

C. V. RAMAN GLOBAL UNIVERSITY

MAHURA - 752054, BHUBANESWAR, ODISHA



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Employee Enhancement Programme

Experiential Learning Report on JAVA

TITLE:

**Advanced Library Management System: A Smart and
Efficient Approach**

Under the Guidance of

Mr. Anuj Sharma

CRANES VARSITY

SUBMITTED BY:

GROUP – 4

Team No - 16

GROUP MEMBERS:

CL2025010601911753: PRITANSHU KUMAR MALLICK (TL)
CL2025010601874855: AMAN PRATIK PATTNAIK
CL2025010601936670: ANUPAM UAPAMANYU SATAPATHY
CL2025010601932327: SOURAV KUMAR NAYAK
CL2025010601904199: *SUVAM SUBHADARSHI DAS*

CONTENTS

| TITLE | PAGE NO. |
|----------------------------------|-----------------|
| ❖ ACKNOWLEDGEMENT | 5 |
| ❖ INTRODUCTION | 6 |
| ❖ REQUIREMENTS | 8 |
| ❖ DESIGN | 8 |
| ❖ IMPLEMENTATION | 9 |
| ❖ TESTING | 9 |
| ❖ DEPLOYMENT | 10 |
| ❖ MAINTENANCE | 10 |
| ❖ LOGIN PAGE | 11 |
| ❖ HOME PAGE | 12 |
| ❖ STUDENT REGISTRATION INTERFACE | 12 |
| ❖ STUDENT DETAILS INTERFACE | 13 |
| ❖ ADD NEW BOOK INTERFACE | 13 |
| ❖ ISSUE BOOK INTERFACE | 14 |
| ❖ RETURN BOOK INTERFACE | 14 |
| ❖ BOOKS DETAILS | 15 |
| ❖ STATISTICS INTERFACE | 15 |
| ❖ FLOW CHART | 16 |
| ❖ SOURCE CODE | 17 - 119 |
| ❖ CHALLENGES FACED | 120 |
| ❖ CONCLUSION | 120 |
| ❖ REFERENCES | 121 |

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our **Java Trainer, Mr. Anuj Sharma**, from the **CRANES VARSITY**, for providing us with the opportunity to work on this project. His guidance and support have been invaluable in enhancing our research skills and deepening our understanding of the subject.

We would also like to extend our heartfelt appreciation to other trainers of **Cranes Varsity** for equipping us with the necessary knowledge and resources that played a crucial role in the successful completion of this project. Their training and expertise have significantly contributed to our learning experience.

Furthermore, we express our gratitude to our parents and friends for their unwavering support and encouragement throughout this journey.

This project has been instrumental in bridging the gap between theoretical knowledge and practical application. We sincerely acknowledge the contributions of everyone involved in making this project a success.

INTRODUCTION

Advanced Library Management Software (ALMS) uses the latest technology to manage library functions. In today's modern time, many other digital software have been introduced by different companies to meet today's competition. **For Example: The popular library management software is RFID-based library software also known as library automation software.** (It is important to note that the library automation system's advanced features have changed the look of the library and convenient for students to use it.)

The advanced features of Library Management Software have increased the demand in schools, institutions, universities and many more places. Highly advanced Library Management Software caters to the needs of the technology and value of the research area. It has overcome all the limitations of the traditional library system.

➤ **KEY-FEATURES OF ALMS:**

- Modern integrated system
- Easier access
- Can print barcodes
- Simple, clear search interface
- Multilingual
- Export and import records
- Web-based OPAC System

Library Management Software is based on various modules. Library Management Software system helps in the smooth functioning of the library. E.g.,

- Circulation
- Acquisition
- Periodicate
- Database Management
- Reports
- Budget
- Journal control
- Search OPAC
- System Administrator

Advance Library Management Software is based on two types of technologies:

- **Barcode System:** In this system, a student can return or issue a book with the help of a unique barcode number assigned to every book.
- **RFID System:** In this system, electromagnetic fields are used to identify and track tags attached to books. The chip and antenna attached are called an RFID Tag.

****NB:** The demand for RFID-based Library Management Software has increased with the blend of technology. It has converted the traditional library system to the modern library system. It has been designed in such a way to enhance the system as well as various functions of the library and keep in control the inventory system of the library.

- ❖ **In this project, we have implemented a very small fraction of the Advanced Library Management System i.e., Database Management, System Administrator, Search Online Public Access Catalogue (OPAC) and Reports: Member statistics.**
- ❖ **This LMS Project is only for prototype purposes.**

Technical Details:

Programming Language: **Java**

GUI Framework: **Java Swing and JFrame**

Database Management System: **MySQL Database**

IDE: **NetBeans**

Deployment: **The application is deployed locally**

REQUIREMENTS

BOOK MANAGEMENT:

Add, remove, and update books in the library inventory.

Each book should have a unique identifier (barcode or ISBN), title, author, genre, publication year, and availability status.

MEMBER MANAGEMENT:

Register new library members.

Maintain member records including name, address, contact information, and borrowing history.

Borrowing and Returning Books:

Allow members to borrow books.

Record borrowing and returning transactions.

Calculate overdue fines for late returns.

SEARCH AND BROWSE:

Enable users to search and browse the library catalogue by title, author, genre, and publication year.

REPORTING:

Generate reports on book inventory, member statistics, borrowing trends, and overdue fines

DESIGN

CLASS DESIGN:

Book: Represents a book with properties such as title, author, genre, publication year, and availability status.

MEMBER:

Represents a library member with attributes like name, address, contact information, and borrowing history.

TRANSACTION:

Represents borrowing and returning transactions, including the book borrowed, the member who borrowed it, and the date of the transaction.

LIBRARY:

Manages the library operations such as adding/removing books, registering members, handling transactions, and generating reports.

DATABASE DESIGN:

Design a database schema to store information about books, members, transactions, and other relevant data.

Tables: Books, Members, Transactions, etc.

USER INTERFACE:

Design a user-friendly interface for library staff and members to interact with the system. It include screens/forms for book management, member registration, borrowing/returning books, searching/browsing the catalogue, and generating reports.

IMPLEMENTATION

BACKEND:

The backend of the application handles data storage and retrieval using a relational database management system (RDBMS) like MySQL. Java Database Connectivity (JDBC) is used to connect to the MySQL database and execute SQL queries.

The ConnectMySQL class contains methods to establish connection to the MySQL database. Prepared statements (PreparedStatement) are used to execute parameterized SQL queries to retrieve user information for authentication.

FRONTEND:

The frontend of the application is built using Java Swing, a GUI toolkit for Java. Swing components such as JFrame, JPanel, JLabel, JTextField, JPasswordField, and JButton are used to create the user interface.

The graphical user interface (GUI) includes panels for login (jPanel3) and additional information (jPanel2). Various Swing components are used to create a visually appealing and interactive interface for users to enter their username and password and perform actions such as logging in or closing the application.

INTEGRATION:

The integration between the frontend and backend is achieved through Java code that connects to the MySQL database to authenticate users. When the user clicks the "LOGIN" button (jButton1), an action event is triggered, and the corresponding action listener executes Java code to query the database and authenticate the user. If the authentication is successful, a new window (Home) is opened to proceed with the application. Otherwise, an error message is displayed using JOptionPane.

TESTING

- ❖ Conduct unit tests to verify the functionality of individual components (classes, methods).
- ❖ Perform integration tests to ensure that different parts of the system work together as expected.
- ❖ Conduct user acceptance testing (UAT) with library staff and members to validate the system's usability and functionality.

DEPLOYMENT

- ❖ Deploy the library management system to a production environment.
- ❖ Ensure that the system meets performance, security, and scalability requirements.
- ❖ Provide training to library staff on using the system effectively.

MAINTENANCE

- ❖ Regularly update the system to add new features, fix bugs, and improve performance.
- ❖ Provide ongoing support to address any issues or concerns raised by library staff and members.
- ❖ Stay informed about emerging technologies and trends in library management to keep the system up-to-date.
- ❖ This case study outlines the basic steps involved in designing, implementing, and deploying a library management system in Java. Actual implementation details may vary depending on specific requirements, technology choices, and constraints.

LOGIN PAGE

Purpose of Login Page: The login page serves as the gateway for users to access the Advanced Library Management System (ALMS). It ensures that only authorized users with valid credentials can log in and perform actions within the system.

User Authentication: The login page verifies the identity of users by requesting their username and password. It validates the provided credentials against the database to authenticate users.

