Abstract:

Library management system is a project which aims in developing a computerized system to maintain all the daily work of library .This project has many features which are generally like facility of user login and a facility of teachers login. It also has a facility of admin login through which the admin can monitor the whole system. The librarian after logging into his account i.e. admin account can
generate various reports such as student report , issue report, teacher report and book report.

Overall this project of mine is being developed to help the students as well as staff of library to maintain the library in the best way possible and also to reduce the human efforts.

Introduction :

Requirements Analysis :

List of Tables :

- Librarian
- Book
- Students
- Issue
- Breturn

List of Attributes and their Domain Types :

1. Librarian :
    a. Login ID : id – varchar(20)
    b. Password : password – varchar(20)
2. Book :
    a. Book ID : bookid – varchar(20)
    b. Title : Title – varchar(200)
    c. Author : Author – varchar(200)
3. Students :
    a. Name : name – varchar(100)
    b. Student ID : rollno – varchar(20)
    c. Branch : Branch – varchar(20)
    d. Course : Course – varchar(20)
4. Issue :
    a. Issued date : issue_date – date
    b. Book ID : bookid – varchar(20)
    c. Student ID : rollno – varchar(20)

5. Return :

    a. Returned date : breturn_date – date
    b. Book ID : bookid – varchar(20)
    c. Student ID : rollno – varchar(20)

THROUGH THE PROJECT:

This project helps to store data in an efficient way and it can be achieved through various SQL commands and we can also store this for any future use and also we can save our data in a many different areas so we cannot lost all the data at once. This project stores details about books and students in database so that whenever information is required, we can check immediately.

ARCHITECTURE AND TECHNOLOGY USED:

Software Used:

Java Eclipse, MySQL V8.0.29, Java SE version 17.

Java SWING:

SWING is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs. Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT.

Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing also provides several advanced components such as tabbed panel, scroll panes, trees, tables and lists.

Java-MySQL Connectivity using JDBC:

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

```
try{

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

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/college_library","root","kiran08");
```

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

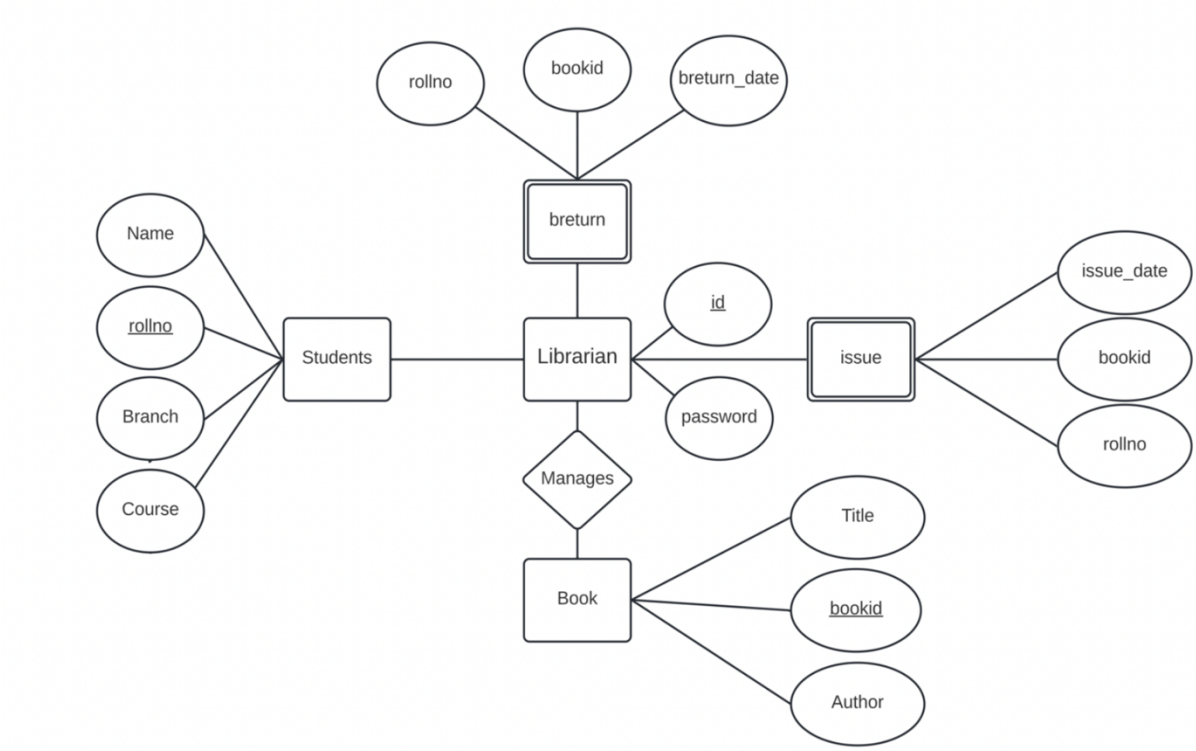
Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

Table Created in MySQL for above mentioned purpose is as:

```
mysql> create table Librarian(id varchar(20), password varchar(20) NOT NULL, primary key(id));

Query OK, 0 rows affected (0.01 sec)

mysql> create table students( name varchar(200) NOT NULL, rollno varchar(20), Branch varchar(20) NOT NULL, Semester varchar(20) NOT NULL, primary key(rollno));

Query OK, 0 rows affected (0.01 sec)

mysql> create table book( bookid varchar(20) PRIMARY KEY, Title varchar(200) NOT NULL, Author varchar(20));

Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql>create table issue( issue_date date not null, bookid varchar(20), rollno varchar(20), foreign key(bookid) references book(bookid) ON DELETE CASCADE, foreign key(rollno) references students(rollno) ON DELETE CASCADE);

Query OK, 0 rows affected (0.02 sec)

mysql> create table breturn(breturn_date date not null, bookid varchar(20) not null, rollno varchar(20) not null, foreign key (bookid) references book(bookid) ON DELETE CASCADE, foreign key (rollno) references students(rollno) ON DELETE CASCADE);

Query OK, 0 rows affected (0.01 sec)
```

Design

ER Diagram:



Database Design:

```
mysql> show tables;
+--------------------------+
| Tables_in_college_library |
+--------------------------+
| book                     |
| breturn                  |
| issue                    |
| Librarian                |
| students                 |
+--------------------------+
5 rows in set (0.01 sec)

mysql> desc Librarian;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| id       | varchar(20) | NO   | PRI | NULL    |       |
| password | varchar(20) | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

```
mysql> desc book;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| bookid | varchar(20) | NO   | PRI | NULL    |       |
| Title  | varchar(200)| YES  |     | NULL    |       |
| Author | varchar(200)| YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)

mysql> desc students;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| name   | varchar(100)| YES  |     | NULL    |       |
| rollno | varchar(20) | NO   | PRI | NULL    |       |
| Branch | varchar(20) | NO   |     | NULL    |       |
| Course | varchar(100)| YES  |     | B.E     |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> desc issue;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| issue_date | date        | NO   |     | NULL    |       |
| bookid     | varchar(20) | YES  | MUL | NULL    |       |
| rollno     | varchar(20) | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> desc breturn;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| breturn_date | date        | NO   |     | NULL    |       |
| bookid       | varchar(20) | NO   | MUL | NULL    |       |
| rollno       | varchar(20) | NO   | MUL | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Implementation

Program:

AddBook.java:

```
import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import java.awt.Color;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.util.regex.Pattern;

import java.awt.event.ActionEvent;


public class AddBook extends JFrame {

	private static final long serialVersionUID = 1L;

	static AddBook frame;

	private JPanel contentPane;

	private JTextField TextField;

	private JTextField TextField_1;

	private JTextField TextField_2;

	/**
```

```
     * Launch the application.
     */
    public static void main(String[] args) {
            EventQueue.invokeLater(new Runnable() {
                    public void run() {
                            try {
                                    frame = new AddBook();
                                    frame.setTitle("Book Info.");
                                    frame.setVisible(true);
                            } catch (Exception e) {
                                    e.printStackTrace();
                            }
                    }
            });
    }
    public AddBook() {
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(110, 95, 650, 400);
setResizable(false);
contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

TextField = new JTextField();
```

```
TextField.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField.setBounds(290, 125, 326, 40);
contentPane.add(TextField);
TextField.setColumns(10);

TextField_1 = new JTextField();
TextField_1.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField_1.setBounds(290, 175, 326, 40);
contentPane.add(TextField_1);
TextField_1.setColumns(10);

TextField_2 = new JTextField();
TextField_2.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField_2.setBounds(290, 225, 326, 40);
contentPane.add(TextField_2);
TextField_2.setColumns(10);

TextField.setText("");
        TextField_1.setText("");
        TextField_2.setText("");

        JLabel lblAddBooks = new JLabel("Add Books");
lblAddBooks.setFont(new Font("Courier New", Font.BOLD, 36));
lblAddBooks.setForeground(Color.BLACK);
lblAddBooks.setBounds(275, 27, 200, 40);
```

```
contentPane.add(lblAddBooks);


		JLabel lblbookid = new JLabel("Book ID :");
lblbookid.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblbookid.setBounds(45, 125, 326, 40);
contentPane.add(lblbookid);


		JLabel lblName = new JLabel("Title :");
lblName.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblName.setBounds(45, 175, 326, 40);
contentPane.add(lblName);
		JLabel lblAuthor = new JLabel("Author :");
lblAuthor.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblAuthor.setBounds(45, 225, 326, 40);
contentPane.add(lblAuthor);
		JButton btnAddBooks = new JButton("Submit");
		btnAddBooks.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
			String id=TextField.getText();
			String name=TextField_1.getText();
			String author=TextField_2.getText();
			boolean checkid =Pattern.matches("0L[0-9][0-9][0-9][0-9][0-9]",id);
			if(checkid) {
			if(id.equals("") || name.equals("")|| author.equals("") ) {
				JOptionPane.showMessageDialog(AddBook.this,"Error !!\n
Kindly Fill all the fields ");
```

```
                    }
                    else {
                    if(Bookdetails.checkbook(id)) {
                            JOptionPane.showMessageDialog(AddBook.this,"Duplication
Error !!\n Inserton Failed");
                    }
                    else {
                    int i=Bookdetails.insertbook(id, name, author);
                    if(i>0){
                            JOptionPane.showMessageDialog(AddBook.this,"Books added
successfully!");
                    }else{
                            JOptionPane.showMessageDialog(AddBook.this,"Unknown
Error !!!\n Insertion not completed");
                    }
                    }
                    }}
                    else
                            JOptionPane.showMessageDialog(AddBook.this,"Format Error
!!\nBook ID - 0LXXXXX");
                    }
                });
                btnAddBooks.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnAddBooks.setBackground(new Color(0,0, 0));
        btnAddBooks.setBounds(300, 300, 130, 50);
        contentPane.add(btnAddBooks);
```

```
                JButton btnBack = new JButton("Back");
                btnBack.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e1) {
                                Book.main(new String[]{});
                                frame.dispose();
                        }
                        });
                  btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
              btnBack.setBackground(new Color(240, 240, 240));
              btnBack.setBounds(450, 300, 130, 50);
              contentPane.add(btnBack);
        }
}
```

AddStudent.java:

```
import java.awt.EventQueue;


import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.border.EmptyBorder;
import javax.swing.ButtonGroup;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
```

```
import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JComboBox;

import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.util.regex.Pattern;
import java.awt.event.ActionEvent;

public class AddStudent extends JFrame {
	private static final long serialVersionUID = 1L;
	static AddStudent frame;
	private JPanel contentPane;
	private JTextField TextField;
	private JTextField TextField_1;
	private JRadioButton rb1;
	private JRadioButton rb2;

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
```

```
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					frame = new AddStudent();
					frame.setTitle("Student Info.");
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the frame.
	 */
	public AddStudent() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 650, 480);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);
		contentPane.setLayout(null);

		JLabel lblAddStudents = new JLabel("Add Students");
```

```
lblAddStudents.setFont(new Font("Courier New", Font.BOLD, 42));
lblAddStudents.setForeground(Color.BLACK);
lblAddStudents.setBounds(225, 27, 326, 40);
contentPane.add(lblAddStudents);

JLabel lblname = new JLabel("Name :");
lblname.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblname.setBounds(45, 125, 326, 40);
contentPane.add(lblname);

JLabel lblrollno = new JLabel("Roll Number :");
lblrollno.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblrollno.setBounds(45, 175, 326, 40);
contentPane.add(lblrollno);

JLabel lblcourse = new JLabel("Course :");
lblcourse.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblcourse.setBounds(45, 225, 326, 40);
contentPane.add(lblcourse);
JLabel lblbranch = new JLabel("Branch :");
lblbranch.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblbranch.setBounds(45, 275, 326, 40);
contentPane.add(lblbranch);

TextField = new JTextField();
```

```
TextField.setFont(new Font("Big Calson", Font.PLAIN, 18));
TextField.setBounds(290, 125, 326, 40);
contentPane.add(TextField);
TextField.setColumns(10);

TextField_1 = new JTextField();
TextField_1.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField_1.setBounds(290, 175, 326, 40);
contentPane.add(TextField_1);
TextField_1.setColumns(10);

      rb1 = new JRadioButton("B.E");
rb1.setFont(new Font("Tahoma", Font.PLAIN, 18));
rb1.setBounds(290, 225, 100, 40);
contentPane.add(rb1);

rb2 = new JRadioButton("M.E");
rb2.setFont(new Font("Tahoma", Font.PLAIN, 18));
rb2.setBounds(400, 225, 100, 40);
contentPane.add(rb2);

ButtonGroup bg=new ButtonGroup();
bg.add(rb1);bg.add(rb2);

String[] be = {"      ---- Select ----"}; ;
```

```
JComboBox<String> jComboBox = new JComboBox<>(be);
jComboBox.setBounds(290, 275, 200, 60);
contentPane.add(jComboBox);

rb1.addItemListener(new ItemListener()
{
      public void itemStateChanged(ItemEvent e)
      {
         if (e.getStateChange() == ItemEvent.SELECTED) {
               jComboBox.removeAllItems();
               jComboBox.addItem("      ---- Select ----");
               jComboBox.addItem("Civil");
               jComboBox.addItem("CSE");
               jComboBox.addItem("EEE");
               jComboBox.addItem("ECE");
               jComboBox.addItem("Mechanical");
               jComboBox.addItem("IT");
         }
         else if (e.getStateChange() == ItemEvent.DESELECTED) {
               jComboBox.removeAllItems();
         }
   }
});

rb2.addItemListener(new ItemListener() {
```

```
        public void itemStateChanged(ItemEvent e) {
           if (e.getStateChange() == ItemEvent.SELECTED) {
           jComboBox.removeAllItems();
           jComboBox.addItem("      ---- Select ----");
           jComboBox.addItem("ADM");
           jComboBox.addItem("ESVLSID");
           jComboBox.addItem("PSPE");
           }
           else if (e.getStateChange() == ItemEvent.DESELECTED) {
                 jComboBox.removeAllItems();
           }
        }
});


        JButton btnAdd = new JButton("Submit");
        btnAdd.addActionListener(new ActionListener() {
                 public void actionPerformed(ActionEvent e) {
                 String name=TextField.getText();
                 String rollno=TextField_1.getText();
                 String branch = null;
                 if(rb1.isSelected()) {
                          branch = rb1.getText();
                 }
                 if(rb2.isSelected()) {
                          branch = rb2.getText();
```

```
                }

                String course =
jComboBox.getItemAt(jComboBox.getSelectedIndex());

                boolean check = Pattern.matches("1602-[0-9][0-9]-[0-9][0-9][0-9]-[0-
9][0-9][0-9]", rollno);

                if(check) {

                if(rollno.equals("") || name.equals("")|| course.equals("      ---- Select ---
-")|| branch.equals(null)) {

                        JOptionPane.showMessageDialog(AddStudent.this,"Error !!\n
Kindly Fill all fields");

                }

                else {

                if(Studentdetails.checkstudent(rollno)) {


        JOptionPane.showMessageDialog(AddStudent.this,"Duplication Error !!\nInsertion
Failed");

                }

                else {

                int i=Studentdetails.insertstudent(name, rollno, branch, course);

                System.out.println(branch);

                if(i>0){

                        JOptionPane.showMessageDialog(AddStudent.this,"Added
Student successfully!");

                }else{

                        JOptionPane.showMessageDialog(AddStudent.this,"Unknown
Error !!\nInsertion Failed");

                }

                }

                }
```

```
                }
                else
                        JOptionPane.showMessageDialog(AddStudent.this,"Format
Error !!\nRoll Number : 1602-XX-XXX-XXX ");
                }
        });

        btnAdd.setFont(new Font("Tahoma", Font.PLAIN, 20));
    btnAdd.setBackground(new Color(0,0, 0));
    btnAdd.setBounds(300, 375, 130, 50);
    contentPane.add(btnAdd);

        JButton btnBack = new JButton("Back");
        btnBack.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e1) {
                        Student.main(new String[]{});
                        frame.dispose();
                }
                });

        btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnBack.setBackground(new Color(240, 240, 240));
        btnBack.setBounds(475, 375, 130, 50);
        contentPane.add(btnBack);
    }
}
```

