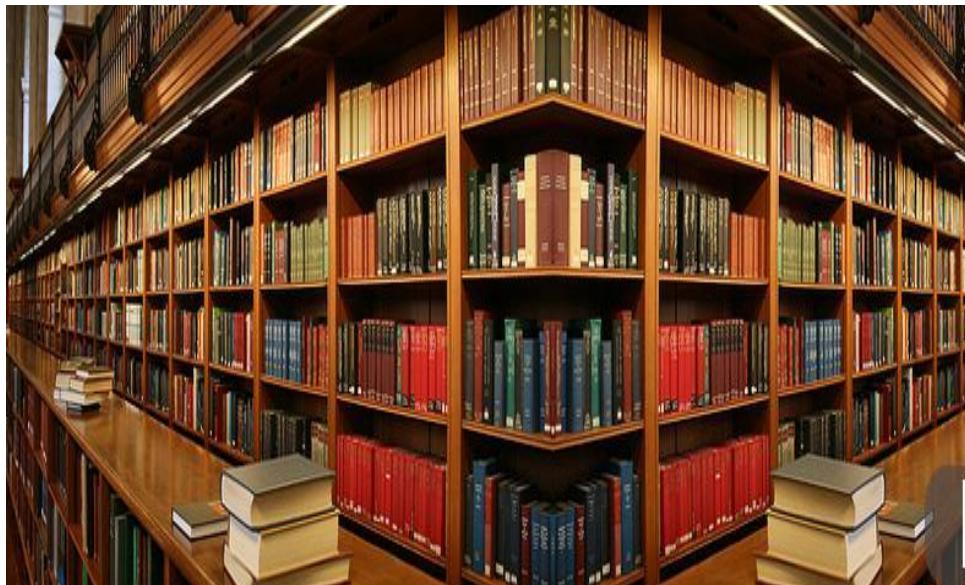


Informatics Practices Project on Library Management System



RITA ABANI , CLASS XII A

CHIREC INTERNATIONAL SCHOOL



**THE NEW MILLENIUM
BIBLIOTHEQUE**

ACKNOWLEDGEMENT

Firstly, I would like to thank the Central Board of Secondary Education (C.B.S.E) for having given us this magnanimous opportunity to utilize and apply the pristine knowledge imparted to us.

Thanks to my Informatics Practices teacher, my inspiration, Naseema ma'am for having embellished the GUI of my life with all the controls and navigations through her unflagging encouragement and timely support.

Thank you to my Principal, Mrs. Iffat Ibrahim for having mentored students like us with great alacrity and fervency.

Thank you to my school for having provided us with all the resources whenever we required them.

Once again, a big thanks to all the tables, integrity constraints, transactions of my database (my friends and family) for having helped me throughout. Thank you all!

INDEX

S.NO	TOPIC	PAGE NO.
1	Abstract	3
2	Hardware Configurations	4
3	Software Requirements (Introduction to Java, Netbeans and Mysql)	5
4	Project Explorer	7
5	Screenshots and code	8
6	SQL tables	47
7	References	48

ABSTRACT

Our Project – “The New Millennium Bibliotheque” is an application based on a library management system.

It consists of two sections: one for the admin (or the librarian) and the other for the user.

Once successfully logged into the system, the admin can make new additions of books to the library, and also remove any book from the library records in case the book has become unavailable due to any reason.

The admin also has the authority to monitor the activities of the user. The admin can issue any book to the user that he / she wishes to read, and ensures that the book is also returned to the library in time.

Any person who wishes to be a member of the library can become one by simply filling a sign up form, thereby adding their names and other details to the library user records.

The user may edit his/her profile at any time, and can look up any new additions to the library, helping them decide any book they might want to borrow from there in the future. It helps the user check if any book he/she wishes to borrow is currently available in the library or not.

MINIMUM HARDWARE CONFIGURATIONS

- **Microsoft Windows Vista SP1/Windows 7 Professional:**

Processor: 800Mhz Intel Pentium III or equivalent.

Memory : 512 MB

Disk Space:750 MB of free disk space.

- **Ubuntu 9.10:**

Processor: 800Mhz Intel Pentium III or equivalent

Memory:512 MB

Disk Space:650 MB of free disk space

- **Macintosh OS X 10.7 Intel:**

Processor: Dual-Core Intel

Memory:2GB

Disk Space: 650 MB of free disk space.

SOFTWARE REQUIREMENTS



INTRODUCTION TO JAVA & NETBEANS

Java is a general-purpose computer-programming language that is concurrent, class-based, and object-oriented and platform independent.

Developed by James Gosling at Sun Microsystems in 1995, Java was originally aimed at allowing entertainment appliances such as video game consoles and VCRs to connect.