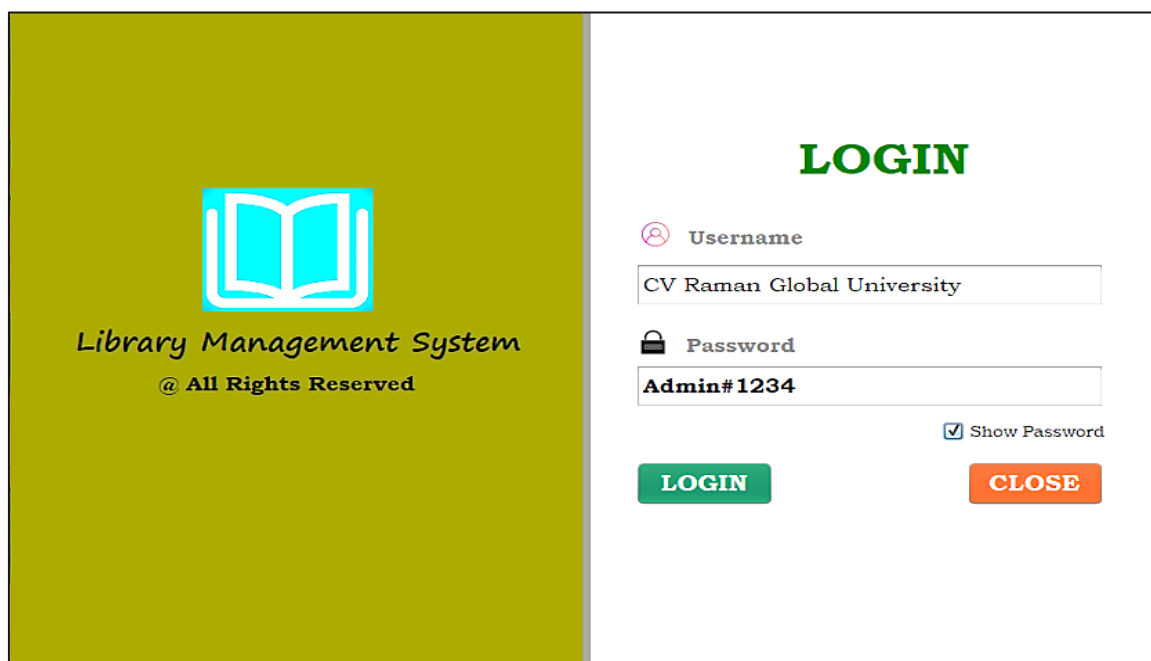
Input Validation: Users are prompted to enter their username and password. Input validation ensures that both fields are filled before proceeding with the login process. If either field is left blank, users are prompted with a message indicating the requirement to fill in both fields.


Connection to Database: Upon entering valid credentials, the login page establishes a connection to the MySQL database. It executes a SQL query to retrieve user information from the database for authentication.

Handling Authentication-Results: If the provided credentials match those stored in the database, users are granted access to the main application interface. In case of invalid credentials, users are notified with a message indicating that the username and password do not match.

User Experience: The design of the login page aims to provide a user-friendly experience, with intuitive input fields and clear instructions. Feedback messages inform users about the outcome of their login attempt, guiding them on how to proceed.


Security Measures: The login page implements security measures such as password hashing and secure database connections to protect user data. It plays a crucial role in safeguarding the ALMS against unauthorized access and potential security threats.






Library Management System
@ All Rights Reserved

LOGIN

 Username
CV Raman Global University

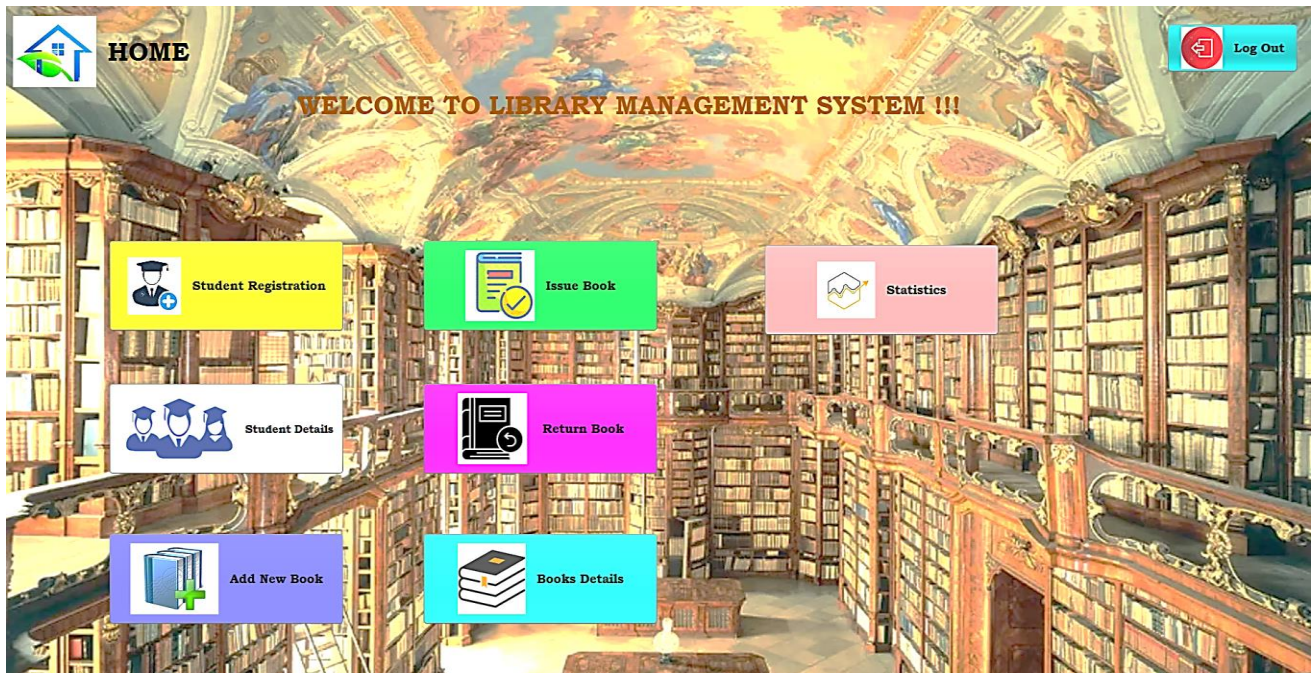
 Password
Admin#1234

☒ Show Password

HOME PAGE

Navigation Menu: The main page serves as the central hub for navigating different functionalities of the Library Management System (AMS). It features a user-friendly navigation menu that categorizes various tasks and operations.

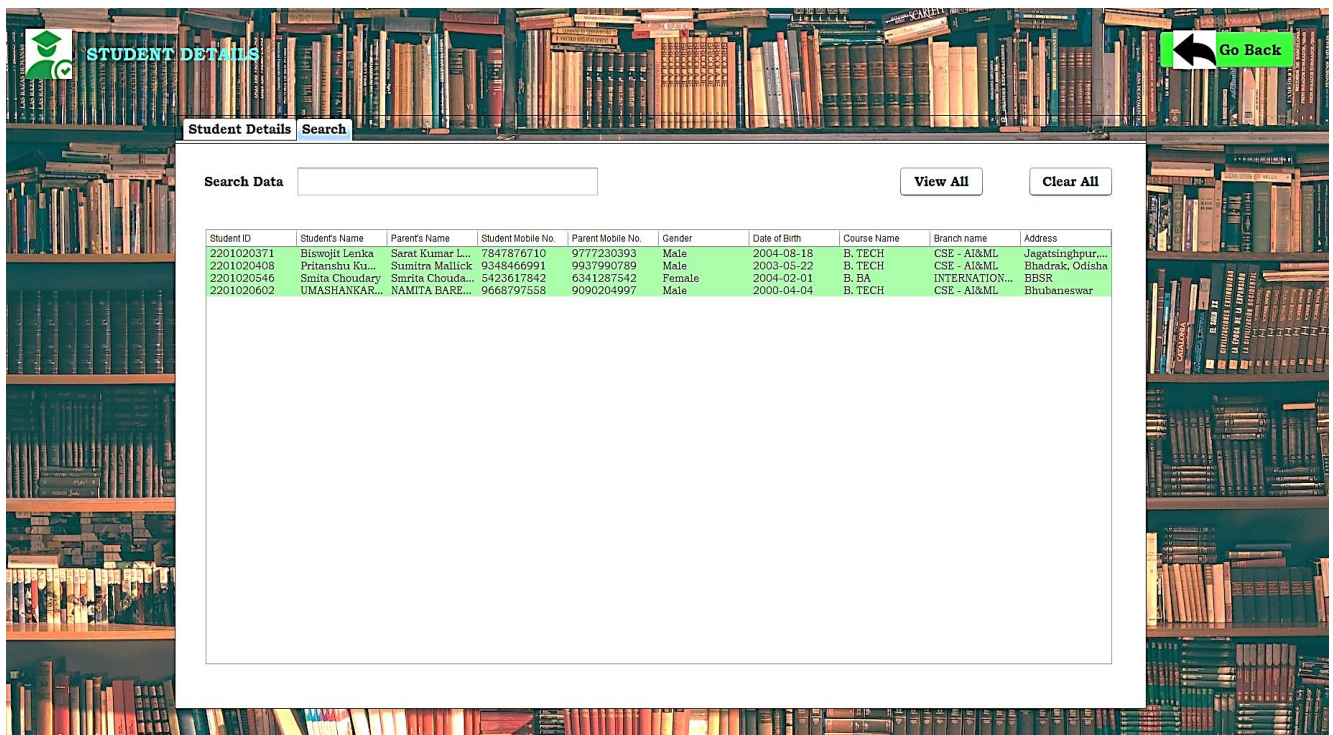
The features are **Student Registration**, **Student Details**, **Add New Book**, **Issue Book**, **Return Book**, **Book Details**, and **Statistics**.




