

Library Management Systems
A MINI-PROJECT REPORT
Submitted by

Rohinth T 230701267

Ritika Taphasvi G 230701266

In partial fulfillment of the award of the degree
of
BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING



RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute
CHENNAI
NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project “Library System Management” is the
bonafide work of “Rohinth T and Ritika Taphasvi G” who carried out the
project work under my supervision.

SIGNATURE

MR. SARAVANA GOKUL

ASSISTANT PROFESSOR

Dept. of Computer Science and Engg,

Rajalakshmi Engineering College

Chennai

SIGNATURE

MRS V JANANEE

ASSISTANT PROFESSOR

Dept. of Computer Science and Engg,

Rajalakshmi Engineering College

Chennai

This mini project report is submitted for the viva voce examination to be held
on _____

TABLE OF CONTENTS

1. INTRODUCTION

- 1.1 INTRODUCTION
- 1.2 IMPLEMENTATION
- 1.3 SCOPE OF THE PROJECT

2. SYSTEM SPECIFICATION

- 2.1 HARDWARE SPECIFICATION
- 2.2 SOFTWARE SPECIFICATION

3. SAMPLE CODE

- 3.1 LOGIN
- 3.2 DASHBOARD
- 3.3 AVAILABLE BOOKS
- 3.4 ADD BOOKS
- 3.5 DELETE BOOKS
- 3.6 AVAILABLE STAFF
- 3.7 ADD STAFFS
- 3.8 DELETE STAFFS

4. SNAPSHOTS

- 4.1 HOME PAGE
- 4.2 BOOK DETAILS
- 4.3 ADD BOOKS
- 4.4 REMOVE BOOKS
- 4.5 STAFF DETAILS
- 4.6 ADD STAFF
- 4.7 REMOVE STAFFS
- 4.8 EDIT ADMIN
- 4.9 LOGIN PAGE

5. CONCLUSION

6. REFERENCES

ABSTRACT

The Library Management System in JAVA is an advanced and efficient solution designed to streamline library operations by providing a secure, user-friendly, and organized platform for managing library resources. This software system aims to enhance the overall library experience, ensuring efficient management of books, members, and inventory while simplifying administrative tasks.

The Library Management System in JAVA is a powerful tool for modernizing how libraries handle resources, increase operational efficiency, and reduce the administrative burden associated with traditional library management. It addresses the challenges of manual record-keeping and improves the accessibility of library services while maintaining the integrity and accuracy of data.

This system is not just limited to traditional libraries; it can be customized to accommodate various library types, such as educational institutions, corporate libraries, or specialized libraries. With its strong focus on inventory management, user experience, and ease of use, this system significantly enhances the way libraries function, making them more efficient and accessible to all users.

1.1 INTRODUCTION

The Library Management System in JAVA addresses these challenges by offering a robust, user-friendly, and secure platform for managing library activities. With this system, librarians can automate essential tasks, such as book cataloguing, member registration, and overdue fine calculations, significantly reducing their workload. Users, on the other hand, gain access to a digital interface where they can search for books, check availability, and even reserve items from the comfort of their homes or on the go.

1.2 IMPLEMENTATION

The **Library Management Systems** project discussed here is implemented using the concepts of **JAVA SWINGS** and **MYSQL**.

1.3 SCOPE OF THE PROJECT

The scope of the **Library Management System** in Java using MySQL includes managing the complete lifecycle of books, such as adding, updating, and deleting records, while also supporting user registration and authentication for members and librarians. It facilitates efficient book borrowing, returning, and overdue tracking, ensuring smooth library operations. The system provides a searchable catalog with filters for author, genre, and availability, making it user-friendly.

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS:

PROCESSOR : Intel i5

MEMORY SIZE : 4GB(Minimum)

