Report of Mini Project

Title of project: **Library Management System**

Name of student: **Divya Rajendra Bhadane**

Roll no: **13118.**

Aim: To design and implement a GUI-based **Library Management System** using **Java (Swing)** as frontend and **MySQL** as backend to manage books, members, and book issue-return records efficiently.

Use Case:

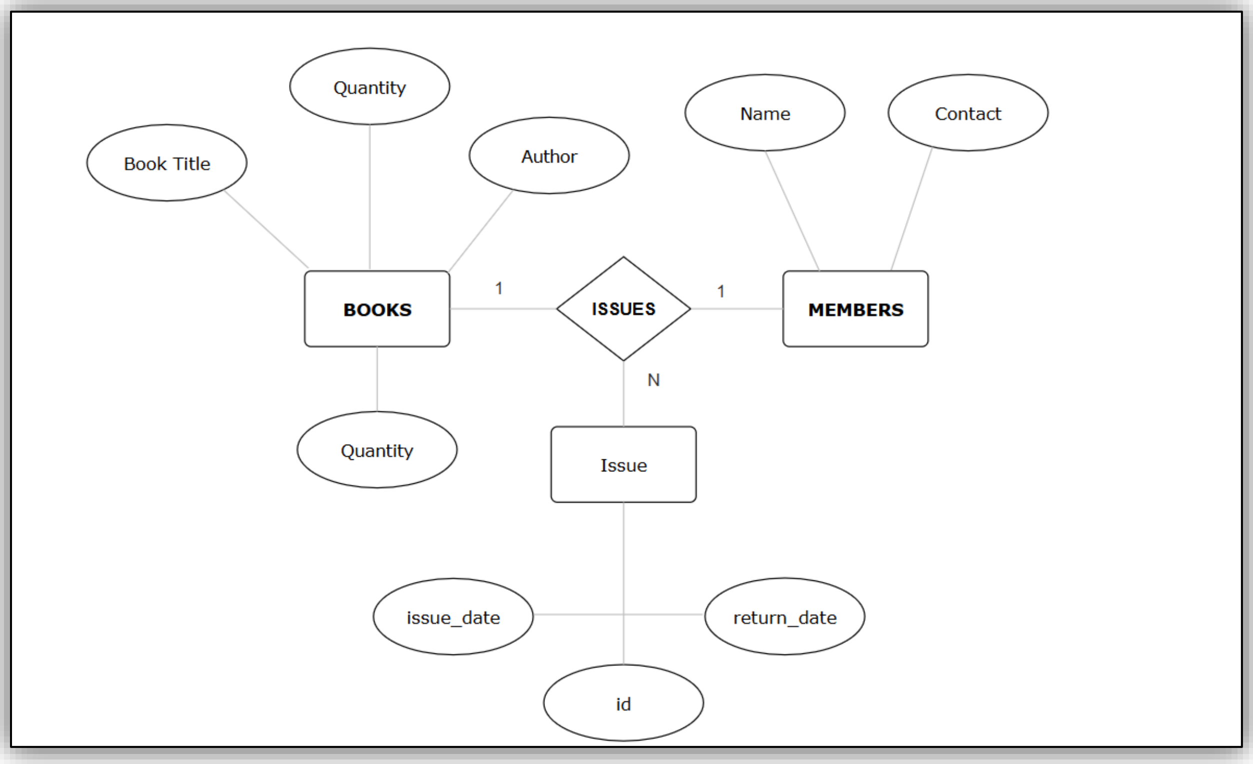
This system allows the librarian to:

- Add, view, and delete **book records**
- Add, view, and delete **member details**
- Issue and return books to members
- Maintain complete issue history in the database

Software Requirements:

Front-End: Java (Swing/AWT GUI)
Back-End: MySQL Database
IDE Used: NetBeans IDE
JDK Version: JDK 8 or above
Connector: MySQL Connector JAR (mysql-connector-j-9.5.0.jar)

ER diagram:



Entities and Attributes:

1. **Books** – Stores book details.
   - **Attributes:** Book ID, Title, Author, Quantity
2. **Members** – Stores member details.
   - **Attributes:** Member ID, Name, Contact
3. **Issue** – Records issued books.
   - **Attributes:** Issue ID, Book ID, Member ID, Issue Date, Return Date

Relationships:

- A **Member** can issue **many Books**.
- A **Book** can be issued by **many Members**.
- The **Issue** table acts as a linking entity (many-to-many relationship between Books and Members).