It is one of the most popular languages in the field of software programming. The reason behind its wide acceptance in the digital world is its large number of useful and robust features:

- Java is simple: It is easy to write and understand. Its virtual machine is able to also handle the memory management efficiently.
- Java is Object-Oriented: Java treats everything as objects and follows all the rules of OOPs like Inheritance, Encapsulation, Polymorphism,etc.
- Java is Distributed: Widely used protocols such as HTTP and FTP are developed in Java.
- Java is Interpreted: Java is also known as interpreted language as its byte code, which is generated after compiling the source code, is interpreted by the Java Virtual Machine(JVM) and is converted to machine dependent code or native code.
- Java is Platform Independent: Unlike many other programming languages like C and C++, when Java is compiled, it is not complied into platform specific machine, but into platform independent byte code, following the “write-once-run-anywhere” concept.
- Garbage Collection in Java: Java also has a feature for automatic garbage collection which prevents inadvertent corruption of memory.

The platform used in this project is NetBeans IDE.

NETBEANS IDE:

Netbeans IDE is a free, open source, cross platform IDE with built-in support for Java.

It runs anywhere there's a JVM and like Java, offers many unique features such as

- Drag and drop GUI creation
- Excellent editing
- Web services
- Debugging capabilities
- Wizards, code generation and management tools. It is user friendly and allows easy creation of any GUI (Graphical User Interface) in the form of Dialog boxes and Frames



INTRODUCTION TO MySQL

A DBMS is a database management system that is used not only for storing and maintaining data, but also for providing the enterprise with centralized control of its operational data, especially the sensitive and crucial data.

There are many DBMSs available for storing and maintaining the data of an enterprise. However, the DBMS used in this investigatory project is MySQL .

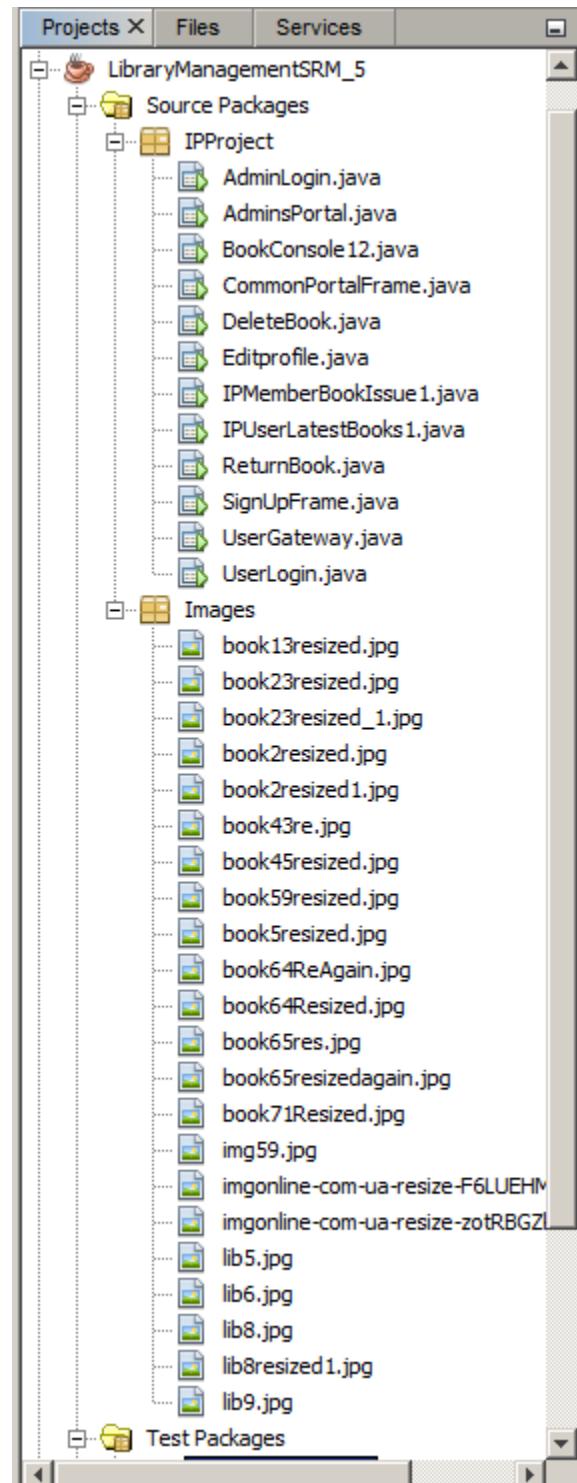
MySQL is a freely available open source Relational Database Management System (RDBMS) that uses SQL (Structured Query Language) in its functioning.