Book.java:

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JLabel;
import java.awt.Font;
import java.awt.Color;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;


public class Book extends JFrame {
	private static final long serialVersionUID = 1L;
	static Book frame;
	private JPanel contentPane;


	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
```

```
                public void run() {
                        try {
                                frame = new Book();
                                frame.setTitle("Book Info.");
                                frame.setVisible(true);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
        });
}

/**
 * Create the frame.
 */
public Book() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 590, 440);
        contentPane = new JPanel();
        contentPane.setForeground(Color.GRAY);
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblInfo = new JLabel("--  Books   --");
```

```
		lblInfo.setFont(new Font("American Typewriter", Font.BOLD, 42));
		lblInfo.setForeground(Color.DARK_GRAY);
		lblInfo.setBounds(200, 50, 410, 60);
		contentPane.add(lblInfo);

		JButton btnAddBook = new JButton("Add Book");
		btnAddBook.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
			AddBook.main(new String[]{});
			frame.dispose();
			}
		});
		btnAddBook.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnAddBook.setBackground(new Color(0,0, 0));
btnAddBook.setBounds(100, 150, 200, 50);
contentPane.add(btnAddBook);

		JButton btnUpdateBook = new JButton("Modify Book Details");
		btnUpdateBook.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent arg0) {
				Updatebook.main(new String[]{});
				frame.dispose();
			}
		});
		btnUpdateBook.setFont(new Font("Tahoma", Font.PLAIN, 20));
```

```
btnUpdateBook.setBackground(new Color(0,0, 0));
btnUpdateBook.setBounds(335, 150, 200, 50);
contentPane.add(btnUpdateBook);

        JButton btnDeleteBook = new JButton("Delete Book");
        btnDeleteBook.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                DeleteBook.main(new String[]{});
                frame.dispose();
            }
        });
        btnDeleteBook.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnDeleteBook.setBackground(new Color(0,0, 0));
btnDeleteBook.setBounds(100, 250, 200, 50);
contentPane.add(btnDeleteBook);

        JButton btnView = new JButton("View All Books");
        btnView.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                ViewBook.main(new String[]{});
            }
        });
        btnView.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnView.setBackground(new Color(0,0, 0));
btnView.setBounds(335, 250, 200, 50);
```

```
	contentPane.add(btnView);


		JButton btnBack = new JButton("Back");
		btnBack.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				Menu.main(new String[]{});
				frame.dispose();
			}
		});
		btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
	btnBack.setBackground(new Color(0,0, 0));
	btnBack.setBounds(215, 335, 200, 50);
	contentPane.add(btnBack);


}
}
```

BookDetails.java:

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;


public class Bookdetails {


	public static boolean checkbook(String bookid){
		boolean status=false;
```

```
            try{
                    Connection con=SQLCon.getConnection();
                    PreparedStatement ps=con.prepareStatement("select * from book
where bookid=?");
                    ps.setString(1,bookid);
                ResultSet rs=ps.executeQuery();
                    status=rs.next();
                    con.close();
            }catch(Exception e){System.out.println(e);}
            return status;
        }
        public static int insertbook(String bookid,String Title,String Author){
            int status=0;
            try{
                    Connection con=SQLCon.getConnection();
                    PreparedStatement ps=con.prepareStatement("insert into
Book(bookid,Title,Author) values(?,?,?)");
                    ps.setString(1,bookid);
                    ps.setString(2,Title);
                    ps.setString(3,Author);
                    status=ps.executeUpdate();
                    con.close();
            }catch(Exception e){System.out.println(e);}
            return status;
        }
        public static int deletebook(String id){
```

```
		int status=0;
		try{
			Connection con=SQLCon.getConnection();
			PreparedStatement ps=con.prepareStatement("delete from book where bookid=?");
			ps.setString(1,id);
			status=ps.executeUpdate();
			con.close();
		}catch(Exception e){System.out.println(e);}
		return status;
	}
	public static int issuebook(String date,String bookid,String rollno){
		int status=0;
		try{
			Connection con=SQLCon.getConnection();
			PreparedStatement ps=con.prepareStatement("insert into issue(issue_date,bookid,rollno) values(?,?,?)");
			ps.setString(2,bookid);
			ps.setString(3,rollno);
			ps.setString(1,date);
			status=ps.executeUpdate();
			con.close();
		}catch(Exception e){System.out.println(e);}
		return status;
	}
	public static int returnbook(String date,String bookid,String rollno){
```

```
                int status=0;
                try{
                        Connection con=SQLCon.getConnection();
                        PreparedStatement ps=con.prepareStatement("insert into
breturn(breturn_date,bookid,rollno) values(?,?,?)");
                        ps.setString(2,bookid);
                        ps.setString(3,rollno);
                        ps.setString(1,date);
                        status=ps.executeUpdate();
                        con.close();
                }catch(Exception e){System.out.println(e);}
                return status;
        }
}
```

ChangePassword.java:

```
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
```

```
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.border.EmptyBorder;

public class ChangePassword extends JFrame {

    private static final long serialVersionUID = 1L;
    static ChangePassword frame;
    private JPanel contentPane;
    private JPasswordField PasswordField;
    private JLabel lblEnterOldPassword;
    private JPasswordField PasswordField1;
    private JLabel lblEnterNewPassword;
    private JPasswordField PasswordField2;
    private JLabel lblReEnterNewPassword;

    /**
     * Launch the application.
     */

    /**
     * Create the frame.
     */
    public ChangePassword(String name) {
```

```
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(110, 95, 650, 400);
setResizable(false);

contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblAdSQLConooks = new JLabel("Change Password");
lblAdSQLConooks.setFont(new Font("Courier New", Font.BOLD, 42));
lblAdSQLConooks.setForeground(Color.BLACK);
lblAdSQLConooks.setBounds(150, 30, 500, 60);
contentPane.add(lblAdSQLConooks);

lblEnterOldPassword = new JLabel("Enter Current Password :");
lblEnterOldPassword.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblEnterOldPassword.setBounds(45, 125, 326, 40);
contentPane.add(lblEnterOldPassword);

lblEnterNewPassword = new JLabel("Enter New Password :");
lblEnterNewPassword.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblEnterNewPassword.setBounds(45, 175, 326, 40);
contentPane.add(lblEnterNewPassword);
```

```
lblReEnterNewPassword = new JLabel("Re-enter New Password :");
lblReEnterNewPassword.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblReEnterNewPassword.setBounds(45, 225, 326, 40);
contentPane.add(lblReEnterNewPassword);

PasswordField = new JPasswordField();
PasswordField.setFont(new Font("Tahoma", Font.PLAIN, 18));
PasswordField.setBounds(290, 125, 326, 40);
contentPane.add(PasswordField);
PasswordField.setColumns(10);

PasswordField1 = new JPasswordField();
PasswordField1.setFont(new Font("Tahoma", Font.PLAIN, 18));
PasswordField1.setBounds(290, 175, 326, 40);
contentPane.add(PasswordField1);
PasswordField.setColumns(10);

PasswordField2 = new JPasswordField();
PasswordField2.setFont(new Font("Tahoma", Font.PLAIN, 18));
PasswordField2.setBounds(290, 225, 326, 40);
contentPane.add(PasswordField2);
PasswordField2.setColumns(10);

JButton btnBack = new JButton("Back");
btnBack.addActionListener(new ActionListener() {
```

```
        public void actionPerformed(ActionEvent e) {
                Menu.main(new String[]{});
                frame.dispose();
        }});
      btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
      btnBack.setBackground(new Color(240, 240, 240));
      btnBack.setBounds(410, 300, 170, 50);
      contentPane.add(btnBack);

      JButton btnSave = new JButton("Save");
      btnSave.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

          String old = String.valueOf(PasswordField.getPassword());
          String pnew = String.valueOf(PasswordField1.getPassword());
          String renew = String.valueOf(PasswordField2.getPassword());
          if(Logindetails.checkpassword(name, old)) {
                if(pnew.equals(renew)) {
          try {
            Connection con = SQLCon.getConnection();

            PreparedStatement st = con.prepareStatement("Update librarian set password=?
where id=?");

            st.setString(1, renew);
            st.setString(2, name);
```

```
            st.executeUpdate();

            System.out.println("updated password for id : " + name);

            JOptionPane.showMessageDialog(btnSave, "Password has been Changed
Successfully\nLogin Again :)");

            frame.dispose();

            Login.main(new String[]{});


          } catch (SQLException sqlException) {

            sqlException.printStackTrace();

          }

        }else {


        JOptionPane.showMessageDialog(ChangePassword.this, "Both Passwords doesn't
match \nEnter again !");

                                PasswordField1.setText("");

                                PasswordField2.setText("");

                                }

                        }

                        else{


        JOptionPane.showMessageDialog(ChangePassword.this,"Current Password is
Incorrect !!");

                        PasswordField.setText("");

                        PasswordField1.setText("");

                        PasswordField2.setText("");

        };
```

```
            }
        });
        btnSave.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnSave.setBackground(new Color(240, 240, 240));
        btnSave.setBounds(210, 300, 170, 50);
        contentPane.add(btnSave);

    }
}
```

DeleteBook.java:

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Font;
```

```
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.awt.event.ActionEvent;

public class DeleteBook extends JFrame {
	private static final long serialVersionUID = 1L;
	static DeleteBook frame;
	private JPanel contentPane;
   private JTextField TextField;
   private JLabel lblEnterOldPassword;
   private JTextField TextField1;
   private JLabel lblEnterNewPassword;
   private JTextField TextField2;
   private JLabel lblReEnterNewPassword;

	/**
	 * Launch the application.
	 */
```

```
    public static void main(String[] args) {
            EventQueue.invokeLater(new Runnable() {
                    public void run() {
                            try {
                                    frame = new DeleteBook();
                                    frame.setTitle("Delete book Info.");
                                    frame.setVisible(true);
                            } catch (Exception e) {
                                    e.printStackTrace();
                            }
                    }
            });
    }

    /**
     * Create the frame.
     */
    public DeleteBook() {
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(110, 95, 650, 420);
setResizable(false);

contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
```

```
contentPane.setLayout(null);
JLabel lblAdSQLConooks = new JLabel("Delete Book");
lblAdSQLConooks.setFont(new Font("Courier New", Font.BOLD, 36));
lblAdSQLConooks.setForeground(Color.BLACK);
lblAdSQLConooks.setBounds(275, 27, 350, 40);
contentPane.add(lblAdSQLConooks);

TextField = new JTextField();
TextField.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField.setBounds(290, 125, 326, 40);
contentPane.add(TextField);
TextField.setColumns(10);

TextField1 = new JTextField();
TextField1.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField1.setBounds(290, 175, 350, 40);
TextField1.setEditable(false);
contentPane.add(TextField1);
TextField.setColumns(20);

TextField2 = new JTextField();
TextField2.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField2.setBounds(290, 225, 350, 40);
contentPane.add(TextField2);
TextField2.setEditable(false);
```

```
TextField2.setColumns(20);


TextField.setText("");
            TextField1.setText("");
            TextField2.setText("");


      lblEnterOldPassword = new JLabel("Book ID :");
lblEnterOldPassword.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblEnterOldPassword.setBounds(45, 125, 326, 40);
contentPane.add(lblEnterOldPassword);


lblEnterNewPassword = new JLabel("Title :");
lblEnterNewPassword.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblEnterNewPassword.setBounds(45, 175, 326, 40);
contentPane.add(lblEnterNewPassword);


lblReEnterNewPassword = new JLabel("Author :");
lblReEnterNewPassword.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblReEnterNewPassword.setBounds(45, 225, 326, 40);
contentPane.add(lblReEnterNewPassword);


            Connection con = SQLCon.getConnection();


            JButton btnBack = new JButton("Back");
          btnBack.addActionListener(new ActionListener() {
```

```
                public void actionPerformed(ActionEvent e) {
                        frame.dispose();
                        Book.main(new String[]{});
                }});
            btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
            btnBack.setBackground(new Color(240, 240, 240));
            btnBack.setBounds(450, 300, 130, 50);
            contentPane.add(btnBack);
        JButton btnDelete = new JButton("Delete");
        btnDelete.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                String sid=TextField.getText();
                                if(sid==null||sid.trim().equals("")){
                                        JOptionPane.showMessageDialog(DeleteBook.this,"Id
can't be blank");
                                }else{

                                        int i=Bookdetails.deletebook(sid);
                                        if(i>0){

            JOptionPane.showMessageDialog(DeleteBook.this,"Book Record deleted
successfully!");
                                                System.out.println("Deleted record of Book-id :
"+sid);
                                        }else{

            JOptionPane.showMessageDialog(DeleteBook.this,"Unable to delete given id!");
                                        }
```

```
                    }
                    TextField.setText("");
                    TextField1.setText("");
                    TextField2.setText("");
                }
            });
        btnDelete.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnDelete.setBackground(new Color(240, 240, 240));
        btnDelete.setBounds(300, 300, 130, 50);
        contentPane.add(btnDelete);

        TextField.addKeyListener((KeyListener) new KeyAdapter() {
           @Override
           public void keyPressed(KeyEvent e) {
              if(e.getKeyCode() == KeyEvent.VK_ENTER){
                 String roll = TextField.getText();

                 Connection con = SQLCon.getConnection();
                 if(Bookdetails.checkbook(roll)) {
                 Statement st;
                    try {
                       PreparedStatement stmt = con.prepareStatement("select
* from book where bookid=?");
                       stmt.setString(1,roll);
                       ResultSet rs = stmt.executeQuery();
                       while(rs.next())
```

```
                    {       TextField.setText(rs.getString(1));
                TextField1.setText(rs.getString(2));
                TextField2.setText(rs.getString(3));
                }
                } catch (SQLException e1) {
                        e1.printStackTrace();
                }}else {
                        JOptionPane.showMessageDialog(DeleteBook.this,"book ID is
Invalid !!!");
                        TextField.setText("");
                        TextField1.setText("");
                        TextField2.setText("");
                }

        }}});


    JButton btnLoad = new JButton("Load");
    btnLoad.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
                String roll = TextField.getText();

            Connection con = SQLCon.getConnection();
            if(Bookdetails.checkbook(roll)) {
            Statement st;
                        try {
```

```
                                        PreparedStatement stmt = con.prepareStatement("select
* from book where bookid=?");
                                        stmt.setString(1,roll);
                                        ResultSet rs = stmt.executeQuery();
                                        while(rs.next())
                                        {       TextField.setText(rs.getString(1));
                        TextField1.setText(rs.getString(2));
                        TextField2.setText(rs.getString(3));
                        }
                        } catch (SQLException e1) {
                                e1.printStackTrace();
                        }}else {
                                JOptionPane.showMessageDialog(DeleteBook.this,"book ID is
Invalid !!!");
                                TextField.setText("");
                                TextField1.setText("");
                                TextField2.setText("");
                        }


            }});
        btnLoad.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnLoad.setBackground(new Color(240, 240, 240));
        btnLoad.setBounds(150, 300, 130, 50);
        contentPane.add(btnLoad);

    }
```

```
}
```

DeleteStudent.java:

```
import java.awt.Color;
import java.awt.EventQueue;


import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Font;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.awt.event.ActionEvent;


public class DeleteStudent extends JFrame {
```

```
	private static final long serialVersionUID = 1L;
	static DeleteStudent frame;
	private JPanel contentPane;
private JTextField TextField;
private JLabel lblrollno;
private JLabel lblDeleteStudents;
private JTextField TextField1;
private JLabel lblName;
private JTextField TextField2;
private JLabel lblcourse;
private JTextField TextField3;
private JLabel lblbranch;


	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					frame = new DeleteStudent();
					frame.setTitle("Delete Student Info.");
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
```

```
                        }
                    }
                });
    }

    /**
     * Create the frame.
     */
    public DeleteStudent() {
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(110, 95, 650, 450);
setResizable(false);

contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);
lblDeleteStudents = new JLabel("Delete Student");
            lblDeleteStudents.setFont(new Font("Courier New", Font.BOLD, 42));
            lblDeleteStudents.setForeground(Color.BLACK);
            lblDeleteStudents.setBounds(200, 27, 400, 40);
            contentPane.add(lblDeleteStudents);

TextField = new JTextField();
TextField.setFont(new Font("Tahoma", Font.PLAIN, 18));
```

```
TextField.setBounds(290, 125, 326, 40);
contentPane.add(TextField);
TextField.setColumns(10);

TextField1 = new JTextField();
TextField1.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField1.setBounds(290, 175, 326, 40);
TextField1.setEditable(false);
contentPane.add(TextField1);
TextField.setColumns(10);

TextField2 = new JTextField();
TextField2.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField2.setBounds(290, 225, 326, 40);
contentPane.add(TextField2);
TextField2.setEditable(false);
TextField2.setColumns(10);

TextField3 = new JTextField();
TextField3.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField3.setBounds(290, 275, 326, 40);
contentPane.add(TextField3);
TextField3.setEditable(false);
TextField3.setColumns(10);
```