HARD DISK : 500 GB of free space

2.2 SOFTWARE SPECIFICATIONS:

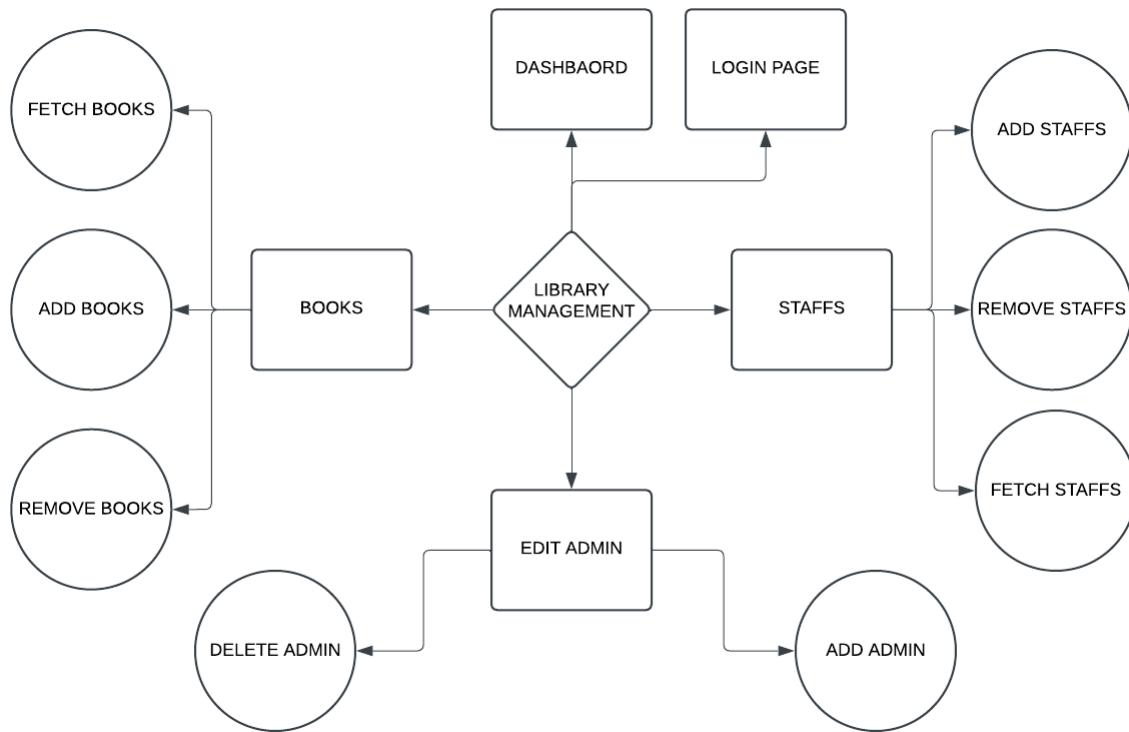
PROGRAMMING LANGUAGE : Java, MySQL

FRONT-END : Java

BACK-END : MySQL

OPERATING SYSTEM : Windows 11

ER DIAGRAM



Source code

1) LOGIN

```
import java.sql.*;
import javax.swing.*;

public class LoginPage extends javax.swing.JFrame {

    public LoginPage() {
        initComponents();
        login.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String url = "jdbc:mysql://localhost:3306/library";
        String mysqluser = "root";
        String mysqlpwd = "Taphasvi@266";
        String pswrd = new String(password.getPassword());
        String username = user.getText();
        String query = "SELECT PASSWORD FROM ADMIN WHERE USER_ID = '" +
        username + "'";
        try {
            Connection conn = DriverManager.getConnection(url, mysqluser, mysqlpwd);
            Statement stm = conn.createStatement();
            ResultSet rs = stm.executeQuery(query);
```

```

        if (rs.next()) {

            String realpswrd = rs.getString("PASSWORD");

            if (realpswrd.equals(pswrd)) {

                Dashboard dsh = new Dashboard();

                dsh.setVisible(true);

                this.dispose();

            } else {

                JOptionPane.showMessageDialog(this, "Username or Password is incorrect!");

            }

        } else {

            JOptionPane.showMessageDialog(this, "Wrong Username!");

        }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(this, e.getMessage());

    }

}

public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new LoginPage().setVisible(true);

        }

    });

}

}

```

2) DASHBOARD

```
import javax.swing.*;
```

```

public class Dashboard extends javax.swing.JFrame {

    public Dashboard() {
        initComponents();
        setDefaultCloseOperation(Dashboard.DISPOSE_ON_CLOSE);
    }

    private void initComponents() {
        add = new javax.swing.JButton();
        update = new javax.swing.JButton();
        delete = new javax.swing.JButton();
        search = new javax.swing.JButton();
        logout = new javax.swing.JButton();
        home = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        add.setText("Add Books");
        add.addActionListener(evt -> addActionPerformed(evt));
        getContentPane().add(add, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 110, 280, 110));

        update.setText("Update Books");
        update.addActionListener(evt -> updateActionPerformed(evt));
        getContentPane().add(update, new
            org.netbeans.lib.awtextra.AbsoluteConstraints(230, 460, 280, 110));
    }

    private void addActionPerformed(java.awt.event.ActionEvent evt) {
        // Add code here
    }

    private void updateActionPerformed(java.awt.event.ActionEvent evt) {
        // Add code here
    }
}

```

```

        delete.setText("Delete Books");

        delete.addActionListener(evt -> deleteActionPerformed(evt));

        getContentPane().add(delete, new org.netbeans.lib.awtextra.AbsoluteConstraints(730,
310, 280, 110));


        search.setText("Search Books");

        search.addActionListener(evt -> searchActionPerformed(evt));

        getContentPane().add(search, new
org.netbeans.lib.awtextra.AbsoluteConstraints(730, 460, 280, 110));


        logout.setText("Logout");

        logout.addActionListener(evt -> logoutActionPerformed(evt));

        getContentPane().add(logout, new
org.netbeans.lib.awtextra.AbsoluteConstraints(960, 40, 140, 60));


        home.setFont(new java.awt.Font("Georgia", 1, 48));

        home.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

        home.setText("DASHBOARD");

        getContentPane().add(home, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
20, 1160, 120));


        pack();
    }

private void actionPerformed(java.awt.event.ActionEvent evt) {

    new AddBooks().setVisible(true);

}

```

```
private void updateActionPerformed(java.awt.event.ActionEvent evt) {  
    // Code to open Update Books frame  
}  
  
private void deleteActionPerformed(java.awt.event.ActionEvent evt) {  
    // Code to open Delete Books frame  
}  
  
private void searchActionPerformed(java.awt.event.ActionEvent evt) {  
    // Code to open Search Books frame  
}  
  
private void logoutActionPerformed(java.awt.event.ActionEvent evt) {  
    this.dispose();  
}  
  
public static void main(String args[]) {  
    java.awt.EventQueue.invokeLater(() -> new Dashboard().setVisible(true));  
}  
  
private javax.swing.JButton add;  
private javax.swing.JButton update;  
private javax.swing.JButton delete;  
private javax.swing.JButton search;  
private javax.swing.JButton logout;  
private javax.swing.JLabel home;  
}
```