Some of the key features present in MySQL include:

- **Speed:** MySQL runs very fast and supports clustered servers for demanding applications.
- **Cost:** MySQL is available free of cost and is preferred for its easy availability, reliability and documentation.
- **Query Language Support:** MySQL understands standards based SQL
- **Portability:** MySQL provides portability and has been tested with a wide range of different compliers and can also work on various different platforms.
- **Data Types:** MySQL provides many datatypes to support different types of data.
- **Security:** MySQL offers a very flexible and secure privilege and password system.
- **Connectivity:** MySQL can connect to server through various protocols.
- **Scalability:** MySQL can handle large databases of over 60,000 tables.
- **Localization:** MySQL can provide error messages to the clients in many languages.

MySQL and Netbeans IDE can be used together and information entered in a NetBeans application can be stored in a database in MySQL by establishing a connection between them, using the concept of **Database Connectivity, which is one of the fundamental elements used in this project.**

PROJECT EXPLORER



SCREENS AND CODE:

THE COMMON PORTAL FRAME

This frame is common to both the user and the admin.

For almost all the frames , we make use of the following import statements:

```
Import java.sql.*;  
import javax.swing.JOptionPane;  
import javax.swing.table.DefaultTableModel;
```



Code for admin button:

```
IP1 adlogin = new IP1();
adlogin.setVisible(false);

IP2_AdminLogin1 page2 = new IP2_AdminLogin1();
page2.setVisible(true);

this.dispose();
```

Code for User Button:

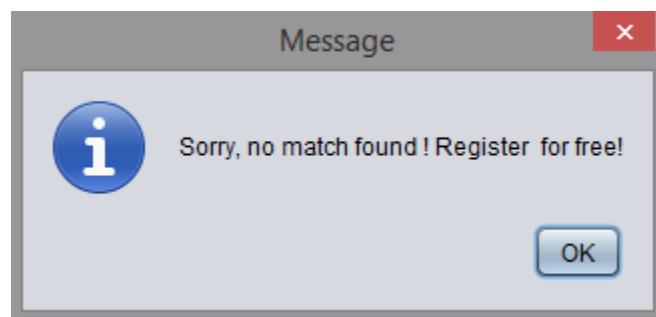
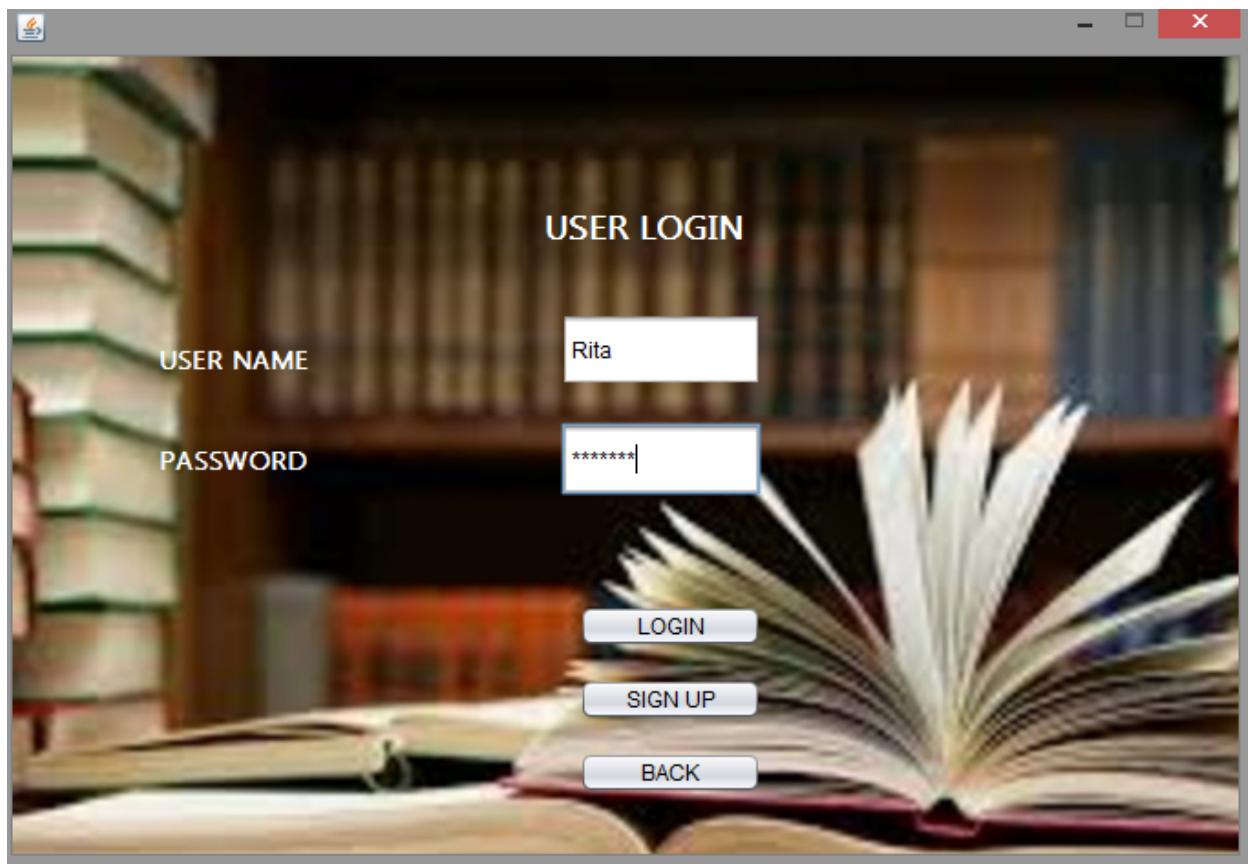
```
Ipmain login = new Ipmain();
login.setVisible(false);

Ipmain page = new Ipmain();
page.setVisible(true);

this.dispose();
```