```
TextField.setText("");
                TextField1.setText("");
                TextField2.setText("");

        lblrollno = new JLabel("Roll No. :");
lblrollno.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblrollno.setBounds(45, 125, 326, 40);
contentPane.add(lblrollno);

lblName = new JLabel("Name :");
lblName.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblName.setBounds(45, 175, 326, 40);
contentPane.add(lblName);

lblcourse = new JLabel("Course :");
lblcourse.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblcourse.setBounds(45, 225, 326, 40);
contentPane.add(lblcourse);

lblbranch = new JLabel("Branch :");
lblbranch.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblbranch.setBounds(45, 275, 326, 40);
contentPane.add(lblbranch);

                Connection con = SQLCon.getConnection();
```

```
                JButton btnBack = new JButton("Back");
            btnBack.addActionListener(new ActionListener() {
               public void actionPerformed(ActionEvent e) {
                        frame.dispose();
                        Student.main(new String[]{});
               }});
            btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
            btnBack.setBackground(new Color(240, 240, 240));
            btnBack.setBounds(500, 350, 100, 50);
            contentPane.add(btnBack);
      JButton btnSave = new JButton("Delete");
      btnSave.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
                  String r = TextField.getText();
           if(Studentdetails.checkstudent(r)) {
           try {

                int i=Studentdetails.deletestudent(r);
                                        if(i>0){
        JOptionPane.showMessageDialog(DeleteStudent.this,"Student Record deleted
successfully!");
                                                System.out.println("Deleted Record of Student
ID : "+r);
                                        }else{
```

```
        JOptionPane.showMessageDialog(DeleteStudent.this,"Unable to delete given
rollno.!");
                                        }


            } catch (Exception sqlException) {
                sqlException.printStackTrace();
            }}
                                    else{
        JOptionPane.showMessageDialog(DeleteStudent.this,"Student ID is Invalid !!!");
        }

            TextField.setText("");
                                    TextField1.setText("");
                                    TextField2.setText("");
                                    TextField3.setText("");
        }
    });
    btnSave.setFont(new Font("Tahoma", Font.PLAIN, 20));
    btnSave.setBackground(new Color(240, 240, 240));
    btnSave.setBounds(350, 350, 100, 50);
    contentPane.add(btnSave);

    TextField.addKeyListener((KeyListener) new KeyAdapter() {
        @Override
        public void keyPressed(KeyEvent e) {
```

```
            if(e.getKeyCode() == KeyEvent.VK_ENTER){
                String roll = TextField.getText();

                Connection con = SQLCon.getConnection();
                if(Studentdetails.checkstudent(roll)) {
                Statement st;
                                try {
                                        PreparedStatement stmt = con.prepareStatement("select
* from students where rollno=?");
                                        stmt.setString(1,roll);
                                        ResultSet rs = stmt.executeQuery();
                                        while(rs.next())
                                        {       TextField1.setText(rs.getString(1));
                        TextField.setText(rs.getString(2));
                        TextField2.setText(rs.getString(3));
                        TextField3.setText(rs.getString(4));
                        }
                        } catch (SQLException e1) {
                                e1.printStackTrace();
                        }}else {
                                JOptionPane.showMessageDialog(DeleteStudent.this,"Student
ID is Invalid !!!");
                                TextField.setText("");
                                TextField1.setText("");
                                TextField2.setText("");
                                TextField3.setText("");
```

```
                }


        }}});


    JButton btnLoad = new JButton("Load");
    btnLoad.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
                String roll = TextField.getText();


            Connection con = SQLCon.getConnection();
            if(Studentdetails.checkstudent(roll)) {
            Statement st;
                        try {
                            PreparedStatement stmt = con.prepareStatement("select
* from students where rollno=?");
                            stmt.setString(1,roll);
                            ResultSet rs = stmt.executeQuery();
                            while(rs.next())
                            {       TextField1.setText(rs.getString(1));
                    TextField.setText(rs.getString(2));
                    TextField2.setText(rs.getString(3));
                    TextField3.setText(rs.getString(4));
                    }
                    } catch (SQLException e1) {
                            e1.printStackTrace();
```

```
                    }}else {
                            JOptionPane.showMessageDialog(DeleteStudent.this,"Student
ID is Invalid !!!");
                            TextField.setText("");
                            TextField1.setText("");
                            TextField2.setText("");
                            TextField3.setText("");
                    }


        }});
      btnLoad.setFont(new Font("Tahoma", Font.PLAIN, 20));
      btnLoad.setBackground(new Color(240, 240, 240));
      btnLoad.setBounds(200, 350, 100, 50);
      contentPane.add(btnLoad);


   }
}
```

Issue.java:

```
import java.awt.EventQueue;


import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
```

```
import java.awt.Color;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JComboBox;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class Issue extends JFrame {
	private static final long serialVersionUID = 1L;
	static Issue frame;
	private JPanel contentPane;
	private JTextField TextField;
	private JTextField TextField1;



	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					frame = new Issue();
					frame.setTitle("Issuing Book");
```

```
                                frame.setVisible(true);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
        });
}

/**
 * Create the frame.
 */
public Issue() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setResizable(false);
        setBounds(100, 100, 650, 480);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblissue = new JLabel("Issue Book ");
        lblissue.setFont(new Font("Courier New", Font.BOLD, 42));
        lblissue.setForeground(Color.DARK_GRAY);
        lblissue.setBounds(225, 27, 326, 40);
        contentPane.add(lblissue);
```

```
TextField = new JTextField();
TextField.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField.setBounds(290, 125, 326, 40);
contentPane.add(TextField);
TextField.setColumns(10);

TextField1 = new JTextField();
TextField1.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField1.setBounds(290, 175, 326, 40);
contentPane.add(TextField1);
TextField.setColumns(10);

      JLabel lblrollno = new JLabel("Student Roll Number :");
lblrollno.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblrollno.setBounds(45, 175, 326, 40);
contentPane.add(lblrollno);

JLabel lblbid = new JLabel("Book-ID :");
lblbid.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblbid.setBounds(45, 125, 326, 40);
contentPane.add(lblbid);

JLabel lbldate = new JLabel("Issuing Date :");
lbldate.setFont(new Font("Big Calson", Font.PLAIN, 18));
```

```
lbldate.setBounds(45, 225, 326, 40);

contentPane.add(lbldate);


JLabel lblnote = new JLabel("Note: Please check Student ID Carefully before issuing book!");

lblnote.setFont(new Font("Tahoma", Font.PLAIN, 16));

lblnote.setForeground(Color.RED);

lblnote.setBounds(100, 400,550 , 40);

contentPane.add(lblnote);


String[] da = {"Day"};

JComboBox<String> d = new JComboBox<>(da);

for (int dayCount = 01; dayCount <= 31; dayCount++)

d.addItem(Integer.toString(dayCount));

d.setBounds(290, 225, 100, 60);

contentPane.add(d);


String[] m = {"Month"};

JComboBox<String> mo = new JComboBox<>(m);

for (int dayCount = 01; dayCount <= 12; dayCount++)

mo.addItem(Integer.toString(dayCount));

mo.setBounds(390, 225, 100, 60);

contentPane.add(mo);


String[] y = {"Year"};

JComboBox<String> ye = new JComboBox<>(y);
```

```
            for (int dayCount = 2022; dayCount >=1990; dayCount--)
           ye.addItem(Integer.toString(dayCount));
            ye.setBounds(490, 225, 100, 60);
            contentPane.add(ye);

           JButton btnIssueBook = new JButton("Issue Book");
           btnIssueBook.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {

                  String bookid=TextField.getText();
                  String studentid=TextField1.getText();
                  String date= ye.getItemAt(ye.getSelectedIndex())+"-
"+mo.getItemAt(mo.getSelectedIndex())+"-"+d.getItemAt(d.getSelectedIndex());
                  if(Bookdetails.checkbook(bookid) ){
                        if(Studentdetails.checkstudent(studentid) ) {
                              if(ye.getItemAt(ye.getSelectedIndex()).equals("Year") ||
mo.getItemAt(mo.getSelectedIndex()).equals("Month")
||d.getItemAt(d.getSelectedIndex()).equals("Day") )
                              {

      JOptionPane.showMessageDialog(Issue.this,"Date Error !!\nSelect a valid date");
                              }
                              else {
                              Bookdetails.issuebook(date,bookid, studentid);
                              JOptionPane.showMessageDialog(Issue.this,"Book
issued successfully!");
                              System.out.println("Book-ID : "+bookid+" issued to
"+studentid+" successfully");
```

```
                                }
                            }
                            else {
                                JOptionPane.showMessageDialog(Issue.this,"Error !!\nEnter a valid Roll Number");
                            }

                        }else{
                            JOptionPane.showMessageDialog(Issue.this,"Error !!\nEnter a valid Book-ID");
                        }//end of checkbook if-else
                        }
                    });

                     JButton btnview = new JButton("View Issued Books");
                    btnview.addActionListener(new ActionListener() {
                       public void actionPerformed(ActionEvent e) {
                            Viewissue.main(new String[]{});
                       }});
                    btnview.setFont(new Font("Tahoma", Font.PLAIN, 20));
                    btnview.setBackground(new Color(240, 240, 240));
                    btnview.setBounds(100, 305, 200, 50);
                    contentPane.add(btnview);

                       JButton btnBack = new JButton("Back");
                       btnBack.addActionListener(new ActionListener() {
```

```
                public void actionPerformed(ActionEvent e) {
                Menu.main(new String[]{});
                        frame.dispose();
                }
        });

                btnIssueBook.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnIssueBook.setBackground(new Color(240, 240, 240));
        btnIssueBook.setBounds(320, 305, 130, 50);
        contentPane.add(btnIssueBook);

                btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnBack.setBackground(new Color(240, 240, 240));
        btnBack.setBounds(475, 305, 130, 50);
        contentPane.add(btnBack);

    }
}
```

Login.java:

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
```

```
import javax.swing.GroupLayout.Alignment;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JPasswordField;

public class Login extends JFrame {
	private static final long serialVersionUID = 1L;
	static Login frame;
	private JPanel contentPane;
	private JTextField TextField;
	private JPasswordField PasswordField;

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
```

```
                public void run() {
                        try {
                                frame = new Login();
                                frame.setTitle("College Library DataBase");
                                frame.setVisible(true);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
        });
}

/**
 * Create the frame.
 */
public Login() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(103, 95, 700, 600);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        ImageIcon i = new ImageIcon("src/vl.png");
        JLabel lbllogo = new JLabel(i);
        lbllogo.setForeground(Color.GRAY);
```

```
            lbllogo.setFont(new Font("Tahoma", Font.PLAIN, 18));
            lbllogo.setBounds(150, 30, 400, 300);
            contentPane.add(lbllogo);

            JLabel lblName = new JLabel("Login ID : ");
lblName.setFont(new Font("Courier New", Font.BOLD, 20));
lblName.setBounds(155, 375, 200, 40);
contentPane.add(lblName);

   JLabel lblName1 = new JLabel("Password : ");
lblName1.setFont(new Font("Courier New", Font.BOLD, 20));
lblName1.setBounds(155, 425, 200, 40);
contentPane.add(lblName1);

TextField = new JTextField();
       TextField.setFont(new Font("Tahoma", Font.PLAIN, 18));
         TextField.setBounds(290, 375, 250, 40);
         contentPane.add(TextField);
         TextField.setColumns(10);

         PasswordField = new JPasswordField();
         PasswordField.setFont(new Font("Tahoma", Font.PLAIN, 18));
         PasswordField.setBounds(290, 425, 250, 40);
                  contentPane.add(PasswordField);
                  PasswordField.setColumns(10);
```

```
            JButton btnClose = new JButton("Close");
        btnClose.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
                frame.dispose();
           }});
        btnClose.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnClose.setBackground(new Color(240, 240, 240));
        btnClose.setBounds(450, 500, 130, 50);
        contentPane.add(btnClose);


            JButton btnLogin = new JButton("Login");
            btnLogin.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                String name=TextField.getText();
                String password=String.valueOf(PasswordField.getPassword());
                if(Logindetails.checklogin(name)){
                    if(Logindetails.checkpassword(name,password)){
                    Menu.main(new String[]{});
                    Menu.id(name);
                    frame.dispose();
                }
                    else{
                        JOptionPane.showMessageDialog(Login.this, "Login
Error !!\n Password was Incorrect");
```

```
                                        PasswordField.setText("");
                                        }
                                }
                                else{
                                        JOptionPane.showMessageDialog(Login.this,"User
Error !!\n User doesn't exist ");
                                TextField.setText("");
                                PasswordField.setText("");
                        }
                        }
                });
                btnLogin.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnLogin.setBackground(new Color(240, 240, 240));
        btnLogin.setBounds(250, 500, 130, 50);
        contentPane.add(btnLogin);

        }
}
```

Logindetails.java:

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class Logindetails {
        public static boolean checklogin(String id){
                boolean allow=false;
```

```
                try{
                        Connection con=SQLCon.getConnection();
                        PreparedStatement ps=con.prepareStatement("select * from Librarian
where id=?");
                        ps.setString(1,id);
                   ResultSet rs=ps.executeQuery();
                        allow=rs.next();
                        con.close();
                }catch(Exception e){System.out.println(e);}
                return allow;
        }
        public static boolean checkpassword(String id,String pass){
                boolean state=false;
                try{
                        Connection con=SQLCon.getConnection();
                        PreparedStatement ps=con.prepareStatement("select * from Librarian
where id =? and password=?");
                        ps.setString(1,id);
                        ps.setString(2,pass);
                   ResultSet rs=ps.executeQuery();
                        state=rs.next();
                        con.close();
                }catch(Exception e){System.out.println(e);}
                return state;
        }
}
```

Menu.java:

```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JLabel;
import java.awt.Font;
import java.awt.Color;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;


public class Menu extends JFrame {
	private static final long serialVersionUID = 1L;
	static Menu frame;
	private JPanel contentPane;
	static String idd;


	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
```

```
                public void run() {
                        try {
                                frame = new Menu();
                                frame.setTitle("College Library Database");
                                frame.setVisible(true);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
        });
    }
public static void id(String id) {
    idd=id;
}
    /**
     * Create the frame.
     */
    public Menu() {
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            setBounds(100, 100, 590, 650);
            contentPane = new JPanel();
            contentPane.setForeground(Color.GRAY);
            contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
            setContentPane(contentPane);
            contentPane.setLayout(null);
```

```
		JLabel lblLibrarianSection = new JLabel("Welcome, Librarian - "+idd);
		lblLibrarianSection.setFont(new Font("Academy Engraved LET", Font.BOLD
, 40));
		lblLibrarianSection.setBounds(75, 50, 500, 60);
		contentPane.add(lblLibrarianSection);

		JButton btnBInfo = new JButton("Books Info");
		btnBInfo.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
			Book.main(new String[]{});
			frame.dispose();
			}
		});
		btnBInfo.setFont(new Font("Tahoma", Font.PLAIN, 20));
	btnBInfo.setBackground(new Color(0,0, 0));
	btnBInfo.setBounds(100, 150, 200, 50);
	contentPane.add(btnBInfo);

		JButton btnSInfo = new JButton("Students Info");
		btnSInfo.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent arg0) {
				Student.main(new String[]{});
				frame.dispose();
			}
		});
```

```
            btnSInfo.setFont(new Font("Tahoma", Font.PLAIN, 20));
            btnSInfo.setBackground(new Color(0,0, 0));
            btnSInfo.setBounds(335, 150, 200, 50);
contentPane.add(btnSInfo);

            JButton btnIssueBook = new JButton("Issue Books");
            btnIssueBook.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        Issue.main(new String[]{});
                        frame.dispose();
                  }
            });
            btnIssueBook.setFont(new Font("Tahoma", Font.PLAIN, 20));
            btnIssueBook.setBackground(new Color(0,0, 0));
            btnIssueBook.setBounds(100, 250, 200, 50);
contentPane.add(btnIssueBook);


            JButton btnChange = new JButton("Change Password");
            btnChange.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        ChangePassword.frame = new ChangePassword(idd);
                        ChangePassword.frame.setTitle("College Library Datbase");
                        ChangePassword.frame.setVisible(true);
                        frame.dispose();
```

```
                }
            });
            btnChange.setFont(new Font("Tahoma", Font.PLAIN, 20));
            btnChange.setBackground(new Color(0,0, 0));
            btnChange.setBounds(215, 485, 200, 50);
contentPane.add(btnChange);

JButton btnSearch = new JButton("Search");
            btnSearch.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    Search.main(new String[]{});
                    frame.dispose();
                }
            });
            btnSearch.setFont(new Font("Tahoma", Font.PLAIN, 20));
            btnSearch.setBackground(new Color(0,0, 0));
            btnSearch.setBounds(215, 335, 200, 50);
contentPane.add(btnSearch);

JButton btnStat = new JButton("Statistics");
btnStat.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    Statistics.main(new String[]{});
                    frame.dispose();
                }
```

```
        });
btnStat.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnStat.setBackground(new Color(0,0, 0));
btnStat.setBounds(215, 410, 200, 50);
contentPane.add(btnStat);


