नास्ति विद्या समम् चक्षुः

BABASAHEB BHIM RAO AMBEDKAR

BIHAR UNIVERSITY

PROJECT REPORT

ON

LIBRARY MANAGEMENT SYSTEM

SUBMITTED TO

M.P. Sinha Science College

BACHELOR OF COMPUTER APPLICATION

UNDER THE GUIDANCE OF

Mr. Binay kr Singh

SUBMITED BY

Rajesh Kumar -17380200001

PREFACE

In our BCA (Bachelor Of Computer Application) course it is obligatory on part of every student to prepare a Project Report in partial fulfillment requirement of the degree course. Students by preparing the project experience the actual working situation and have a deep view of Computer Applications in practical work.

I have made my project on "LIBRARY MANAGEMENT SYSTEM". What I have done on this project, a report of that is being presented in this project report.

ACKNOWLEDGEMENT

I feel immense pleasure in presenting valuable possession of my "course co-ordination" to **BRAB** University.

I sincerely thank to my project guide **Mr.Binay kumar singh** (Teacher of ICETL institute, Patna) for guidance and encouragement in carrying out this project work I also wish to express my gratitude to the officials and other staff members of "ICETL institute,Patna" who rendered their help during the period of my project work. For their kind co-operation to the completion of my project work .I wish to express my profound gratitude and sincere thanks to my esteemed learned Director **MR. Himanshu Jaishwal**. who allowed me to join summer training Last but not least I wish to avail myself of this opportunity, express a sense of gratitude and love to my friends and my beloved parents for their manual support, strength, help and for everything.

I have been fortunate in having dedicated colleagues who took pains and tried to help in completing this project

INDEX

HARDWARE AND SOFTWARE REQUIREMENTS:

- Any Operating System
- INTEL™ Pentium 3, and above with speeds more than 1.6 GHz
- Minimum RAM – 256 Mb @ 32 – bit or 512 Mb @ 64 - bit
- Secondary Memory, Minimum 300 Mb
- JAVA Runtime Environment 6
- Admin Privileges On the System (Server Side)
- Dedicated Network link with speeds more than 64 Kbps
- Firewall policies unblocked for "Remote Database Client"
- Screen Resolution : Any Resolution, Recommended : 1024 x 2160

DESCRIPTION OF LIBRARY MANAGEMENT SYSTEM

Library management system plays a great role in any company and college. This is used to keep the data and details of company members of the company and in college this is used to keep the track of different books and students in the library. Student who has the book of library is put with his roll number in the database.

To overcome the drawbacks of file system, we use the library database.

Library management is used because this is simple to use and also no need of maintaining the records on papers, because computers will keep the track of every book and library.

Library software is developed in "**JAVA Swings**". "**JDBC**" is used for connectivity between software and database.

This contains Modules with different functions.

LIBRARY RECORD MAIN PAGE:-

There are four options on the main library page. These are:

1. Member Details.
2. Book Details.
3. Issue Details.
4. Exit.

MEMBER DETAILS OPTION :-

List:-

This module is used to see the already entered names in the database of the library.

Add Member:-

This button is used to add the members in the database of library.

When we click on the “Add Member” button then a new frame will open and in this we have different options like “Mem_Name, Mem_Address, Renewal_Date”. Then we have two buttons “Add and Close”. “Add” to add the members and “Close”to close the frame. We can enter many records at a time in the record.

Delete Member:-

This button is used to delete the entries from the list of library database.

When we click on the “Delete Member” button then a new frame open and this will have a text box that will ask to enter the Member Code of the member whom we want to delete.There are two buttons in this frame also. One is “Ok” means to delete.

And another is “Cancel” means to cancel without deleting entry.

Close:-

When we click on this button then main frame opens that contains four other buttons named Member Details, Book Details, Issue Details and Exit. Fourth is exit button that is to exit the complete frame.

BOOK DETAILS OPTION :-

List:-

This module is used to see the already entered names in the database of the library.

Add Book:-

This button is used to add the book in the database of library.

When we click on the "Add Book" button then a new frame will open and in this we have different options like "Book_Name, Author,Publisher". Then we have two buttons "Add and Close". "Add" to add the books and "Close" to close the frame. We can enter many records at a time in the record.

Delete Book:-

This button is used to delete the books from the list of library database.

When we click on the "Delete Book" button then a new frame open and this will have a text box that will ask to enter the Book Code of the books which we want to delete.

There are two buttons in this frame also. One is "Ok" means to delete.

And another is "Cancel" means to cancel without deleting entry.

Close:-

When we click on this button then another frame opens that contains four other buttons named Member Details, Book Details, Issue Details and Exit. Fourth is exit button that is to exit the complete frame.

ISSUE DETAILS OPTION :-

List:-

This module is used to see the already entered names in the database of the library.

Issue:-

This button is used to issue the book and add its record in the database of library.

When we click on the "Issue" button then a new frame will open and in this we have different optionslike "Issue_Id,Date_of_Issue, Date_of_Return, Mem_Code, Book_Code". Then we have two buttons "Issue and Close". "Issue" to issue the books and "Close" to close the frame. We can enter many records at a time in the record.

Return:-

This button is used to update the list of returned books in the library database.

When we click on the "Return" button then a new frame open and in this we have different options like "Issue_Id,Date_of_Issue,Date_of_Return,Mem_Code,Book_Code". There are two buttons in this frame also. One is “Return” means that the book has been returned.And another is “Close” to close the frame.

Close:-

When we click on this button then another frame opens that contains four other buttons named Member Details, Book Details, Issue Details and Exit. Fourth is exit button that is to exit the complete frame.

EXIT OPTION:-

Exit button that is fourth on the main window is used to exit the complete frame.

INTRODUCTION OF JAVA

Java

– was created in 1991

– by James Gosling et al. of Sun Microsystems.

– Initially called Oak, in honor of the tree outside Gosling's window, its name
was changed to Java because there was already a language called Oak.

The Java technology is:

– A programming language

– A development environment

– An application environment

– A deployment environment

Java Technology:

Programming Language

- As a programming language, Java can create all kinds of applications that you could create using any conventional programming language.

Java Technology:

A Development Environment

- **As a development environment, Java technology provides you with a large suite of tools:**

– A compiler (javac)

– An interpreter (java)

– A documentation generator (javadoc)

– A class file packaging tool

and so on.

Java Technology:

An Application and Runtime Environment

- Java technology applications are typically general-purpose programs that run on any machine where the Java runtime environment (JRE) is installed.

- **There are two main deployment environments:**

1. The JRE supplied by the Java 2 Software Development Kit (SDK) contains the complete set of class files for all the Java technology packages, which includes basic language classes, GUI component classes, and so on.

2. The other main deployment environment is on your web browser. Most commercial browsers supply a Java technology interpreter and runtime environment.

PRIMARY GOALS OF JAVA TECHNOLOGY

Java technology provides the following:

1. It is "simple, object oriented and familiar".
2. It is "robust and secure".

3. It is "architecture neutral and portable".

4. It executes with "high performance".

5. It can be "interpreted, threaded, and use dynamically".

6. Eliminates many pitfalls of other language like memory management and pointer
Arithmetic.

7. It is object oriented to help you to visualize the program in a real life terms.

8. Enables us to streamline the code.

Java Features

- **Some features of Java:**

– The Java Virtual Machine

– Garbage Collection

– Code Security

The Java Virtual Machine

- **Java Virtual Machine (JVM)**

– An imaginary machine that is implemented by emulating software on a real machine.

– It provides the hardware platform specifications to which you compile all Java technology code.

- **Byte code**

– A special machine language that can be understood by the Java Virtual Machine (JVM)

– Independent of any particular computer hardware, so any computer with a Java interpreter can execute the compiled Java program, no matter what type of computer the program was compiled on.

Garbage Collection

- **Garbage collection thread**

– Responsible for freeing any memory that can be freed. This happens automatically during the lifetime of the Java program.

– Programmer is freed from the burden of having to deallocate that memory themselves.

Code Security

- **Code**

-Security is attained in Java through the implementation of its Java Runtime Environment (JRE).

- **JRE**

– Runs code compiled for a JVM and performs class loading (through the class loader), code verification (through the byte code verifier) and finally code execution.

- **Class Loader**

– Responsible for loading all classes needed for the Java program.

– Adds security by separating the namespaces for the classes of the local file system from those that are imported from network sources.

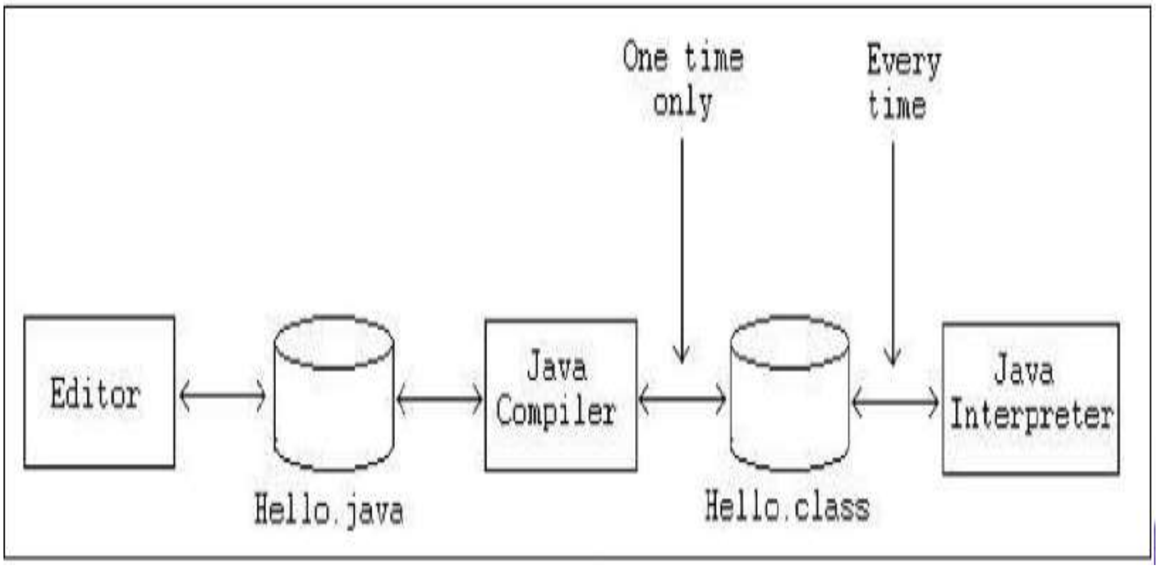
– After loading all the classes, the memory layout of the executable is then determined. This adds protection against unauthorized access to restricted areas of the code since the memory layout is determined during runtime

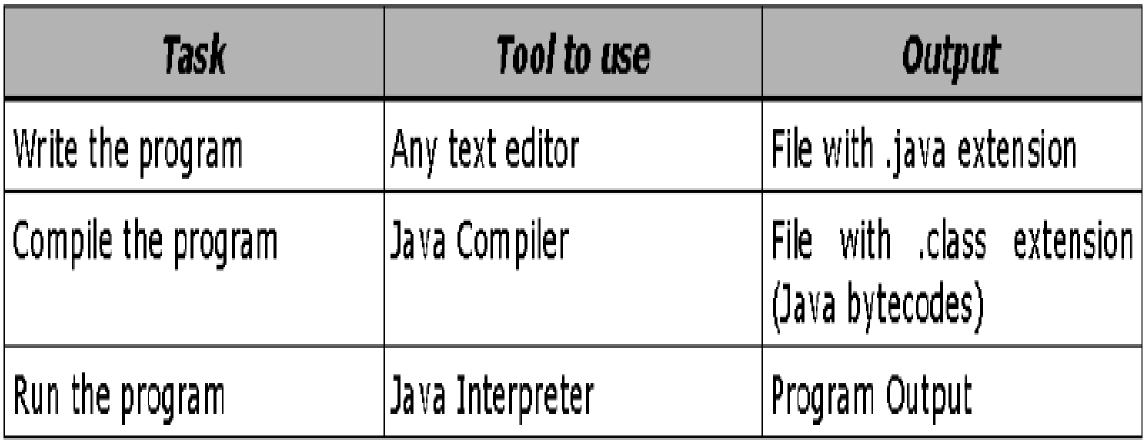
- **Byte code verifier**

– Tests the format of the code fragments and checks the code fragments for illegal code that can violate access rights to objects.