USER SECTION

This frame takes in the user name and password of the user, incase, the user is not an existing member, he or she can become a new member by signing up for free.



Code for login button:

```
String uname = jTextField1.getText();
String pass = new String(jPasswordField1.getPassword());
System.out.println(pass);
try{
Class.forName("java.sql.Driver");
Connection con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
Statement stmt = con.createStatement();
String query ="select uname,pswd from user_records;";
System.out.println(query);
ResultSets= stmt.executeQuery(query);
String DBuname;
String DBpswd;
booleanlflag= false;
while(rs.next())
{
DBuname=rs.getString(1);
DBpswd=rs.getString(2);

if(uname.equals(DBuname) && pass.equals(DBpswd))
{lflag=true;
JOptionPane.showMessageDialog(null,"You have sucessfully logged in");
Ipmain login = new Ipmain();
login.setVisible(false);
IPUserData1 user = new IPUserData1();
user.setVisible(true);
```

```

this.dispose()
}

}

if(lflag==false){

JOptionPane.showMessageDialog(null,"Sorry, no match found !Register for free!");

}}

catch(Exception e){

System.out.println(e);

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

Ipmain login = new Ipmain();

login.setVisible(false);

IP3signUp1 signup = new IP3signUp1();

signup.setVisible(true);

this.dispose();

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

Ipmain login1 = new Ipmain();

login1.setVisible(false);

IP1 main1 = new IP1();

main1.setVisible(true);

this.dispose()

}

```

Code for Sign Up button:

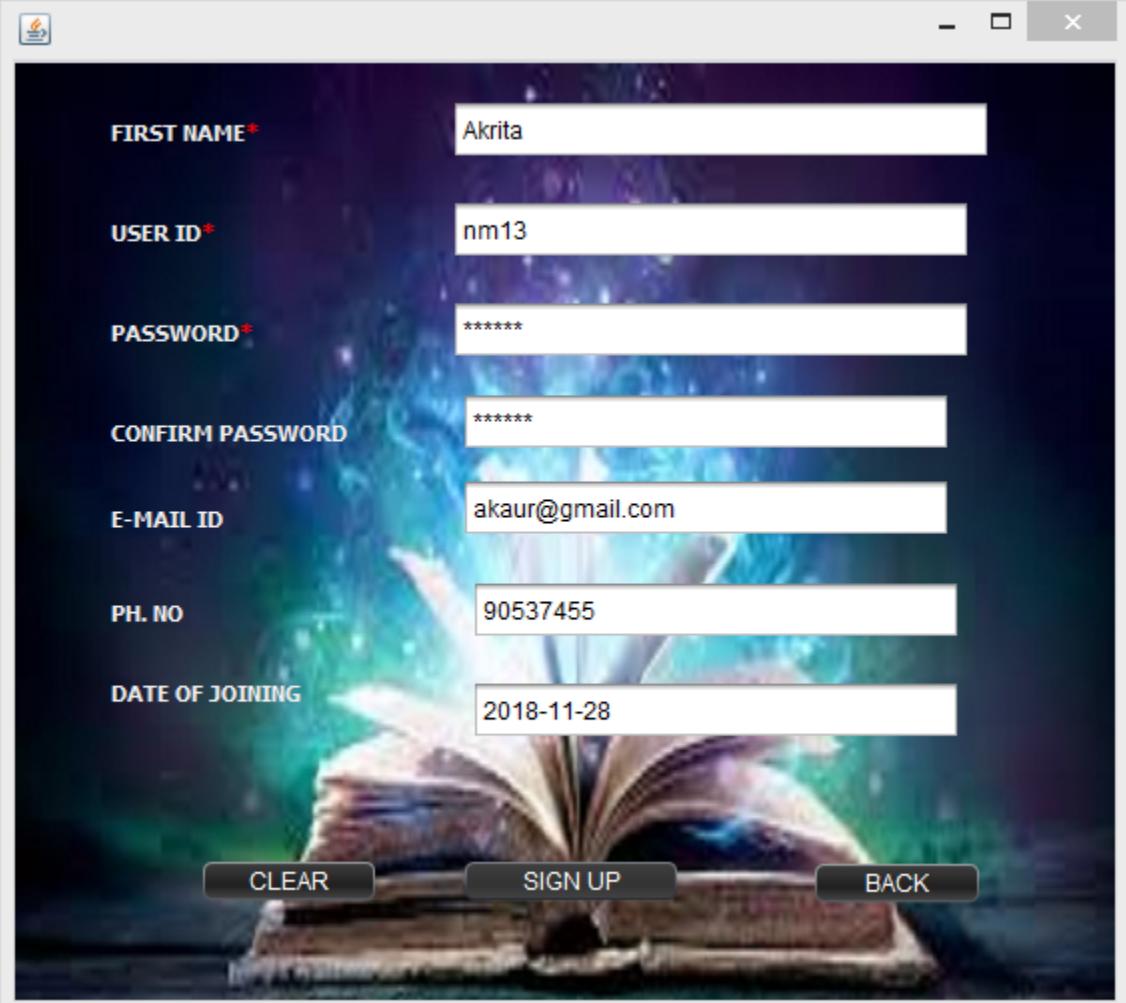
```
Ipmain login = new Ipmain();
login.setVisible(false);
IP3signUp1 signup = new IP3signUp1();
signup.setVisible(true);
this.dispose();
```