3) AVAILABLE BOOKS

```
import java.sql.*;  
import javax.swing.*;  
import javax.swing.table.DefaultTableModel;  
  
public class BooksAvailable extends javax.swing.JFrame {  
  
    public BooksAvailable() {  
        initComponents();  
        setDefaultCloseOperation(BooksAvailable.DISPOSE_ON_CLOSE);  
    }  
  
    private void initComponents() {  
        jTable2 = new javax.swing.JTable();  
        fetch = new javax.swing.JButton();  
        back = new javax.swing.JButton();  
  
        jTable2.setModel(new javax.swing.table.DefaultTableModel(  
            new Object [][] {},  
            new String [] { "Book ID", "Category", "Name", "Author", "Copies" }  
        ));  
        fetch.setText("Fetch");  
        fetch.addActionListener(evt -> fetchActionPerformed(evt));  
        back.setText("Back");  
        back.addActionListener(evt -> backActionPerformed(evt));  
        pack();  
    }  
}
```

```
}
```

```
private void fetchActionPerformed(java.awt.event.ActionEvent evt) {  
  
    DefaultTableModel model = (DefaultTableModel)jTable2.getModel();  
  
    String url = "jdbc:mysql://localhost:3306/library";  
  
    String user = "root";  
  
    String pwd = "Taphasvi@266";  
  
    String query = "SELECT * FROM BOOKS;";  
  
    try {  
  
        Connection conn = DriverManager.getConnection(url, user, pwd);  
  
        Statement stm = conn.createStatement();  
  
        ResultSet rs = stm.executeQuery(query);  
  
        while (rs.next()) {  
  
            model.addRow(new Object[] {  
  
                rs.getString("BOOK_ID"), rs.getString("CATEGORY"),  
  
                rs.getString("TITLE"), rs.getString("AUTHOR"),  
  
                rs.getString("COPIES")  
            });  
  
        }  
  
    } catch (Exception e) {  
  
        JOptionPane.showMessageDialog(this, e.getMessage());  
  
    }  
  
}
```

```
private void backActionPerformed(java.awt.event.ActionEvent evt) {  
  
    this.dispose();
```

```

    }

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(() -> new BooksAvailable().setVisible(true));
}

private javax.swing.JButton back;
private javax.swing.JButton fetch;
private javax.swing.JTable jTable2;
}

```

4) 1) ADD BOOKS

```
import java.sql.*;
```

```
import javax.swing.*;
```

```
public class AddBooks extends javax.swing.JFrame {
```

```
    public AddBooks() {
```

```
        initComponents();
```

```
        setDefaultCloseOperation(AddBooks.DISPOSE_ON_CLOSE);
```

```
}
```

```
    private void addActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        String url = "jdbc:mysql://localhost:3306/library";
```

```
        String user = "root";
```

```
        String pwd = "Taphasvi@266";
```

```
        String insertQuery = "INSERT INTO BOOKS VALUES(?,?,?,?,?)";
```

```
        String updateQuery = "UPDATE BOOKS SET COPIES=COPIES+? WHERE
TITLE=? AND CATEGORY=? AND AUTHOR=?";
```

```
        String id = t1.getText();
```

```

String category = t2.getText();

String title = t3.getText();

String author = t4.getText();

int copies = Integer.parseInt(t5.getText());

try {

    Connection conn = DriverManager.getConnection(url, user, pwd);

    PreparedStatement checkStm = conn.prepareStatement(updateQuery);

    checkStm.setInt(1, copies);

    checkStm.setString(2, title);

    checkStm.setString(3, category);

    checkStm.setString(4, author);

    int rows = checkStm.executeUpdate();

    if (rows > 0) {

        JOptionPane.showMessageDialog(this, "One record updated successfully");

    } else {

        PreparedStatement insertStm = conn.prepareStatement(insertQuery);

        insertStm.setString(1, id);

        insertStm.setString(2, category);

        insertStm.setString(3, title);

        insertStm.setString(4, author);

        insertStm.setInt(5, copies);

        insertStm.execute();

        JOptionPane.showMessageDialog(this, "One record added successfully");

    }

    t1.setText(null);

    t2.setText(null);

    t3.setText(null);
}

```

```

t4.setText(null);

t5.setText(null);

} catch (Exception e) {

JOptionPane.showMessageDialog(this, e);

}

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(() -> new AddBooks().setVisible(true));

}

}

```

2) DELETE BOOKS

```

import java.sql.*;

import javax.swing.*;

public class RemoveBooks extends javax.swing.JFrame {

    public RemoveBooks() {

        initComponents();

        setDefaultCloseOperation(RemoveBooks.DISPOSE_ON_CLOSE);

    }

    private void deleteActionPerformed(java.awt.event.ActionEvent evt) {

        String url = "jdbc:mysql://localhost:3306/library";

        String user = "root";

        String pwd = "Taphasvi@266";

```

```

String input = t1.getText();

String query = "DELETE FROM BOOKS WHERE book_id ='" + input + "' OR title
='" + input + "';";

try {

    Connection conn = DriverManager.getConnection(url, user, pwd);

    Statement stm = conn.createStatement();

    int rows = stm.executeUpdate(query);

    if (rows > 0)

        JOptionPane.showMessageDialog(this, "Book removed from library!");

    else

        JOptionPane.showMessageDialog(this, "No such book available!");

    stm.close();

} catch (Exception e) {

    JOptionPane.showMessageDialog(this, e.getMessage());

}

}

private void cancelActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(() -> new RemoveBooks().setVisible(true));

}

}