Mysql tables:

Library Management System: Backend Mysql Database Creation and Table Creation

```
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\MySQL\MySQL Server 5.5\bin>mysql -h localhost -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.5.16 MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

--CREATE DATABASE lms;
Query OK, 1 row affected (0.00 sec)
```

```
USE lms;
Database changed
--CREATE TABLE books (
id INT AUTO_INCREMENT PRIMARY KEY,
title VARCHAR(100),
author VARCHAR(100),
quantity INT);
Query OK, 0 rows affected (0.02 sec)
--CREATE TABLE members (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50),
    contact VARCHAR(15));
Query OK, 0 rows affected (0.01 sec)
--CREATE TABLE issue (
    id INT AUTO_INCREMENT PRIMARY KEY,
    book_id INT,
    member_id INT,
    issue_date DATE,
    return_date DATE,
    FOREIGN KEY (book_id) REFERENCES books(id),
    FOREIGN KEY (member_id) REFERENCES members(id)
Query OK, 0 rows affected (0.02 sec)
```

Front end code:

Library Management System: Frontend Java Code

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

public class LibraryManagement extends JFrame {

    static final String URL = "jdbc:mysql://localhost:3306/lms";
    static final String USER = "root";
    static final String PASSWORD = "root";

    Connection con;
    PreparedStatement pst;
    ResultSet rs;

    DefaultTableModel bookModel, memberModel, issueModel;

    public LibraryManagement() {
        setTitle("Library Management System");
        setSize(850, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JTabbedPane tabs = new JTabbedPane();
        tabs.addTab("Books", createBookPanel());
        tabs.addTab("Members", createMemberPanel());
        tabs.addTab("Issue Books", createIssuePanel());

        add(tabs);

        try {
            con = DriverManager.getConnection(URL, USER,
PASSWORD);
            System.out.println("Connected to Database Successfully!");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Database Connection
Failed: " + e.getMessage());
        }

        setVisible(true);
    }
```

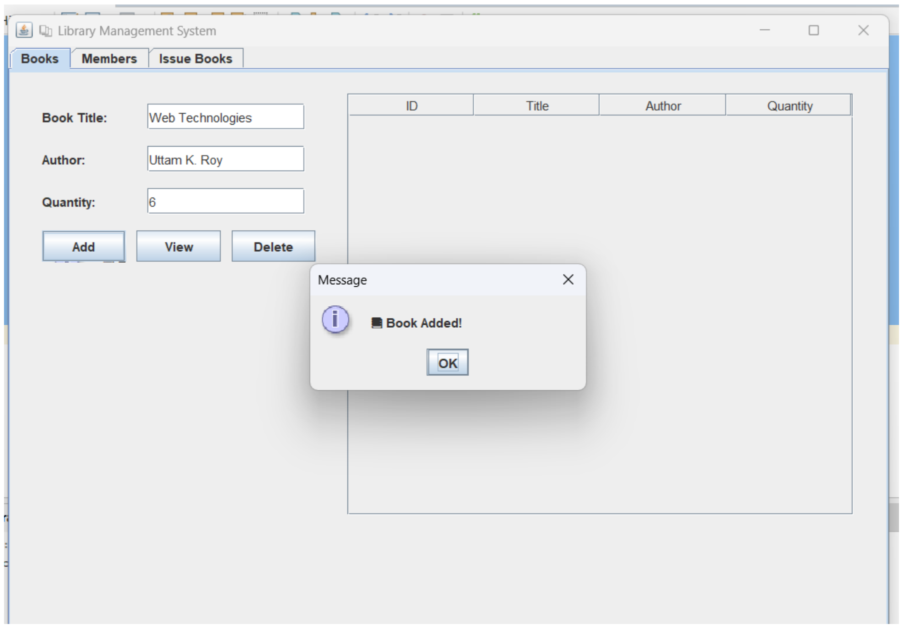
```
JPanel createBookPanel() {
    JPanel panel = new JPanel(null);

    JLabel lblTitle = new JLabel("Book Title:");
    lblTitle.setBounds(30, 30, 100, 25);
    JTextField txtTitle = new JTextField();
    txtTitle.setBounds(130, 30, 150, 25);