Code for back button:

```
Ipmain login1 = new Ipmain();
login1.setVisible(false);
IP1 main1 = new IP1();
main1.setVisible(true);
this.dispose();
```

REGISTRATION

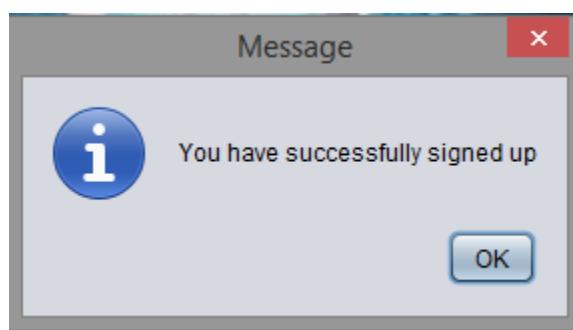
An unregistered user is directed to this page on the click of the Register button. The field marked with a * are mandatory to be filled.



A screenshot of a Windows-style application window titled "REGISTRATION". The background features a photograph of an open book with its pages fanned out. The form contains the following fields:

FIRST NAME*	Akrita
USER ID*	nm13
PASSWORD*	*****
CONFIRM PASSWORD	*****
E-MAIL ID	akaur@gmail.com
PH. NO	90537455
DATE OF JOINING	2018-11-28

At the bottom are three buttons: "CLEAR", "SIGN UP", and "BACK".



Code for Sign Up button:

```
String uid = jTextField8.getText();
String uname = jTextField1.getText();
String pswd = new String(jPasswordField1.getPassword());
String pswd2 = new String(jPasswordField2.getPassword());
String email = jTextField3.getText();
String phno = jTextField4.getText();
String jdate= jTextField7.getText();
if(uid.equals(""))||uname.equals("")||pswd.equals("")||pswd2.equals(""))
{
    JOptionPane.showMessageDialog(null,"Please enter required details");
}
else{
    if (pswd2.equals(pswd)){
        try{
            Class.forName("java.sql.Driver");
            Connection con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
            Statement stmt = con.createStatement();
            String query = "insert into user_records values ('"+uid+"','"+uname+"','"+pswd+
"+email+"','"+phno+"','"+jdate+"','nil')";
            System.out.println(query);
            intsucc= stmt.executeUpdate(query);
            if(succ>0){
                JOptionPane.showMessageDialog(null,"You have successfully signed up");
            }
        }
    }
}
```

```

else{
    JOptionPane.showMessageDialog(null,"Please try again ");
}

}

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

else {
    JOptionPane.showMessageDialog(null,"Please try again the passwords do not match !!!!!!");
}
}
}

```

Code for clear button:

```

jTextField1.setText(" ");
jTextField4.setText(" ");
jTextField3.setText(" ");
jTextField8.setText(" ");
jTextField7.setText(" ");
jPasswordField1.setText("");
jPasswordField2.setText("");

```

Code for back button:

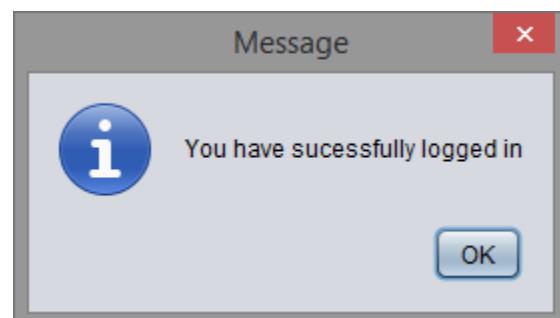
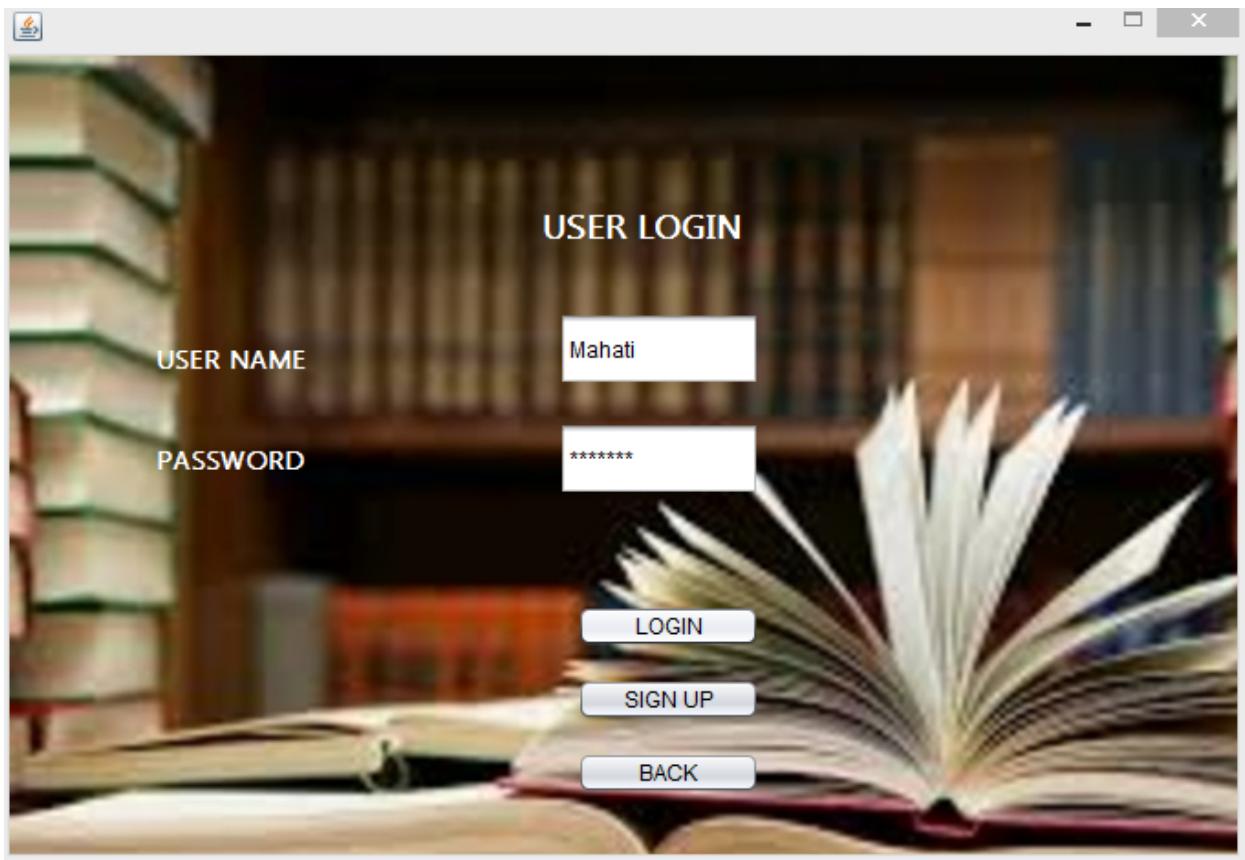
```

IP3signUp1 back = new IP3signUp1();
back.setVisible(false);
Ipmain diff = new Ipmain();
diff.setVisible(true);
this.dispose();