```

3) AVAILABLE STAFF

```

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

public class StaffAvailable extends javax.swing.JFrame {

    public StaffAvailable() {
        initComponents();
        setDefaultCloseOperation(StaffAvailable.DISPOSE_ON_CLOSE);
    }

    private void fetchActionPerformed(java.awt.event.ActionEvent evt) {
        DefaultTableModel model = (DefaultTableModel) jTable2.getModel();
        String url = "jdbc:mysql://localhost:3306/library";
        String user = "root";
        String pwd = "Taphasvi@266";
        String query = "SELECT * FROM STAFF";
        try {
            Connection conn = DriverManager.getConnection(url, user, pwd);
            Statement stm = conn.createStatement();
            ResultSet rs = stm.executeQuery(query);
            while (rs.next()) {
                String staffid = rs.getString("STAFF_ID");
                String name = rs.getString("name");
                long contact = rs.getLong("CONTACT");
                model.addRow(new Object[] { staffid, name, contact });
            }
            rs.close();
        }
    }
}

```

```

        stm.close();

    } catch (Exception e) {

        JOptionPane.showMessageDialog(this, e.getMessage());

    }

}

private void backActionPerformed(java.awt.event.ActionEvent evt) {

    this.dispose();

}

public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(() -> new StaffAvailable().setVisible(true));

}

private javax.swing.JButton back, fetch;

private javax.swing.JTable jTable2;

}

```

4) ADD STAFF

```

import java.sql.*;

import javax.swing.*;

public class AddStaff extends javax.swing.JFrame {

    public AddStaff() {

        initComponents();

        setDefaultCloseOperation(AddStaff.DISPOSE_ON_CLOSE);
    }
}

```

```

}

private void actionPerformed(java.awt.event.ActionEvent evt) {

    String url = "jdbc:mysql://localhost:3306/library";
    String user = "root";
    String pwd = "Taphasvi@266";
    String query = "INSERT INTO STAFF VALUES(?, ?, ?)";
    String id = t1.getText();
    String name = t2.getText();
    int contact = Integer.parseInt(t3.getText());
    try {
        Connection conn = DriverManager.getConnection(url, user, pwd);
        PreparedStatement stm = conn.prepareStatement(query);
        stm.setString(1, id);
        stm.setString(2, name);
        stm.setInt(3, contact);
        stm.execute();
        JOptionPane.showMessageDialog(this, "One staff added successfully");
        t1.setText(null);
        t2.setText(null);
        t3.setText(null);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e);
    }
}

private void cancelActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

```

```

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(() -> new AddStaff().setVisible(true));
}

private javax.swing.JButton add, cancel;
private javax.swing.JTextField t1, t2, t3;
}

```

5) DELETE STAFF

```

import java.sql.*;
import javax.swing.*;

public class RemoveStaff extends javax.swing.JFrame {
    public RemoveStaff() {
        initComponents();
        setDefaultCloseOperation(RemoveStaff.DISPOSE_ON_CLOSE);
    }

    private void deleteActionPerformed(java.awt.event.ActionEvent evt) {
        String url = "jdbc:mysql://localhost:3306/library";
        String user = "root";
        String pwd = "Taphasvi@266";
        String input = t1.getText();
        String query = "DELETE FROM STAFF WHERE staff_id ='" + input + "' OR name ='" + input + "'";
        try (Connection conn = DriverManager.getConnection(url, user, pwd)) {
            Statement stm = conn.createStatement();
            int rows = stm.executeUpdate(query);
        }
    }
}

```

```

        JOptionPane.showMessageDialog(this, rows > 0 ? "Staff removed from library!" :
"No staff available!");

    } catch (Exception e) {

        JOptionPane.showMessageDialog(this, e.getMessage());

    }

}

private void cancelActionPerformed(java.awt.event.ActionEvent evt) {

    this.dispose();

}

public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(() -> new RemoveStaff().setVisible(true));

}

private javax.swing.JButton cancel;

private javax.swing.JButton delete;

private javax.swing.JTextField t1;

}

```

6) EDIT ADMIN

```

import java.sql.*;

import javax.swing.*;

```



```

public class EditAdmin extends javax.swing.JFrame {

    public EditAdmin() {

        initComponents();