        JButton btnReturnBook = new JButton("Return Books");
        btnReturnBook.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Return.main(new String[]{});
                frame.dispose();
            }
        });
        btnReturnBook.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnReturnBook.setBackground(new Color(0,0, 0));
        btnReturnBook.setBounds(335, 250, 200, 50);
contentPane.add(btnReturnBook);


        JButton btnLogout = new JButton("Logout");
        btnLogout.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Login.main(new String[]{});
                frame.dispose();
```

```
                }
            });
            btnLogout.setFont(new Font("Tahoma", Font.PLAIN, 20));
            btnLogout.setBackground(new Color(0,0, 0));
            btnLogout.setBounds(215, 555, 200, 50);
        contentPane.add(btnLogout);


        }
}
```

Return.java:

```
import java.awt.EventQueue;


import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;


import java.awt.Color;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JComboBox;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
```

```
public class Return extends JFrame {
	private static final long serialVersionUID = 1L;
	static Return frame;
	private JPanel contentPane;
	private JTextField TextField;
	private JTextField TextField1;



	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					frame = new Return();
					frame.setTitle("Returning Book");
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}
```

```
/**
 * Create the frame.
 */
public Return() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setResizable(false);
        setBounds(100, 100, 650, 480);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblReturn = new JLabel("Return Book ");
        lblReturn.setFont(new Font("Courier New", Font.BOLD, 42));
        lblReturn.setForeground(Color.DARK_GRAY);
        lblReturn.setBounds(225, 27, 326, 40);
        contentPane.add(lblReturn);

      TextField = new JTextField();
      TextField.setFont(new Font("Tahoma", Font.PLAIN, 18));
      TextField.setBounds(290, 125, 326, 40);
      contentPane.add(TextField);
      TextField.setColumns(10);
```

```
TextField1 = new JTextField();
TextField1.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField1.setBounds(290, 175, 326, 40);
contentPane.add(TextField1);
TextField.setColumns(10);

    JLabel lblrollno = new JLabel("Student Roll Number :");
lblrollno.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblrollno.setBounds(45, 175, 326, 40);
contentPane.add(lblrollno);

JLabel lblbid = new JLabel("Book-ID :");
lblbid.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblbid.setBounds(45, 125, 326, 40);
contentPane.add(lblbid);

JLabel lbldate = new JLabel("Returning Date :");
lbldate.setFont(new Font("Big Calson", Font.PLAIN, 18));
lbldate.setBounds(45, 225, 326, 40);
contentPane.add(lbldate);

JLabel lblnote = new JLabel("Note: Please check Condition of Book before
returning book!");
        lblnote.setFont(new Font("Tahoma", Font.PLAIN, 16));
        lblnote.setForeground(Color.RED);
        lblnote.setBounds(100, 400,550 , 40);
```

```
contentPane.add(lblnote);


String[] da = {"Day"};
    JComboBox<String> d = new JComboBox<>(da);
    for (int dayCount = 01; dayCount <= 31; dayCount++)
  d.addItem(Integer.toString(dayCount));
    d.setBounds(290, 225, 100, 60);
    contentPane.add(d);


    String[] m = {"Month"};
    JComboBox<String> mo = new JComboBox<>(m);
    for (int dayCount = 01; dayCount <= 12; dayCount++)
  mo.addItem(Integer.toString(dayCount));
    mo.setBounds(390, 225, 100, 60);
    contentPane.add(mo);


    String[] y = {"Year"};
    JComboBox<String> ye = new JComboBox<>(y);
    for (int dayCount = 2022; dayCount >=1990; dayCount--)
  ye.addItem(Integer.toString(dayCount));
    ye.setBounds(490, 225, 100, 60);
    contentPane.add(ye);


  JButton btnReturnBook = new JButton("Return Book");
  btnReturnBook.addActionListener(new ActionListener() {
```

```
                public void actionPerformed(ActionEvent e) {


                String bookid=TextField.getText();

                String studentid=TextField1.getText();

                String date= ye.getItemAt(ye.getSelectedIndex())+"-
"+mo.getItemAt(mo.getSelectedIndex())+"-"+d.getItemAt(d.getSelectedIndex());

                if(Bookdetails.checkbook(bookid) ){

                        if(Studentdetails.checkstudent(studentid) ) {

                                if(ye.getItemAt(ye.getSelectedIndex()).equals("Year") ||
mo.getItemAt(mo.getSelectedIndex()).equals("Month")
||d.getItemAt(d.getSelectedIndex()).equals("Day") )

                                {


        JOptionPane.showMessageDialog(Return.this,"Date Error !!\nSelect a valid date");

                                }

                                else {

                                Bookdetails.returnbook(date,bookid,studentid);

                                JOptionPane.showMessageDialog(Return.this,"Book
Returned successfully!");

                                System.out.println("Book-ID : "+bookid+" returned by
"+studentid+" successfully");

                                }

                        }

                        else {

                                JOptionPane.showMessageDialog(Return.this,"Error
!!\nEnter a valid Roll Number");

                        }


                }else{
```

```
                    JOptionPane.showMessageDialog(Return.this,"Error !!\nEnter
a valid Book-ID");
                }//end of checkbook if-else


                }
            });

             JButton btnview = new JButton("View Returned Books");
            btnview.addActionListener(new ActionListener() {
               public void actionPerformed(ActionEvent e) {
                     Viewreturn.main(new String[]{});
               }});
            btnview.setFont(new Font("Tahoma", Font.PLAIN, 20));
            btnview.setBackground(new Color(240, 240, 240));
            btnview.setBounds(100, 305, 200, 50);
            contentPane.add(btnview);

               JButton btnBack = new JButton("Back");
               btnBack.addActionListener(new ActionListener() {
                     public void actionPerformed(ActionEvent e) {
                     Menu.main(new String[]{});
                           frame.dispose();
                     }
               });

               btnReturnBook.setFont(new Font("Tahoma", Font.PLAIN, 20));
```

```
            btnReturnBook.setBackground(new Color(240, 240, 240));
            btnReturnBook.setBounds(330, 305, 130, 50);
            contentPane.add(btnReturnBook);


                btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
            btnBack.setBackground(new Color(240, 240, 240));
            btnBack.setBounds(490, 305, 130, 50);
            contentPane.add(btnBack);


        }
}
```

Search.java:

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
```

```
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

public class Search extends JFrame{
	private static final long serialVersionUID = 1L;
	static Search frame;
	private JPanel contentPane;
	private JTextField TextField_1;
	private JTextField TextField_2;
	private JTextField TextField_3;
	private JTextArea TextField_5;
	private JTextArea TextField_4;

	public static void main(String [] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					frame = new Search();
					frame.setTitle("Search");
					frame.setVisible(true);
				}
				catch(Exception e){
					e.printStackTrace();
```

```
                                    }
                        }
    });
    }
    public Search() {
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(110, 30, 700, 900);
setResizable(false);

contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);


       String[] op = {"      ---- Select ----","Book","Student"}; ;
       JComboBox<String> jComboBox = new JComboBox<>(op);
       jComboBox.setFont(new Font("Tahoma", Font.PLAIN, 18));
       jComboBox.setSelectedIndex(0);
       jComboBox.setBounds(240, 100, 326, 100);
       contentPane.add(jComboBox);

TextField_1 = new JTextField();
TextField_1.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField_1.setBounds(240, 200, 400, 40);
```

```
contentPane.add(TextField_1);
TextField_1.setColumns(10);

TextField_2 = new JTextField();
TextField_2.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField_2.setBounds(240, 370, 400, 40);
contentPane.add(TextField_2);
TextField_2.setColumns(10);
TextField_2.setVisible(false);

TextField_3 = new JTextField();
TextField_3.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField_3.setBounds(240, 430, 400, 40);
contentPane.add(TextField_3);
TextField_3.setColumns(10);
TextField_3.setVisible(false);

TextField_4 = new JTextArea();
TextField_4.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField_4.setBounds(240, 490, 400, 100);
contentPane.add(TextField_4);
TextField_4.setColumns(10);
TextField_4.setVisible(false);

TextField_5 = new JTextArea();
```

```
TextField_5.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField_5.setBounds(240, 640, 400, 100);
contentPane.add(TextField_5);
TextField_5.setColumns(10);
TextField_5.setVisible(false);

            TextField_1.setText("");
            TextField_2.setText("");
            TextField_3.setText("");
            TextField_4.setText("");
            TextField_5.setText("");



            JLabel lblSearchs = new JLabel("-- Search --");
lblSearchs.setFont(new Font("American Typewriter", Font.BOLD, 42));
lblSearchs.setForeground(Color.DARK_GRAY);
lblSearchs.setBounds(180, 50, 326, 60);
contentPane.add(lblSearchs);

JLabel lblName = new JLabel("ID : ");
lblName.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblName.setBounds(45, 200, 326, 40);
contentPane.add(lblName);
```

```
JLabel lblNam = new JLabel();
lblNam.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblNam.setBounds(45, 370, 326, 40);
contentPane.add(lblNam);

JLabel lblAuthor = new JLabel();
lblAuthor.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblAuthor.setBounds(45, 430, 326, 40);
contentPane.add(lblAuthor);

JLabel lblissue = new JLabel();
lblissue.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblissue.setBounds(45, 490, 326, 40);
contentPane.add(lblissue);

JLabel lblreturn = new JLabel();
lblreturn.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblreturn.setBounds(45, 640, 326, 40);
contentPane.add(lblreturn);

        JLabel lblbookid = new JLabel("Search By : ");
lblbookid.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblbookid.setBounds(45, 125, 326, 40);
contentPane.add(lblbookid);
```

```
JButton btnBac = new JButton("Clear");
                    btnBac.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e1) {
                            TextField_2.setText("");
                            TextField_3.setText("");
                            TextField_4.setText("");
                            TextField_5.setText("");
                            TextField_1.setText("");
                          jComboBox.setSelectedIndex(0);
                            TextField_2.setVisible(false);
                            TextField_3.setVisible(false);
                            TextField_4.setVisible(false);
                            TextField_5.setVisible(false);
                            lblAuthor.setText("");
                            btnBac.setVisible(false);
                            lblreturn.setText("");
                            lblissue.setText("");
                            lblNam.setText("");
                        }
                        });
                     btnBac.setFont(new Font("Tahoma", Font.PLAIN, 20));
                   btnBac.setBackground(new Color(240, 240, 240));
                   btnBac.setBounds(450, 780, 130, 50);
                   contentPane.add(btnBac);
                btnBac.setVisible(false);
```

```
JButton btnSearch = new JButton("Search");
btnSearch.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
String item =
jComboBox.getItemAt(jComboBox.getSelectedIndex());
if(item.equals("      ---- Select ----" ) ||
TextField_1.getText().equals("")){
JOptionPane.showMessageDialog(Search.this,"Error
!!\n Please Select an Option !");
}
else {
if(item.equals("Book")) {

if(Bookdetails.checkbook(TextField_1.getText())) {
lblAuthor.setText("Author :");
lblNam.setText("Title :");
lblissue.setText("Lended By : ");
lblreturn.setText("Returned By : ");
TextField_2.setVisible(true);
TextField_3.setVisible(true);
TextField_4.setVisible(true);
TextField_5.setVisible(true);
TextField_4.setText("");
TextField_5.setText("");
btnBac.setVisible(true);
```

```
try{
        Connection con=SQLCon.getConnection();
        PreparedStatement ps=con.prepareStatement("select title,author from book where bookid =
'"+TextField_1.getText()+"'",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_
UPDATABLE);
        ResultSet rs=ps.executeQuery();
        String r = null;
        while(rs.next())  {

    TextField_2.setText(rs.getString(1));

    TextField_3.setText(rs.getString(2));

        }
        String S = TextField_1.getText();
        PreparedStatement ps1=con.prepareStatement("select name from students where rollno in (SELECT rollno from
issue where bookid =
'"+S+"')",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
        ResultSet rs1=ps1.executeQuery();
        while(rs1.next())

    TextField_4.append(rs1.getString(1)+"\n");
        PreparedStatement ps2=con.prepareStatement("select name from students where rollno in (SELECT rollno from
breturn where bookid =
'"+S+"')",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
        ResultSet rs2=ps2.executeQuery();
        while(rs2.next())

    TextField_5.append(rs2.getString(1)+"\n");
```

```
                                        con.close();

                                        }catch(Exception e1){
System.out.println(e1);}

                                    }else {

    JOptionPane.showMessageDialog(Search.this,"Error !!\n Book-ID not found !");
                                    }
                                }
                                else {

    if(Studentdetails.checkstudent(TextField_1.getText())){
                                        lblAuthor.setText("Branch :");
                                        lblNam.setText("Name :");
                                        lblissue.setText("Books Lended : ");
                                        lblreturn.setText("Books Returned : ");
                                        TextField_2.setVisible(true);
                                        TextField_3.setVisible(true);
                                        TextField_4.setVisible(true);
                                        TextField_5.setVisible(true);
                                        TextField_4.setText("");
                                        TextField_5.setText("");
                                        btnBac.setVisible(true);
                                        try{
```

```
                                                    Connection
con=SQLCon.getConnection();

                                                    PreparedStatement
ps=con.prepareStatement("select name,branch from students where rollno =
'"+TextField_1.getText()+"'",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_
UPDATABLE);

                                                    ResultSet rs=ps.executeQuery();

                                                    String r = null;

                                                    while(rs.next())  {


        TextField_2.setText(rs.getString(1));


        TextField_3.setText(rs.getString(2));

                                                    }

                                                    String S = TextField_1.getText();

                                                    PreparedStatement
ps1=con.prepareStatement("select title from book where bookid in (SELECT bookid from
issue where rollno =
'"+S+"')",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);

                                                    ResultSet
rs1=ps1.executeQuery();

                                                    while(rs1.next())


        TextField_4.append(rs1.getString(1)+"\n");

                                                    PreparedStatement
ps2=con.prepareStatement("select title from book where bookid in (SELECT bookid from
breturn where rollno =
'"+S+"')",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);

                                                    ResultSet
rs2=ps2.executeQuery();

                                                    while(rs2.next())


        TextField_5.append(rs2.getString(1)+"\n");
```

```
                                                con.close();

                                            }catch(Exception e1){
System.out.println(e1);}
                                        }else {
        JOptionPane.showMessageDialog(Search.this,"Error !!\n Student-ID not Found !");
                                        }
                                        }

                                    }
                            }
                        });
                        btnSearch.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnSearch.setBackground(new Color(0,0, 0));
        btnSearch.setBounds(300, 290, 130, 50);
        contentPane.add(btnSearch);


                        JButton btnBack = new JButton("Back");
                        btnBack.addActionListener(new ActionListener() {
                                public void actionPerformed(ActionEvent e1) {
                                        Menu.main(new String[]{});
                                        frame.dispose();
                                }
```

```
                });
                btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
                btnBack.setBackground(new Color(240, 240, 240));
                btnBack.setBounds(450, 290, 130, 50);
                contentPane.add(btnBack);

        }

}
```

SQLConn.java:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

public class SQLCon{
        public static Connection getConnection(){
                Connection con=null;
                try{
                        Class.forName("com.mysql.cj.jdbc.Driver");

        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/college_library","ro
ot","kiran08");
                }catch(Exception e){System.out.println(e);}
                return con;
        }
```