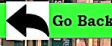
STUDENT REGISTRATION INTERFACE

The screenshot displays the 'STUDENT REGISTRATION' interface. The title 'STUDENT REGISTRATION' is at the top left, and a 'Go Back' button is at the top right. The main form is titled 'Enter Student Details...'. It contains the following fields: 'Student ID' (text input), 'Student's Name' (text input), 'Parent's Name' (text input), 'Student's Mobile Number' (text input), 'Parent's Mobile Number' (text input), 'Gender' (dropdown menu with '--- Select your Gender ---'), 'Date of Birth' (calendar icon), 'Course Name' (dropdown menu with '--- Select your Course ---'), 'Branch Name' (dropdown menu with '--- Select your Branch ---'), and 'Address (Dist./State)' (text input). A 'Save' button with a floppy disk icon is at the bottom of the form. The background is a dense arrangement of books on shelves.

STUDENT DETAILS INTERFACE



**STUDENT DETAILS**



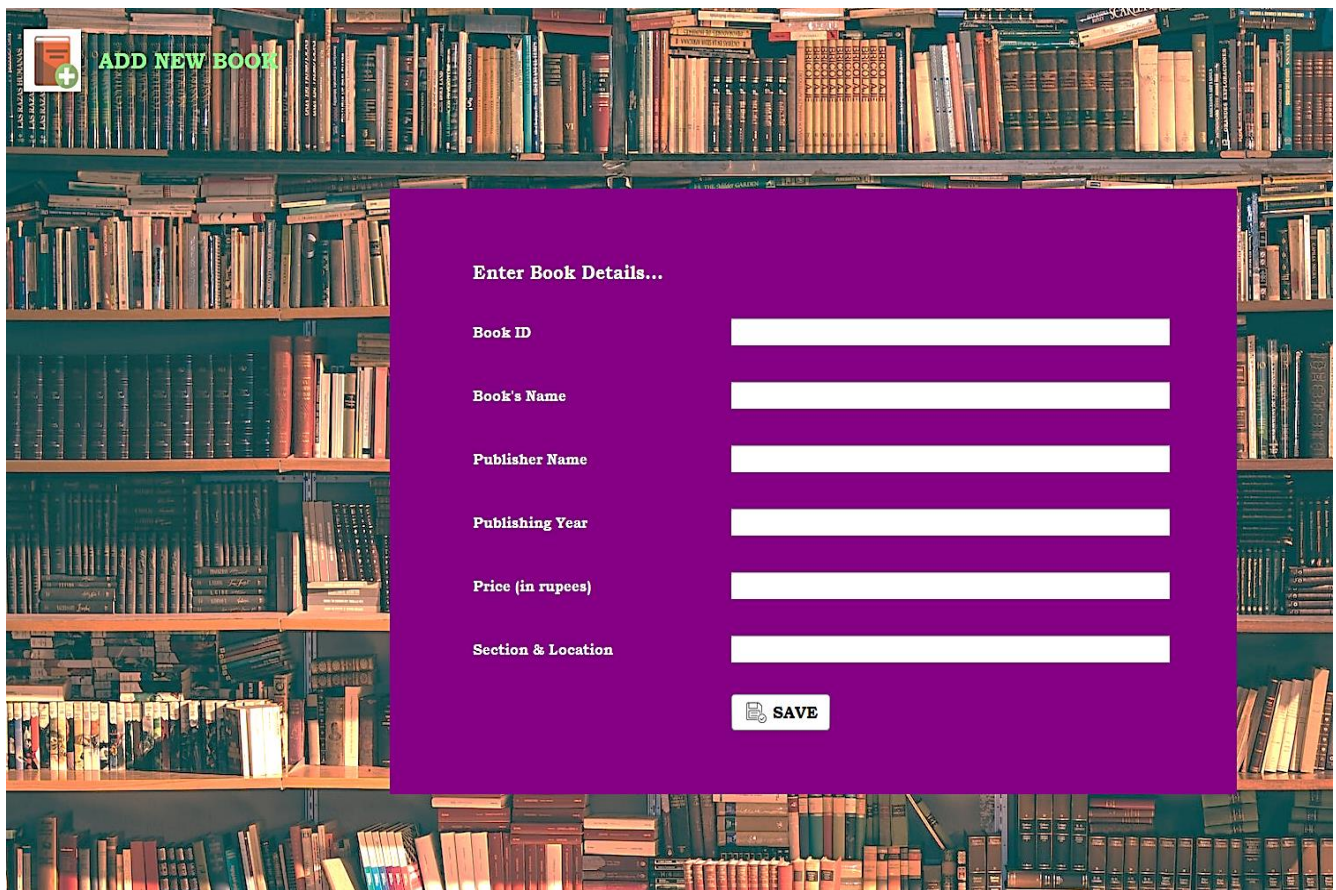
Student Details **Search**


Search Data

View All **Clear All**

| Student ID | Student's Name | Parent's Name | Student Mobile No. | Parent Mobile No. | Gender | Date of Birth | Course Name | Branch name | Address |
|------------|-----------------|------------------|--------------------|-------------------|--------|---------------|-------------|----------------|------------------|
| 2201020371 | Eliswajit Lenka | Sarat Kumar L... | 7847876710 | 9777230393 | Male | 2004-08-18 | B. TECH | CSE - AI&ML | Jagatsinghpur... |
| 2201020408 | Pritanshu Ku... | Sumitra Mallick | 9348466991 | 9937990789 | Male | 2003-05-22 | B. TECH | CSE - AI&ML | Bhadrak, Odisha |
| 2201020546 | Smita Choudary | Smrita Chouda... | 5423617842 | 6341287542 | Female | 2004-02-01 | B. BA | INTERNATION... | BBSR |
| 2201020602 | UMASHANKAR... | NAMITA BARE... | 9668797558 | 9090204997 | Male | 2000-04-04 | B. TECH | CSE - AI&ML | Bhubaneswar |

ADD NEW BOOK INTERFACE



**ADD NEW BOOK**

Enter Book Details...

Book ID


Book's Name

Publisher Name

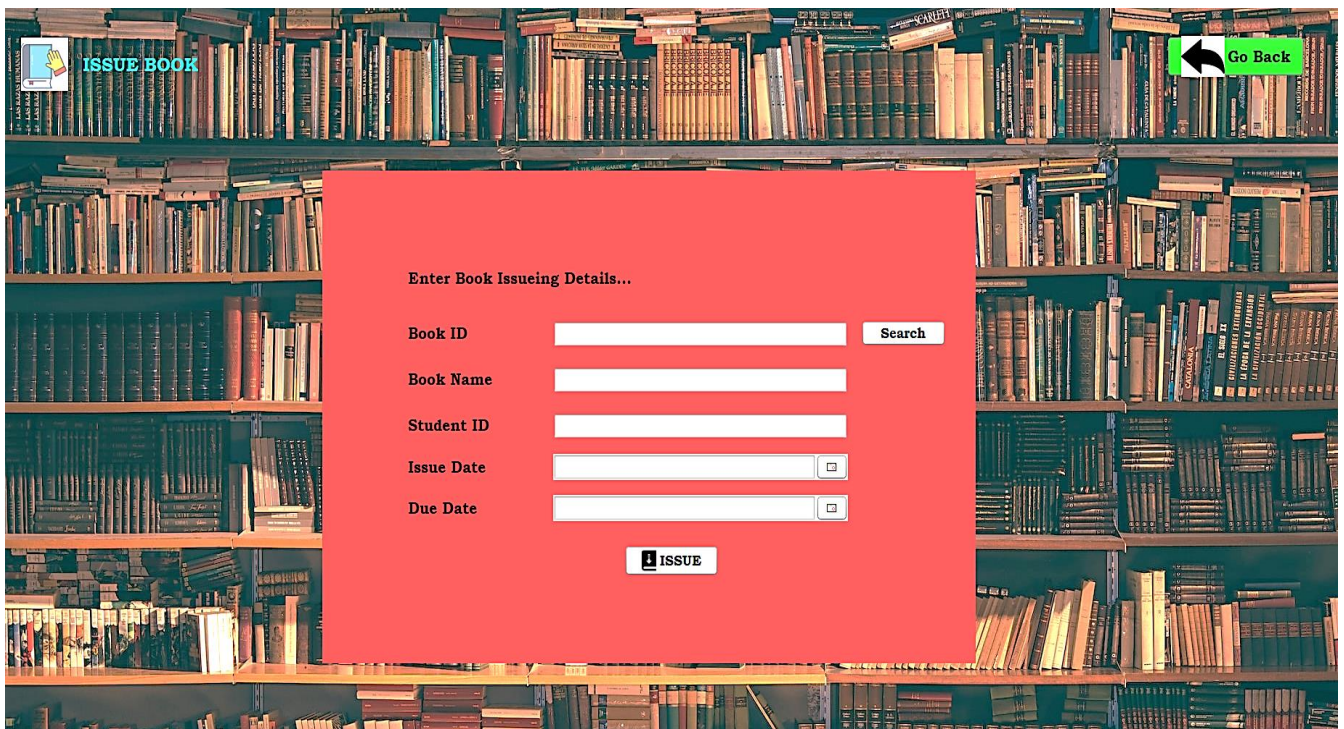
Publishing Year

Price (in rupees)

Section & Location

 **SAVE**

ISSUE BOOK INTERFACE



The interface for issuing a book is displayed over a background of bookshelves. It features a red central form with a title bar, input fields for book and student details, a search button, and an issue button. A 'Go Back' button is in the top right corner.