```

```

        setDefaultCloseOperation(EditAdmin.DISPOSE_ON_CLOSE);

    }

private void updateActionPerformed(java.awt.event.ActionEvent evt) {
    String url = "jdbc:mysql://localhost:3306/library";
    String user = "root";
    String pwd = "Taphasvi@266";
    String id = t1.getText();
    String column = jComboBox1.getSelectedItem().toString();
    String query = "UPDATE admin SET " + column + " = '" + id + "';";
    try (Connection conn = DriverManager.getConnection(url, user, pwd);
         Statement stmt = conn.createStatement()) {
        int rows = stmt.executeUpdate(query);
        if (rows > 0) {
            JOptionPane.showMessageDialog(this, "Credentials Updated Successfully");
        }
        t1.setText("");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e);
    }
}

private void cancelActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

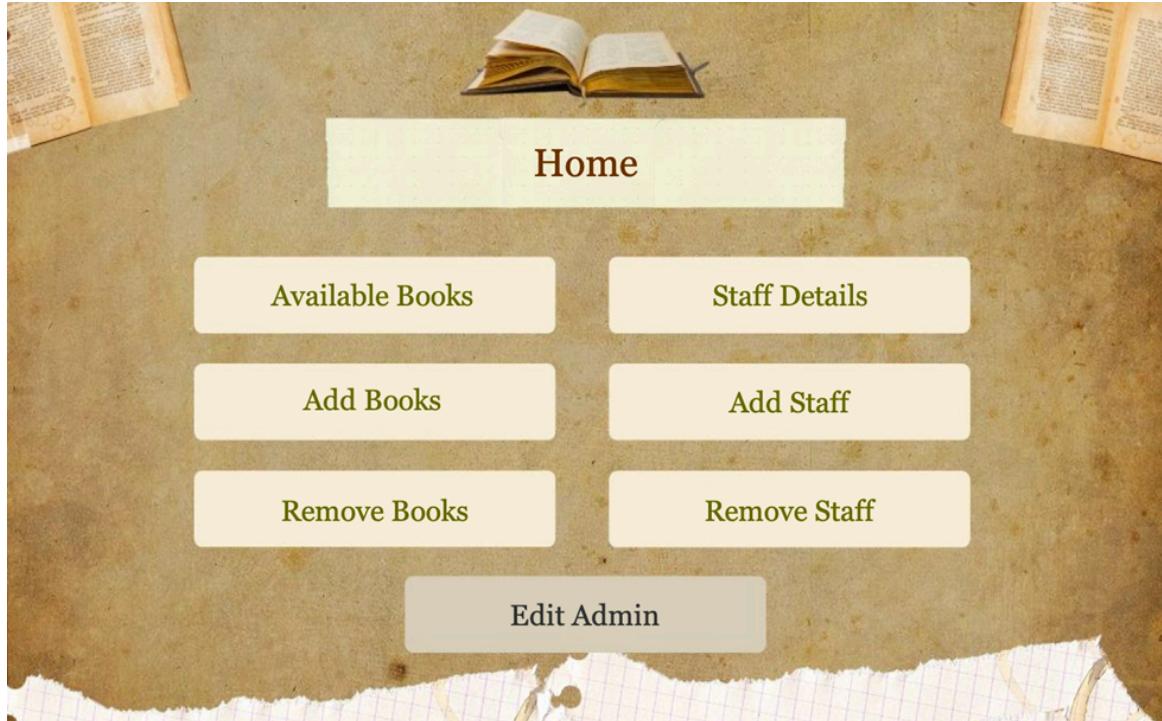
public static void main(String args[]) {

```

```
        java.awt.EventQueue.invokeLater(() -> new EditAdmin().setVisible(true));  
    }  
  
    private javax.swing.JButton cancel;  
    private javax.swing.JComboBox<String> jComboBox1;  
    private javax.swing.JTextField t1;  
    private javax.swing.JButton update;  
}
```