Phases of a Java Program

- **The following figure describes the process of compiling and executing a Java program**





Task	Tool to use	Output
Write the program	Any text editor	File with .java extension
Compile the program	Java Compiler	File with .class extension (Java bytecodes)
Run the program	Java Interpreter	Program Output

JAVA TOOLS

JAVAC COMPILER:-

Java programs are created under any text editor , say notepad just open the text editor , type your program and save this program with the extension **.java**.

For example: - javac filename.java

JAVA INTERPRETER:-

The java interpreter, java is used to execute the java class file produced by the javac complier.

For example: - java filename

JDB TOOL :-

The jdb tool is used to debug your program.

For example: - jdb filename.class

JAVAP DISASSEMBLER:-

If you do not have a java source file but have a byte code then you can get back the java

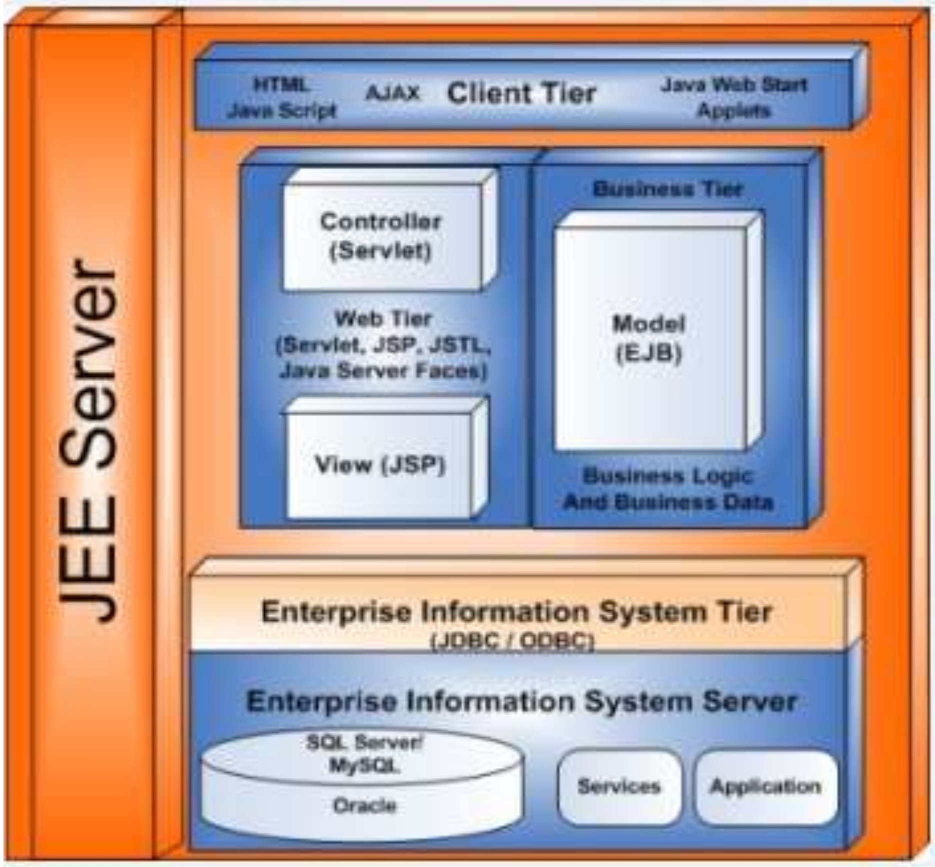
dissassembler ,javap.

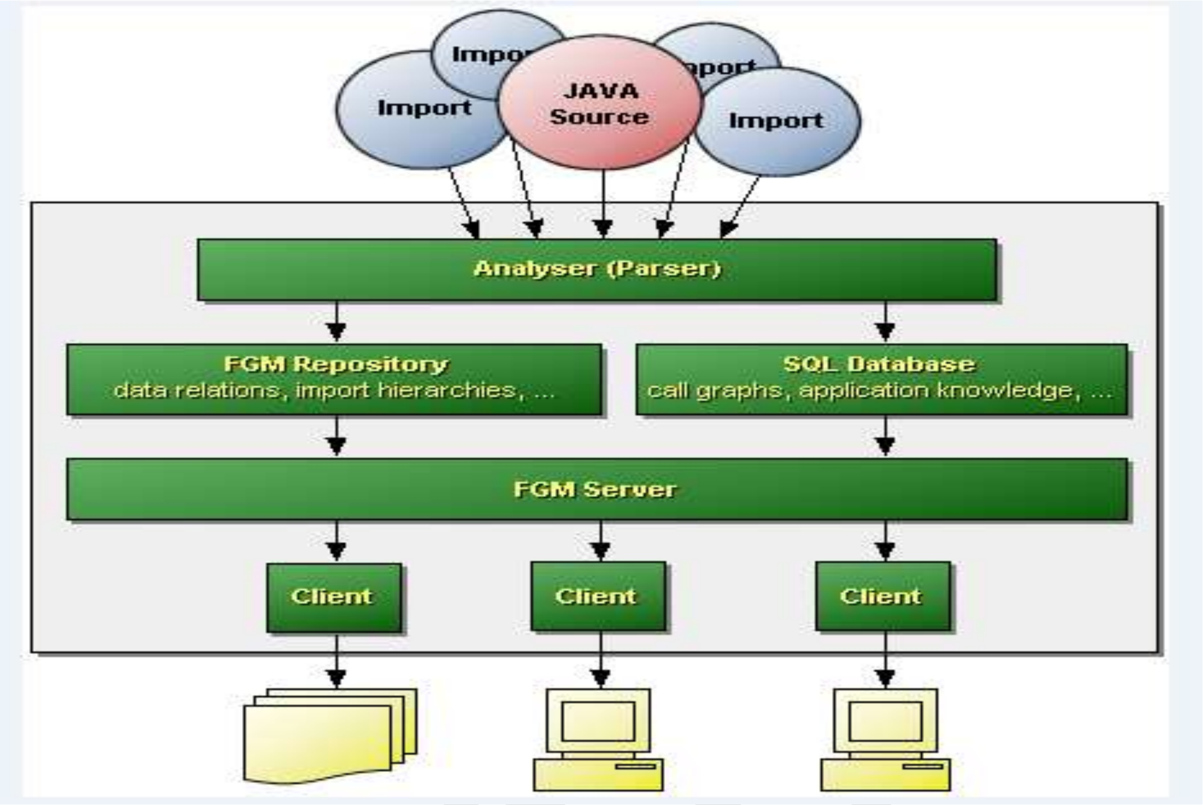
For example :- javap filename.class

JAVA DOC TOOL :-

The java doc is a document generator that creates html page documentations for the classes that you create

JAVA ARCHITECTURE





<u>The java technology architecture uses the following features to fulfill the previously listed goals :</u>

- The JVM
- Garbage Collection
- The JRE
- JVM tool interface

Advantages

1. The main advantages are that multiple os can coexist on the same computer in strong isolation from each other and that it can provide an instruction set architecture (ISA) different from that of the real machine.

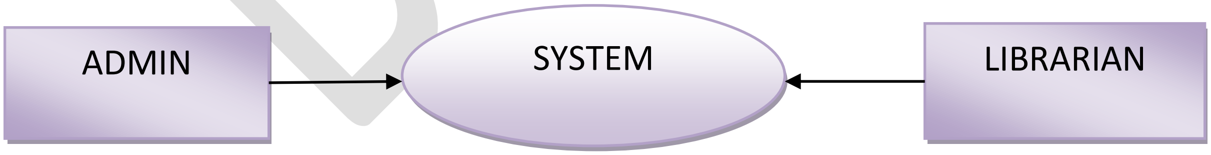
2. The main advantages are that multiple os can coexist on the same computer in strong isolation from each other and that it can provide an instruction set architecture (ISA) different from that of the real machine.

SYSTEM MODEL

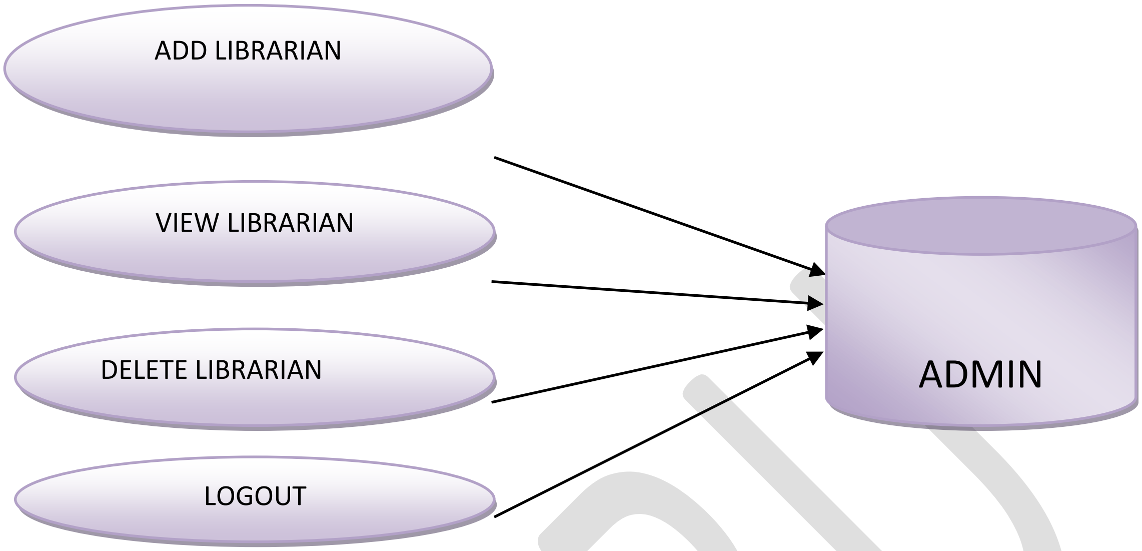
The structure of the system can the system can be divided into three main logical components. The first component must provide some form of menu management, allowing the institute to control what can be added or return by member.