ISSUE BOOK

Go Back

Enter Book Issuing Details...

Book ID Search

Book Name

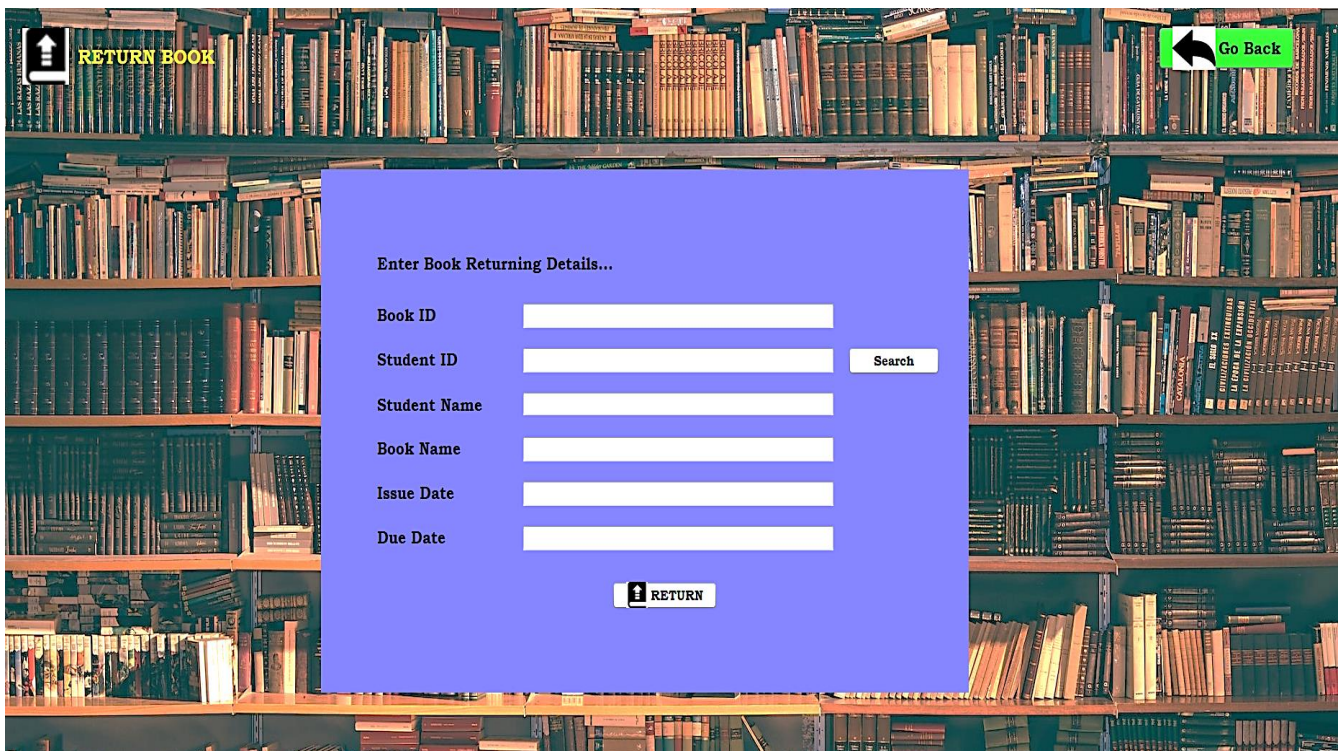
Student ID

Issue Date

Due Date

ISSUE

RETURN BOOK INTERFACE



The interface for returning a book is displayed over a background of bookshelves. It features a blue central form with a title bar, input fields for book and student details, a search button, and a return button. A 'Go Back' button is in the top right corner.

RETURN BOOK

Go Back

Enter Book Returning Details...

Book ID

Student ID Search

Student Name


Book Name


Issue Date

Due Date

RETURN

BOOK DETAILS INTERFACE

 **BOOK DETAILS**




Book Details **Search**

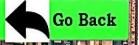
Search Data

View All **Clear All**

| Book ID | Book Name | Publisher | Publishing Year | Price | Section |
|---------|--------------------------------|------------------|-----------------|-------|---------------------|
| 1001 | Harry Potter and the Philos... | J. K. Rowling | 1997 | 500 | Novel, NV/2/23 |
| 1002 | The Alchemist (O Alquimis... | Paulo Coelho | 1988 | 200 | FICTION, FC/6/21 |
| 1003 | Dream of the Red Chamber | Cao Xueqin | 1791 | 600 | Classical, CS/5/32 |
| 1021 | The Hobbit | J. R. R. Tolkien | 1937 | 400 | FANTASY, F/4/1/56/4 |
| 1023 | Harry Potter and the Cham... | J. K. Rowling | 1998 | 500 | NOVEL, NV/2/24 |
| 1035 | Harry Potter and the Order... | J. K. Rowling | 2003 | 500 | NOVEL, NV/2/27 |
| 1042 | The Da Vinci Code | Dan Brown | 2003 | 600 | NOVEL, MN/23/21 |
| 1056 | Harry Potter and the Priso... | J. K. Rowling | 1999 | 500 | NOVEL, NV/2/25 |
| 1089 | Harry Potter and the Goble... | J. K. Rowling | 2000 | 500 | NOVEL, NV/2/26 |

STATISTICS INTERFACE

 **STATISTICS**



ISSUED AND RETURNED DETAILS

Search Issued

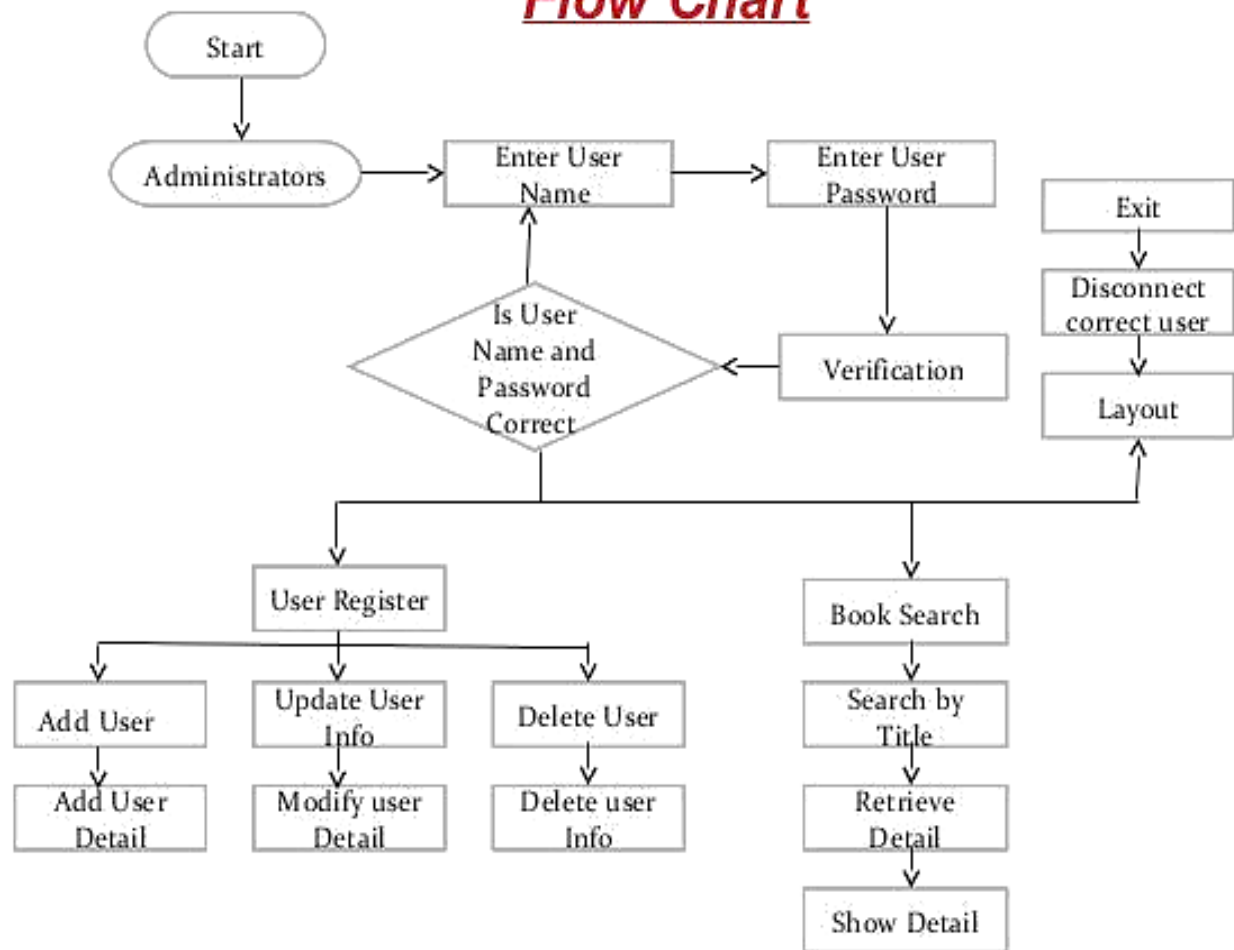
| S_ID | S_Name | Book_ID | Book_Name | Issue_Date | Due_Date |
|------------|----------------|---------|--------------------------|------------|------------|
| 2201020371 | Biswojit Lenka | 1003 | Dream of the Red Chamber | 2024-04-25 | 2024-04-30 |