    JLabel lblAuthor = new JLabel("Author:");
    lblAuthor.setBounds(30, 70, 100, 25);
    JTextField txtAuthor = new JTextField();
    txtAuthor.setBounds(130, 70, 150, 25);

    JLabel lblQty = new JLabel("Quantity:");
    lblQty.setBounds(30, 110, 100, 25);
    JTextField txtQty = new JTextField();
    txtQty.setBounds(130, 110, 150, 25);

    JButton btnAdd = new JButton("Add");
    btnAdd.setBounds(30, 150, 80, 30);
    JButton btnView = new JButton("View");
```

```
        btnView.setBounds(120, 150, 80, 30);
        JButton btnDelete = new JButton("Delete");
        btnDelete.setBounds(210, 150, 80, 30);

        bookModel = new DefaultTableModel(new String[]{"ID", "Title",
"Author", "Quantity"}, 0);
        JTable table = new JTable(bookModel);
        JScrollPane scroll = new JScrollPane(table);
        scroll.setBounds(320, 20, 480, 400);

        panel.add(lblTitle); panel.add(txtTitle);
        panel.add(lblAuthor); panel.add(txtAuthor);
        panel.add(lblQty); panel.add(txtQty);
        panel.add(btnAdd); panel.add(btnView); panel.add(btnDelete);
        panel.add(scroll);

        btnAdd.addActionListener(e -> {
            try {
                pst = con.prepareStatement("INSERT INTO books(title,
author, quantity) VALUES (?, ?, ?)");
                pst.setString(1, txtTitle.getText());
                pst.setString(2, txtAuthor.getText());
                pst.setInt(3, Integer.parseInt(txtQty.getText()));
                pst.executeUpdate();
                JOptionPane.showMessageDialog(this, "Book Added!");
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(this, ex.getMessage());
            }
        });
        btnView.addActionListener(e -> loadTable("books", bookModel));
        btnDelete.addActionListener(e -> deleteRow("books", table,
bookModel));

        return panel;
}
JPanel createMemberPanel() {
        JPanel panel = new JPanel(null);

        JLabel lblName = new JLabel("Name:");
        lblName.setBounds(30, 30, 100, 25);
        JTextField txtName = new JTextField();
        txtName.setBounds(130, 30, 150, 25);

        JLabel lblContact = new JLabel("Contact:");
        lblContact.setBounds(30, 70, 100, 25);
```

```
        JTextField txtContact = new JTextField();
        txtContact.setBounds(130, 70, 150, 25);

        JButton btnAdd = new JButton("Add");
        btnAdd.setBounds(30, 110, 80, 30);
        JButton btnView = new JButton("View");
        btnView.setBounds(120, 110, 80, 30);
        JButton btnDelete = new JButton("Delete");
        btnDelete.setBounds(210, 110, 80, 30);

        memberModel = new DefaultTableModel(new String[]{"ID",
"Name", "Contact"}, 0);
        JTable table = new JTable(memberModel);
        JScrollPane scroll = new JScrollPane(table);
        scroll.setBounds(320, 20, 480, 400);

        panel.add(lblName); panel.add(txtName);
        panel.add(lblContact); panel.add(txtContact);
        panel.add(btnAdd); panel.add(btnView); panel.add(btnDelete);
        panel.add(scroll);

        btnAdd.addActionListener(e -> {
            try {
                pst = con.prepareStatement("INSERT INTO members(name,
contact) VALUES (?, ?)");
                pst.setString(1, txtName.getText());
                pst.setString(2, txtContact.getText());
                pst.executeUpdate();
                JOptionPane.showMessageDialog(this, "Member Added!");
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(this, ex.getMessage());
            }
        });
        btnView.addActionListener(e -> loadTable("members",
memberModel));
        btnDelete.addActionListener(e -> deleteRow("members", table,
memberModel));

        return panel;
    }

    JPanel createIssuePanel() {
        JPanel panel = new JPanel(null);

        JLabel lblBookId = new JLabel("Book ID:");
        lblBookId.setBounds(30, 30, 100, 25);
        JTextField txtBookId = new JTextField();
        txtBookId.setBounds(130, 30, 150, 25);

        JLabel lblMemberId = new JLabel("Member ID:");
        lblMemberId.setBounds(30, 70, 100, 25);
        JTextField txtMemberId = new JTextField();
        txtMemberId.setBounds(130, 70, 150, 25);