DATA FLOW DIAGRAM(DFD)

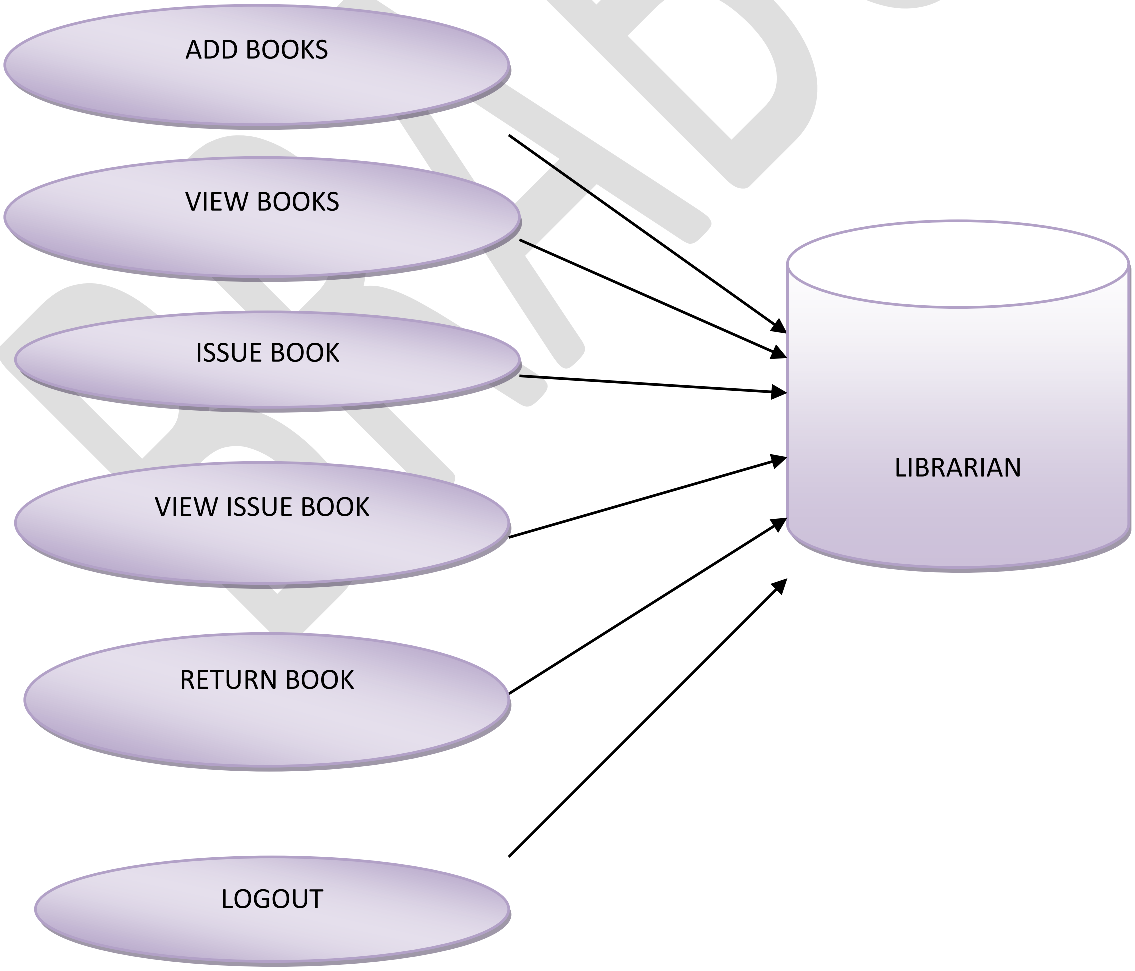
Level 0:-



Level 1:-



Level 1:



Admin Login Table :-

Field Name	Data Type
USERNAME	VARCHAR2(15)
PASSWORD	VARCHAR2(10)

Librarian Profile Table :-

Field Name	Data Type
ID	VARCHAR2(6)
NAME	VARCHAR2(25)
GENDER	VARCHAR2(6)
FATHER_ NAME	VARCHAR2(25)
PASSWORD	VARCHAR2(10)
CONFIRM_PASSWORD	VARCHAR2(10)
CITY	VARCHAR2(25)
EMAIL	VARCHAR2(25)
PHONE NO	NUMBER(12)

Login Librarian:-

Field Name	Data Type
USERNAME	VARCHAR2(15)
PASSWORD	VARCHAR2(10)

Add Book:-

Field Name	Data Type
CODE	VARCHAR2(6)
NAME	VARCHAR2(25)
AUTHOR	VARCHAR2(25)
PUBLISHER	VARCHAR2(25)
STOCK	NUMBER(3)
RACK	NUMBER(2)

Issue_Book:-

Field Name	Data Type
CODE	VARCHAR2(6)
ID	VARCHAR2(5)
NAME	VARCHAR2(25)
CONTACT NO	NUMBER(12)
ISSUE	DATE
REISSUE	DATE

PROJECT CODE-

CODE FOR MAKING THE MAIN LIBRARY PAGE-

```
package adminlogin;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class lms extends JFrame implements ActionListener {

    lms(){

            JFrame frame = new JFrame("LIBARARY MANAGEMENT SYSTEM");
            JButton b = new JButton("ADMIN");
            b.setBounds(150,100,130,50);
            Container c = frame.getContentPane();
            c.setLayout(null);
            c.setBackground(Color. blue);
            JButton b1 = new JButton("LIBRARIAN");
            b1.setBounds(150,170,130,50);
            frame.add(b);
            frame.add(b1);
            frame.setSize(450, 400);
            frame.setLayout(null);
            frame.setVisible(true);
            frame.setResizable(false);
            frame.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
            b.addActionListener(this);
            b1.addActionListener(this);

        }
        public static void main(String[] args) {
            new lms();

        }

        public void actionPerformed(ActionEvent ae)
        {
            if(ae.getActionCommand().equals("ADMIN")){
        AdminLogin login = new AdminLogin();
            }
            if(ae.getActionCommand().equals("LIBRARIAN")){
                LibrarianLogin login = new LibrarianLogin();
            }
        }
}
```

CODE FOR ADMINISTRATION DETAILS BUTTON-

```
package adminlogin;

import java.awt.*;
import java.awt.event.*;
import java.sql.*;

import javax.swing.*;

import dbcon.DBCon;
//import javafx.scene.paint.Color;


public class AdminLogin extends JFrame implements ActionListener{

        JFrame frame;
        JLabel idLabel = new JLabel("Enter ID:");
        JLabel passwordLabel = new JLabel("Enter Password:");
        JLabel message = new JLabel();
        JTextField idTextField = new JTextField();
        JPasswordField passwordField = new JPasswordField();
        JButton addButton = new JButton("LOGIN");
        AdminLogin(){
                createWindow();
                setLocationAndSize();
                setColorAndFont();
                addComponentsToFrame();
                actionEvent();
        }
        private void setColorAndFont() {
                idLabel.setForeground(java.awt.Color.BLUE);
                passwordLabel.setForeground(java.awt.Color.BLUE);
        }
        public void createWindow(){
                frame = new JFrame();
                frame.setTitle("ADMIN LOGIN FORM");
                frame.setBounds(60, 40, 500, 500);
                frame.getContentPane().setBackground(java.awt.Color.red);
                frame.getContentPane().setLayout(null);
                frame.setVisible(true);
                frame.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
                frame.setResizable(false);
        }
        public void setLocationAndSize() {

                idLabel.setBounds(20, 70, 60, 70);
                passwordLabel.setBounds(20, 220, 100, 70);
                idTextField.setBounds(180, 93, 165, 23);
                passwordField.setBounds(180, 243, 165, 23);
                addButton.setBounds(180, 343, 165, 23);
        }
        public void addComponentsToFrame() {
                frame.add(idLabel);
                frame.add(passwordLabel);
                frame.add(idTextField);
                frame.add(passwordField);
                frame.add(addButton);
                frame.add(message);
        }
        public void actionEvent(){
                addButton.addActionListener(this);
        }
        public static void main(String[] args) {
                new AdminLogin();
        }
        public void actionPerformed(ActionEvent ae)          {
           Connection conn=null;



                try {

                 if(ae.getActionCommand().equals("LOGIN")) {
                        String userName = idTextField.getText();
                        char password[] = passwordField.getPassword();
                        String  pass1=new String(password);
                        String pass=pass1.toString();
                        conn =DBCon.getCon();
                        Statement stmt = conn.createStatement();
                   System.out.println("Create statement");
                   System.out.println("user  "+userName+"  pass  "+pass);
                   ResultSet rs = stmt.executeQuery("select * from login_Admin where
username='"+userName+"'and password='"+pass+"' ");
                   System.out.println(rs);
                   if(rs.next()) {
```

```
                System.out.println("if");
                adminsection sec = new adminsection();
             }
             else {
                JOptionPane.showMessageDialog(null,"invalid id & password");
                }
          }


        } catch (Exception e) {
                e.printStackTrace();
        }


    }
}
```

CODE FOR ADMIN MEMBER OPTION -

```
package adminlogin;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;

public class adminsection extends JFrame implements ActionListener {



    adminsection(){

        JFrame frame = new JFrame("ADMIN SECTION");

        JButton b = new JButton("Add Librarian");

        b.setBounds(110,100,130,50);


  Container c = frame.getContentPane();

        c.setLayout(null);
        c.setBackground(Color.PINK);


        JButton b1 = new JButton("View Librarian");

        b1.setBounds(110,170,130,50);

        JButton b2 = new JButton("Delete Librarian");

        b2.setBounds(110,240,130,50);


        JButton b3 = new JButton("Logout");

        b3.setBounds(110,310,130,50);

        frame.add(b);

        frame.add(b1);

        frame.add(b2);

        frame.add(b3);

        frame.setSize(400, 500);

        frame.setLayout(null);

        frame.setVisible(true);

        frame.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);

        b.addActionListener((ActionListener) this);
        b1.addActionListener((ActionListener) this);
        b2.addActionListener((ActionListener) this);
        b3.addActionListener((ActionListener) this);


    }

    public static void main(String[] args) {
        new adminsection();



    }

public void actionPerformed(ActionEvent ae) {
```

```
                    if(ae.getActionCommand().equals("Add Librarian"))
                    {
                              LibForm frm = new LibForm();


                              }

                    if(ae.getActionCommand().equals("View Librarian"))
                    {
                              Viewlibrarian lib = new Viewlibrarian();
                              //lib.createUI();
                              lib.showTableData();
                               }

                    if(ae.getActionCommand().equals("Delete Librarian"))
                    {
                              DelLib lib = new DelLib();
                              }
                    if(ae.getActionCommand().equals("Logout"))
                    {
                              Logout lg = new Logout();
                                Logout.main("aaa");
                    }
    }
}
```