SNAPSHOTS

4.1 HOME PAGE

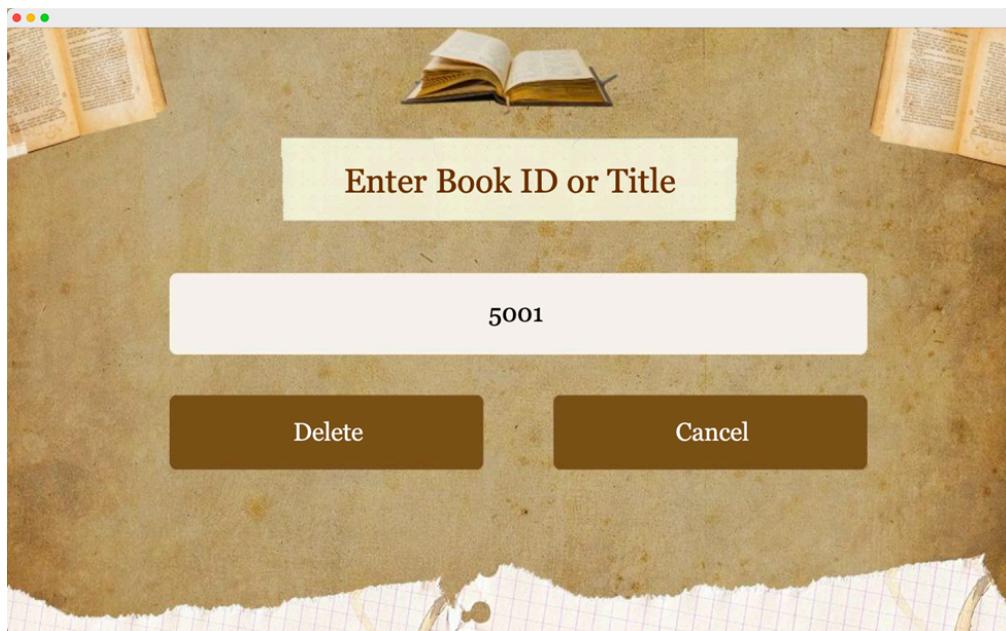


4.2 BOOK DETAILS

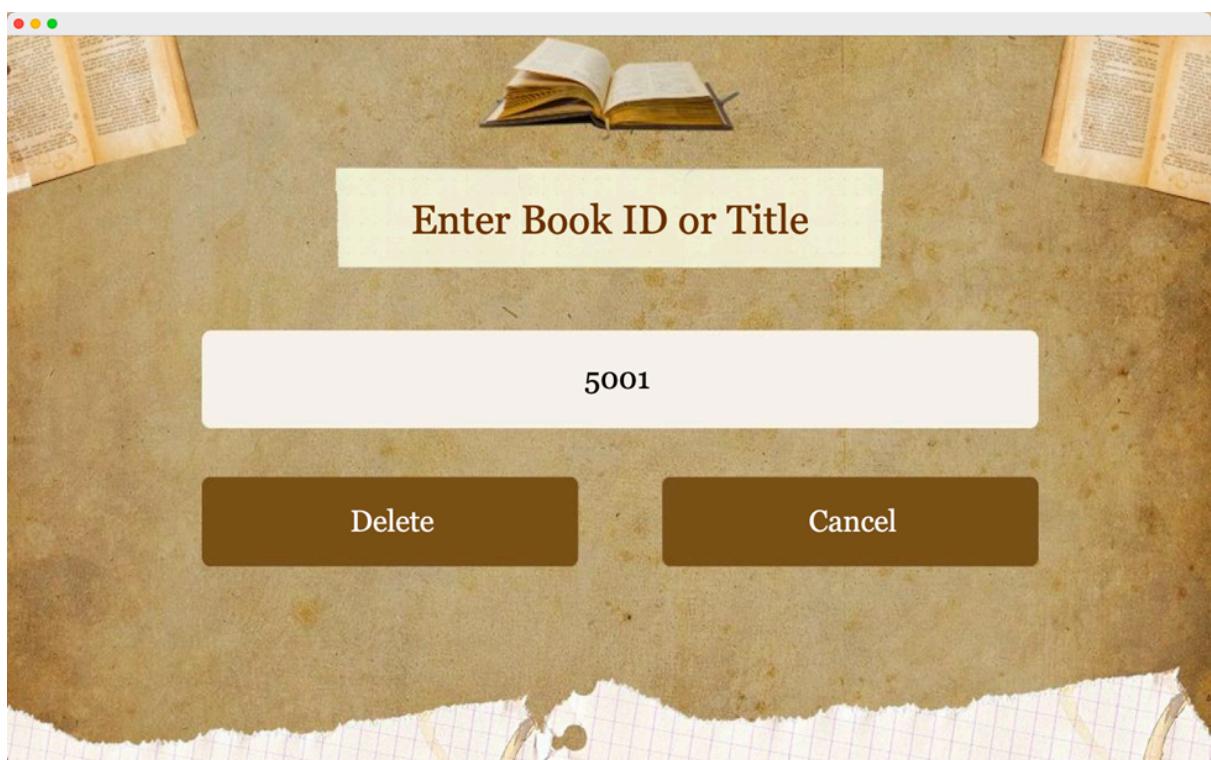
Book ID	Category	Name	Author	Copies
8001	DSA	ALGORITHMS MADE EASY	NARSIKAR KARNAWAT	18
8002	DSA	HEAD FIRST JAVA	KATHY SIERRA BERT BATES	8
8003	ANCIENT HISTORY	THE ANCIENT PAST	R.S SHARMA	20
8004	NOVEL	THE GAME OF VOTES	PRASHANT BASRA KHAN	10
8005	NOVEL	THE GREAT GATSBY	F. SCOTT FITZGERALD	5
8006	NOVEL	GONE WITH THE WIND	DOH MCUSKIL	7
8007	BIOGRAPHY	STEVE JOBS	WALTER ISAACSON	6
8008	BIOGRAPHY	THE SECRET LIFE OF DEBBIE G.	VINA BATRA	15
8009	SCIENCE	COSMOS	CARL SAGAN	24
8010	SCIENCE	CONCEPTS OF BIOLOGY	REBECCA ROUSH	29
8011	SCIENCE	A BRIEF HISTORY OF TIME	STEPHEN HAWKING	25
8012	NOVEL	THE SILMARILLION	J.R.R. TOLKIEN	18
8013	BIOGRAPHY	MALCOLM X	YUVAL NOAH HARARI	22
8014	HISTORY	SAPIENS	RICHARD DAWKINS	20
8015	SCIENCE	THE SELFISH GENE	J.R.R. TOLKIEN	15
8016	NOVEL	THE LORD OF THE RINGS	EINSTEIN: HIS LIFE AND UNIVERSE	30
8017	BIOGRAPHY	EINSTEIN: HIS LIFE AND UNIVERSE	WALTER ISAACSON	28
8018	HISTORY	GUNS, GERMS, AND STEEL	JARED DIAMOND	21
8019	SCIENCE	SILENCE OF THE FRONTIERS	SABINE HOUSSENFELDER	17
8020	NOVEL	THE HOBBIT	J.R.R. TOLKIEN	24
8021	BIOGRAPHY	STEVE JOBS	WALTER ISAACSON	19
8022	HISTORY	THE RISE AND FALL OF THE ROMAN E...	EDWARDS GIBBON	23
8023	SCIENCE	A BRIEF HISTORY OF TIME FROM THE...	STEPHEN HAWKING	26
8024	NOVEL	THE SILMARILLION	J.R.R. TOLKIEN	16
8025	BIOGRAPHY	MALCOLM X	MALCOLM X	20
8026	HISTORY	SAPIENS	YUVAL NOAH HARARI	18
8027	SCIENCE	THE SELFISH GENE	RICHARD DAWKINS	14
8028	NOVEL	THE LORD OF THE RINGS	J.R.R. TOLKIEN	28
8029	BIOGRAPHY	EINSTEIN: HIS LIFE AND UNIVERSE	WALTER ISAACSON	26
8030	HISTORY	GUNS, GERMS, AND STEEL	JARED DIAMOND	20
8031	NOVEL	THE SILMARILLION	J.R.R. TOLKIEN	18
8032	BIOGRAPHY	MALCOLM X	MALCOLM X	22
8033	HISTORY	SAPIENS	YUVAL NOAH HARARI	20
8034	SCIENCE	THE SELFISH GENE	RICHARD DAWKINS	15
8035	NOVEL	THE LORD OF THE RINGS	J.R.R. TOLKIEN	30
8036	BIOGRAPHY	EINSTEIN: HIS LIFE AND UNIVERSE	WALTER ISAACSON	28

Fetch Back

4.3 ADD BOOKS



4.4 REMOVE BOOKS



4.5 STAFF DETAILS

A screenshot of a software application window titled "Staff Details". The window features a table with three columns: "Staff ID", "Name", and "Contact". The "Staff ID" column lists IDs from 1001 to 1016. The "Name" column lists names such as Shreya, Sathya, Vruksha, Preethi, Leo, Garnacho, Amad, Messi, Mainoo, Rashford, Onana, Rukesh, Shruti, Reshma, Ronald, and Sahana. The "Contact" column lists corresponding phone numbers. Below the table are two brown rectangular buttons labeled "Fetch" and "Back". The background of the window has a textured, parchment-like appearance.