        JLabel lblDate = new JLabel("Return Date (YYYY-MM-DD):");
        lblDate.setBounds(30, 110, 200, 25);
        JTextField txtDate = new JTextField();
        txtDate.setBounds(230, 110, 100, 25);

        JButton btnIssue = new JButton("Issue Book");
        btnIssue.setBounds(30, 150, 100, 30);
        JButton btnView = new JButton("View");
        btnView.setBounds(140, 150, 80, 30);

        issueModel = new DefaultTableModel(new String[]{"ID", "Book
ID", "Member ID", "Issue Date", "Return Date"}, 0);
        JTable table = new JTable(issueModel);
        JScrollPane scroll = new JScrollPane(table);
        scroll.setBounds(320, 20, 480, 400);

        panel.add(lblBookId); panel.add(txtBookId);
        panel.add(lblMemberId); panel.add(txtMemberId);
        panel.add(lblDate); panel.add(txtDate);
        panel.add(btnIssue); panel.add(btnView);
        panel.add(scroll);
        btnIssue.addActionListener(e -> {
            try {
                pst = con.prepareStatement("INSERT INTO issue(book_id,
    member_id, issue_date, return_date) VALUES (?, ?, CURDATE(), ?)");
                pst.setInt(1, Integer.parseInt(txtBookId.getText()));
                pst.setInt(2, Integer.parseInt(txtMemberId.getText()));
                pst.setString(3, txtDate.getText());
                pst.executeUpdate();
                JOptionPane.showMessageDialog(this, "Book Issued!");
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(this, ex.getMessage());
            }
        });
        btnView.addActionListener(e -> loadTable("issue", issueModel));
        return panel;
    }
    void loadTable(String tableName, DefaultTableModel model) {
        try {
            model.setRowCount(0);
            pst = con.prepareStatement("SELECT * FROM " + tableName);
            rs = pst.executeQuery();
            ResultSetMetaData rsmd = rs.getMetaData();
            int cols = rsmd.getColumnCount();
            while (rs.next()) {
                Object[] row = new Object[cols];
                for (int i = 0; i < cols; i++) row[i] = rs.getObject(i + 1);
                model.addRow(row);
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
    }

