successfully");
        id_txt.setText(null);
        nm_txt.setText(null);
    }
    catch(Exception ex)
    {
        error.setText("Not Updated");
    }
}
else if(e.getSource().equals(clear))
{
    id_txt.setText(null);
    nm_txt.setText(null);
    error.setText("Welcome");
}
else if(e.getSource().equals(exit))
{
    frmAuthorDatabase.dispose();
}
}

```

```

    }
}
public class Authors
{
    public static void main(String[] args)
    {
        //Auth window = new Auth();
    }
}

```

ISBN TABLE.JAVA

```

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author sasi
 */

import javax.swing.JFrame;
import java.awt.Color;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.Statement;

```



```

class Isbn implements ActionListener
{
    public JFrame frmIsbnDatabase;
    public JTextField auth_txt,isbn_txt;
    public JButton add,update,clear,add_auth;
    public JLabel error;
    public String sqlquery,id,isbn;
    public Connection conn;
    Isbn()
    {
        frmIsbnDatabase = new JFrame();
        frmIsbnDatabase.setTitle("ISBN Database");
        frmIsbnDatabase.getContentPane().setFont(new Font("Bookman Old
Style", Font.BOLD, 15));
        frmIsbnDatabase.getContentPane().setBackground(Color.WHITE);
        frmIsbnDatabase.setBounds(100, 100, 328, 283);
        frmIsbnDatabase.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frmIsbnDatabase.getContentPane().setLayout(null);

        JLabel isbntable = new JLabel("ISBN Database");
        isbntable.setFont(new Font("Times New Roman", Font.BOLD, 24));
        isbntable.setBounds(80, 11, 172, 39);
        frmIsbnDatabase.getContentPane().add(isbntable);

        JLabel authid_lbl = new JLabel("Author Id");
        authid_lbl.setFont(new Font("Bookman Old Style", Font.BOLD, 15));
        authid_lbl.setBounds(10, 61, 89, 24);
        frmIsbnDatabase.getContentPane().add(authid_lbl);

        JLabel isbn_lbl = new JLabel("ISBN");
        isbn_lbl.setFont(new Font("Bookman Old Style", Font.BOLD, 15));
        isbn_lbl.setBounds(10, 98, 46, 24);
        frmIsbnDatabase.getContentPane().add(isbn_lbl);

        auth_txt = new JTextField();
        auth_txt.setFont(new Font("Bookman Old Style", Font.BOLD, 14));
        auth_txt.setBounds(116, 61, 188, 24);
        frmIsbnDatabase.getContentPane().add(auth_txt);
        auth_txt.setColumns(10);
    }
}

```

```
isbn_txt = new JTextField();
isbn_txt.setFont(new Font("Bookman Old Style", Font.BOLD, 14));
isbn_txt.setBounds(116, 98, 188, 24);
frmIsbnDatabase.getContentPane().add(isbn_txt);
isbn_txt.setColumns(10);

add = new JButton("Add");
add.setBackground(Color.WHITE);
add.setFont(new Font("Algerian", Font.PLAIN, 17));
add.setBounds(10, 141, 142, 28);
add.addActionListener(this);
frmIsbnDatabase.getContentPane().add(add);

update = new JButton("Update");
update.setBackground(Color.WHITE);
update.setFont(new Font("Algerian", Font.PLAIN, 17));
update.setBounds(162, 141, 142, 28);
update.addActionListener(this);
frmIsbnDatabase.getContentPane().add(update);

add_auth = new JButton("Add Author");
add_auth.setBackground(Color.WHITE);
add_auth.setFont(new Font("Algerian", Font.PLAIN, 17));
add_auth.setBounds(10, 180, 142, 28);
add_auth.addActionListener(this);
frmIsbnDatabase.getContentPane().add(add_auth);

clear = new JButton("Clear");
clear.setBackground(Color.WHITE);
clear.setFont(new Font("Algerian", Font.PLAIN, 17));
clear.setBounds(162, 180, 142, 28);
clear.addActionListener(this);
frmIsbnDatabase.getContentPane().add(clear);

error = new JLabel("Welcome");
error.setForeground(Color.RED);
error.setFont(new Font("Bookman Old Style", Font.BOLD, 13));
error.setBounds(116, 212, 94, 24);
frmIsbnDatabase.getContentPane().add(error);
```