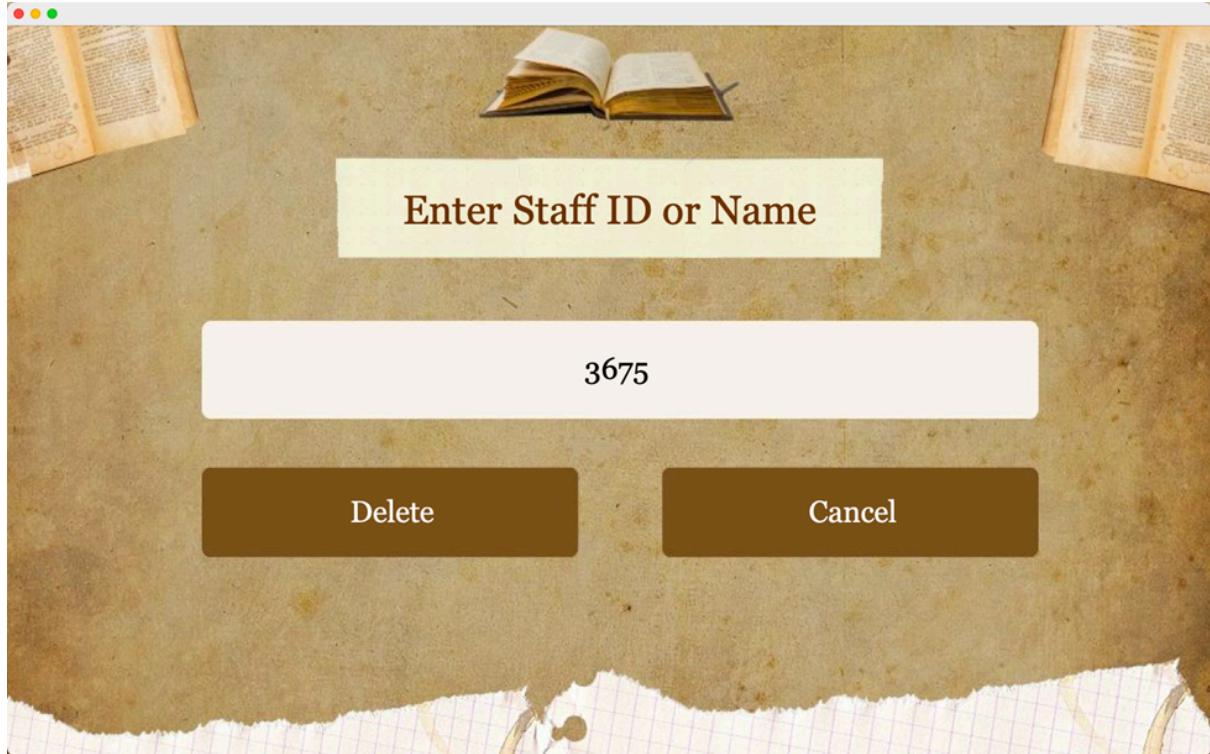
Staff ID	Name	Contact
1001	Shreya	765843412
1002	Sathya	890574312
1003	Vruksha	954231432
1004	Preethi	950314312
1005	Leo	889056734
1006	Garnacho	864778356
1007	Amad	873567652
1008	Messi	789568903
1009	Mainoo	890267489
1010	Rashford	908765678
1011	Onana	908767894
1012	Rukesh	908765678
1013	Shruti	980278493
1014	Reshma	908765456
1015	Ronald	983561728
1016	Sahana	678236746