```
}
```

Statistics.java:

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;


public class Statistics extends JFrame{
	private static final long serialVersionUID = 1L;
	static Statistics frame;
	private JPanel contentPane;
	private JTextField TextField_1;
```

```
    private JTextArea TextField_2;


    public static void main(String [] args) {
            EventQueue.invokeLater(new Runnable() {
                    public void run() {
                            try {
                                    frame = new Statistics();
                                    frame.setTitle("Statistics");
                                    frame.setVisible(true);
                            }
                            catch(Exception e){
                                    e.printStackTrace();
                            }
                    }
    });
    }
    public Statistics() {
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(110, 95, 670, 600);
setResizable(false);

contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);
```

```
        String[] op = {"      ---- Select ----","Book","Student"}; ;
        JComboBox<String> jComboBox = new JComboBox<>(op);
        jComboBox.setFont(new Font("Tahoma", Font.PLAIN, 18));
        jComboBox.setSelectedIndex(0);
        jComboBox.setBounds(240, 100, 326, 100);
        contentPane.add(jComboBox);

TextField_1 = new JTextField();
TextField_1.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField_1.setBounds(240, 260, 400, 40);
contentPane.add(TextField_1);
TextField_1.setColumns(10);
TextField_1.setVisible(false);
TextField_1.setEditable(false);

TextField_2 = new JTextArea();
TextField_2.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField_2.setBounds(240, 320, 400, 100);
contentPane.add(TextField_2);
TextField_2.setColumns(10);
TextField_2.setVisible(false);
TextField_2.setEditable(false);
```

```
		TextField_1.setText("");
		TextField_2.setText("");



		JLabel lblStatisticss = new JLabel("-- Statistics --");
lblStatisticss.setFont(new Font("American Typewriter", Font.BOLD, 42));
lblStatisticss.setForeground(Color.DARK_GRAY);
lblStatisticss.setBounds(180, 30, 326, 60);
contentPane.add(lblStatisticss);

JLabel lblName = new JLabel();
lblName.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblName.setBounds(45, 260, 326, 40);
contentPane.add(lblName);
		JLabel lblAuthor = new JLabel();
lblAuthor.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblAuthor.setBounds(45, 310, 326, 40);
contentPane.add(lblAuthor);

		JLabel lblbookid = new JLabel("Show By : ");
lblbookid.setFont(new Font("Big Calson", Font.PLAIN, 18));
lblbookid.setBounds(45, 125, 326, 40);
contentPane.add(lblbookid);
```

```
JButton btnBac = new JButton("Clear");
                    btnBac.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e1) {
                            TextField_1.setText("");
                            TextField_2.setText("");
                            TextField_1.setVisible(false);
                          jComboBox.setSelectedIndex(0);
                            TextField_2.setVisible(false);
                            lblAuthor.setVisible(false);
                            lblName.setVisible(false);
                            btnBac.setVisible(false);
                        }
                        });
                      btnBac.setFont(new Font("Tahoma", Font.PLAIN, 20));
                    btnBac.setBackground(new Color(240, 240, 240));
                    btnBac.setBounds(450, 500, 130, 50);
                    contentPane.add(btnBac);
                  btnBac.setVisible(false);


            JButton btnStatistics = new JButton("Submit");
            btnStatistics.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        String item =
jComboBox.getItemAt(jComboBox.getSelectedIndex());
                        if(item.equals("      ---- Select ----")) {
```

```
                                    JOptionPane.showMessageDialog(Statistics.this,"Error
!!\n Please Select an Option !");
                                }
                                else {
                                    if(item.equals("Book")) {
                                lblName.setText("Most Lended Book :");
                                        lblAuthor.setText("Lended By :");
                                        TextField_1.setVisible(true);
                                        TextField_2.setText("");
                                        TextField_2.setVisible(true);
                                        btnBac.setVisible(true);
                                        try{
                                            Connection
con=SQLCon.getConnection();
                                            PreparedStatement
ps=con.prepareStatement("select title from book where bookid = (SELECT bookid FROM
issue  GROUP BY bookid  ORDER BY count(*) DESC LIMIT
1)",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
                                            //step4 execute query
                                            ResultSet rs=ps.executeQuery();
                                            String r = null;
                                            while(rs.next())

        TextField_1.setText(rs.getString(1));
                                            String S = TextField_1.getText();
                                        PreparedStatement
ps2=con.prepareStatement("select bookid from book where title
='"+S+"'",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
                                        //step4 execute query
```

```
                                        ResultSet rs2=ps2.executeQuery();
                                            while(rs2.next())
                                                r = rs2.getString(1);
                                        PreparedStatement
ps1=con.prepareStatement("select name from students where rollno in (SELECT rollno from
issue where bookid =
'"+r+"')",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
                                        //step4 execute query
                                        ResultSet rs1=ps1.executeQuery();
                                        while(rs1.next())

        TextField_2.append(rs1.getString(1)+"\n");
                                        con.close();


                                        }catch(Exception e1){
System.out.println(e1);}



                                }
                                else {
                                        lblName.setText("Most Active Student :");
                                        lblAuthor.setText("Books Lended :");
                                        TextField_1.setVisible(true);
                                        TextField_2.setVisible(true);
                                        btnBac.setVisible(true);
                                        TextField_2.setText("");
                                        try{
```

```
                                                    Connection
con=SQLCon.getConnection();

                                                    PreparedStatement
ps=con.prepareStatement("select name from students where rollno = (SELECT rollno FROM
issue  GROUP BY rollno  ORDER BY count(*) DESC LIMIT
1)",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);

                                                    //step4 execute query

                                                    ResultSet rs=ps.executeQuery();

                                                    String r = null;

                                                    while(rs.next())


        TextField_1.setText(rs.getString(1));

                                                    String S = TextField_1.getText();

                                                PreparedStatement
ps2=con.prepareStatement("select rollno from students  where name
='"+S+"'",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);

                                                //step4 execute query

                                                ResultSet rs2=ps2.executeQuery();

                                                    while(rs2.next())

                                                        r = rs2.getString(1);

                                                    PreparedStatement
ps1=con.prepareStatement("select title from book where bookid in (SELECT bookid from
issue where rollno =
'"+r+"')",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);

                                                    //step4 execute query

                                                    ResultSet rs1=ps1.executeQuery();

                                                    while(rs1.next())


        TextField_2.append(rs1.getString(1)+"\n");

                                                    con.close();
```

```
                                                    }catch(Exception e1){
System.out.println(e1);}
                                                }
                                            }
                                    }
                            });
                            btnStatistics.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnStatistics.setBackground(new Color(0,0, 0));
        btnStatistics.setBounds(300, 200, 130, 50);
        contentPane.add(btnStatistics);


                            JButton btnBack = new JButton("Back");
                            btnBack.addActionListener(new ActionListener() {
                                    public void actionPerformed(ActionEvent e1) {
                                            Menu.main(new String[]{});
                                            frame.dispose();
                                    }
                                    });
                             btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
                    btnBack.setBackground(new Color(240, 240, 240));
                    btnBack.setBounds(450, 200, 130, 50);
                    contentPane.add(btnBack);

            }
```

```
}
```

Student.java:

```
import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import java.awt.Font;

import java.awt.Color;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;


public class Student extends JFrame {

	private static final long serialVersionUID = 1L;

	static Student frame;

	private JPanel contentPane;


	/**

	 * Launch the application.

	 */

	public static void main(String[] args) {

		EventQueue.invokeLater(new Runnable() {
```

```
                public void run() {
                        try {
                                frame = new Student();
                                frame.setTitle("Student Info.");
                                frame.setVisible(true);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
        });
}

/**
 * Create the frame.
 */
public Student() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 590, 440);
        contentPane = new JPanel();
        contentPane.setForeground(Color.GRAY);
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblInfo = new JLabel("--  Students  --");
```

```
		lblInfo.setFont(new Font("American Typewriter", Font.BOLD, 42));
		lblInfo.setForeground(Color.DARK_GRAY);
		lblInfo.setBounds(180, 50, 326, 60);
		contentPane.add(lblInfo);

		JButton btnAddStudent = new JButton("Add Student");
		btnAddStudent.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
			AddStudent.main(new String[]{});
			frame.dispose();
			}
		});
		btnAddStudent.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnAddStudent.setBackground(new Color(0,0, 0));
btnAddStudent.setBounds(100, 150, 200, 50);
contentPane.add(btnAddStudent);

		JButton btnUpdateStudent = new JButton("Modify Details");
		btnUpdateStudent.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent arg0) {
				Updatestudent.main(new String[]{});
				frame.dispose();
			}
		});
		btnUpdateStudent.setFont(new Font("Tahoma", Font.PLAIN, 20));
```

```
btnUpdateStudent.setBackground(new Color(0,0, 0));
btnUpdateStudent.setBounds(335, 150, 200, 50);
contentPane.add(btnUpdateStudent);

        JButton btnDeleteStudent = new JButton("Delete Student");
        btnDeleteStudent.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                DeleteStudent.main(new String[]{});
                frame.dispose();
            }
        });
        btnDeleteStudent.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnDeleteStudent.setBackground(new Color(0,0, 0));
btnDeleteStudent.setBounds(100, 250, 200, 50);
contentPane.add(btnDeleteStudent);

        JButton btnView = new JButton("View Students");
        btnView.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                ViewStudent.main(new String[]{});
            }
        });
        btnView.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnView.setBackground(new Color(0,0, 0));
btnView.setBounds(335, 250, 200, 50);
```

```
        contentPane.add(btnView);


                JButton btnBack = new JButton("Back");
                btnBack.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                Menu.main(new String[]{});
                                frame.dispose();
                        }
                });
                btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnBack.setBackground(new Color(0,0, 0));
        btnBack.setBounds(215, 335, 200, 50);
        contentPane.add(btnBack);


}
}
```

Studentdetails.java:

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;


public class Studentdetails {
        public static boolean checkstudent(String studid){
                boolean stat=false;
                try{
```

```
                    Connection con=SQLCon.getConnection();

                    PreparedStatement ps=con.prepareStatement("select * from students
where rollno=?");

                    ps.setString(1,studid);

                  ResultSet rs=ps.executeQuery();

                    stat=rs.next();

                    con.close();

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

                return stat;

        }

        public static int insertstudent(String name,String rollno,String Branch, String
Course){

                int status=0;

                try{

                    Connection con=SQLCon.getConnection();

                    PreparedStatement ps=con.prepareStatement("insert into
students(name,rollno,Branch,Course) values(?,?,?,?)");

                    ps.setString(1,name);

                    ps.setString(2,rollno);

                    ps.setString(3,Branch);

                    ps.setString(4,Course);

                    status=ps.executeUpdate();

                    con.close();

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

                return status;

        }

        public static int deletestudent(String id){
```

```
                int status=0;
                try{
                        Connection con=SQLCon.getConnection();
                        PreparedStatement ps=con.prepareStatement("delete from students
where rollno=?");
                        ps.setString(1,id);
                        status=ps.executeUpdate();
                        con.close();
                }catch(Exception e){System.out.println(e);}
                return status;
        }

}
```

Updatebook.java:

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.swing.JButton;
```

```
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

public class Updatebook extends JFrame {

    private static final long serialVersionUID = 1L;
    static Updatebook frame;
    private JPanel contentPane;
    private JTextField TextField;
    private JLabel lblBookId;
    private JTextField TextField1;
    private JLabel lblTitle;
    private JTextField TextField2;
    private JLabel lblAuthor;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
                    EventQueue.invokeLater(new Runnable() {
                            public void run() {
```

```
                        try {
                                frame = new Updatebook();
                                frame.setTitle("Book Info.");
                                frame.setVisible(true);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
        });
    }

/**
 * Create the frame.
 */
public Updatebook() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(110, 95, 650, 450);
    setResizable(false);

    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

            JLabel lblUpdateBooks = new JLabel("Modify Details");
```

```
            lblUpdateBooks.setFont(new Font("Courier New", Font.BOLD, 42));
            lblUpdateBooks.setForeground(Color.BLACK);
            lblUpdateBooks.setBounds(150, 27, 500, 60);
contentPane.add(lblUpdateBooks);

TextField = new JTextField();
TextField.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField.setBounds(290, 137, 326, 40);
contentPane.add(TextField);
TextField.setColumns(10);

TextField1 = new JTextField();
TextField1.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField1.setBounds(290, 187, 326, 40);
contentPane.add(TextField1);
TextField.setColumns(10);

TextField2 = new JTextField();
TextField2.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField2.setBounds(290, 237, 326, 40);
contentPane.add(TextField2);
TextField2.setColumns(10);

TextField.setText("");
            TextField1.setText("");
```

```
                TextField2.setText("");


                 Connection con = SQLCon.getConnection();


                 JButton btnBack = new JButton("Back");
               btnBack.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                           frame.dispose();
                           Book.main(new String[]{});
                  }});
               btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
               btnBack.setBackground(new Color(240, 240, 240));
               btnBack.setBounds(450, 325, 130, 50);
               contentPane.add(btnBack);
      JButton btnSave = new JButton("Save");
      btnSave.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
                  String b = TextField.getText();
            String t = TextField1.getText();
            String a = TextField2.getText();
           if(Bookdetails.checkbook(b)) {
           try {


              PreparedStatement st = con.prepareStatement("Update book set title=?, author =
? where bookid=?");
```

```
            st.setString(1, t);
            st.setString(2, a);
            st.setString(3, b);
            st.executeUpdate();
           System.out.println("Updated Info for Book-ID : " + b);
            JOptionPane.showMessageDialog(btnSave, "Data Updated Successfully :)");

        } catch (SQLException sqlException) {
            sqlException.printStackTrace();
        }}
                                else{
    JOptionPane.showMessageDialog(Updatebook.this,"Book ID is Invalid !!!");
                                TextField.setText("");
                                TextField1.setText("");
                                TextField2.setText("");
    }



    }
});
btnSave.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnSave.setBackground(new Color(240, 240, 240));
btnSave.setBounds(300, 325, 130, 50);
contentPane.add(btnSave);
```

```
lblBookId = new JLabel("Book ID :");
lblBookId.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblBookId.setBounds(45, 137, 326, 40);
contentPane.add(lblBookId);

lblTitle = new JLabel("Title :");
lblTitle.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblTitle.setBounds(45, 187, 326, 40);
contentPane.add(lblTitle);

lblAuthor = new JLabel("Author :");
lblAuthor.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblAuthor.setBounds(45, 237, 326, 40);
contentPane.add(lblAuthor);

JButton btnLoad = new JButton("Load");
btnLoad.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e) {
              String book = TextField.getText();
        String title = TextField1.getText();
        String author = TextField2.getText();
            Connection con = SQLCon.getConnection();
            if(Bookdetails.checkbook(book)) {
            Statement st;
                        try {
```

```
                                        PreparedStatement stmt = con.prepareStatement("select
* from book where bookid=?");
                                        stmt.setString(1,book);
                                        ResultSet rs = stmt.executeQuery();
                                        while(rs.next())
                                        {       TextField.setText(rs.getString(1));
                        TextField1.setText(rs.getString(2));
                        TextField2.setText(rs.getString(3));}
                        } catch (SQLException e1) {
                                e1.printStackTrace();
                        }}else {
                                JOptionPane.showMessageDialog(Updatebook.this,"Book ID
is Invalid !!");
                                TextField.setText("");
                                TextField1.setText("");
                                TextField2.setText("");
                        }


            }});
        btnLoad.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnLoad.setBackground(new Color(240, 240, 240));
        btnLoad.setBounds(150, 325, 130, 50);
        contentPane.add(btnLoad);
    }
}
```

Updatestudent.java:

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

public class Updatestudent extends JFrame {

    private static final long serialVersionUID = 1L;
    static Updatestudent frame;
    private JPanel contentPane;
    private JTextField TextField;
```

```
private JTextField TextField1;
private JTextField TextField2;
private JTextField TextField3;
/**
 * Launch the application.
 */
public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					frame = new Updatestudent();
					frame.setTitle("Student Info.");
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

/**
 * Create the frame.
 */
public Updatestudent() {
	setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
setBounds(110, 95, 650, 480);
setResizable(false);

contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblUpdateStudents = new JLabel("Modify Details");
        lblUpdateStudents.setFont(new Font("Courier New", Font.BOLD, 42));
        lblUpdateStudents.setForeground(Color.BLACK);
        lblUpdateStudents.setBounds(150, 27, 500, 60);
        contentPane.add(lblUpdateStudents);

        JLabel lblname = new JLabel("Name :");
        lblname.setFont(new Font("Big Calson", Font.PLAIN, 18));
        lblname.setBounds(45, 175, 326, 40);
        contentPane.add(lblname);

        JLabel lblrollno = new JLabel("Roll Number :");
        lblrollno.setFont(new Font("Big Calson", Font.PLAIN, 18));
        lblrollno.setBounds(45, 125, 326, 40);
        contentPane.add(lblrollno);

        JLabel lblcourse = new JLabel("Course :");
```

```
            lblcourse.setFont(new Font("Big Calson", Font.PLAIN, 18));
            lblcourse.setBounds(45, 225, 326, 40);
            contentPane.add(lblcourse);
            JLabel lblbranch = new JLabel("Branch :");
            lblbranch.setFont(new Font("Big Calson", Font.PLAIN, 18));
            lblbranch.setBounds(45, 275, 326, 40);
            contentPane.add(lblbranch);

TextField = new JTextField();
TextField.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField.setBounds(290, 125, 326, 40);
contentPane.add(TextField);
TextField.setColumns(10);

TextField1 = new JTextField();
TextField1.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField1.setBounds(290, 175, 326, 40);
contentPane.add(TextField1);
TextField.setColumns(10);

TextField2 = new JTextField();
TextField2.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField2.setBounds(290, 225, 326, 40);
contentPane.add(TextField2);
TextField2.setColumns(10);
```

```
TextField3 = new JTextField();
TextField3.setFont(new Font("Tahoma", Font.PLAIN, 18));
TextField3.setBounds(290, 275, 326, 40);
contentPane.add(TextField3);
TextField3.setColumns(10);


TextField.setText("");
            TextField1.setText("");
            TextField2.setText("");


             Connection con = SQLCon.getConnection();


             JButton btnBack = new JButton("Back");
          btnBack.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                    frame.dispose();
                    Student.main(new String[]{});
            }});
          btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
          btnBack.setBackground(new Color(240, 240, 240));
          btnBack.setBounds(450, 375, 130, 50);
          contentPane.add(btnBack);
JButton btnSave = new JButton("Save");
btnSave.addActionListener(new ActionListener() {
```

```
public void actionPerformed(ActionEvent e) {
        String r = TextField.getText();
    String n = TextField1.getText();
    String b = TextField3.getText();
    String c = TextField2.getText();
   if(Studentdetails.checkstudent(r)) {
   try {