CODE FOR LIB FORM DETAILS BUTTON-

```
package adminlogin;

import javax.swing.*;

import dbcon.DBCon;

import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class LibForm implements ActionListener {

          JFrame frame;

          String[] gender = {"","Male","Female"};

          JLabel idLabel = new JLabel("ID NO.");

          JLabel nameLabel = new JLabel("NAME");
```

```
        JLabel genderLabel = new JLabel("GENDER");

        JLabel fatherNameLabel = new JLabel("FATHER NAME");

        JLabel passwordLabel = new JLabel("PASSWORD");

        JLabel confirmPasswordLabel = new JLabel("CONFIRM PASSWORD");

        JLabel cityLabel = new JLabel("CITY");

        JLabel emailLabel = new JLabel("EMAIL");

        JLabel phoneLabel = new JLabel("PHONE NO");

        JTextField idTextField = new JTextField();

        JTextField nameTextField = new JTextField();

        JComboBox genderComboBox = new JComboBox(gender);

        JTextField fatherTextField = new JTextField();

        JPasswordField passwordField = new JPasswordField();

        JPasswordField confirmPasswordField = new JPasswordField();

        JTextField cityTextField = new JTextField();

        JTextField emailTextField = new JTextField();

        JTextField phoneTextField = new JTextField();

        JButton addButton = new JButton("ADD");

        JButton resetButton = new JButton("RESET");

    LibForm(){

        createWindow();

        setLocationAndSize();

        addComponentsToFrame();

        actionEvent();

        }
```

```
public void createWindow()
{

        frame = new JFrame();
        frame.setTitle("Librarian From");
        frame.setBounds(90, 40, 500, 700);
        frame.getContentPane().setBackground(Color.LIGHT_GRAY);
        frame.getContentPane().setLayout(null);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
        frame.setResizable(true);
}

public void setLocationAndSize() {


        idLabel.setBounds(20, 20, 40, 70);

        nameLabel.setBounds(20, 70, 40, 70);

        genderLabel.setBounds(20, 120, 80, 70);

        fatherNameLabel.setBounds(20, 170, 100, 70);

        passwordLabel.setBounds(20, 220, 100, 70);

        confirmPasswordLabel.setBounds(20, 270, 140, 70);

        cityLabel.setBounds(20, 320, 100, 70);

        emailLabel.setBounds(20, 370, 100, 70);

        phoneLabel.setBounds(20, 420, 100, 70);

        idTextField.setBounds(180, 43, 165, 23);

        nameTextField.setBounds(180, 93, 165, 23);

        genderComboBox.setBounds(180, 143, 165, 23);

        fatherTextField.setBounds(180,193,165,23);

        passwordField.setBounds(180,243,165,23);

        confirmPasswordField.setBounds(180,293,165,23);

        cityTextField.setBounds(180,343,165,23);
```

```
            emailTextField.setBounds(180,393,165,23);

     phoneTextField.setBounds(180,443,165,23);

     addButton.setBounds(70,550,100,35);

     resetButton.setBounds(220,550,100,35);

}
public void addComponentsToFrame()
{
   frame.add(idLabel);

     frame.add(nameLabel);

     frame.add(genderLabel);

     frame.add(fatherNameLabel);

     frame.add(passwordLabel);

     frame.add(confirmPasswordLabel);

     frame.add(cityLabel);

     frame.add(emailLabel);

     frame.add(phoneLabel);

     frame.add(idTextField);

     frame.add(nameTextField);

     frame.add(genderComboBox);

     frame.add(fatherTextField);

      frame.add(passwordField);

      frame.add(confirmPasswordField);

      frame.add(cityTextField);

      frame.add(emailTextField);

      frame.add(phoneTextField);

      frame.add(addButton);
```

```
    frame.add(resetButton);

}

public void actionEvent()

{
   addButton.addActionListener(this);
   resetButton.addActionListener(this);
}


public void actionPerformed(ActionEvent ae) {
            Connection conn=null;
   if(ae.getSource()==addButton)

   {

           try {

                      conn =DBCon.getCon();
           PreparedStatement ps=conn.prepareStatement("insert into librarian values(?,?,?,?,?,?,?,?,?)");

           //Specifying the values of it's parameter

           ps.setString(1,idTextField.getText());
           ps.setString(2,nameTextField.getText());
         ps.setString(3,genderComboBox.getSelectedItem().toString());
         ps.setString(4,fatherTextField.getText());
         ps.setString(5,passwordField.getText());
         ps.setString(6,confirmPasswordField.getText());
         ps.setString(7,cityTextField.getText());
         ps.setString(8,emailTextField.getText());
         String ph=phoneTextField.getText();
         int phno= Integer.parseInt(ph);
         ps.setInt(9,phno);

         //Checking for the Password match

         if(passwordField.getText().equals(confirmPasswordField.getText()))
         {
            //Executing query

           ps.executeUpdate();

            JOptionPane.showMessageDialog(null,"Librarian Added Successfully");
```

```
      }
      else
      {
        JOptionPane.showMessageDialog(null,"password did not match");
      }

      conn.close();


       } catch (SQLException e1) {

               e1.printStackTrace();
    }


  }
  if(ae.getSource()==resetButton)
  {
    //Clearing Fields

    idTextField.setText("");

    nameTextField.setText("");

       genderComboBox.setSelectedItem("Male");

       fatherTextField.setText("");

       passwordField.setText("");

       confirmPasswordField.setText("");

       cityTextField.setText("");

       emailTextField.setText("");

       phoneTextField.setText("");
  }
}
public static void main(String[] args) {
       new LibForm();
}
}
```

CODE FOR VIEW LIBRAIAN OPTION-

```
package adminlogin;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import dbcon.DBCon;
public class Viewlibrarian  extends JFrame implements ActionListener{

        JFrame frame, frame1;
        JTextField textbox;
        JLabel label;
        JButton button;
        JPanel panel;
        static JTable table;
        String[] columnNames
={"Id","Name","Gender","Father_name","Password","cpassword","City","Email","Phone No"};

        public void createUI()
        {
        frame = new JFrame("Database Search Result");
        frame.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
        frame.setLayout(null);

        button = new JButton("search");
        button.setBounds(120,130,150,20);
        button.addActionListener(this);

        frame.add(button);
        frame.setVisible(true);
        frame.setSize(500, 400);
        }

        public void actionPerformed(ActionEvent ae)
        {
        button = (JButton)ae.getSource();
        System.out.println("Showing Table Data.......");
        showTableData();
        }

        public void showTableData()
        {
        frame1 = new JFrame("VIEW LIBRARIAN");
        frame1.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
        frame1.setLayout(new BorderLayout());

        DefaultTableModel model = new DefaultTableModel();
        model.setColumnIdentifiers(columnNames);
        table = new JTable();
        table.setModel(model);
        table.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
        table.setFillsViewportHeight(true);
        JScrollPane scroll = new JScrollPane(table);
        scroll.setHorizontalScrollBarPolicy(
        JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
        scroll.setVerticalScrollBarPolicy(
        JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);

        String id= "";
        String name= "";
        String  gender= "";
        String father_name = "";
        String password = "";
        String cpassword = "";
        String city = "";
        String email = "";
        String phno = "";
        try
        {

        Connection con = DBCon.getCon();
        String sql = "select * from librarian";
        PreparedStatement ps = con.prepareStatement(sql);
        ResultSet rs = ps.executeQuery();
        int i =0;
        while(rs.next())
        {
        id = rs.getString(1);
        name = rs.getString(2);
        gender = rs.getString(3);
        father_name = rs.getString(4);
        password = rs.getString(5);
         cpassword = rs.getString(6);
         city = rs.getString(7);
        email = rs.getString(8);
        phno= rs.getString(9);
        model.addRow(new Object[]{id , name, gender,
father_name,password,cpassword,city,email,phno});
        i++;
        }
```

```
        if(i <1)
        {
        JOptionPane.showMessageDialog(null, "No Record Found","Error",
        JOptionPane.ERROR_MESSAGE);
        }
        if(i ==1)
        {
        System.out.println(i+" Record Found");
        }
        else
        {
        System.out.println(i+" Records Found");
        }
        }
        catch(Exception ex)
        {
        JOptionPane.showMessageDialog(null, ex.getMessage(),"Error",
        JOptionPane.ERROR_MESSAGE);
        }
        frame1.add(scroll);
        frame1.setVisible(true);
        frame1.setSize(400,300);
        }

        public static void main(String args[])
        {
                Viewlibrarian sr = new Viewlibrarian();
        sr.createUI();
        }
}
```

CODE FOR DELETE LIBRARIAN OPTION-

```
package adminlogin;

import java.awt.event.*;
import javax.swing.*;
import dbcon.DBCon;
import java.awt.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class DelLib implements ActionListener {

  JFrame frame;
```

```
    JLabel idLabel = new JLabel("Enter Id");

    JTextField idTextField = new JTextField();

     JButton delButton = new JButton("Delete");
    DelLib(){
     createWindow();
    setLocationAndSize();
    addComponentsToFrame();
    actionEvent();
    }

    public void createWindow()
    {

             frame = new JFrame();
             frame.setTitle("Delete Librarian");
             frame.setBounds(40, 40, 380, 400);
             frame.getContentPane().setBackground(Color.LIGHT_GRAY);
             frame.getContentPane().setLayout(null);
             frame.setVisible(true);
             frame.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
             frame.setResizable(false);
    }

public void setLocationAndSize() {
             idLabel.setBounds(20, 20, 90, 70);
             idTextField.setBounds(180, 43, 165, 23);
              delButton.setBounds(110,115,100,35);

}

public void addComponentsToFrame()

    {
      frame.add(idLabel);
      frame.add(idTextField);
       frame.add(delButton);
    }

public void actionEvent()

    {
      delButton.addActionListener(this);

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

       new DelLib();
           }
           @Override
           public void actionPerformed(ActionEvent ae) {

                      Connection conn=null;
              if(ae.getSource()==delButton)  {

                      try {
                      conn =DBCon.getCon();

            Statement stmt = conn.createStatement();

                                            System.out.println("Create Statement");

                                            String id = idTextField.getText();

                                            ResultSet rs = stmt.executeQuery("delete from librarian where
id='"+id+"'");

                                            System.out.println(rs);

                                            if(rs.next()) {
                                            System.out.println("if");
                                            JOptionPane.showMessageDialog(null,"Record Deleted Successfully ");
                                       adminsection sec = new adminsection();
                               }


                                    else {
                                            JOptionPane.showMessageDialog(null,"Invalid user id ");
                                    }


                                             conn.close();


                      } catch (SQLException e1) {

                                e1.printStackTrace();
                 }

              }
           }
}
```

CODE FOR LOGOUT BUTTON-

```
package adminlogin;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.Statement;
import javax.swing.event.*;
import dbcon.DBCon;
public class Logout implements ActionListener {

    public static  void main(String ...args) {
        JDialog.setDefaultLookAndFeelDecorated(true);

        int response = JOptionPane.showConfirmDialog(null,"Do you want to exit???","Confirm
Exit",

                JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);

                int confirmed = response;

        if (confirmed == JOptionPane.YES_OPTION) {
            lms s = new lms();
          dispose();
         }
        }



        private static void dispose() {

            new Logout();


    }


        @Override
        public void actionPerformed(ActionEvent ae) {

            Connection conn=null;




            try {
```

```
            if(ae.getActionCommand().equals("LOGOUT")) {
             conn =DBCon.getCon();
                Statement stmt = conn.createStatement();
             System.out.println("Create statement");


        }
            }

                catch (Exception e) {
                    e.printStackTrace();
            }

        }
}
```