4.6 ADD STAFFS

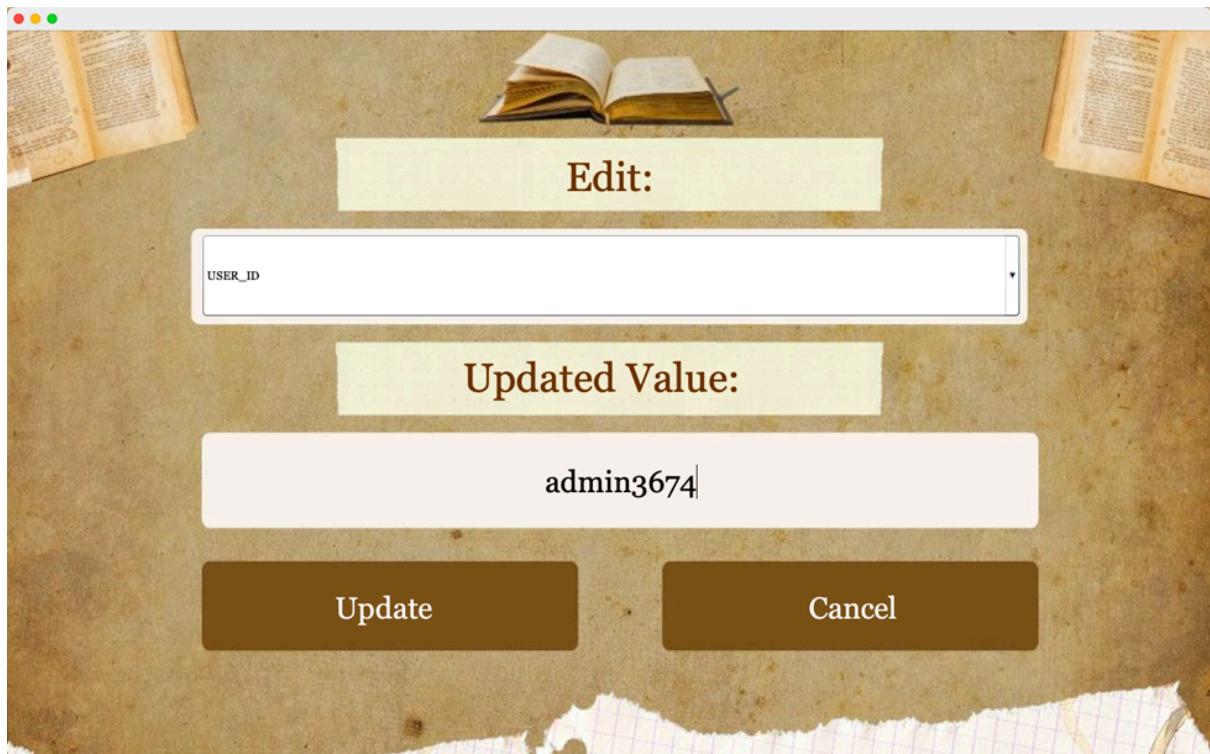
A screenshot of a software application window titled "Add Staff". The window has a parchment-like background with open books. It contains three input fields: "Staff ID" with value "2334", "Name" with value "George", and "Contact" with value "98367829". At the bottom are two brown rectangular buttons labeled "Add" and "Cancel".

Staff ID:	2334
Name:	George
Contact:	98367829

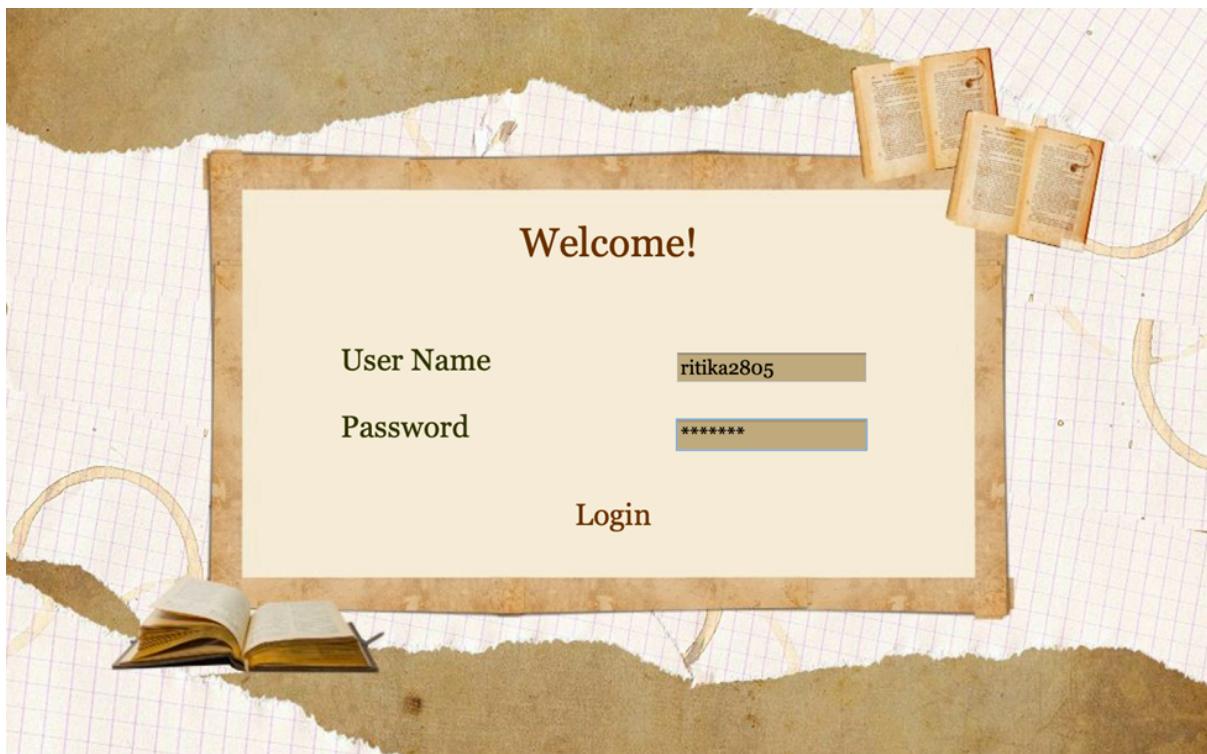
4.7 REMOVE STAFFS



4.8 EDIT ADMIN



4.9 LOGIN PAGE



CONCLUSION

The Java-based Library Management System is a robust application designed to streamline and automate library operations such as book lending, returns, catalog management, and member tracking. This project demonstrates the efficient use of Java's object-oriented programming features and database integration to enhance library workflows.

REFERENCES

1. <https://www.javatpoint.com/java-tutorial>
2. <https://www.wikipedia.org/>
3. <https://www.w3schools.com/sql/>