      PreparedStatement st = con.prepareStatement("Update students set
Name=?,branch = ?,course=? where rollno=?");

      st.setString(1, n);
      st.setString(3, c);
      st.setString(2, b);
      st.setString(4, r);
      st.executeUpdate();
    System.out.println("Updated Info for Roll No. : " + r);
      JOptionPane.showMessageDialog(btnSave, "Data Updated Successfully :)");

   } catch (SQLException sqlException) {
      sqlException.printStackTrace();
   }}
                        else{
JOptionPane.showMessageDialog(Updatestudent.this,"Student ID is Invalid !!!");
                        TextField.setText("");
```

```
                                TextField1.setText("");
                                TextField2.setText("");
                                TextField3.setText("");
        }



        }
    });
    btnSave.setFont(new Font("Tahoma", Font.PLAIN, 20));
    btnSave.setBackground(new Color(240, 240, 240));
    btnSave.setBounds(300, 375, 130, 50);
    contentPane.add(btnSave);

    JButton btnLoad = new JButton("Load");
    btnLoad.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                String roll = TextField.getText();
            String name = TextField1.getText();
            String branch = TextField3.getText();
            String course = TextField2.getText();
                Connection con = SQLCon.getConnection();
                if(Studentdetails.checkstudent(roll)) {
                Statement st;
                                try {
                                        PreparedStatement stmt = con.prepareStatement("select
* from students where rollno=?");
```

```
                                stmt.setString(1,roll);
                                ResultSet rs = stmt.executeQuery();
                                while(rs.next())
                                {       TextField1.setText(rs.getString(1));
                        TextField.setText(rs.getString(2));
                        TextField3.setText(rs.getString(3));
                        TextField2.setText(rs.getString(4));
                        }
                        } catch (SQLException e1) {
                                e1.printStackTrace();
                        }}else {
                                JOptionPane.showMessageDialog(Updatestudent.this,"Student
ID is Invalid !!!");
                                TextField.setText("");
                                TextField1.setText("");
                                TextField2.setText("");
                                TextField3.setText("");
                        }

            }});
        btnLoad.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnLoad.setBackground(new Color(240, 240, 240));
        btnLoad.setBounds(150, 375, 130, 50);
        contentPane.add(btnLoad);

    }
```

```
}
```

ViewBook.java:

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.border.EmptyBorder;
import javax.swing.JTable;

public class ViewBook extends JFrame {

	private JPanel contentPane;
	private JTable table;

	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
```

```
                                ViewBook frame = new ViewBook();
                                frame.setTitle("Books");
                                frame.setVisible(true);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
        });
}

/**
 * Create the frame.
 */
public ViewBook() {
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        contentPane.setLayout(new BorderLayout(0, 0));
        setContentPane(contentPane);

        String data[][]=null;
        String column[]=null;
        try{
                Connection con=SQLCon.getConnection();
```

```
                PreparedStatement ps=con.prepareStatement("select * from
book",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);

                ResultSet rs=ps.executeQuery();


                ResultSetMetaData rsmd=rs.getMetaData();

                int cols=rsmd.getColumnCount();

                column=new String[cols];

                for(int i=1;i<=cols;i++){

                        column[i-1]=rsmd.getColumnName(i);

                }


                rs.last();

                int rows=rs.getRow();

                rs.beforeFirst();


                data=new String[rows][cols];

                int count=0;

                while(rs.next()){

                        for(int i=1;i<=cols;i++){

                                data[count][i-1]=rs.getString(i);

                        }

                        count++;

                }

                con.close();

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

```
                table = new JTable(data,column);
                table.setDefaultEditor(Object.class, null);
                JScrollPane sp=new JScrollPane(table);

                contentPane.add(sp, BorderLayout.CENTER);
            }

        }
```

Viewissue.java:

```
        import java.awt.BorderLayout;
        import java.awt.EventQueue;
        import java.sql.Connection;
        import java.sql.PreparedStatement;
        import java.sql.ResultSet;
        import java.sql.ResultSetMetaData;

        import javax.swing.JFrame;
        import javax.swing.JPanel;
        import javax.swing.JScrollPane;
        import javax.swing.border.EmptyBorder;
        import javax.swing.JTable;

        public class Viewissue extends JFrame {
```

```
private JPanel contentPane;
private JTable table;

public static void main(String[] args) {
	EventQueue.invokeLater(new Runnable() {
		public void run() {
			try {
				Viewissue frame = new Viewissue();
				frame.setTitle("Books");
				frame.setVisible(true);
			} catch (Exception e) {
				e.printStackTrace();
			}
		}
	});
}

/**
 * Create the frame.
 */
public Viewissue() {
	setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
	setBounds(100, 100, 450, 300);
	contentPane = new JPanel();
	contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
```

```
contentPane.setLayout(new BorderLayout(0, 0));
setContentPane(contentPane);

String data[][]=null;
String column[]=null;
try{
	Connection con=SQLCon.getConnection();
	PreparedStatement ps=con.prepareStatement("select * from
issue",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
	ResultSet rs=ps.executeQuery();

	ResultSetMetaData rsmd=rs.getMetaData();
	int cols=rsmd.getColumnCount();
	column=new String[cols];
	for(int i=1;i<=cols;i++){
		column[i-1]=rsmd.getColumnName(i);
	}

	rs.last();
	int rows=rs.getRow();
	rs.beforeFirst();

	data=new String[rows][cols];
	int count=0;
	while(rs.next()){
		for(int i=1;i<=cols;i++){
```

```
                                        data[count][i-1]=rs.getString(i);
                                }
                                count++;
                        }
                        con.close();
                }catch(Exception e){System.out.println(e);}

                table = new JTable(data,column);
                table.setDefaultEditor(Object.class, null);
                JScrollPane sp=new JScrollPane(table);
                contentPane.add(sp, BorderLayout.CENTER);
        }
    }
```

Viewreturn.java:

```
    import java.awt.BorderLayout;
    import java.awt.EventQueue;
    import java.sql.Connection;
    import java.sql.PreparedStatement;
    import java.sql.ResultSet;
    import java.sql.ResultSetMetaData;

    import javax.swing.JFrame;
    import javax.swing.JPanel;
    import javax.swing.JScrollPane;
```

```
import javax.swing.border.EmptyBorder;
import javax.swing.JTable;

public class Viewreturn extends JFrame {

	private JPanel contentPane;
	private JTable table;

	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					Viewreturn frame = new Viewreturn();
					frame.setTitle("Books");
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the frame.
	 */
```

```
public Viewreturn() {
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        contentPane.setLayout(new BorderLayout(0, 0));
        setContentPane(contentPane);

        String data[][]=null;
        String column[]=null;
        try{
                Connection con=SQLCon.getConnection();
                PreparedStatement ps=con.prepareStatement("select * from
breturn",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
                ResultSet rs=ps.executeQuery();

                ResultSetMetaData rsmd=rs.getMetaData();
                int cols=rsmd.getColumnCount();
                column=new String[cols];
                for(int i=1;i<=cols;i++){
                        column[i-1]=rsmd.getColumnName(i);
                }

                rs.last();
                int rows=rs.getRow();
                rs.beforeFirst();
```

```
                    data=new String[rows][cols];
                    int count=0;
                    while(rs.next()){
                        for(int i=1;i<=cols;i++){
                            data[count][i-1]=rs.getString(i);
                        }
                        count++;
                    }
                    con.close();
                }catch(Exception e){System.out.println(e);}

                table = new JTable(data,column);
                table.setDefaultEditor(Object.class, null);
                JScrollPane sp=new JScrollPane(table);
                contentPane.add(sp, BorderLayout.CENTER);
            }
        }
```

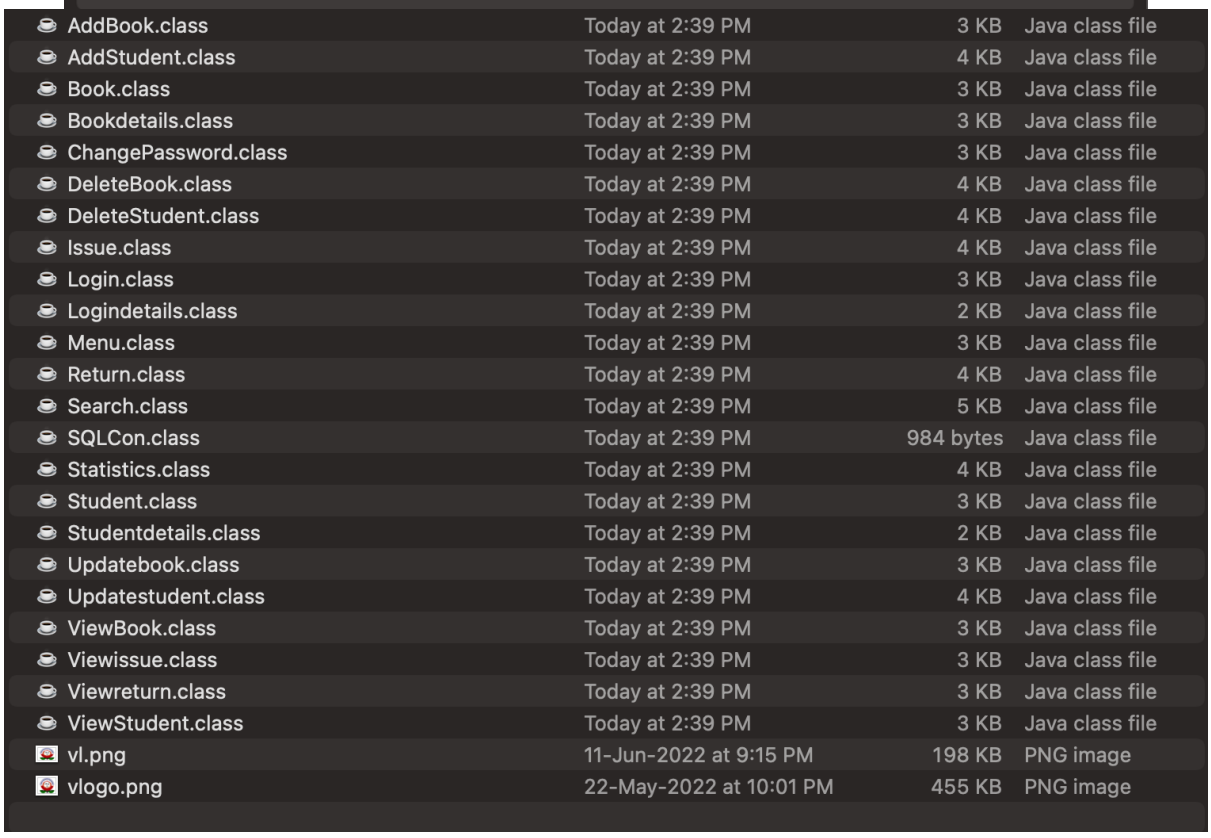
Viewstudent.java:

```
        import java.awt.BorderLayout;
        import java.awt.EventQueue;
        import java.sql.Connection;
        import java.sql.PreparedStatement;
        import java.sql.ResultSet;
        import java.sql.ResultSetMetaData;
```

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.border.EmptyBorder;
import javax.swing.JTable;

public class ViewStudent extends JFrame {

	private JPanel contentPane;
	private JTable table;

	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					ViewStudent frame = new ViewStudent();
					frame.setTitle("Students");
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}
```

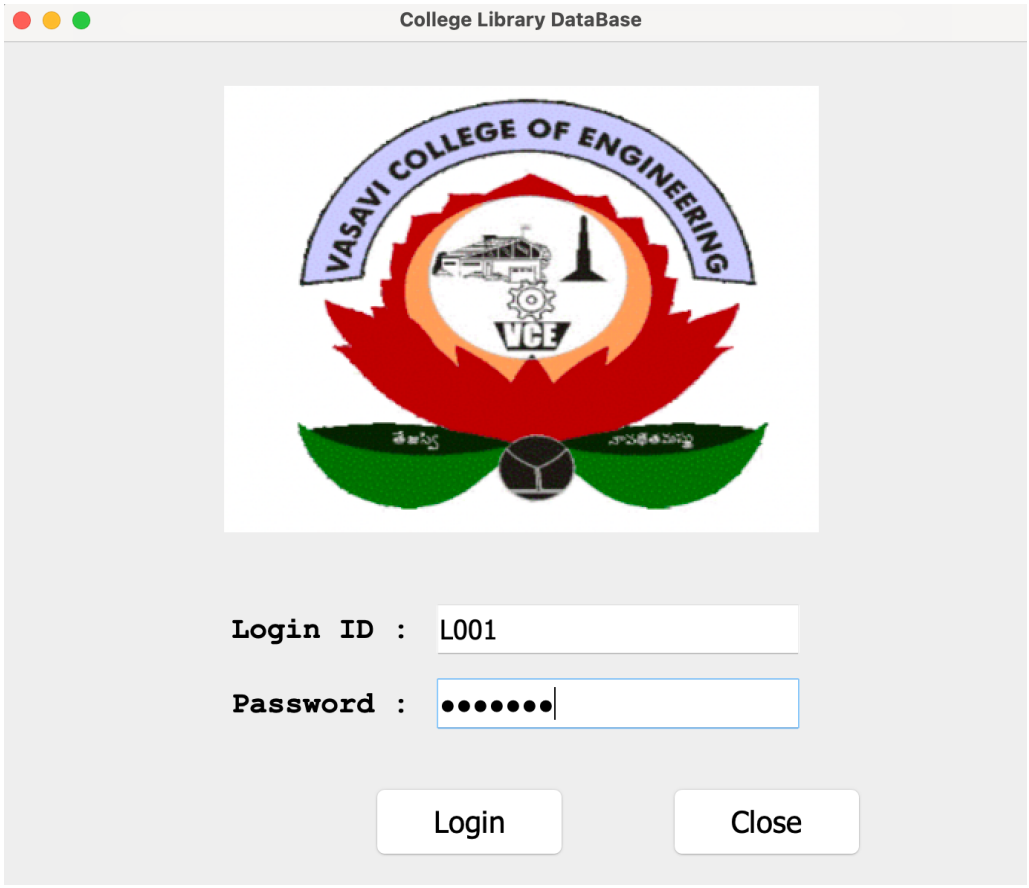
```
/**
 * Create the frame.
 */
public ViewStudent() {
	setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
	setBounds(100, 100, 450, 300);
	contentPane = new JPanel();
	contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
	contentPane.setLayout(new BorderLayout(0, 0));
	setContentPane(contentPane);
	String data[][]=null;
	String column[]=null;
	try{
		Connection con=SQLCon.getConnection();
		PreparedStatement ps = con.prepareStatement("select * from
students",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
		ResultSet rs=ps.executeQuery();

		ResultSetMetaData rsmd=rs.getMetaData();
		int cols=rsmd.getColumnCount();
		column=new String[cols];
		for(int i=1;i<=cols;i++){
			column[i-1]=rsmd.getColumnName(i);
		}
```