```

        frmIsbnDatabase.setVisible(true);

    }

    @Override
    public void actionPerformed(ActionEvent e)
    {
        id = auth_txt.getText().toUpperCase();
        isbn = isbn_txt.getText().toUpperCase();
        if(e.getSource().equals(add))
        {
            conn = Connections.getConnections();
            try
            {
                Statement st = conn.createStatement();
                st.execute("INSERT INTO ISBN (AUTHOR_ID,ISBN) VALUES ('" +
id + "','" + isbn + "')");
                error.setText("Add Successfully");
                auth_txt.setText(null);
                isbn_txt.setText(null);
            }
            catch(Exception ex)
            {
                error.setText("Not Inserted");
            }
        }
        else if(e.getSource().equals(update))
        {
            conn = Connections.getConnections();
            if(conn != null)
            {
                System.out.print("Connected");
            }
            try
            {
                Statement st = conn.createStatement();
                st.execute("UPDATE ISBN SET ISBN='" + isbn + "'" WHERE
AUTHOR_ID='" + id + "'");
                error.setText("Update Successfully");
                auth_txt.setText(null);
                isbn_txt.setText(null);
            }
        }
    }
}

```

```

        }
        catch(Exception ex)
        {
            error.setText("Not Updated");
        }
    }
    else if(e.getSource().equals(clear))
    {
        auth_txt.setText(null);
        isbn_txt.setText(null);
        error.setText("Welcome");
    }
    else if(e.getSource().equals(add_auth))
    {
        new Auth();
    }
}
}

public class IsbnTable
{
    public static void main(String[] args)
    {
        //Isbn window = new Isbn();
    }
}

```

BOOK DATABASE .JAVA

```

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

```

```

/**
 *
 * @author sasi
 */

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.Statement;
import java.awt.event.ActionEvent;

class Book implements ActionListener
{
    public JFrame frmBookTable;
    public JTextField title_txt;
    public JTextField isbn_txt;
    public JTextField cpy_txt;
    public JButton add,update,add_isbn,clear;
    public JLabel errorlbl;
    public String title,isbn,cpright;
    public Connection conn;
    Book()
    {
        frmBookTable = new JFrame();
        frmBookTable.setTitle("Book Table");
        frmBookTable.setBackground(Color.WHITE);
        frmBookTable.getContentPane().setBackground(Color.WHITE);
        frmBookTable.setBounds(100, 100, 330, 285);
        frmBookTable.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frmBookTable.getContentPane().setLayout(null);

        JLabel bookdatabase = new JLabel("Book Database");
        bookdatabase.setFont(new Font("Times New Roman", Font.BOLD, 25));
        bookdatabase.setBounds(78, 11, 168, 24);
    }
}

```

```
frmBookTable.getContentPane().add(bookdatabase);

JLabel title = new JLabel("Title");
title.setFont(new Font("Bookman Old Style", Font.BOLD, 17));
title.setBounds(10, 49, 68, 24);
frmBookTable.getContentPane().add(title);

JLabel isbn = new JLabel("ISBN");
isbn.setFont(new Font("Bookman Old Style", Font.BOLD, 17));
isbn.setBounds(10, 80, 52, 24);
frmBookTable.getContentPane().add(isbn);

JLabel copyright = new JLabel("Copyright");
copyright.setFont(new Font("Bookman Old Style", Font.BOLD, 17));
copyright.setBounds(10, 111, 104, 24);
frmBookTable.getContentPane().add(copyright);

title_txt = new JTextField();
title_txt.setFont(new Font("Bookman Old Style", Font.BOLD, 14));
title_txt.setBounds(137, 49, 168, 24);
frmBookTable.getContentPane().add(title_txt);
title_txt.setColumns(10);

isbn_txt = new JTextField();
isbn_txt.setFont(new Font("Bookman Old Style", Font.BOLD, 14));
isbn_txt.setBounds(137, 80, 168, 24);
frmBookTable.getContentPane().add(isbn_txt);
isbn_txt.setColumns(10);

cpy_txt = new JTextField();
cpy_txt.setFont(new Font("Bookman Old Style", Font.BOLD, 14));
cpy_txt.setBounds(137, 111, 168, 24);
frmBookTable.getContentPane().add(cpy_txt);
cpy_txt.setColumns(10);

add = new JButton("Add");
add.addActionListener(this);
add.setBackground(Color.WHITE);
add.setFont(new Font("Algerian", Font.PLAIN, 14));
add.setBounds(10, 153, 132, 23);
```