CODE FOR LABRARIAN LOGIN OPTION-

```
package adminlogin;
import java.awt.event.*;
import java.sql.*;
import dbcon.DBCon;
import java.awt.*;
import javax.swing.*;
public class LibrarianLogin extends JFrame  implements ActionListener{
JFrame frame;

    JLabel idLabel = new JLabel("Enter ID:");
    JLabel passwordLabel = new JLabel("Enter Password:");
    JLabel message = new JLabel();
    JTextField idTextField = new JTextField();
    JPasswordField passwordField = new JPasswordField();
    JButton addButton = new JButton("LOGIN");
    LibrarianLogin(){

        createWindow();
        setLocationAndSize();
        setColorAndFont();
        addComponentsToFrame();
        actionEvent();
    }

    private void setColorAndFont() {
        idLabel.setForeground(java.awt.Color.BLUE);
        passwordLabel.setForeground(java.awt.Color.BLUE);


    }
```

```
public void createWindow()
{
        frame = new JFrame();
        frame.setTitle("LIBRARIAN LOGIN FORM");
        frame.setBounds(60, 40, 500, 500);
        frame.getContentPane().setBackground(java.awt.Color.RED);
        frame.getContentPane().setLayout(null);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
        frame.setResizable(false);
}

public void setLocationAndSize() {

        idLabel.setBounds(20, 70, 60, 70);
        passwordLabel.setBounds(20, 220, 100, 70);
        idTextField.setBounds(180, 93, 165, 23);
        passwordField.setBounds(180, 243, 165, 23);
        addButton.setBounds(180, 343, 165, 23);
}

public void addComponentsToFrame() {

        frame.add(idLabel);
        frame.add(passwordLabel);
        frame.add(idTextField);
        frame.add(passwordField);
        frame.add(addButton);
        frame.add(message);



}
public void actionEvent()
{
        addButton.addActionListener(this);
        }


public static void main(String[] args) {

        new LibrarianLogin();


}

public void actionPerformed(ActionEvent ae) {
```

```
                Connection conn=null;

                try {

                        if(ae.getActionCommand().equals("LOGIN")) {

                                String userName = idTextField.getText();
                                char password[] = passwordField.getPassword();
                                String pass1 = new String(password);
                                String pass = pass1.toString();
                                conn = DBCon.getCon();
                                Statement stmt = conn.createStatement();

                                System.out.println("Create Statement");

                                System.out.println("user"+userName+"pass"+pass);

                                 ResultSet rs = stmt.executeQuery("select * from librarian where
id='"+userName+"'and password='"+pass+"' ");
                                  System.out.println(rs);
                                  if(rs.next()) {
                                        System.out.println("if");
                                        librariansection sec = new librariansection();
                                  }
                                  else {
                                        JOptionPane.showMessageDialog(null,"Invalid user id &
password");
                                        }
                                 }


                                } catch (Exception e) {
                                        e.printStackTrace();
                                }
                       }

                /*if(ae.getActionCommand().equals("LOGIN"))
                {
                       librariansection sec = new librariansection();


                }

                String userName = idTextField.getText();
                String password = passwordField.getText();

                if(userName.equals("lib") && password.equals("lib123")) {
                        message.setText("<html><font color='green'> SUCCESSFULLY LOGIN
</font></html>");

                }
                else {
                        message.setText("<html><font color='blue'>INVALID USER...</font></html>");
                }*/



}
```

CODE FOR ADD BOOK-

```
package adminlogin;
import javax.swing.*;
import dbcon.DBCon;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class Add_Books implements ActionListener {
JFrame frame;

        JLabel codeLabel = new JLabel("Book_Code");
        JLabel nameLabel = new JLabel("Book_Name");
        JLabel authorLabel = new JLabel("Author");
        JLabel publisherLabel = new JLabel("Publisher");
        JLabel stockLabel = new JLabel("Stock");
        JLabel rackLabel = new JLabel("Rack no.");
        JTextField codeTextField = new JTextField();
        JTextField nameTextField = new JTextField();
        JTextField authorTextField = new JTextField();
        JTextField publisherTextField = new JTextField();
        JTextField stockTextField = new JTextField();
        JTextField rackTextField = new JTextField();
        JButton addButton = new JButton("ADD");
        JButton resetButton = new JButton("RESET");
               Add_Books(){
                createWindow();
                setLocationAndSize();
                addComponentsToFrame();
                actionEvent();
        }
        public void createWindow()
        {
                frame = new JFrame();
                frame.setTitle("ADD BOOKS");
                frame.setBounds(40, 40, 380, 600);
                frame.getContentPane().setBackground(Color.GRAY);
                frame.getContentPane().setLayout(null);
                frame.setVisible(true);
                frame.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
                frame.setResizable(true);
        }
        public void setLocationAndSize() {
                codeLabel.setBounds(20, 20, 90, 70);
                nameLabel.setBounds(20, 70, 80, 70);
                authorLabel.setBounds(20, 120, 100, 70);
                publisherLabel.setBounds(20, 170, 100, 70);
                stockLabel.setBounds(20, 220, 140, 70);
                rackLabel.setBounds(20, 270, 100, 70);
                codeTextField.setBounds(180, 43, 165, 23);
                nameTextField.setBounds(180, 93, 165, 23);
                authorTextField.setBounds(180,143,165,23);
                publisherTextField.setBounds(180,193,165,23);
                stockTextField.setBounds(180,243,165,23);
                rackTextField.setBounds(180,293,165,23);

          addButton.setBounds(70,500,100,35);
           resetButton.setBounds(220,500,100,35);
          }
        public void addComponentsToFrame()
        {
          frame.add(codeLabel);
          frame.add(nameLabel);
          frame.add(authorLabel);
          frame.add(publisherLabel);
          frame.add(stockLabel);
          frame.add(rackLabel);
          frame.add(codeTextField);
           frame.add(nameTextField);
           frame.add(authorTextField);
           frame.add(publisherTextField);
           frame.add(stockTextField);
          frame.add(rackTextField);
          frame.add(addButton);
          frame.add(resetButton);
         }
         public void actionEvent()

        {
         addButton.addActionListener(this);
          resetButton.addActionListener(this);
        }

        public void actionPerformed(ActionEvent ae) {
                Connection conn=null;
```

```
                if(ae.getSource()==addButton)

                {
                    try {

                             conn =DBCon.getCon();
                             PreparedStatement ps=conn.prepareStatement("insert into add_books
         values(?,?,?,?,?,?)");
                   ps.setString(1,codeTextField.getText());
                   ps.setString(2,nameTextField.getText());
                   ps.setString(3,authorTextField.getText());
                   ps.setString(4,publisherTextField.getText());
                  // ps.setString(5,stockTextField.getText());
                  // ps.setString(6,rackTextField.getText());

                   String st=stockTextField.getText();
                   int stno= Integer.parseInt(st);
                   ps.setInt(5,stno);

                   String rt=rackTextField.getText();
                   int rtno= Integer.parseInt(rt);
                   ps.setInt(6,rtno);

                   Object obj = ae.getSource();

                   if(obj == addButton ) {

                     ps.executeUpdate();

                      JOptionPane.showMessageDialog(null,"Librarian Added Book Successfully");

                   }
                   else
                   {
                      JOptionPane.showMessageDialog(null,"Book Added Not Successfully");
                   }

                   conn.close();
                     } catch (SQLException e1) {
                     e1.printStackTrace();
                  }
                }
                if(ae.getSource()==resetButton)
                {
                     codeTextField.setText("");
                     nameTextField.setText("");
                     authorTextField.setText("");
                     publisherTextField.setText("");
                stockTextField.setText("");
                rackTextField.setText("");

          }
              }
          public static void main(String[] args) {
                new Add_Books();
        }

}
```

CODE FOR VIEW BOOK-

```
package adminlogin;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import dbcon.DBCon;
public class Viewbooks extends JFrame implements ActionListener {
        JFrame frame, frame1;
        JTextField textbox;
        JLabel label;
        JButton button;
        JPanel panel;
        static JTable table;
        String[] columnNames ={"Code","Name","Author","Publisher","Stock","Rack"};

        public void createUI()
        {
        frame = new JFrame("View Books");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);

        button = new JButton("search");
        button.setBounds(120,130,150,20);
        button.addActionListener(this);
        frame.add(button);
        frame.setVisible(true);
        frame.setSize(500, 400);
        }

        public void actionPerformed(ActionEvent ae)
        {
```

```
button = (JButton)ae.getSource();
System.out.println("Showing Table Data.......");
showTableData();
}

public void showTableData()
{

        frame1 = new JFrame("VIEW BOOKS");
        frame1.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
        frame1.setLayout(new BorderLayout());

        DefaultTableModel model = new DefaultTableModel();
        model.setColumnIdentifiers(columnNames);
        table = new JTable();
        table.setModel(model);
        table.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
        table.setFillsViewportHeight(true);
        JScrollPane scroll = new JScrollPane(table);
        scroll.setHorizontalScrollBarPolicy(
        JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
        scroll.setVerticalScrollBarPolicy(
        JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);

        String code= "";
        String name= "";
        String  author= "";
        String publisher = "";
        String stock = "";
        String rack = "";

        try
        {
         Connection con = DBCon.getCon();
        String sql = "select * from add_books";
        PreparedStatement ps = con.prepareStatement(sql);
        ResultSet rs = ps.executeQuery();
        int i =0;
        while(rs.next())
        {
                code = rs.getString(1);
                name = rs.getString(2);
                author = rs.getString(3);
                publisher = rs.getString(4);
                stock = rs.getString(5);
                rack = rs.getString(6);
                model.addRow(new Object[]{code , name, author, publisher,stock,rack});
                i++;
```

```
            }
    if(i <1)
    {
    JOptionPane.showMessageDialog(null, "No Record Found","Error",
    JOptionPane.ERROR_MESSAGE);
    }
    if(i ==1)
    {
    System.out.println(i+" Record Found");
    }
    else
    {
    System.out.println(i+" Records Found");
    }
    }
    catch(Exception ex)
    {
    JOptionPane.showMessageDialog(null, ex.getMessage(),"Error",
    JOptionPane.ERROR_MESSAGE);
    }
    frame1.add(scroll);
    frame1.setVisible(true);
    frame1.setSize(400,300);
    }

    public static void main(String args[])
    {
            Viewbooks sr = new Viewbooks();
    sr.createUI();
    }

    }
```

CODE FOR ISSUE BOOK-

```
package adminlogin;
import javax.swing.*;
import dbcon.DBCon;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
public class Issue_Books implements ActionListener {
        JFrame frame;
         JLabel codeLabel = new JLabel("Book_Code");
          JLabel idLabel = new JLabel("Student_Id");
         JLabel nameLabel = new JLabel("Student_Name");
         JLabel contactLabel = new JLabel("Student_Contact");
```

```
            JLabel issueLabel = new JLabel("Issued Date");
            JLabel reissueLabel = new JLabel("Reissued Date");
            JTextField codeTextField = new JTextField();
            JTextField idTextField = new JTextField();
            JTextField nameTextField = new JTextField();
            JTextField contactTextField = new JTextField();
            JTextField issueTextField = new JTextField();
            JTextField reissueTextField = new JTextField();
            JButton addButton = new JButton("Issue Book");
            JButton resetButton = new JButton("RESET");

    Issue_Books(){
                createWindow();
                setLocationAndSize();
                addComponentsToFrame();
                actionEvent();
                        }

            public void createWindow()
            {
                frame = new JFrame();
                frame.setTitle("ISSUE BOOKS");
                frame.setBounds(40, 40, 380, 600);
                frame.getContentPane().setBackground(Color.LIGHT_GRAY);
                frame.getContentPane().setLayout(null);
                frame.setVisible(true);
                frame.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
                frame.setResizable(false);
            }
            public void setLocationAndSize() {
                codeLabel.setBounds(20, 20, 90, 70);
                idLabel.setBounds(20, 70, 80, 70);
                nameLabel.setBounds(20, 120, 100, 70);
                contactLabel.setBounds(20, 170, 100, 70);
                issueLabel.setBounds(20, 220, 140, 70);
                reissueLabel.setBounds(20, 270, 100, 70);
                codeTextField.setBounds(180, 43, 165, 23);
                idTextField.setBounds(180, 93, 165, 23);
                nameTextField.setBounds(180,143,165,23);
                contactTextField.setBounds(180,193,165,23);
                issueTextField.setBounds(180,243,165,23);
                reissueTextField.setBounds(180,293,165,23);

              addButton.setBounds(70,500,100,35);
               resetButton.setBounds(220,500,100,35);
             }
            public void addComponentsToFrame()
            {
```

```
        frame.add(codeLabel);
        frame.add(idLabel);
        frame.add(nameLabel);
        frame.add(contactLabel);
        frame.add(issueLabel);
        frame.add(reissueLabel);
        frame.add(codeTextField);
         frame.add(idTextField);
         frame.add(nameTextField);
        frame.add(contactTextField);
         frame.add(issueTextField);
         frame.add(reissueTextField);
          frame.add(addButton);
          frame.add(resetButton);
         }
        public void actionEvent()

       {
          addButton.addActionListener(this);
          resetButton.addActionListener(this);
       }
       @Override
       public void actionPerformed(ActionEvent ae) {

                  Connection conn=null;
          if(ae.getSource()==addButton)

          {
                  try {
                  conn =DBCon.getCon();
                  PreparedStatement ps=conn.prepareStatement("insert into issue_books
values(?,?,?,?,?,?)");

                              ps.setString(1,codeTextField.getText());
                              ps.setString(2,idTextField.getText());
                              ps.setString(3,nameTextField.getText());
                              String st=contactTextField.getText();
                int ctno= Integer.parseInt(st);
                ps.setInt(4,ctno);

                ps.setString(5,issueTextField.getText());

                ps.setString(6,reissueTextField.getText());
                  Object obj = ae.getSource();

                if(obj == addButton ) {

                  ps.executeUpdate();
        JOptionPane.showMessageDialog(null,"Book Issued Successfully");

      }
      else
      {
        JOptionPane.showMessageDialog(null,"Book Issued Not Successfully");
      }

      conn.close();


       } catch (SQLException e1) {

              e1.printStackTrace();
    }
  }

  if(ae.getSource()==resetButton)
  {

       codeTextField.setText("");

       idTextField.setText("");

       nameTextField.setText("");

       contactTextField.setText("");

       issueTextField.setText("");

       reissueTextField.setText("");

  }
      }

      public static void main(String[] args) {
              new Issue_Books();

       }


}
```