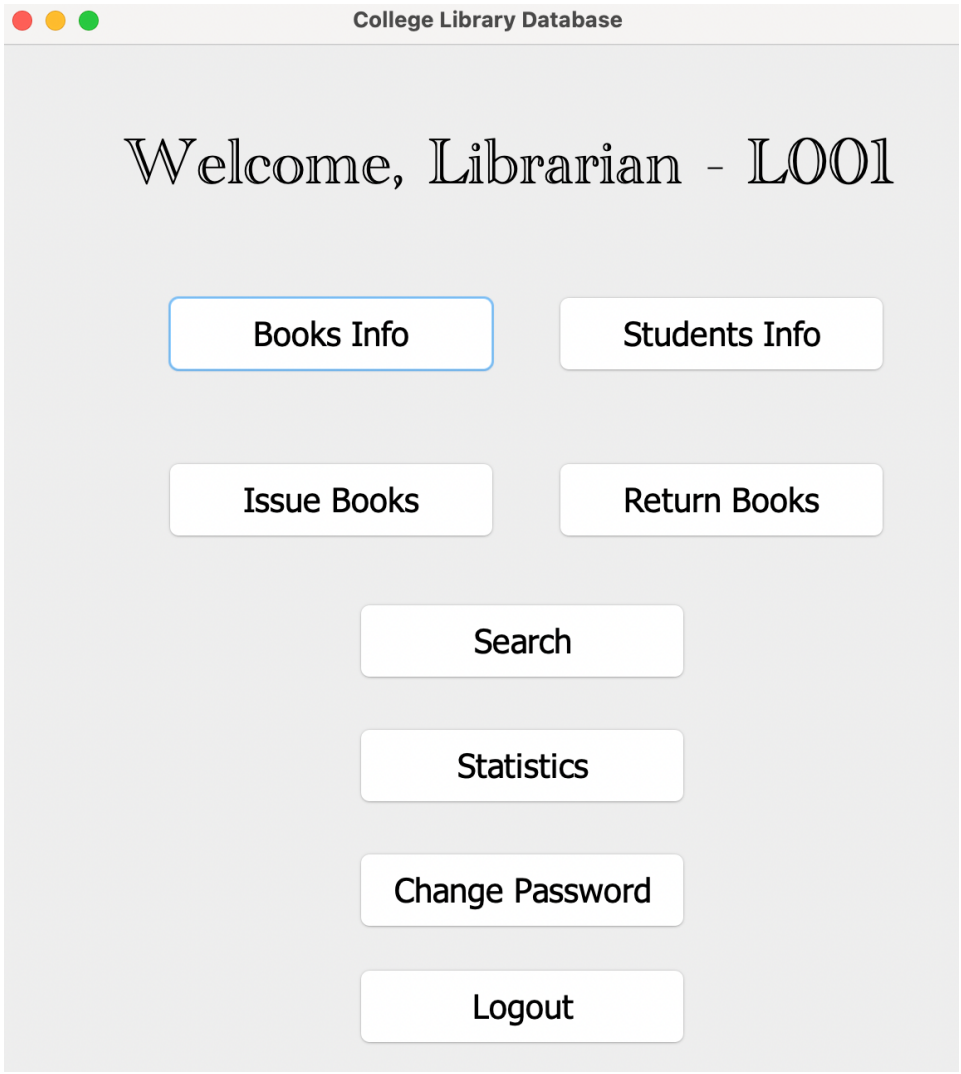
Name	Date Modified	Size	Kind
AddBook.java	17-Jun-2022 at 8:36 AM	5 KB	Plain Text
AddStudent.java	16-Jun-2022 at 7:14 PM	7 KB	Plain Text
Book.java	16-Jun-2022 at 6:44 PM	3 KB	Plain Text
Bookdetails.java	17-Jun-2022 at 8:29 AM	2 KB	Plain Text
ChangePassword.java	17-Jun-2022 at 8:29 AM	5 KB	Plain Text
DeleteBook.java	17-Jun-2022 at 8:29 AM	7 KB	Plain Text
DeleteStudent.java	17-Jun-2022 at 8:29 AM	8 KB	Plain Text
Issue.java	17-Jun-2022 at 8:40 AM	6 KB	Plain Text
Login.java	16-Jun-2022 at 6:35 PM	4 KB	Plain Text
Logindetails.java	17-Jun-2022 at 8:31 AM	905 bytes	Plain Text
Menu.java	Yesterday at 10:43 AM	5 KB	Plain Text
Return.java	17-Jun-2022 at 8:39 AM	6 KB	Plain Text
Search.java	Yesterday at 10:49 AM	10 KB	Plain Text
SQLCon.java	17-Jun-2022 at 8:28 AM	393 bytes	Plain Text
Statistics.java	Yesterday at 9:20 AM	8 KB	Plain Text
Student.java	17-Jun-2022 at 8:28 AM	4 KB	Plain Text
Studentdetails.java	17-Jun-2022 at 8:31 AM	1 KB	Plain Text
Updatebook.java	17-Jun-2022 at 8:30 AM	6 KB	Plain Text
Updatestudent.java	20-Jun-2022 at 10:29 PM	7 KB	Plain Text
ViewBook.java	17-Jun-2022 at 8:30 AM	2 KB	Plain Text
Viewissue.java	17-Jun-2022 at 8:30 AM	2 KB	Plain Text
Viewreturn.java	17-Jun-2022 at 8:30 AM	2 KB	Plain Text
ViewStudent.java	17-Jun-2022 at 8:30 AM	2 KB	Plain Text
vl.png	11-Jun-2022 at 9:15 PM	198 KB	PNG image
vlogo	22-May-2022 at 10:01 PM	455 KB	PNG image

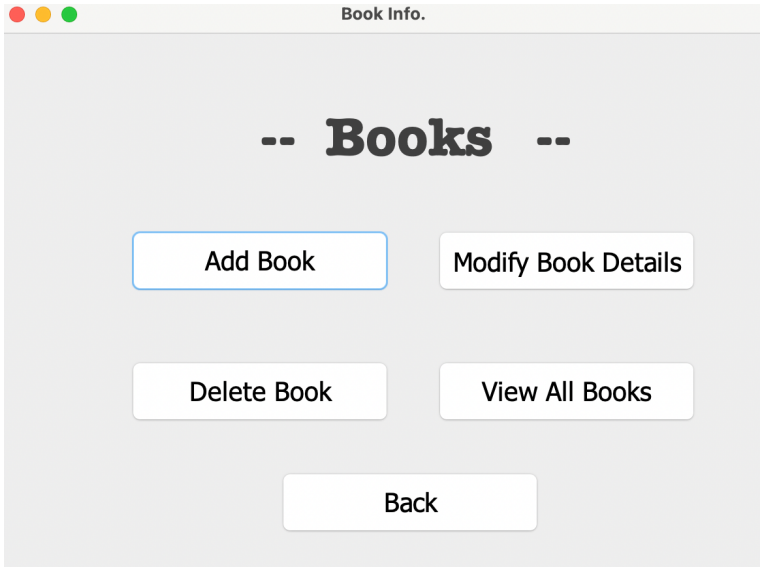


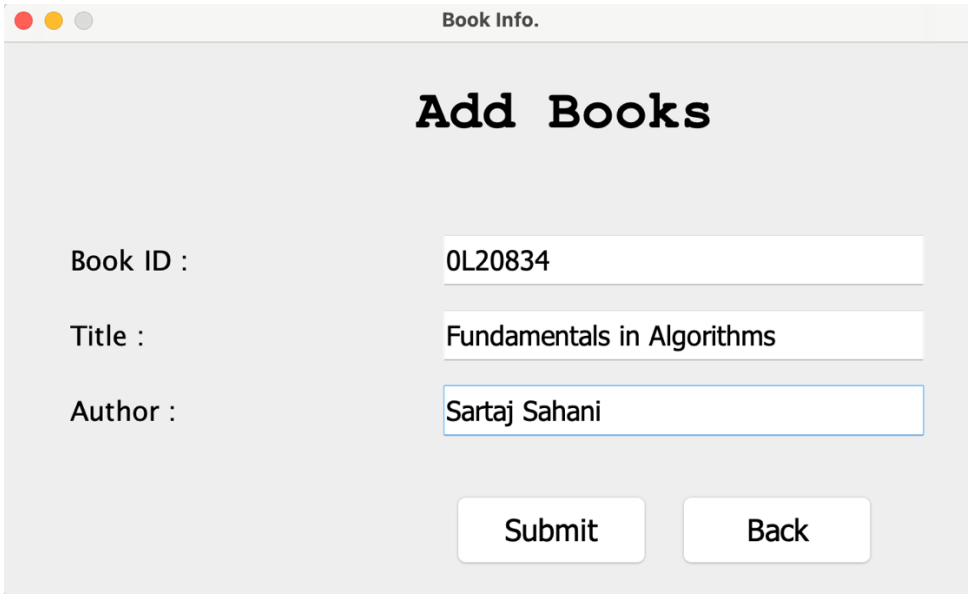
AddBook.class	Today at 2:39 PM	3 KB	Java class file
AddStudent.class	Today at 2:39 PM	4 KB	Java class file
Book.class	Today at 2:39 PM	3 KB	Java class file
Bookdetails.class	Today at 2:39 PM	3 KB	Java class file
ChangePassword.class	Today at 2:39 PM	3 KB	Java class file
DeleteBook.class	Today at 2:39 PM	4 KB	Java class file
DeleteStudent.class	Today at 2:39 PM	4 KB	Java class file
Issue.class	Today at 2:39 PM	4 KB	Java class file
Login.class	Today at 2:39 PM	3 KB	Java class file
Logindetails.class	Today at 2:39 PM	2 KB	Java class file
Menu.class	Today at 2:39 PM	3 KB	Java class file
Return.class	Today at 2:39 PM	4 KB	Java class file
Search.class	Today at 2:39 PM	5 KB	Java class file
SQLCon.class	Today at 2:39 PM	984 bytes	Java class file
Statistics.class	Today at 2:39 PM	4 KB	Java class file
Student.class	Today at 2:39 PM	3 KB	Java class file
Studentdetails.class	Today at 2:39 PM	2 KB	Java class file
Updatebook.class	Today at 2:39 PM	3 KB	Java class file
Updatestudent.class	Today at 2:39 PM	4 KB	Java class file
ViewBook.class	Today at 2:39 PM	3 KB	Java class file
Viewissue.class	Today at 2:39 PM	3 KB	Java class file
Viewreturn.class	Today at 2:39 PM	3 KB	Java class file
ViewStudent.class	Today at 2:39 PM	3 KB	Java class file
vl.png	11-Jun-2022 at 9:15 PM	198 KB	PNG image
vlogo.png	22-May-2022 at 10:01 PM	455 KB	PNG image

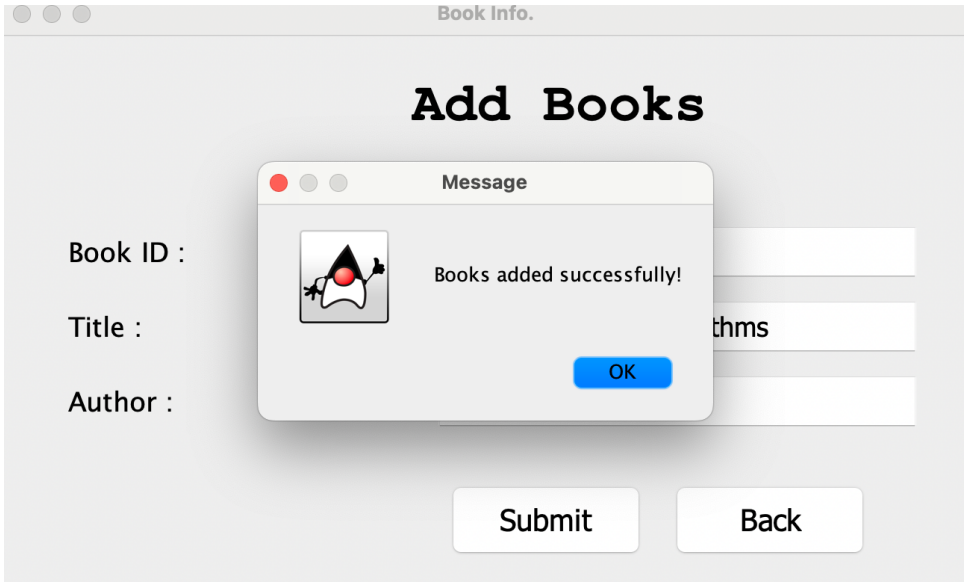
Testing:
Java GUI Testing:




College Library Database
Welcome, Librarian - L001
Books Info
Students Info
Issue Books
Return Books
Search
Statistics
Change Password
Logout

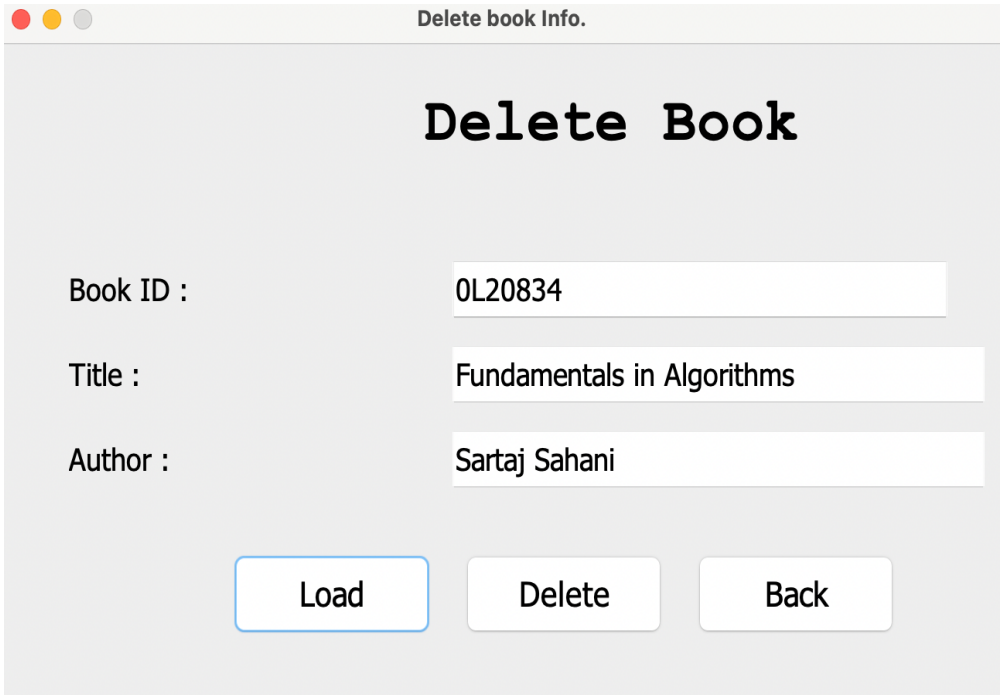

Book Info.
-- Books --
Add Book
Modify Book Details
Delete Book
View All Books
Back