```

frmBookTable.getContentPane().add(add);

update = new JButton("Update");
update.addActionListener(this);
update.setBackground(Color.WHITE);
update.setFont(new Font("Algerian", Font.PLAIN, 14));
update.setBounds(153, 153, 152, 23);
frmBookTable.getContentPane().add(update);

add_isbn = new JButton("Add ISBN");
add_isbn.addActionListener(this);
add_isbn.setBackground(Color.WHITE);
add_isbn.setFont(new Font("Algerian", Font.PLAIN, 14));
add_isbn.setBounds(10, 187, 132, 23);
frmBookTable.getContentPane().add(add_isbn);

clear = new JButton("Clear");
clear.addActionListener(this);
clear.setBackground(Color.WHITE);
clear.setFont(new Font("Algerian", Font.PLAIN, 14));
clear.setBounds(153, 187, 152, 23);
frmBookTable.getContentPane().add(clear);

errorlbl = new JLabel("Welcome");
errorlbl.setFont(new Font("Bookman Old Style", Font.BOLD, 13));
errorlbl.setForeground(Color.RED);
errorlbl.setBounds(120, 213, 114, 33);
frmBookTable.getContentPane().add(errorlbl);
}

@Override
public void actionPerformed(ActionEvent e)
{
    title = title_txt.getText().toUpperCase();
    isbn = isbn_txt.getText().toUpperCase();
    cpright = cpy_txt.getText().toUpperCase();
    if(e.getSource().equals(add))
    {
        conn = Connections.getConnections();
        if(conn != null)

```

```

        {
            System.out.print("Connected");
        }
        try
        {
            Statement st = conn.createStatement();
            st.execute("INSERT INTO BOOK (TITLE,ISBN,COPYRIGHT) VALUES
('" + title + "','" + isbn + "','" + cpright + "')");
            errorlbl.setText("Add Successfully");
            title_txt.setText(null);
            isbn_txt.setText(null);
            cpy_txt.setText(null);
        }
        catch(Exception ex)
        {
            errorlbl.setText("Not Inserted");
        }
    }
    else if(e.getSource().equals(update))
    {
        conn = Connections.getConnections();
        if(conn != null)
        {
            System.out.print("Connected");
        }
        try
        {
            Statement st = conn.createStatement();
            st.execute("UPDATE BOOK SET TITLE='" + title + "' AND
COPYRIGHT='" + cpright + "' WHERE ISBN='" + isbn + "'");
            errorlbl.setText("Update Successfully");
            title_txt.setText(null);
            isbn_txt.setText(null);
            cpy_txt.setText(null);
        }
        catch(Exception ex)
        {
            errorlbl.setText("Not Updated");
        }
    }
}

```



```

        else if(e.getSource().equals(clear))
        {
            title_txt.setText(null);
            isbn_txt.setText(null);
            cpy_txt.setText(null);
            errorlbl.setText("Welcome");
        }
        else if(e.getSource().equals(add_isbn))
        {
            new Isbn();
        }
    }
}

public class BookDatabase
{
    public static void main(String[] args)
    {
        Book window = new Book();
        window.frmBookTable.setVisible(true);
    }
}

```

CONNECTIONS.JAVA