CODE FOR VIEW ISSUE BOOK-

```
package adminlogin;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import dbcon.DBCon;

public class ViewIsuBook extends JFrame implements ActionListener {

        JFrame frame, frame1;
        JTextField textbox;
        JLabel label;
        JButton button;
        JPanel panel;
        static JTable table;
        String[] columnNames ={"Code","Id","Name","Contact","Issue","Reissue"};

        public void createUI()
        {
        frame = new JFrame("View Issued Books");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);

        button = new JButton("search");
        button.setBounds(120,130,150,20);
        button.addActionListener(this);

        frame.add(button);
        frame.setVisible(true);
        frame.setSize(500, 400);
        }

        public void actionPerformed(ActionEvent ae)
        {
        button = (JButton)ae.getSource();
        System.out.println("Showing Table Data.......");
        showTableData();
        }

        public void showTableData()
        {
```

```
frame1 = new JFrame("VIEW ISSUED BOOKS");
frame1.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
frame1.setLayout(new BorderLayout());

DefaultTableModel model = new DefaultTableModel();
model.setColumnIdentifiers(columnNames);
table = new JTable();
table.setModel(model);
table.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
table.setFillsViewportHeight(true);
JScrollPane scroll = new JScrollPane(table);
scroll.setHorizontalScrollBarPolicy(
JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
scroll.setVerticalScrollBarPolicy(
JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);


String code= "";
String id= "";
String  name= "";
String contact = "";
String issue = "";
String reissue = "";

try {
        Connection con = DBCon.getCon();
        String sql = "select * from issue_books";
        PreparedStatement ps = con.prepareStatement(sql);
        ResultSet rs = ps.executeQuery();
        int i =0;
        while(rs.next())
        {

                code = rs.getString(1);
                id = rs.getString(2);
                name = rs.getString(3);
                contact = rs.getString(4);
                issue = rs.getString(5);
                reissue = rs.getString(6);

                model.addRow(new Object[]{code , id, name, contact, issue,reissue});
        }

        if(i <1)
        {
        JOptionPane.showMessageDialog(null, "No Record Found","Error",
        JOptionPane.ERROR_MESSAGE);
        }
```

```
                                        if(i ==1)
                                        {
                                        System.out.println(i+" Record Found");
                                        }
                                        else
                                        {
                                        System.out.println(i+" Records Found");
                                        }
                                        }
                                        catch(Exception ex)
                                        {
                                        JOptionPane.showMessageDialog(null, ex.getMessage(),"Error",
                                        JOptionPane.ERROR_MESSAGE);
                                        }
                                        frame1.add(scroll);
                                        frame1.setVisible(true);
                                        frame1.setSize(400,300);
                                        }

        public static void main(String[] args) {
                ViewIsuBook vib = new  ViewIsuBook ();
                vib.createUI();

        }
}
```

CODE FOR RETURN BOOK-

```
package adminlogin;
import javax.swing.*;
import dbcon.DBCon;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;


public class Returnbooks extends JFrame implements ActionListener {
   JFrame frame;
   JLabel codeLabel = new JLabel("Book_Code");
   JLabel idLabel = new JLabel("Student_Id");
    JTextField codeTextField = new JTextField();
     JTextField idTextField = new JTextField();
     JButton addButton = new JButton("Return Book");
     JButton bkButton = new JButton("Back");
Returnbooks(){
        createWindow();
```

```
            setLocationAndSize();
            addComponentsToFrame();
            actionEvent();
            }
    public void createWindow()
    {
            frame = new JFrame();
            frame.setTitle("RETURN BOOKS");
            frame.setBounds(40, 40, 380, 450);
            frame.getContentPane().setBackground(Color.LIGHT_GRAY);
            frame.getContentPane().setLayout(null);
            frame.setVisible(true);
            frame.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
            frame.setResizable(false);
    }

    public void setLocationAndSize() {
            codeLabel.setForeground (Color.red);
            codeLabel.setBounds (15, 15, 100, 20);
            idLabel.setForeground (Color.red);
            idLabel.setBounds (15, 45, 100, 20);
            codeTextField.setHorizontalAlignment (JTextField.RIGHT);
            codeTextField.setBounds (120, 15, 175, 25);
            idTextField.setHorizontalAlignment (JTextField.RIGHT);
            idTextField.setEnabled (true);
            idTextField.setBounds (120, 45, 175, 25);
            addButton.setBounds (25, 175, 125, 25);
            addButton.addActionListener (this);
             bkButton.setBounds (165, 175, 125, 25);
             bkButton.addActionListener (this);
     }
    public void addComponentsToFrame()

    {
       frame.add(codeLabel);

       frame.add(idLabel);

       frame.add(codeTextField);

       frame.add(idTextField);

       frame.add(addButton);

        frame.add(bkButton);

    }
```

```
        public void actionEvent()

        {

          addButton.addActionListener(this);
          bkButton.addActionListener(this);
        }





public static void main(String[] args) {
                new Returnbooks();

        }

        @Override
        public void actionPerformed(ActionEvent ae) {

                Connection conn=null;
          if(ae.getSource()==addButton)

          {
                try {

                        conn =DBCon.getCon();

                        PreparedStatement ps=conn.prepareStatement("");

                        ps.setString(1,codeTextField.getText());

                        ps.setString(2,idTextField.getText());

                         Object obj = ae.getSource();

                       if(obj == addButton ) {

                        ps.executeUpdate();

                          JOptionPane.showMessageDialog(null,"Book Returned Successfully");

                       }
                       else
                       {
                          JOptionPane.showMessageDialog(null,"Sorry Unable To Return Book");
                       }
```

```
                        conn.close();
                        } catch (SQLException e1) {
                          e1.printStackTrace();
                         }
                       }

               if(ae.getSource()==bkButton)
               {
                     setVisible (false);
                             dispose();
               }
           }
}
```

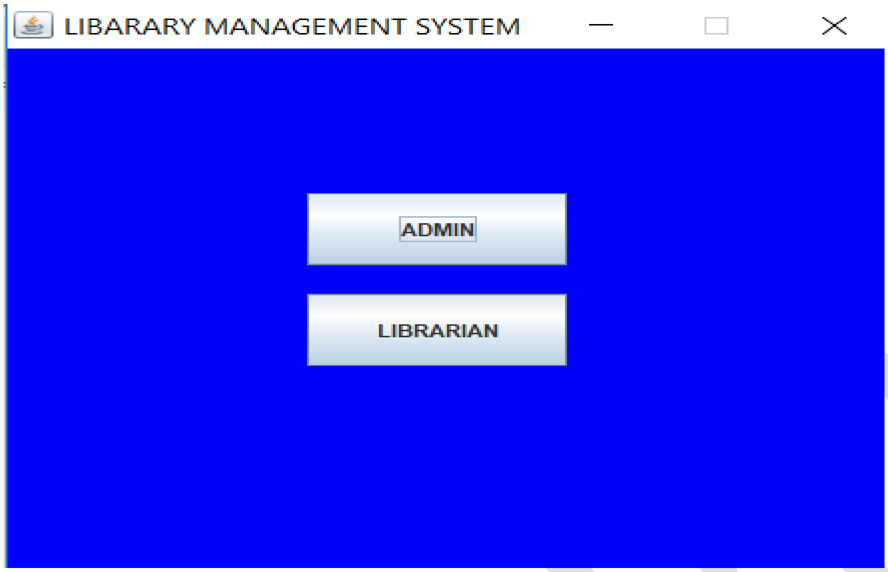
CODE FOR DATABASE CONNECTION-

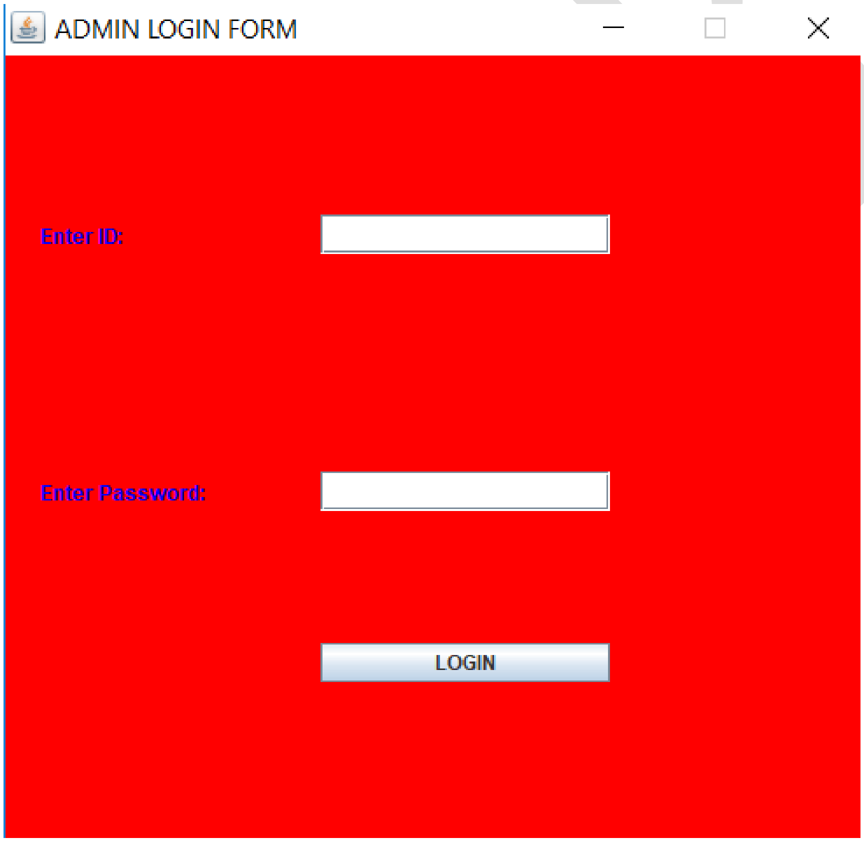
```
package dbcon;
import java.sql.*;
public class DBCon {
        private static Connection conn=null;
     public static Connection getCon() {

       try {
          Class.forName("oracle.jdbc.driver.OracleDriver");
                  System.out.println("Load the Driver class");

                  conn =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
                  System.out.println("Get Connection"+conn);
          // conn.close();
     }catch(ClassNotFoundException e){
             System.out.println("MyClassNotFoundException " +e);
                e.printStackTrace();
         }
         catch(SQLException e){
             System.out.println("SQLException " +e);
                e.printStackTrace();
         }
      catch(Exception e){
             System.out.println("Exception " +e);
                e.printStackTrace();
         }


        return conn;
     }

        public static void main(String[] args) {

        }

}
```