    void deleteRow(String tableName, JTable table, DefaultTableModel
model) {
        int row = table.getSelectedRow();
        if (row == -1) {
            JOptionPane.showMessageDialog(this, "Select a record to
delete!");
            return;
        }
        int id = (int) model.getValueAt(row, 0);
        try {
            pst = con.prepareStatement("DELETE FROM " + tableName + "
WHERE id = ?");
            pst.setInt(1, id);
            pst.executeUpdate();
            model.removeRow(row);
            JOptionPane.showMessageDialog(this, "Record Deleted!");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new LibraryManagement());
    }
}
```

Screenshots:

Book Details:



Library Management System

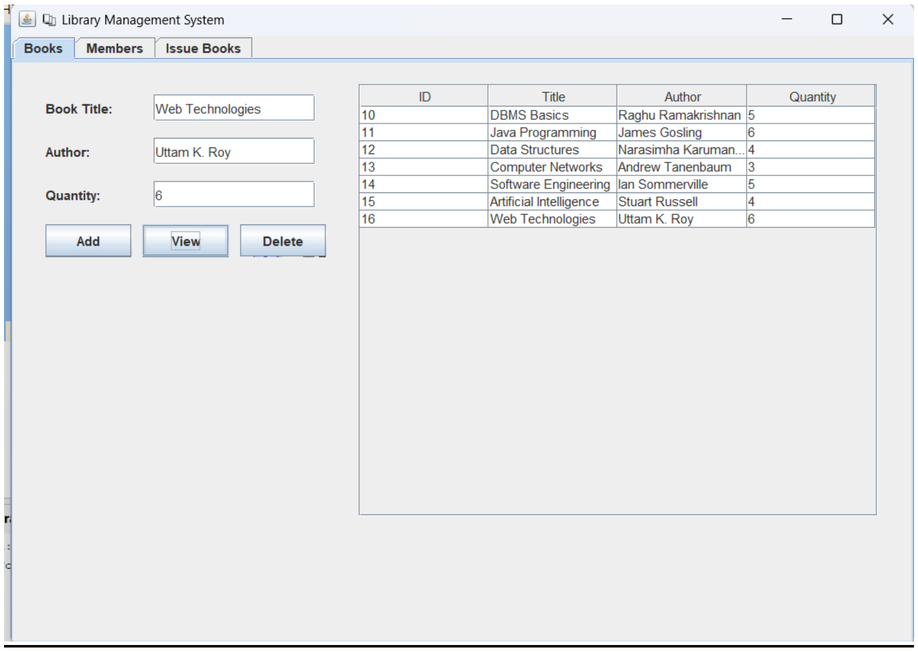
Book Title:	Web Technologies
Author:	Uttam K. Roy
Quantity:	6

ID	Title	Author	Quantity

Message

Book Added!