Search Returned

| S_ID | S_Name | Book_ID | Book_Name | Issue_Date | Due_Date |
|------------|-------------------------|---------|-------------------------------|------------|------------|
| 2201020408 | Pritanshu Kumar Mallick | 1001 | Harry Potter and the Philo... | 2024-04-24 | 2024-04-28 |
| 2201020371 | Biswojit Lenka | 1001 | Harry Potter and the Philo... | 2024-04-24 | 2024-04-27 |

FLOW CHART

Flow Chart



SOURCE CODE

SQL File for creating Database and tables

```
-- Create the database
CREATE DATABASE IF NOT EXISTS advance_lms;
USE library;

-- Create tables
CREATE TABLE IF NOT EXISTS books (
    Book_ID VARCHAR(10) NOT NULL PRIMARY KEY,
    Book_Name VARCHAR(100),
    Publisher VARCHAR(100),
    P_YEAR INT,
    Price INT,
    Section VARCHAR(60)
);

CREATE TABLE IF NOT EXISTS issue (
    Book_ID VARCHAR(10),
    S_ID VARCHAR(10),
    Issue_Date DATE,
    Due_Date DATE,
    ReturnBook VARCHAR(5)
);

CREATE TABLE IF NOT EXISTS login (
    userid VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(20) NOT NULL,
    PRIMARY KEY (password)
);

CREATE TABLE IF NOT EXISTS student_details (
    S_ID BIGINT NOT NULL PRIMARY KEY,
    S_Name VARCHAR(100),
    P_Name VARCHAR(100),
    S_Mob BIGINT UNIQUE,
    P_Mob BIGINT UNIQUE,
    Gender VARCHAR(8) NOT NULL,
    DOB DATE,
    C_Name VARCHAR(70),
    B_Name VARCHAR(70),
    Address VARCHAR(60)
);
```


FOR JDBC Connection

```
package com.cgu.ConnectionProvider;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ConnectMySQL {

    private static final String DB_URL =
"jdbc:mysql://localhost:3306/advance_lms";
    private static final String DB_USER = "root";
    private static final String DB_PASSWORD = "SQL#2023@hxiz408";

    static Connection con = null;

    public static Connection ConnectToDB() {
        try {
            // Load MySQL JDBC Driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Establish Connection
            con = DriverManager.getConnection(DB_URL, DB_USER,
DB_PASSWORD);
            System.out.println("Connection Established
Successfully...");
        }
        catch (ClassNotFoundException ex) {
            System.err.println("MySQL JDBC Driver Not Found!");

            Logger.getLogger(ConnectMySQL.class.getName()).log(Level.SEVERE,
null, ex);
        }
        catch (SQLException ex) {
            System.err.println("Database Connection Failed!");

            Logger.getLogger(ConnectMySQL.class.getName()).log(Level.SEVERE,
null, ex);
        }
        return con;
    }

    public static void runSQLFile(String filePath) {
```

```

        try (Connection conn = ConnectToDB();
            Statement stmt = conn.createStatement();
            BufferedReader reader = new BufferedReader(new
FileReader(filePath))) {

            StringBuilder sql = new StringBuilder();
            String line;

            while ((line = reader.readLine()) != null) {
                line = line.trim();
                // Skip empty lines and comments
                if (line.isEmpty() || line.startsWith("--")) {
                    continue;
                }

                sql.append(line).append(" ");

                // Execute when line ends with semicolon
                if (line.endsWith(";")) {
                    String query = sql.toString();
                    try {
                        stmt.execute(query);
                    } catch (SQLException e) {
                        System.err.println("Error executing query: "
+ query);
                        e.printStackTrace();
                    }
                    sql.setLength(0); // Clear the buffer
                }
            }
            System.out.println("SQL file executed successfully.");
        } catch (IOException e) {
            System.err.println("Error reading SQL file: " +
e.getMessage());
        } catch (SQLException e) {
            System.err.println("SQL execution error: " +
e.getMessage());
        }
    }

    public static void main(String[] args) {
        // Update the path to your SQL file location
        String sqlFilePath =
"C:\\Users\\LENOVO\\Documents\\NetBeansProjects\\Advance Library
Management
System\\src\\com\\cgu\\ConnectionProvider\\DatabaseAndTables.sql";

        // Run the SQL file to create database and tables
        runSQLFile(sqlFilePath);
    }
}

```

FOR LOGIN PAGE

```
package advance.library.management.system;

import ConnectionProvider.ConnectMySQL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

public class Login extends javax.swing.JFrame {

    public Login() {
        initComponents();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
    }

    @SuppressWarnings("unchecked")
    # Self-generated code for JFrame

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton1ActionPerformed
        PreparedStatement pst;
        ResultSet rs;
        Connection c=ConnectMySQL.ConnectToDB();
        try {
            pst=c.prepareStatement("SELECT * FROM advance_lms.login where
userid=? AND password=?");
            pst.setString(1, username.getText());
            pst.setString(2, password.getText());
            rs=pst.executeQuery();
            if(rs.next()){
                setVisible(false);
                new Home().setVisible(true);
            }
            else{
                JOptionPane.showMessageDialog(this, "Please Enter Valied
ID and Password");
            }
        }
        catch (SQLException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE,
null, ex);
        }
    } //GEN-LAST:event_jButton1ActionPerformed

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton2ActionPerformed
        int yes=JOptionPane.showConfirmDialog(this, "Are you really Close
this
```

```

application?", "Exit", JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
    if (yes == JOptionPane.YES_OPTION) {
        System.exit(0);
    }
} //GEN-LAST:event_jButton2ActionPerformed

private void passwordActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_passwordActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_passwordActionPerformed

private void chkboxActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_chkboxActionPerformed
    if (chkbox.isSelected()) {
        password.setEchoChar((char)0); //show
    }
    else {
        password.setEchoChar('\u2022'); //hide
    }
} //GEN-LAST:event_chkboxActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);

```

```

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Login().setVisible(true);
        }
    });
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JCheckBox chkbx;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPasswordField password;
private javax.swing.JTextField username;
// End of variables declaration//GEN-END:variables
}