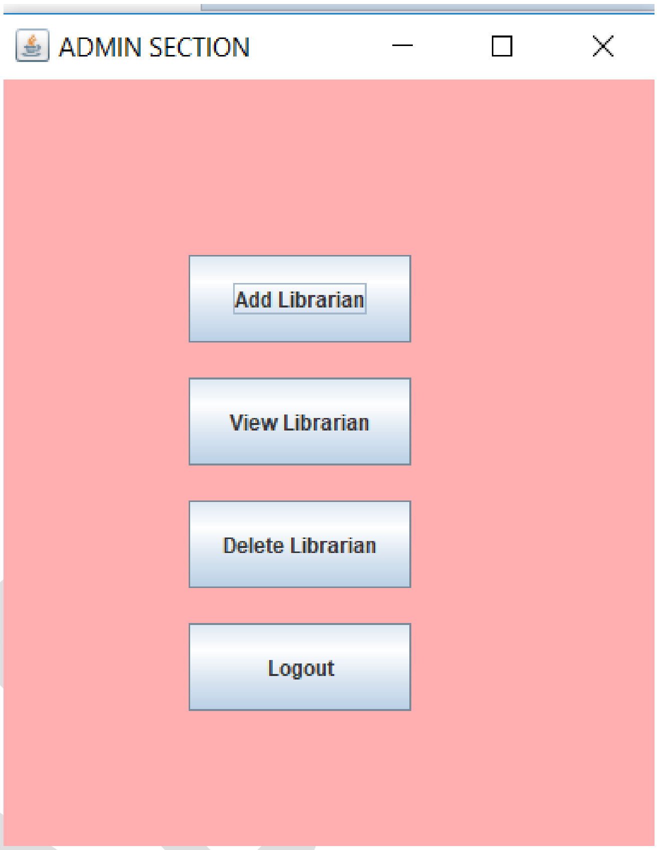
DATABASE SQL-

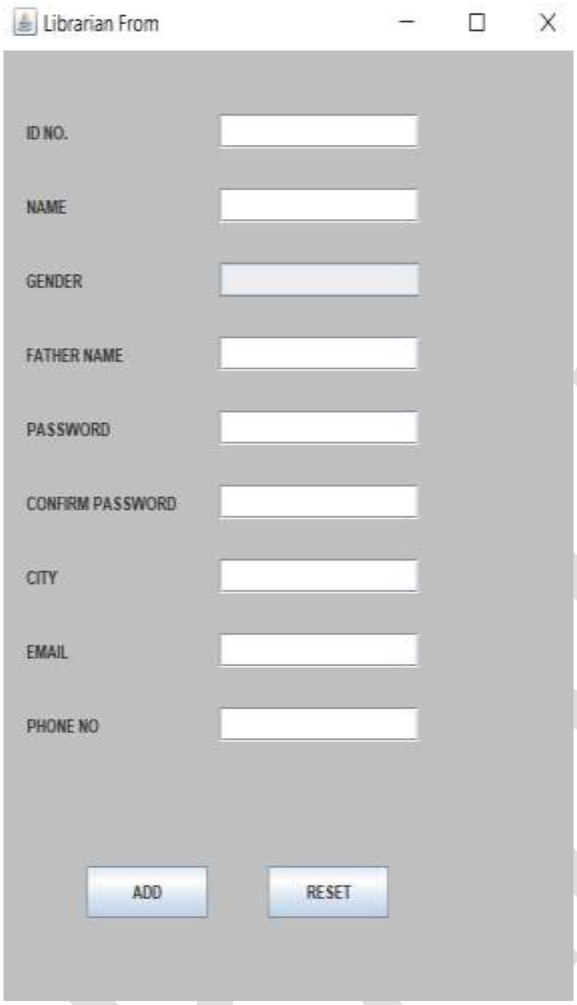
```
create table login_Admin(username varchar2(15),password varchar2(10));

insert into login_Admin values('Iqra',iqra');

desc login_Admin;

select * from login_Admin;

create table librarian(id varchar2(6),name varchar2(25),gender varchar2(6),father_name varchar2(25),password varchar2(10),confirm_password varchar2(10),city varchar2(25),email varchar2(25),phoneno number(12));

create table login_librarian(username varchar2(15),password varchar2(10));

insert into login_librarian values('lib','lib12');

select * from login_librarian;

create table add_books(code varchar2(6),name varchar2(25),author varchar2(25),publisher varchar2(25),stock number(3),rack number(2));

select * from add_books;

create table issue_books(code varchar2(6),id varchar2(5),name varchar2(25),contactno number(12),issue date,reissue date);
```

PREVIEW:-


LIBARARY MANAGEMENT SYSTEM
ADMIN
LIBRARIAN


ADMIN LOGIN FORM
Enter ID:
Enter Password:
LOGIN


ADMIN SECTION
Add Librarian
View Librarian
Delete Librarian
Logout



Librarian From

ID NO.