```

/**
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author sasi

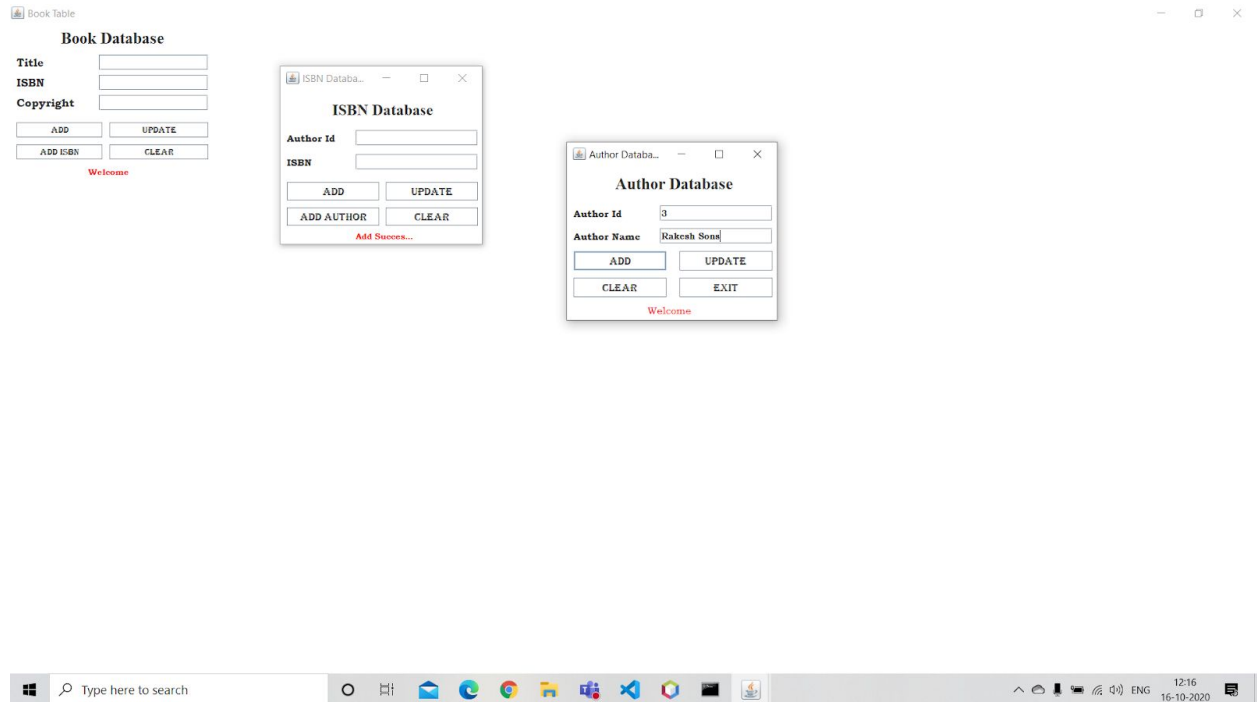
```

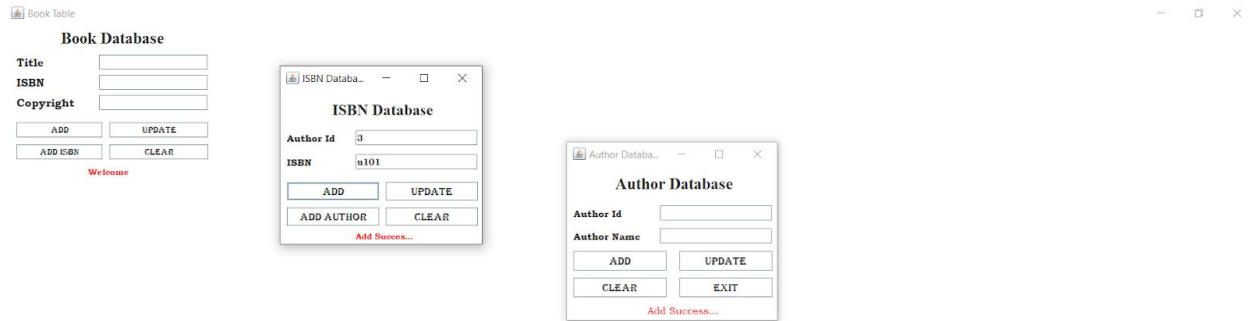
```
*/

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

public class Connections
{
    public static Connection getConnections()
    {
        String url = "jdbc:postgresql://localhost:5432/testdb";
        String username = "postgres";
        String password = "psql";
        Connection conn = null;
        try
        {
            Class.forName("org.postgresql.Driver");
            conn = DriverManager.getConnection(url, username, password);
        }
        catch (ClassNotFoundException e)
        {
            e.printStackTrace();
        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
        return conn;
    }
}
```

OUTPUT:





Book Table

Book Database

Title	<input type="text" value="ubuntu basics"/>
ISBN	<input type="text" value="u101"/>
Copyright	<input type="text" value="Krima and co"/>
<input type="button" value="ADD"/>	<input type="button" value="UPDATE"/>
<input type="button" value="ADD ISBN"/>	<input type="button" value="CLEAR"/>

Add Successful...