```

FOR HOME PAGE

```
package advance.library.management.system;
import javax.swing.JFrame;
public class Home extends javax.swing.JFrame {
    public Home() {
        initComponents();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
    }
    @SuppressWarnings("unchecked")
    # Self-generated code for JFrame
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
GEN-FIRST:event_jButton1ActionPerformed
        setVisible(false);
        new StudentRegistration().setVisible(true);
    }
GEN-LAST:event_jButton1ActionPerformed

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
GEN-FIRST:event_jButton2ActionPerformed
        setVisible(false);
        new AddBook().setVisible(true);
    }
GEN-LAST:event_jButton2ActionPerformed

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
{
GEN-FIRST:event_jButton4ActionPerformed
        setVisible(false);
        new ReturnBook().setVisible(true);
    }
GEN-LAST:event_jButton4ActionPerformed

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)
{
GEN-FIRST:event_jButton5ActionPerformed
        setVisible(false);
        new StudentDetail().setVisible(true);
    }
GEN-LAST:event_jButton5ActionPerformed

    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt)
{
GEN-FIRST:event_jButton6ActionPerformed
        setVisible(false);
        new Login().setVisible(true);
    }
GEN-LAST:event_jButton6ActionPerformed

    private void jButton7ActionPerformed(java.awt.event.ActionEvent evt)
{
GEN-FIRST:event_jButton7ActionPerformed
        setVisible(false);
        new BookDetails().setVisible(true);
    }
GEN-LAST:event_jButton7ActionPerformed

    private void jButton8ActionPerformed(java.awt.event.ActionEvent evt)
{
GEN-FIRST:event_jButton8ActionPerformed
        setVisible(false);
        new Statistics().setVisible(true);
    }
GEN-LAST:event_jButton8ActionPerformed

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{
GEN-FIRST:event_jButton3ActionPerformed
```

```

        setVisible(false);
        new IssueBook().setVisible(true);
    } //GEN-LAST:event_jButton3ActionPerformed

    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
        * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;

                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        }
    } //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Home().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;

```

```
private javax.swing.JButton jButton6;  
private javax.swing.JButton jButton7;  
private javax.swing.JButton jButton8;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
// End of variables declaration//GEN-END:variables  
}
```


FOR STUDENT REGISTRATION PAGE

```
package advance.library.management.system;

import ConnectionProvider.ConnectMySQL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTextField;

public class StudentRegistration extends javax.swing.JFrame {

    public StudentRegistration() {
        initComponents();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
    }

    @SuppressWarnings("unchecked")
    # Self-generated code for JFrame

    private void jTextField1ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField1ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField1ActionPerformed

    private void jTextField2ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField2ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField2ActionPerformed

    private void jTextField3ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField3ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField3ActionPerformed

    private void jTextField4ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField4ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField4ActionPerformed

    private void jTextField5ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField5ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField5ActionPerformed

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton1ActionPerformed

        PreparedStatement pst;
        Connection c = ConnectMySQL.ConnectToDB();
```

```

        if(jTextField1.getText().isEmpty()){
            JOptionPane.showMessageDialog(rootPane, "Please enter Student
ID");
            jTextField1.requestFocusInWindow();
        }
        else if(jTextField2.getText().isEmpty()){
            JOptionPane.showMessageDialog(rootPane, "Please enter Student
Name");
            jTextField2.requestFocusInWindow();
        }
        else if(jTextField3.getText().isEmpty()){
            JOptionPane.showMessageDialog(rootPane, "Please enter Parent
Name");
            jTextField3.requestFocusInWindow();
        }
        else if(jTextField4.getText().isEmpty()){
            JOptionPane.showMessageDialog(rootPane, "Please enter Student
Number");
            jTextField4.requestFocusInWindow();
        }
        else if(jTextField5.getText().isEmpty()){
            JOptionPane.showMessageDialog(rootPane, "Please enter Parent
Number");
            jTextField5.requestFocusInWindow();
        }
        else if(jComboBox3.getSelectedItem().equals("--- Select your
Gender ---")) {
            JOptionPane.showMessageDialog(rootPane, "Please select the
gender of the student");
            jComboBox3.requestFocusInWindow();
        }
        else
            if(((JTextField)jDateChooser1.getDateEditor().getUiComponent()).getText().
isEmpty()) {
                JOptionPane.showMessageDialog(rootPane, "Please select a
date");
                jDateChooser1.requestFocusInWindow();
            }
            else if(jComboBox1.getSelectedItem().equals("--- Select your
Course ---")){
                JOptionPane.showMessageDialog(rootPane, "Please select course
of the Student");
                jComboBox1.requestFocusInWindow();
            }
            else if(jComboBox2.getSelectedItem().equals("--- Select your
Branch ---")){
                JOptionPane.showMessageDialog(rootPane, "Please select Branch
of the Student");
                jComboBox2.requestFocusInWindow();
            }
            else if(jTextField6.getText().isEmpty()){
                JOptionPane.showMessageDialog(rootPane, "Please enter Address
of the Student");
                jTextField6.requestFocusInWindow();
            }
    }

```

```

        else {
            try {
                String sql = "INSERT INTO advance_lms.student_details
(S_ID, S_Name, P_Name, S_Mob, P_Mob, Gender, DOB, C_Name, B_Name, Address)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
                pst = c.prepareStatement(sql);
                pst.setString(1, jTextField1.getText());
                pst.setString(2, jTextField2.getText());
                pst.setString(3, jTextField3.getText());
                pst.setString(4, jTextField4.getText());
                pst.setString(5, jTextField5.getText());
                pst.setString(6, jComboBox3.getSelectedItem().toString());
                pst.setString(7,
((JTextField)jDateChooser1.getDateEditor().getUiComponent()).getText());
                pst.setString(8, jComboBox1.getSelectedItem().toString());
                pst.setString(9, jComboBox2.getSelectedItem().toString());
                pst.setString(10, jTextField6.getText());
                pst.executeUpdate();
                showSuccess("Record Successfully Saved");
                clear(); // Assuming this method clears all text fields
after successful insertion
            }
            catch (SQLException ex) {
                Logger.getLogger(Login.class.getName()).log(Level.SEVERE,
null, ex);
                showError("Failed to save record. Please try again
later.");
                clear();
            }
        }
    } //GEN-LAST:event_jButton1ActionPerformed

    private void showError(String message) {
        JOptionPane.showMessageDialog(rootPane, message, "Error",
JOptionPane.ERROR_MESSAGE);
    }

    private void showSuccess(String message) {
        JOptionPane.showMessageDialog(rootPane, message, "Success",
JOptionPane.INFORMATION_MESSAGE);
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton2ActionPerformed
        setVisible(false);
        new Home().setVisible(true);
    } //GEN-LAST:event_jButton2ActionPerformed

    private void jComboBox3ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jComboBox3ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jComboBox3ActionPerformed

    private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jComboBox1ActionPerformed

```

```

        // TODO add your handling code here:
    } //GEN-LAST:event_jComboBox1ActionPerformed

    private void jComboBox2ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jComboBox2ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jComboBox2ActionPerformed

    private void jTextField6ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField6ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField6ActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel
        setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with
        the default look and feel.
         * For details see
        http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;

                }
            }
        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(StudentRegistration.class.getName()).lo
            g(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(StudentRegistration.class.getName()).lo
            g(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(StudentRegistration.class.getName()).lo
            g(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(StudentRegistration.class.getName()).lo
            g(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {

```

```

        new StudentRegistration().setVisible(true);
    }
});
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JComboBox<String> jComboBox1;
private javax.swing.JComboBox<String> jComboBox2;
private javax.swing.JComboBox<String> jComboBox3;
private com.toedter.calendar.JDateChooser jDateChooser1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
// End of variables declaration//GEN-END:variables
private void clear() {
    jTextField1.setText(""); // Clear Student ID field
    jTextField2.setText(""); // Clear Student Name field
    jTextField3.setText(""); // Clear Parent Name field
    jTextField4.setText(""); // Clear Student Number field
    jTextField5.setText(""); // Clear Parent Number field
    jTextField6.setText(""); // Clear Address field
    jComboBox1.setSelectedIndex(0); // Reset Course ComboBox
    jComboBox2.setSelectedIndex(0); // Reset Branch ComboBox
    jComboBox3.setSelectedIndex(0); // Reset Gender ComboBox
    jDateChooser1.setDate(null); // Clear Date Chooser
}
}

```

FOR STUDENT DETAILS PAGE

```
package advance.library.management.system;

import ConnectionProvider.ConnectMySQL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JFrame;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.RowFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

public class StudentDetail extends javax.swing.JFrame {
    public StudentDetail() {
        initComponents();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
    }

    @SuppressWarnings("unchecked")
    # Self-generated code for JFrame

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_jButton3ActionPerformed
        DefaultTableModel model =
(DefaultTableModel)student_details.getModel();
        model.setRowCount(0);
        model.setRowCount(5);
    //GEN-LAST:event_jButton3ActionPerformed

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_jButton4ActionPerformed
        String sql = "SELECT * FROM advance_lms.student_details";
        try {
            Connection conn = ConnectMySQL.ConnectToDB();
            PreparedStatement pst = conn.prepareStatement(sql);
            ResultSet rs = pst.executeQuery();
            DefaultTableModel obj = (DefaultTableModel)jTable1.getModel();
            obj.setRowCount(0);
            while (rs.next()) {
                obj.addRow(new String[]{
                    rs.getString("S_ID"),
                    rs.getString("S_Name"),
                    rs.getString("P_Name"),
                    rs.getString("S_Mob"),
                    rs.getString("P_Mob"),
                    rs.getString("Gender"),
                    rs.getString("DOB"),
                    rs.getString("C_Name"),
                });
            }
        } catch (SQLException ex) {
            Logger.getLogger(StudentDetail.class.getName()).log(Level.SEVERE, null, ex);
        }
    //GEN-LAST:event_jButton4ActionPerformed
}
```

```

        rs.getString("B_Name"),
        rs.getString("Address")
    });
    }
}
catch (SQLException ex) {

Logger.getLogger(StudentRegistration.class.getName()).log(Level.SEVERE,
null, ex);
    showError("Failed to show record. Please try again later.");
}
} //GEN-LAST:event_jButton4ActionPerformed

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton5ActionPerformed
    DefaultTableModel obj = (DefaultTableModel) jTable1.getModel();
    obj.setRowCount(0);
    obj.setRowCount(5);
} //GEN-LAST:event_jButton5ActionPerformed

private void jTextField1KeyReleased(java.awt.event.KeyEvent evt)
{ //GEN-FIRST:event_jTextField1KeyReleased
    DefaultTableModel obj = (DefaultTableModel) jTable1.getModel();
    TableRowSorter<DefaultTableModel> obj1 = new
TableRowSorter<>(obj);
    jTable1.setRowSorter(obj1);
    obj1.setRowFilter(RowFilter.regexFilter(jTextField1.getText()));
} //GEN-LAST:event_jTextField1KeyReleased

private void showError(String message) {
    JOptionPane.showMessageDialog(rootPane, message, "Error",
JOptionPane.ERROR_MESSAGE);
}

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