NAME

GENDER

FATHER NAME

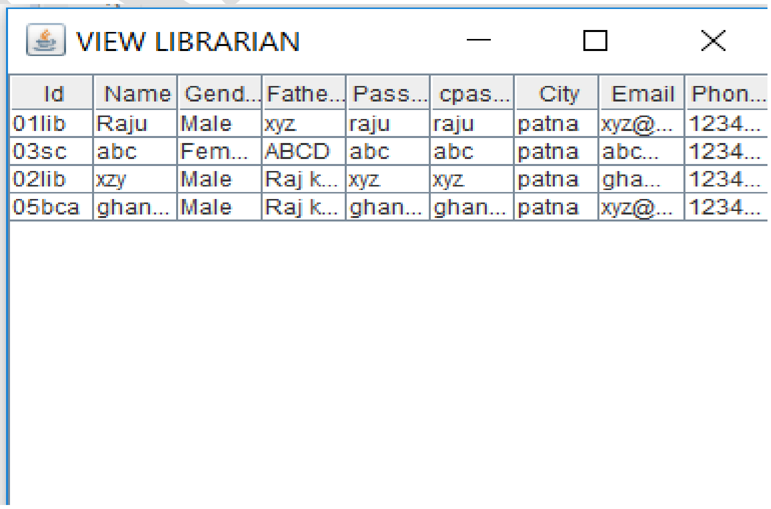
PASSWORD

CONFIRM PASSWORD

CITY

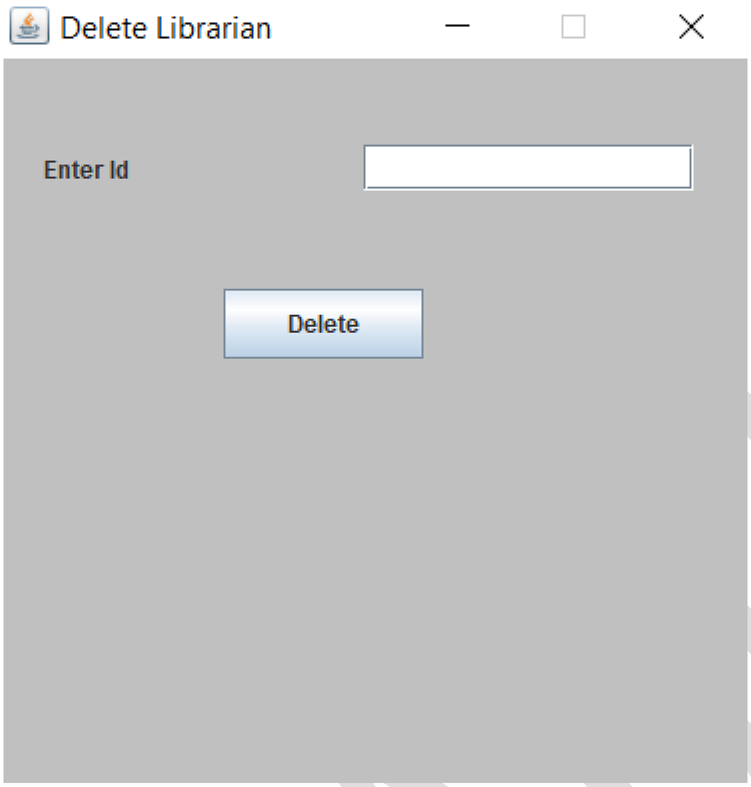
EMAIL

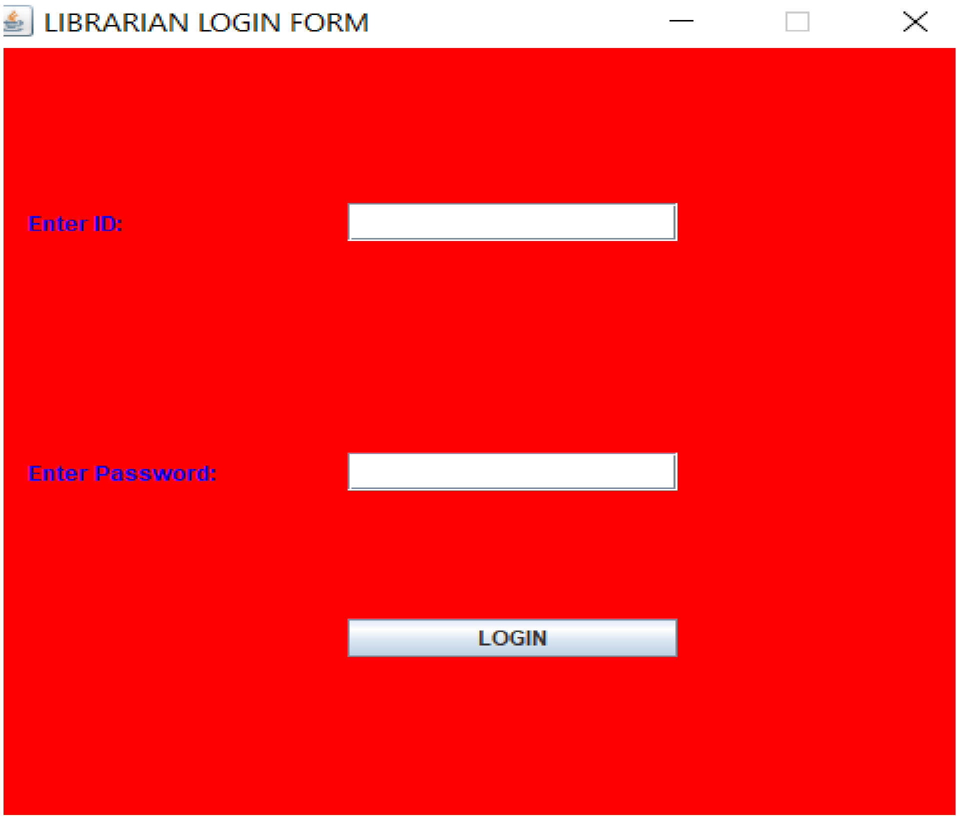
PHONE NO

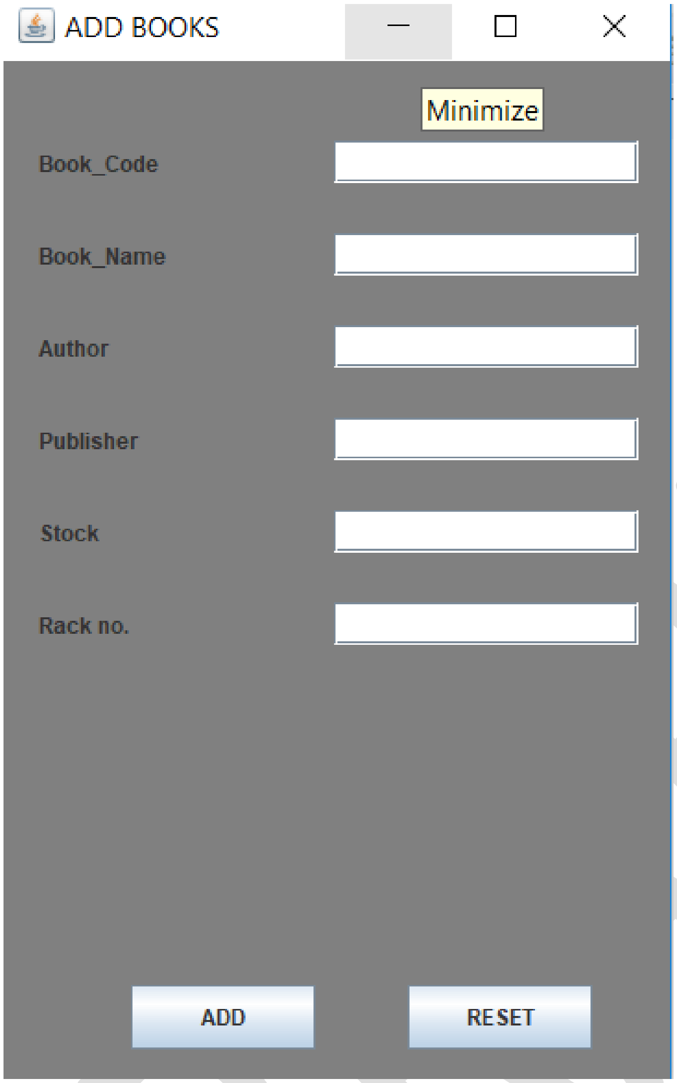


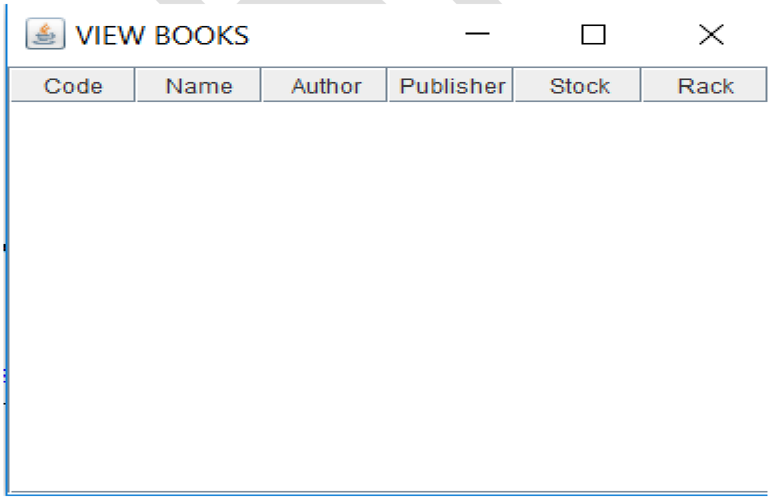
VIEW LIBRARIAN

Id	Name	Gend...	Fathe...	Pass...	cpas...	City	Email	Phon...
01lib	Raju	Male	xyz	raju	raju	patna	xyz@...	1234...
03sc	abc	Fem...	ABCD	abc	abc	patna	abc...	1234...
02lib	xzy	Male	Raj k...	xyz	xyz	patna	gha...	1234...
05bca	ghan...	Male	Raj k...	ghan...	ghan...	patna	xyz@...	1234...

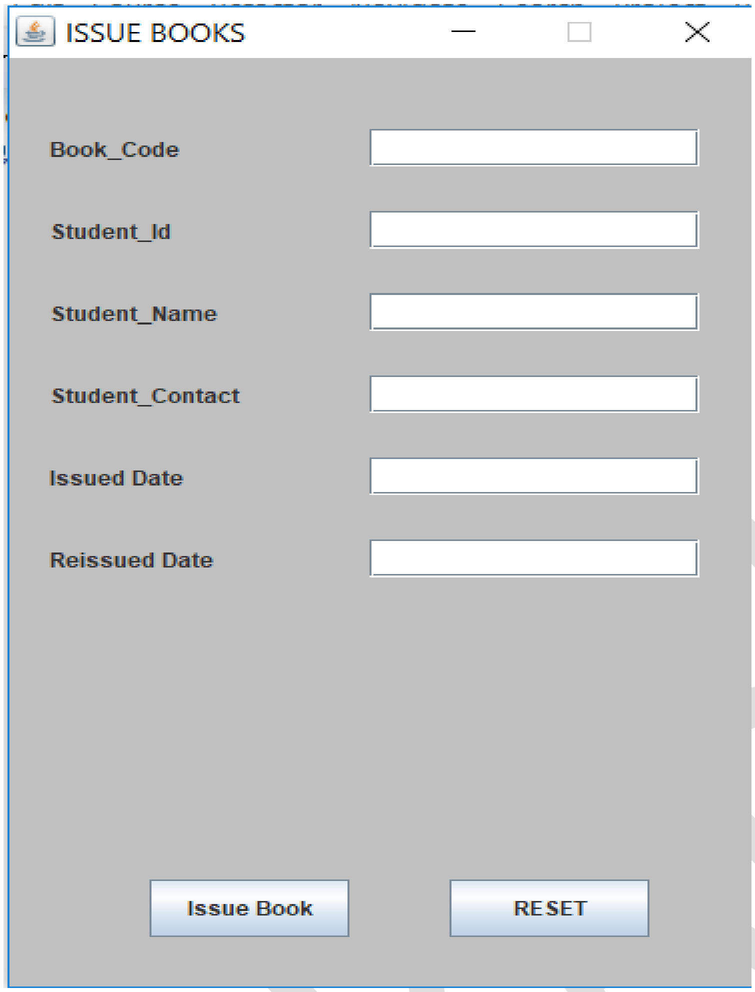

Delete Librarian
Enter Id
Delete


LIBRARIAN LOGIN FORM
Enter ID:
Enter Password:
LOGIN





Code	Name	Author	Publisher	Stock	Rack



ISSUE BOOKS

Book_Code

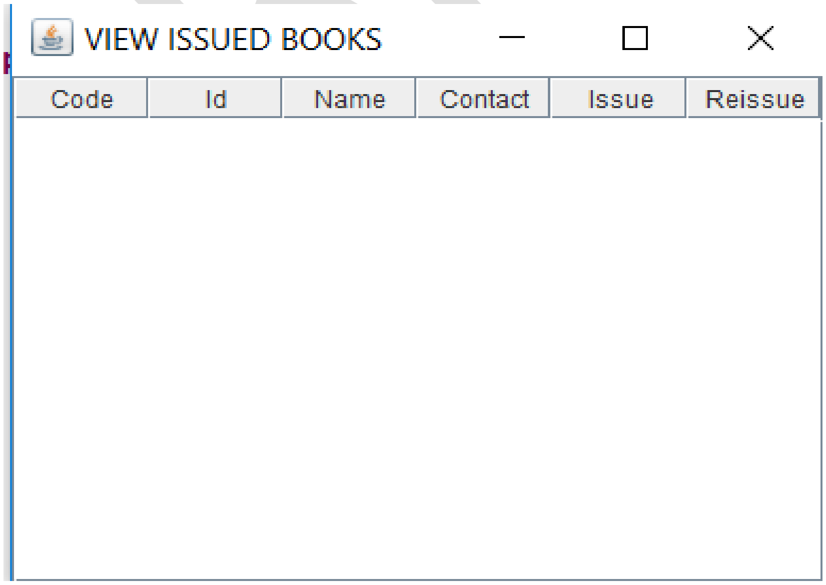
Student_Id

Student_Name

Student_Contact

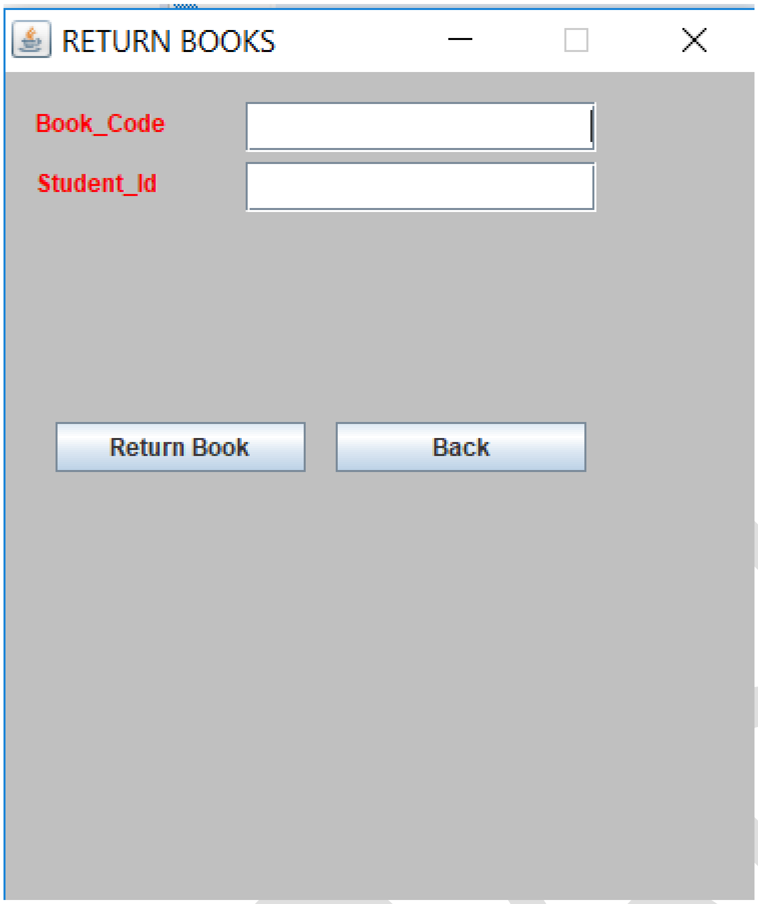
Issued Date

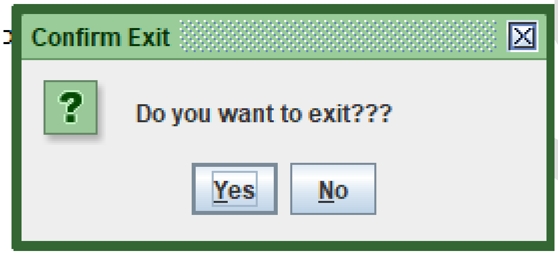
Reissued Date



VIEW ISSUED BOOKS

Code	Id	Name	Contact	Issue	Reissue





Do you want to exit???

BIBLIOGRAPHY

1. Software Engineering	Roger S. Pressman
2. Java	K.A.Mugal
3. Database System Concepts	Henry F. Korth
4. Structured Query Language	Ivan Bayros

5. www.java.sun.com
6. www.google.com