View Books Details:

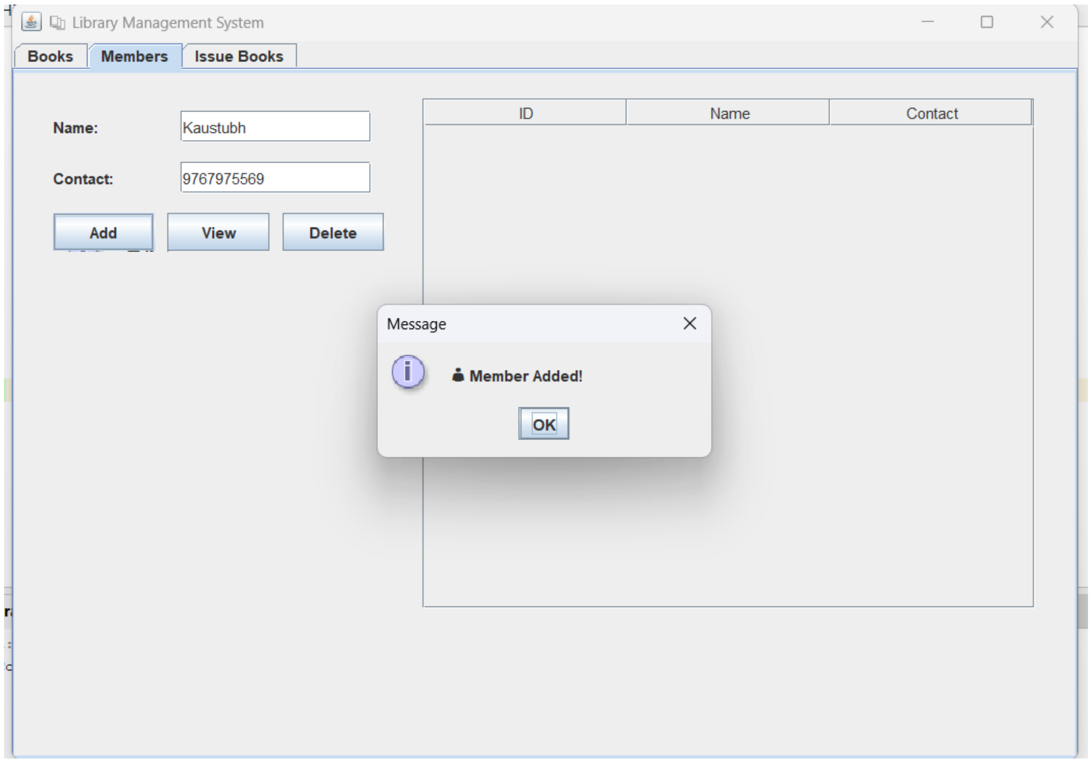


Library Management System

Book Title:	Web Technologies
Author:	Uttam K. Roy
Quantity:	6

ID	Title	Author	Quantity
10	DBMS Basics	Raghu Ramakrishnan	5
11	Java Programming	James Gosling	6
12	Data Structures	Narasimha Karuman...	4
13	Computer Networks	Andrew Tanenbaum	3
14	Software Engineering	Ian Sommerville	5
15	Artificial Intelligence	Stuart Russell	4
16	Web Technologies	Uttam K. Roy	6

View Members:



Name: Kaustubh

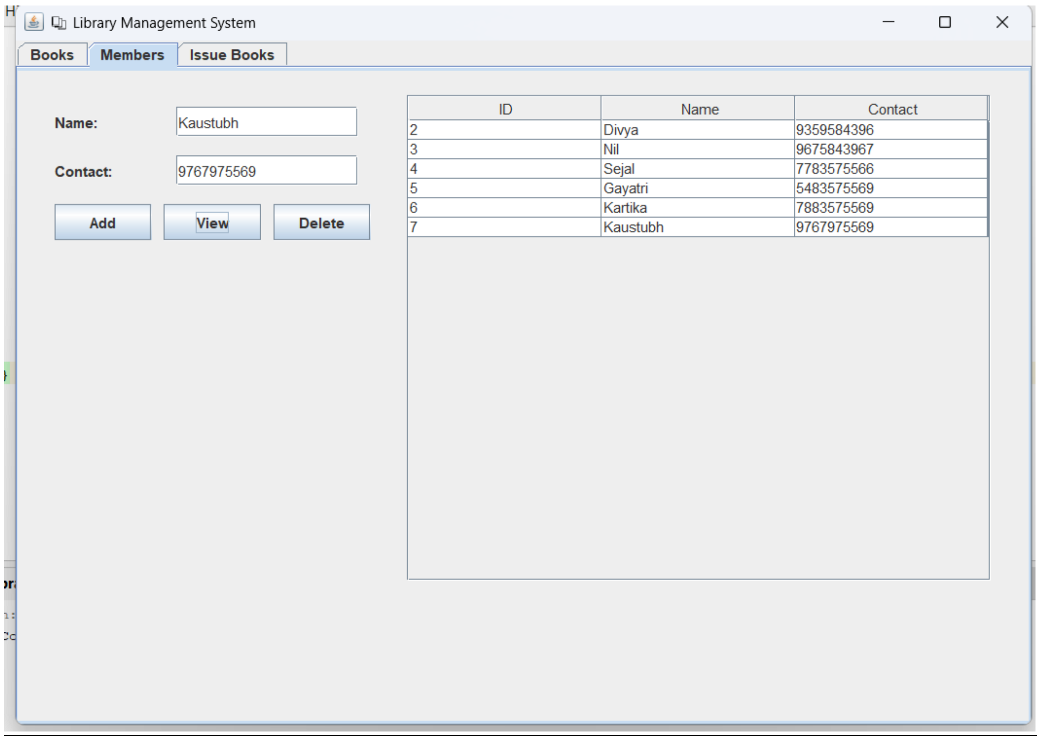
Contact: 9767975569

ID	Name	Contact

Message

Member Added!

View Member Details:

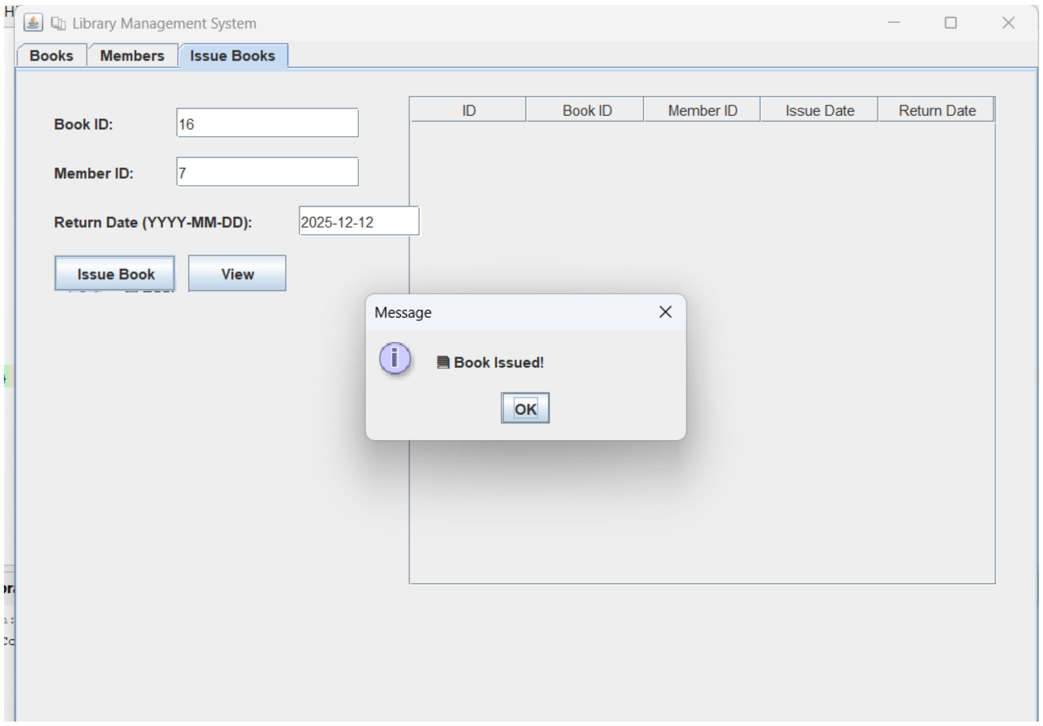


Name: Kaustubh

Contact: 9767975569

ID	Name	Contact
2	Divya	9359584396
3	Nil	9675843967
4	Sejal	7783575566
5	Gayatri	5483575569
6	Kartika	7883575569
7	Kaustubh	9767975569

Issue Details:



Book ID: 16

Member ID: 7