```

```

java.util.logging.Logger.getLogger(StudentDetail.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(StudentDetail.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(StudentDetail.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(StudentDetail.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new StudentDetail().setVisible(true);
        }
    });
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTable student_details;
// End of variables declaration//GEN-END:variables
}

```


FOR ADD BOOK PAGE

```
package advance.library.management.system;

import ConnectionProvider.ConnectMySQL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTextField;

public class AddBook extends javax.swing.JFrame {

    public AddBook() {
        initComponents();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
    }

    @SuppressWarnings("unchecked")
    # Self-generated code for JFrame

    private void showError(String message) {
        JOptionPane.showMessageDialog(rootPane, message, "Error",
JOptionPane.ERROR_MESSAGE);
    }
    private void showSuccess(String message) {
        JOptionPane.showMessageDialog(rootPane, message, "Success",
JOptionPane.INFORMATION_MESSAGE);
    }
    private void clear() {
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");
        jTextField4.setText("");
        jTextField5.setText("");
        jTextField6.setText("");
    }
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
```

```

javax.swing.UIManager.setLookAndFeel(info.getClassName());
        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(AddBook.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(AddBook.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(AddBook.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(AddBook.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
    }
//</editor-fold>
/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new AddBook().setVisible(true);
    }
});
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
// End of variables declaration//GEN-END:variables
}

```

FOR ISSUE BOOK PAGE

```
package advance.library.management.system;

import ConnectionProvider.ConnectMySQL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTextField;

public class IssueBook extends javax.swing.JFrame {
    PreparedStatement pst;
    ResultSet rs;
    Connection c=ConnectMySQL.ConnectToDB();

    public IssueBook() {
        initComponents();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
    }

    @SuppressWarnings("unchecked")
    # Self-generated code for JFrame

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton3ActionPerformed
        try {
            pst=c.prepareStatement("SELECT * FROM advance_lms.books where
Book_ID=?");
            pst.setString(1, jTextField1.getText());
            rs=pst.executeQuery();
            if(rs.next()){
                jTextField3.setText(rs.getString("Book_Name"));
            }
            else{
                JOptionPane.showMessageDialog(this, "Please Enter Valid
Book ID");
            }
        }
        catch (SQLException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE,
null, ex);
        }
    } //GEN-LAST:event_jButton3ActionPerformed

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton1ActionPerformed
        if(jTextField1.getText().equals("")){
            JOptionPane.showMessageDialog(rootPane, "Please enter Book
ID and Search it again");
        }
    }
}
```

```

        jTextField1.requestFocus();
    }
    else{
        try{
            String sql = "SELECT * FROM advance_lms.books WHERE
Book_ID=?";

            pst = c.prepareStatement(sql);
            pst.setString(1, jTextField1.getText());
            rs = pst.executeQuery();
            if(rs.next()){
                String sql1 = "SELECT * FROM
advance_lms.Student_Details WHERE S_ID=?";
                pst = c.prepareStatement(sql1);
                pst.setString(1, jTextField2.getText());
                ResultSet rsl = pst.executeQuery();
                if(rsl.next()){
                    String sql2 = "INSERT INTO advance_lms.Issue
(Book_ID, S_ID, Issue_Date, Due_Date, ReturnBook) VALUES (?, ?, ?, ?, ?)";
                    pst = c.prepareStatement(sql2);
                    pst.setString(1, jTextField1.getText());
                    pst.setString(2, jTextField2.getText());
                    pst.setString(3,
((JTextField)jDateChooser1.getDateEditor().getUiComponent()).getText());
                    pst.setString(4,
((JTextField)jDateChooser2.getDateEditor().getUiComponent()).getText());
                    pst.setString(5, "NO");
                    pst.executeUpdate();
                    JOptionPane.showMessageDialog(null, "Book
Successfully Issued.");
                    clear();
                }
                else{
                    JOptionPane.showMessageDialog(null, "Incorrect
Student ID.");
                }
            }
            else{
                JOptionPane.showMessageDialog(null, "Incorrect Book
ID.");
            }
        }
        catch (SQLException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE,
null, ex);
            JOptionPane.showMessageDialog(null, "Connection Error.");
        }
    }
} //GEN-LAST:event_jButtonActionPerformed

private void clear() {
    jTextField1.setText("");
    jTextField3.setText("");
    jTextField2.setText("");
    jDateChooser1.setDate(null);
    jDateChooser2.setDate(null);
}

```

```

    }

    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
        * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(IssueBook.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(IssueBook.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(IssueBook.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(IssueBook.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new IssueBook().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private com.toedter.calendar.JDateChooser jDateChooser1;
    private com.toedter.calendar.JDateChooser jDateChooser2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;

```

```
private javax.swing.JLabel jLabel14;  
private javax.swing.JLabel jLabel15;  
private javax.swing.JLabel jLabel16;  
private javax.swing.JLabel jLabel17;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JTextField jTextField1;  
private javax.swing.JTextField jTextField2;  
private javax.swing.JTextField jTextField3;  
// End of variables declaration//GEN-END:variables  
}
```

FOR RETURN BOOK PAGE

```
package advance.library.management.system;

import ConnectionProvider.ConnectMySQL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

public class ReturnBook extends javax.swing.JFrame {
    Connection c=ConnectMySQL.ConnectToDB();
    PreparedStatement pst;
    PreparedStatement pst1;
    PreparedStatement pst2;
    ResultSet rs;
    ResultSet rsl;
    ResultSet rsl1;

    public ReturnBook() {
        initComponents();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
    }

    @SuppressWarnings("unchecked")
    # Self-generated code for JFrame

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton3ActionPerformed
        try {
            String sql = "UPDATE advance_lms.issue SET ReturnBook='YES' WHERE
Book_ID=? AND S_ID=?";
            pst = c.prepareStatement(sql);
            pst.setString(1, jTextField1.getText());
            pst.setString(2, jTextField2.getText()); // Set the second
parameter here
            int rowsUpdated = pst.executeUpdate(); // Use executeUpdate for
update queries
            if (rowsUpdated > 0) {
                JOptionPane.showMessageDialog(this, "Book Successfully
returned.");
                clear(); // Clear text fields if the update was successful
            } else {
                JOptionPane.showMessageDialog(this, "No rows were updated.");
            }
        } catch (SQLException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null,
ex);
            JOptionPane.showMessageDialog(null, "Connection Error.");
        }
    } //GEN-LAST:event_jButton3ActionPerformed
```

```

private void clear() {
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    jTextField6.setText("");
}

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(ReturnBook.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ReturnBook.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ReturnBook.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(ReturnBook.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ReturnBook().setVisible(true);
        }
    });
}

```



```
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel2;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
// End of variables declaration//GEN-END:variables
```

```
}
```