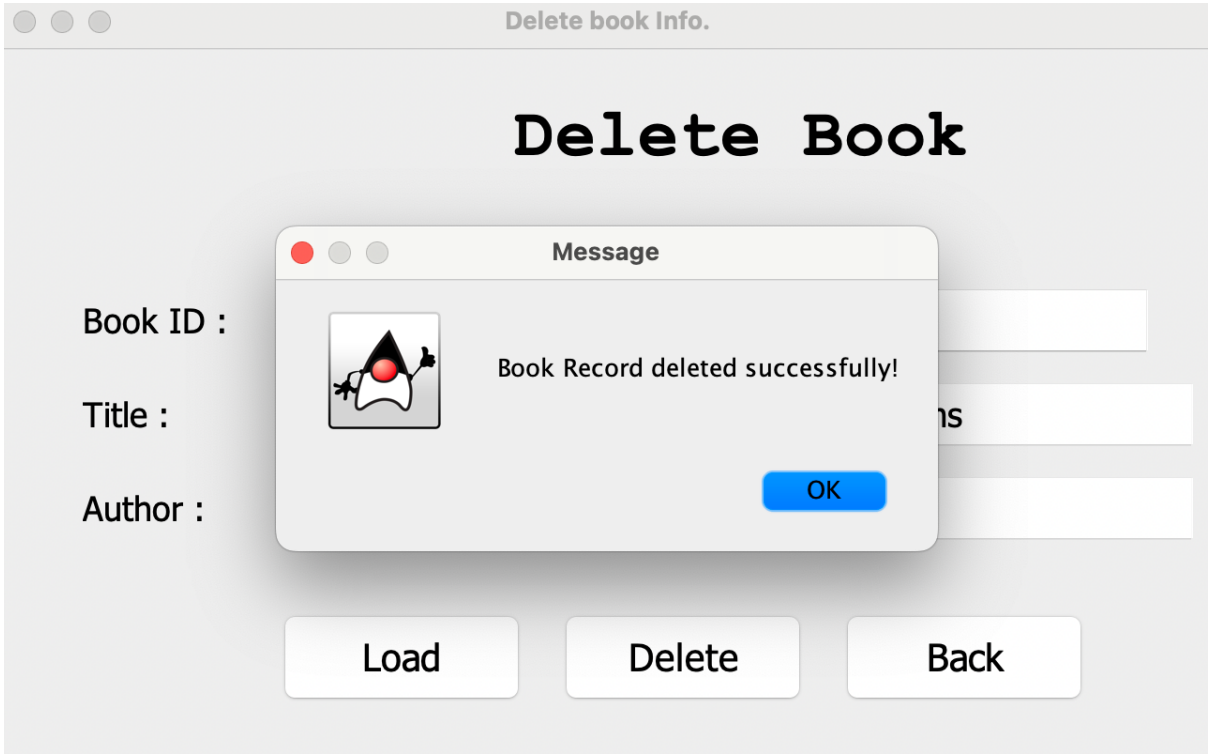


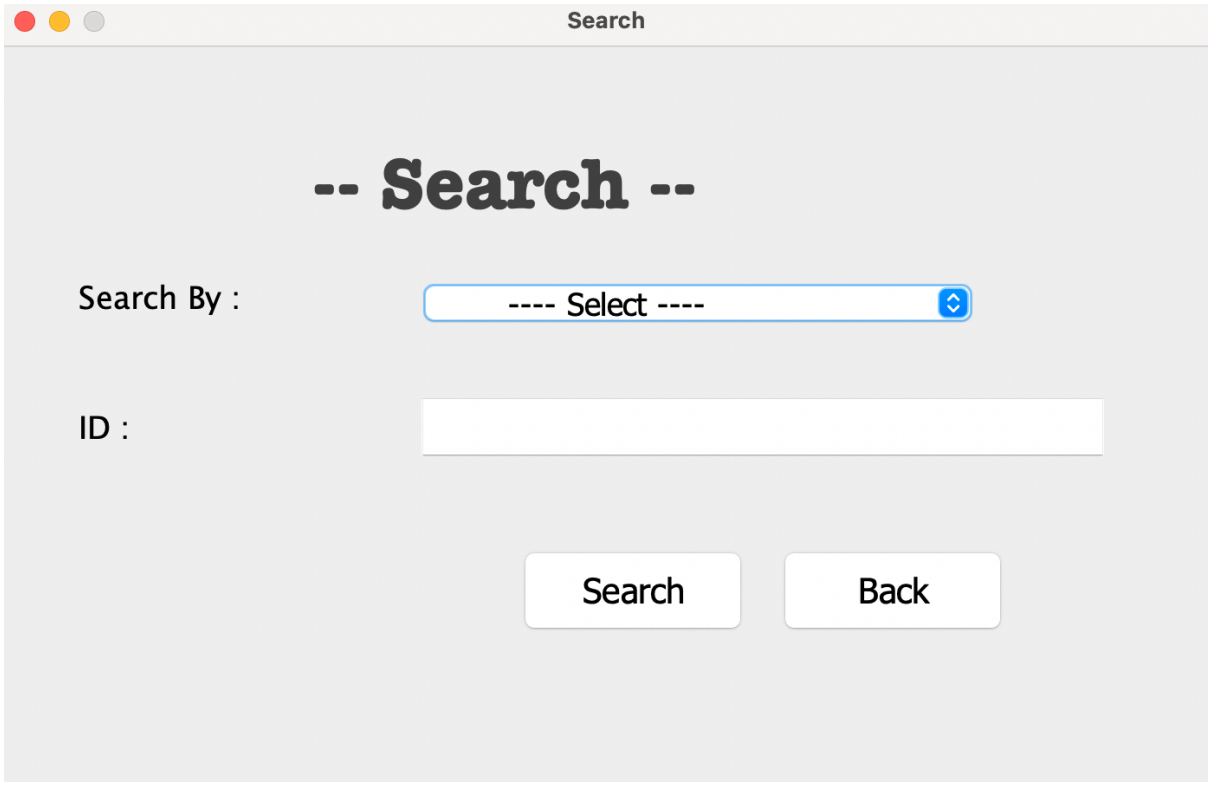


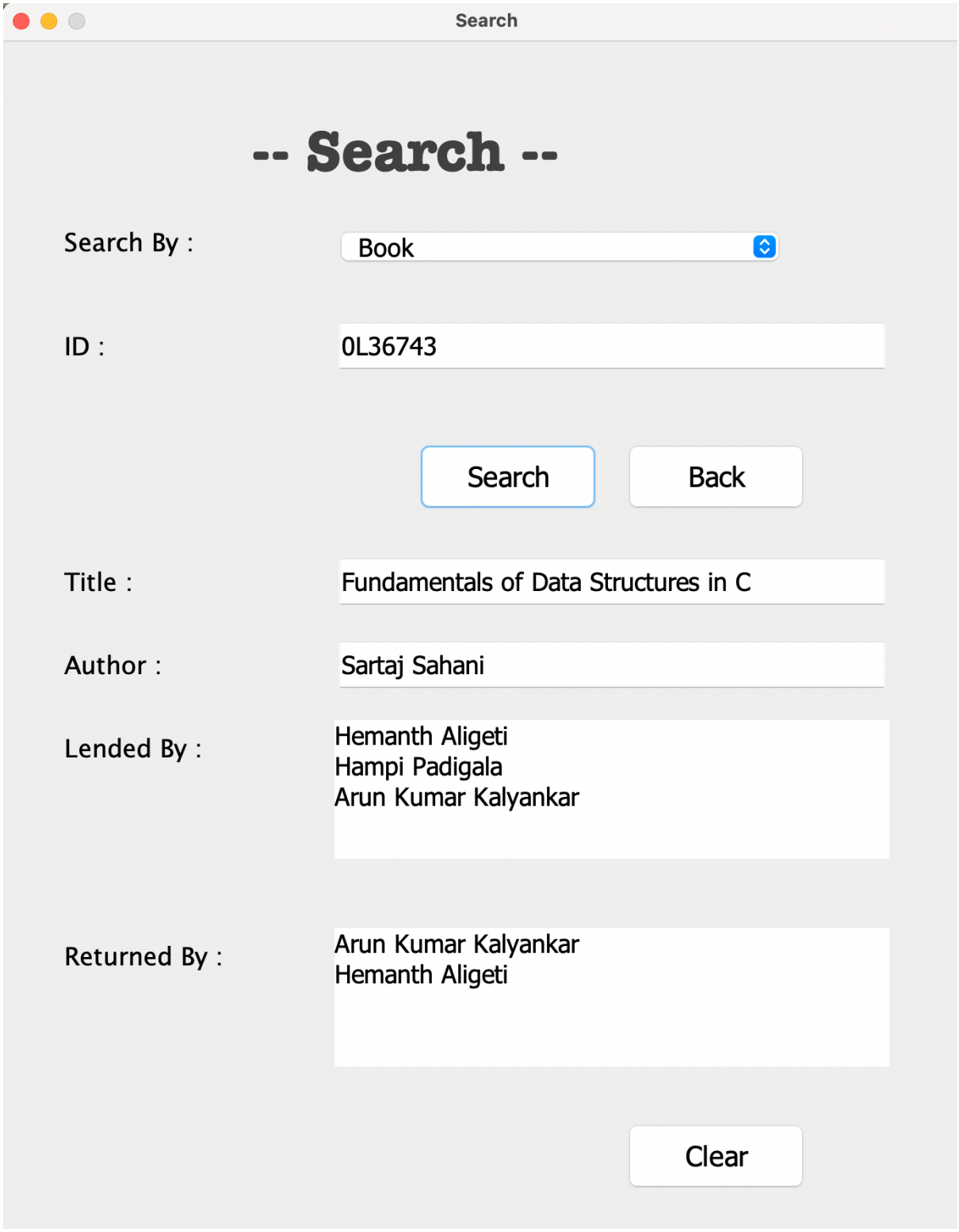
View All Books:

bookid	Title	Author
0L10383	Java : The Complete Reference – 7th Edition	Herbert Schildt
0L12323	Computer Organisation	J.K.Jain
0L13454	Advanced Engineering Mathematics	R K Jain
0L16143	A Basic Course in Environmental Studies	Dr. Surinder Deswal
0L18212	Software Engineering – 6th Edition	Roger S. Pressman
0L20834	Fundamentals in Algorithms	Sartaj Sahani
0L21045	Fundamentals of Computer Algorithms	Sartaj Sahani
0L30057	Fundamentals of Digital Logic with VHDL Design – 3rd Edition	Stephen Brown
0L30101	Computer Organization – 5th Edition	Carl Hamacher
0L30382	Dicrete Mathematics and Its Applications with Combinatorics and Graph Theory ...	Kenneth H Rosen
0L33436	Database System Concepts – 6th Edition	Abraham Silberschatz
0L35085	Fundamentals of Computer Algorithms	Sartaj Sahani
0L35247	Probability, Random Variables andSrtochastic Processes – 4th Edition	Athanasios Papoulis
0L36453	Engineering Ethics includes Human Values	Govindarajan M
0L36743	Fundamentals of Data Structures in C	Sartaj Sahani
0L46882	Database System Concepts – 6th Edition	Abraham Silberschatz

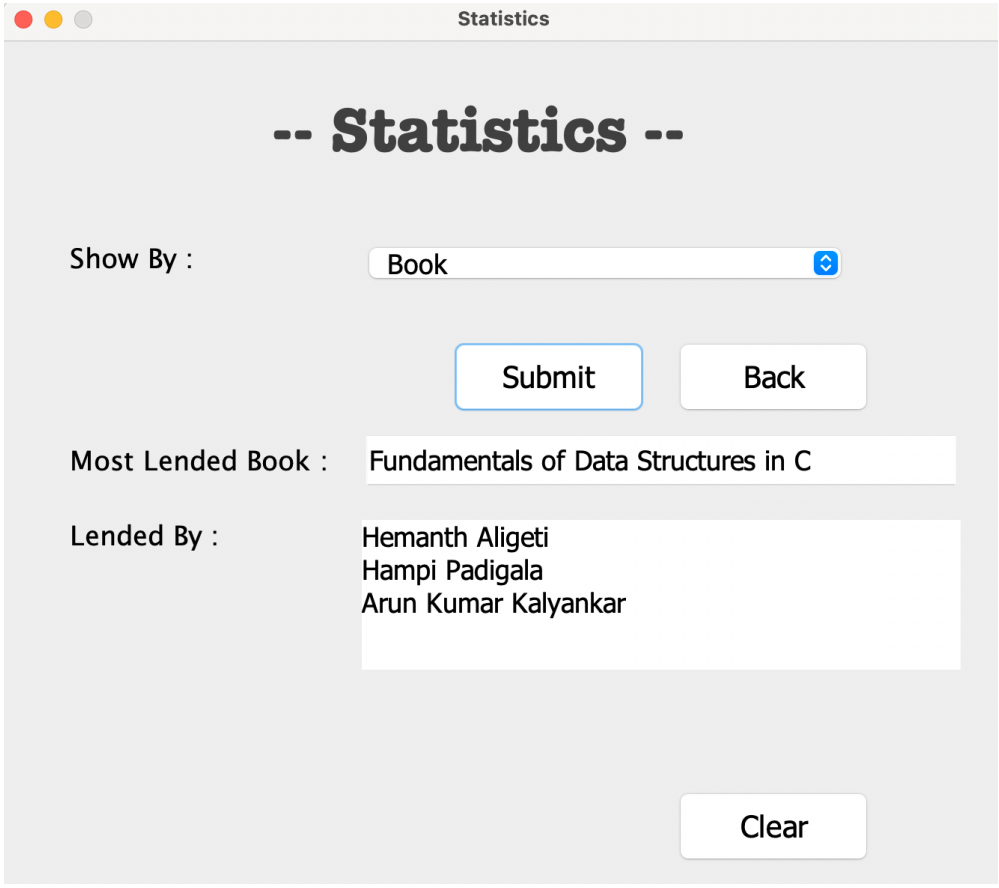

Delete book Info.
Delete Book
Book ID :
0L20834
Title :
Fundamentals in Algorithms
Author :
Sartaj Sahani
Load
Delete
Back

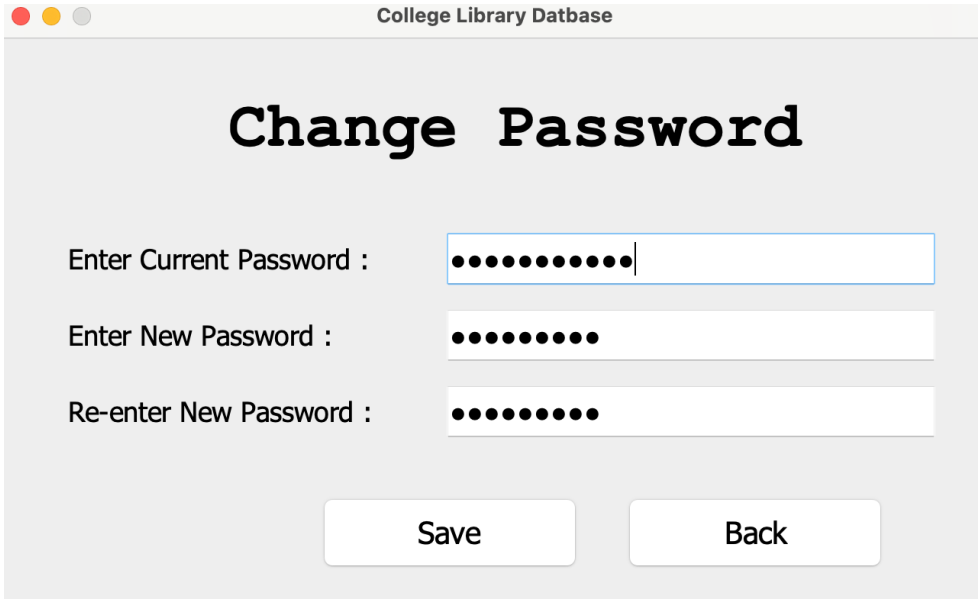

Delete book Info.
Delete Book
Message
Book Record deleted successfully!
OK
Book ID :
Title :
Author :
Load
Delete
Back

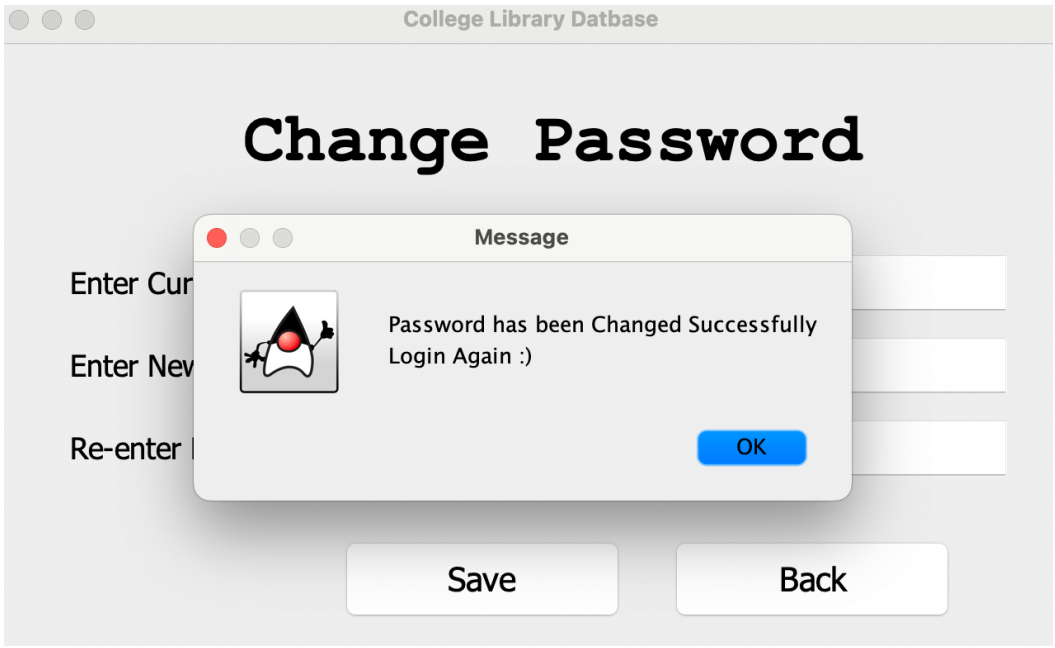

Search
-- Search --
Search By :
---- Select ----
ID :
Search
Back


Search
-- Search --
Search By :
Book
ID :
0L36743
Search
Back
Title :
Fundamentals of Data Structures in C
Author :
Sartaj Sahani
Lended By :
Hemanth Aligeti
Hampi Padigala
Arun Kumar Kalyankar
Returned By :
Arun Kumar Kalyankar
Hemanth Aligeti
Clear


Statistics
-- Statistics --
Show By :
---- Select ----
Submit
Back


Statistics
-- Statistics --
Show By :
Book
Submit
Back
Most Lended Book :
Fundamentals of Data Structures in C
Lended By :
Hemanth Aligeti
Hampi Padigala
Arun Kumar Kalyankar
Clear





The data entered in the above form is updated in the "Book " table of MySQL server V8.0.29

```
mysql> select * from book;
+---------+-----------------------------------------------------------------------------------------+----------------------+
| bookid  | Title                                                                                   | Author               |
+---------+-----------------------------------------------------------------------------------------+----------------------+
| 0L10383 | Java : The Complete Reference - 7th Edition                                             | Herbert Schildt      |
| 0L12323 | Computer Organisation                                                                   | J.K.Jain             |
| 0L13454 | Advanced Engineering Mathematics                                                        | R K Jain             |
| 0L16143 | A Basic Course in Environmental Studies                                                 | Dr. Surinder Deswal  |
| 0L18212 | Software Engineering - 6th Edition                                                      | Roger S. Pressman    |
| 0L20834 | Fundamentals in Algorithms                                                              | Sartaj Sahani        |
| 0L21045 | Fundamentals of Computer Algorithms                                                     | Sartaj Sahani        |
| 0L30057 | Fundamentals of Digital Logic with VHDL Design - 3rd Edition                            | Stephen Brown        |
| 0L30101 | Computer Organization - 5th Edition                                                     | Carl Hamacher        |
| 0L30382 | Dicrete Mathematics and Its Applications with Combinatorics and Graph Theory - 7th Edition | Kenneth H Rosen      |
| 0L33436 | Database System Concepts - 6th Edition                                                  | Abraham Silberschatz |
| 0L35085 | Fundamentals of Computer Algorithms                                                     | Sartaj Sahani        |
| 0L35247 | Probability, Random Variables andSrtochastic Processes - 4th Edition                    | Athanasios Papoulis  |
| 0L36453 | Engineering Ethics includes Human Values                                                | Govindarajan M       |
| 0L36743 | Fundamentals of Data Structures in C                                                    | Sartaj Sahani        |
| 0L46882 | Database System Concepts - 6th Edition                                                  | Abraham Silberschatz |
+---------+-----------------------------------------------------------------------------------------+----------------------+
16 rows in set (0.00 sec)
```

Results:

A Java Swing based – "College Library Database Management System" – SQL connectivity using JDBC project is completed Successfully.

Discussion and Future work:

While doing this project I got new ideas I understood how to work on projects. Now to further extend this project I want to create an android app by which I can control my project on my hand and connect to it.

References:

- https://www.youtube.com/watch?v=jtfuwnN2NoA
- https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html
- https://www.javatpoint.com/mysql-connection
- https://dev.mysql.com/doc/dev/connector-j/8.0/