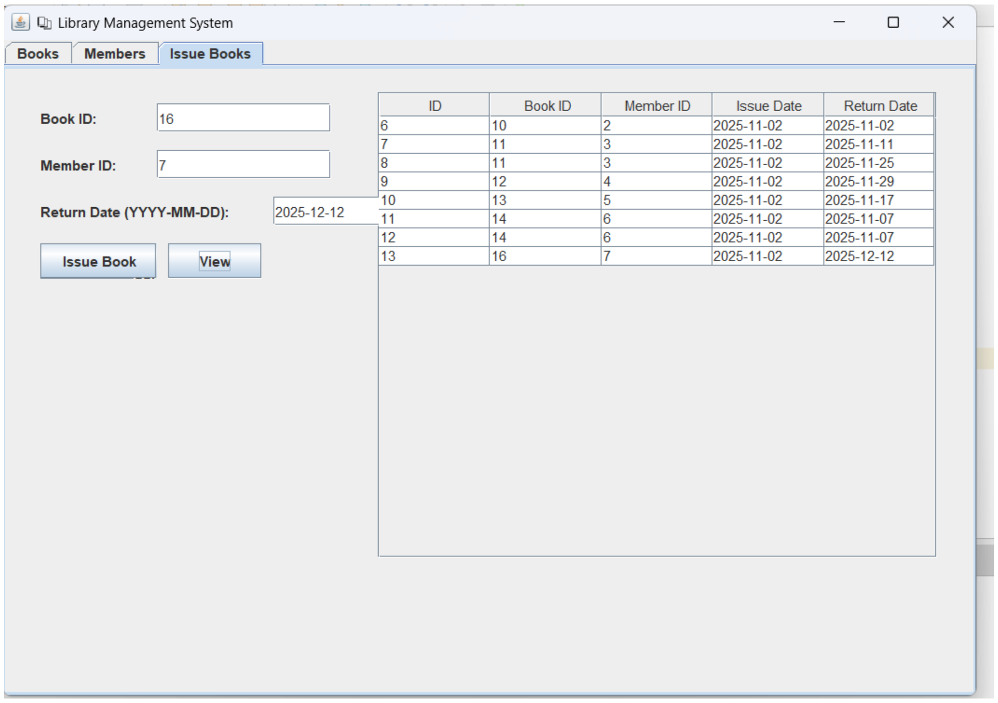
Return Date (YYYY-MM-DD): 2025-12-12

ID	Book ID	Member ID	Issue Date	Return Date

Message

Book Issued!

View Issue Details:



Book ID: 16

Member ID: 7

Return Date (YYYY-MM-DD): 2025-12-12

ID	Book ID	Member ID	Issue Date	Return Date
6	10	2	2025-11-02	2025-11-02
7	11	3	2025-11-02	2025-11-11
8	11	3	2025-11-02	2025-11-25
9	12	4	2025-11-02	2025-11-29
10	13	5	2025-11-02	2025-11-17
11	14	6	2025-11-02	2025-11-07
12	14	6	2025-11-02	2025-11-07
13	16	7	2025-11-02	2025-12-12

Conclusion :

The Library Management System is a simple and efficient application that helps manage books, members, and issue records easily. It reduces manual work by storing all data securely in a MySQL database and provides a user-friendly interface using Java Swing. This project makes library operations faster, accurate, and well-organized.