FOR BOOK DETAILS PAGE

```
package advance.library.management.system;

import ConnectionProvider.ConnectMySQL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JFrame;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.RowFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

public class BookDetails extends javax.swing.JFrame {

    public BookDetails() {
        initComponents();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
    }

    @SuppressWarnings("unchecked")
    # Self-generated code for JFrame

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_jButton1ActionPerformed
        String sql = "SELECT * FROM advance_lms.books";
        try {
            Connection conn = ConnectMySQL.ConnectToDB();
            PreparedStatement pst = conn.prepareStatement(sql);
            ResultSet rs = pst.executeQuery();
            DefaultTableModel model =
(DefaultTableModel)book_details.getModel();
            model.setRowCount(0);
            while (rs.next()) {
                model.addRow(new String[]{
                    rs.getString("Book_ID"),
                    rs.getString("Book_Name"),
                    rs.getString("Publisher"),
                    rs.getString("P_Year"),
                    rs.getString("Price"),
                    rs.getString("Section"),
                });
            }
        }
        catch (SQLException ex) {

            Logger.getLogger(StudentRegistration.class.getName()).log(Level.SEVERE,
            null, ex);

            showError("Failed to show record. Please try again later.");

        }
    }
    //GEN-LAST:event_jButton1ActionPerformed
}
```

```

        private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{
//GEN-FIRST:event_jButton3ActionPerformed
    DefaultTableModel model =
(DefaultTableModel)book_details.getModel();
    model.setRowCount(0);
    model.setRowCount(5);
}
//GEN-LAST:event_jButton3ActionPerformed

        private void jTextField1KeyReleased(java.awt.event.KeyEvent evt)
{
//GEN-FIRST:event_jTextField1KeyReleased
    DefaultTableModel obj = (DefaultTableModel)jTable1.getModel();
    TableRowSorter<DefaultTableModel> obj1 = new
TableRowSorter<>(obj);
    jTable1.setRowSorter(obj1);
    obj1.setRowFilter(RowFilter.regexFilter(jTextField1.getText()));
}
//GEN-LAST:event_jTextField1KeyReleased

        private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
{
//GEN-FIRST:event_jButton4ActionPerformed
    String sql = "SELECT * FROM advance_lms.books";
    try {
        Connection conn = ConnectMySQL.ConnectToDB();
        PreparedStatement pst = conn.prepareStatement(sql);
        ResultSet rs = pst.executeQuery();
        DefaultTableModel obj = (DefaultTableModel)jTable1.getModel();
        obj.setRowCount(0);
        while (rs.next()) {
            obj.addRow(new String[]{
                rs.getString("Book_ID"),
                rs.getString("Book_Name"),
                rs.getString("Publisher"),
                rs.getString("P_Year"),
                rs.getString("Price"),
                rs.getString("Section"),
            });
        }
    }
    catch (SQLException ex) {

Logger.getLogger(StudentRegistration.class.getName()).log(Level.SEVERE,
null, ex);

        showError("Failed to show record. Please try again later.");
    }
}
//GEN-LAST:event_jButton4ActionPerformed

        private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)
{
//GEN-FIRST:event_jButton5ActionPerformed
    DefaultTableModel obj = (DefaultTableModel)jTable1.getModel();
    obj.setRowCount(0);
    obj.setRowCount(5);
}
//GEN-LAST:event_jButton5ActionPerformed

        private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
//GEN-FIRST:event_jButton2ActionPerformed

```

```

        setVisible(false);
        new Home().setVisible(true);
    } //GEN-LAST:event_jButton2ActionPerformed

    private void showError(String message) {
        JOptionPane.showMessageDialog(rootPane, message, "Error",
JOptionPane.ERROR_MESSAGE);
    }

    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;

                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(BookDetails.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(BookDetails.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(BookDetails.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(BookDetails.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new BookDetails().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables

```

```
private javax.swing.JTable book_details;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField1;
// End of variables declaration//GEN-END:variables
```

```
}
```

FOR STATISTICS PAGE

```
package advance.library.management.system;

import ConnectionProvider.ConnectMySQL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JFrame;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.RowFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;
import net.proteanit.sql.DbUtils;

public class Statistics extends javax.swing.JFrame {
    Connection con = ConnectMySQL.ConnectToDB();
    PreparedStatement pst;
    ResultSet rs;
    ResultSet rsl;

    public Statistics() {
        initComponents();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
    }

    @SuppressWarnings("unchecked")
    # Self-generated code for JFrame

    private void jTextField2KeyReleased(java.awt.event.KeyEvent evt)
    { //GEN-FIRST:event_jTextField2KeyReleased
        DefaultTableModel obj =
        (DefaultTableModel) returned_details.getModel();
        TableRowSorter<DefaultTableModel> obj1 = new
        TableRowSorter<>(obj);
        returned_details.setRowSorter(obj1);
        obj1.setRowFilter(RowFilter.regexFilter(jTextField2.getText()));
    } //GEN-LAST:event_jTextField2KeyReleased

    private void jTextField1KeyReleased(java.awt.event.KeyEvent evt)
    { //GEN-FIRST:event_jTextField1KeyReleased
        DefaultTableModel obj =
        (DefaultTableModel) issue_details.getModel();
        TableRowSorter<DefaultTableModel> obj1 = new
        TableRowSorter<>(obj);
        issue_details.setRowSorter(obj1);
        obj1.setRowFilter(RowFilter.regexFilter(jTextField1.getText()));
    } //GEN-LAST:event_jTextField1KeyReleased

    private void formComponentShown(java.awt.event.ComponentEvent evt)
    { //GEN-FIRST:event_formComponentShown
        try{
```

```

        String sql = "SELECT issue.S_ID, student_details.S_Name,
issue.Book_ID, books.Book_Name, issue.Issue_Date, issue.Due_Date FROM
student_details INNER JOIN books INNER JOIN issue WHERE
books.Book_ID=issue.Book_ID AND student_details.S_ID=issue.S_ID AND
issue.ReturnBook='NO'";
        pst = con.prepareStatement(sql);
        rs = pst.executeQuery();
        issue_details.setModel(DbUtils.resultSetToTableModel(rs));
        String sql1 = "SELECT issue.S_ID, student_details.S_Name,
issue.Book_ID, books.Book_Name, issue.Issue_Date, issue.Due_Date FROM
student_details INNER JOIN books INNER JOIN issue WHERE
books.Book_ID=issue.Book_ID AND student_details.S_ID=issue.S_ID AND
issue.ReturnBook='YES'";
        pst = con.prepareStatement(sql1);
        rsl = pst.executeQuery();
        returned_details.setModel(DbUtils.resultSetToTableModel(rsl));
    }
    catch (SQLException ex){
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null,
ex);
        JOptionPane.showMessageDialog(null, "Connection Error.");
    }
}
//GEN-LAST:event_formComponentShown
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Statistics.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Statistics.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Statistics.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```
java.util.logging.Logger.getLogger(Statistics.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Statistics().setVisible(true);
        }
    });
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JTable issue_details;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JTabbedPane jTabbedPane2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTable returned_details;
// End of variables declaration//GEN-END:variables
}
```


CHALLENGES FACED

- ❖ **Database Connectivity:** Establishing a connection to the database (MySQL) might have posed challenges, especially with ensuring the correct JDBC driver is installed and the connection URL is properly configured.
- ❖ **SQL Queries:** Constructing SQL queries for database operations such as inserting, updating, or querying data could have been complex, especially if you encountered syntax errors or issues with parameterized queries.
- ❖ **UI Design and Implementation:** Designing the user interface (UI) using Swing components requires careful planning and layout management. You might have faced challenges with positioning components, managing their properties, or handling user interactions effectively.
- ❖ **Event Handling:** Implementing event listeners for UI components, such as buttons and date choosers, to respond to user actions accurately could have been tricky, especially if you encountered issues with event propagation or handling.
- ❖ **File Handling:** Managing file operations, such as reading images from the file system, could have presented challenges, particularly with handling exceptions, file paths, and byte array conversions.
- ❖ **Exception Handling:** Dealing with exceptions, both checked and unchecked, throughout the project is essential for robustness. You might have faced challenges with identifying and handling exceptions effectively, especially in database operations and file handling.
- ❖ **Testing and Debugging:** Testing the application thoroughly to identify and fix bugs is crucial. Debugging issues related to UI behavior, database interactions, or data persistence could have been time-consuming.

CONCLUSION

In conclusion, this project has provided valuable insights into the process of developing a Java application for the Advanced Library Management System. Through this project, we have achieved the following:

- ❖ **Database Integration:** Successfully integrated MySQL database to store customer information, demonstrating proficiency in database connectivity and SQL operations.
- ❖ **User Interface Design:** Designed a user-friendly interface using Java Swing components, showcasing skills in UI layout management and event handling.
- ❖ **File Handling and Image Processing:** Implemented file handling functionalities to upload customer photos, showcasing proficiency in handling file operations and image processing.
- ❖ **Exception Handling:** Demonstrated the ability to handle exceptions effectively, ensuring the robustness and reliability of the application.

REFERENCES

- ❖ **Stack Overflow & Git-hub:** Online community for programmers
- ❖ **GeeksforGeeks:** Educational website providing articles, tutorials, and coding challenges related to programming.
- ❖ **YouTube Tutorials:** YouTube channels or tutorials referred to for learning Java Swing, MySQL integration, or related topics.
- ❖ **Java Swing Tutorial by Oracle:** Oracle's official tutorial for learning Java Swing GUI components.
- ❖ **MySQL Tutorial by MySQL:** Official tutorials and documentation provided by MySQL for learning SQL and database management.
- ❖ **Java Programming Book:** Thinking in Java.