```

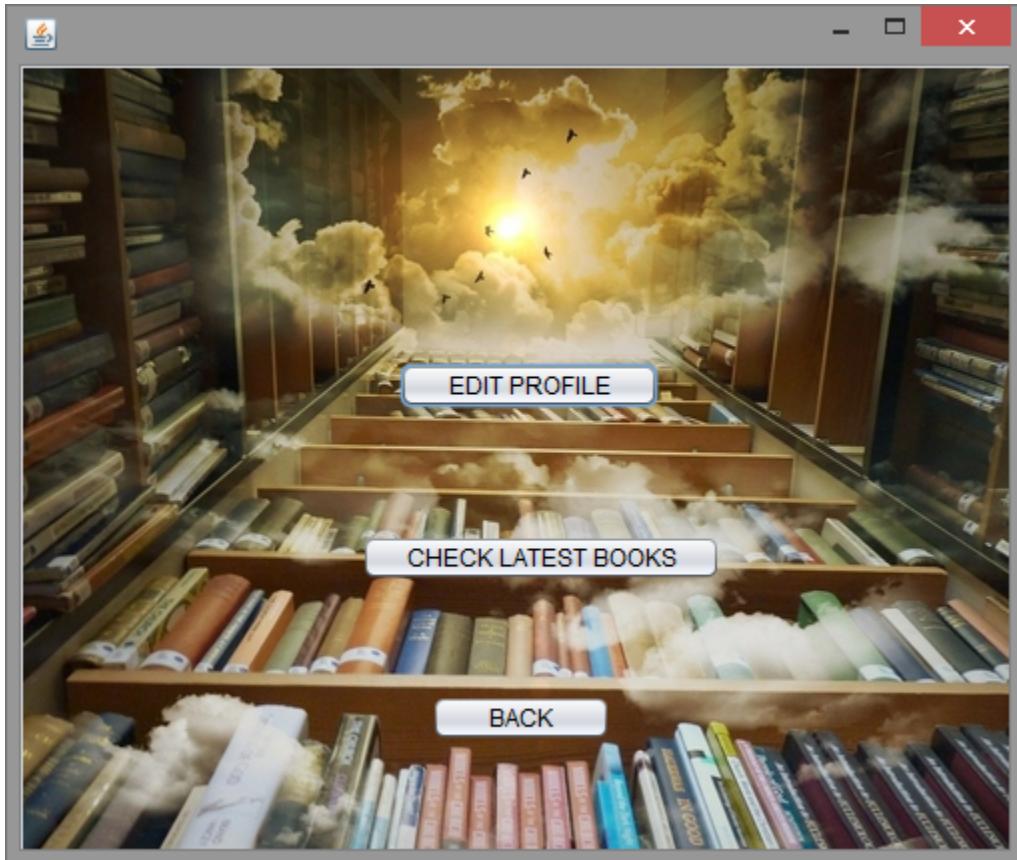
FOR REGISTERED OR EXISTING USERS:

An existing user will be redirected to the user's main portal. A JOptionPane message dialog will be used to display if the user has logged on.



USER GATEWAY

The user gateway leads the user to explore the library under two broad options. A user can edit his or her profile, or can even check out the latest books.



Code for edit profile button :

```
IPUserData1 ep = new IPUserData1(uname);
ep.setVisible(false);
Editprofile ep1 = new Editprofile(uname);
ep1.setVisible(true);
this.dispose();
Editprofileepf = new Editprofile(uname);
```

Code for Check latest books button:

```
IPUserData1 udata = new IPUserData1(uname);
udata.setVisible(false);

IPUserLatestBooks1 latest = new IPUserLatestBooks1(uname);
latest.setVisible(true);

this.dispose();
```

Code for back button:

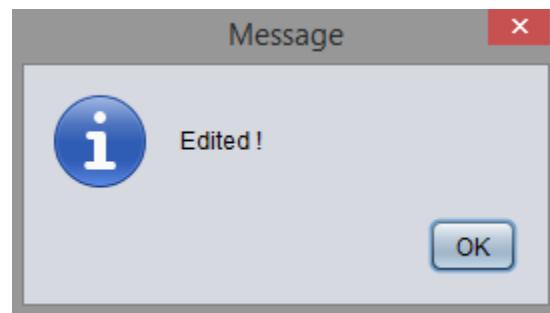
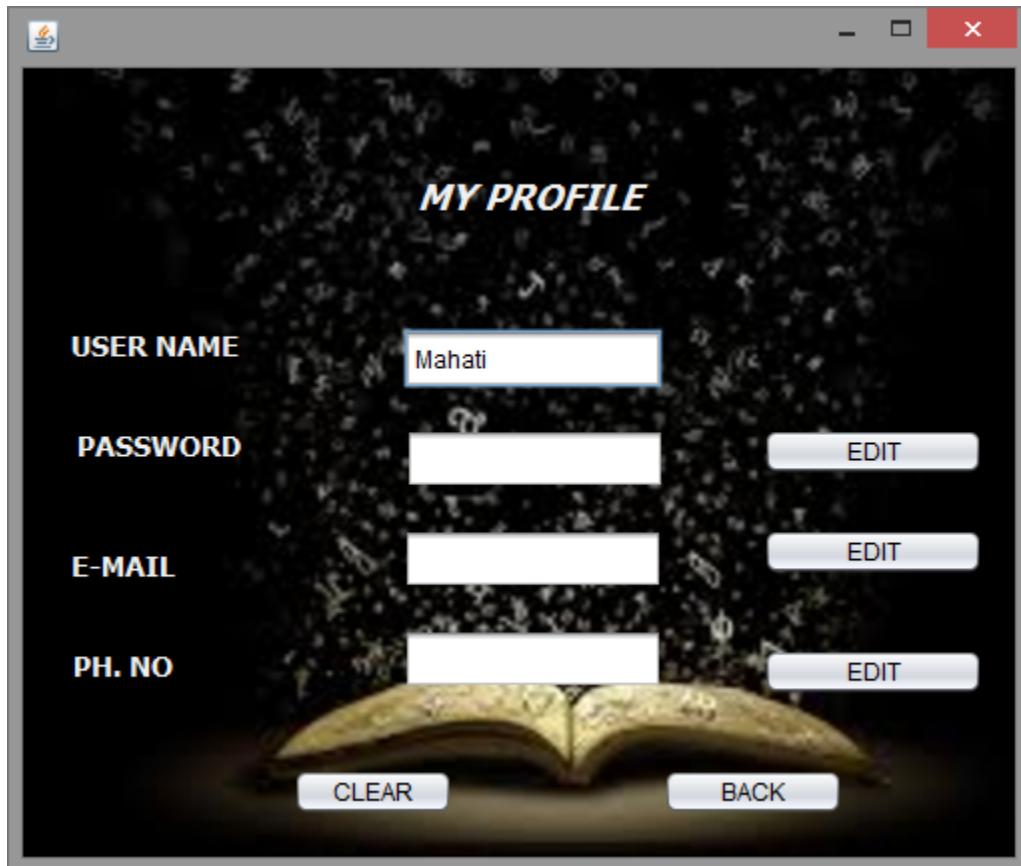
```
IPUserData1 ud1 = new IPUserData1(uname);
ud1.setVisible(false);

Ipmain main2 = new Ipmain();
main2.setVisible(true);

this.dispose();
```

EDIT PROFILE

This frame allows the user to change his or her password, e-mail or phone number which are accordingly changed in the tables. If a user wishes to change only one of the three field, that can be done too.



```

public class Editprofile extends javax.swing.JFrame {

    String uname;
    Ipmainabc = new Ipmain();

    public Editprofile(String uname) {
        initComponents();
        this.uname = uname;
        jTextField1.setText(uname);
    }
}

```

Code for Editing Password Button:

```

String uname1 = jTextField1.getText();
String pwd = new String(jPasswordField1.getPassword());
try {
    if (uname1.length() > 0) {
        Class.forName("java.sql.Driver");
        Connection con =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a", "root", "");
        Statement stmt = con.createStatement();
        if (pwd.length() > 0) {
            String query1 = "update user_records set pswd= " + pwd + "where uname = " +
            uname1 + ";";
            System.out.println(query1);
            int r = stmt.executeUpdate(query1);
            JOptionPane.showMessageDialog(null, " Edited successfully");
        }
    }
}

```

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

```
}
```

Code for editing e-mail button:

```
String uid = jTextField1.getText();
String mail = jTextField3.getText();
try {
if (uid.length() > 0) {
Class.forName("java.sql.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a", "root", "");
Statement stmt = con.createStatement();
if (mail.length() > 0) {
String query2 = "update user_records set email= " + mail + "where uid = " + uid +
"";
System.out.println(query2);
int r = stmt.executeUpdate(query2);
JOptionPane.showMessageDialog(null, " Edited successfully");
}
}
} catch (Exception e) {
System.out.println(e);
}
```

```
}
```

Code for editing phone number button:

```
String uid= jTextField1.getText();
intphno= Integer.parseInt(jTextField4.getText());
try{
    if(uid.length()>0)
    {
        Class.forName("java.sql.Driver");
        Connection con=
        DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
        Statement stmt = con.createStatement();
        if(String.valueOf(phno).length()>0){
            String query3 = "update user_records set phone = " +phno+ " where uid =
"+uid+";";
            System.out.println(query3);
            int r = stmt.executeUpdate(query3);
            JOptionPane.showMessageDialog(null, "Edited !");
        }
    }
}
catch(Exception e){
    System.out.println(e);
}
```

Code for clear button:

```
jTextField1.setText(" ");  
jTextField3.setText(" ");  
jTextField4.setText(" ");  
jPasswordField1.setText(" ");
```

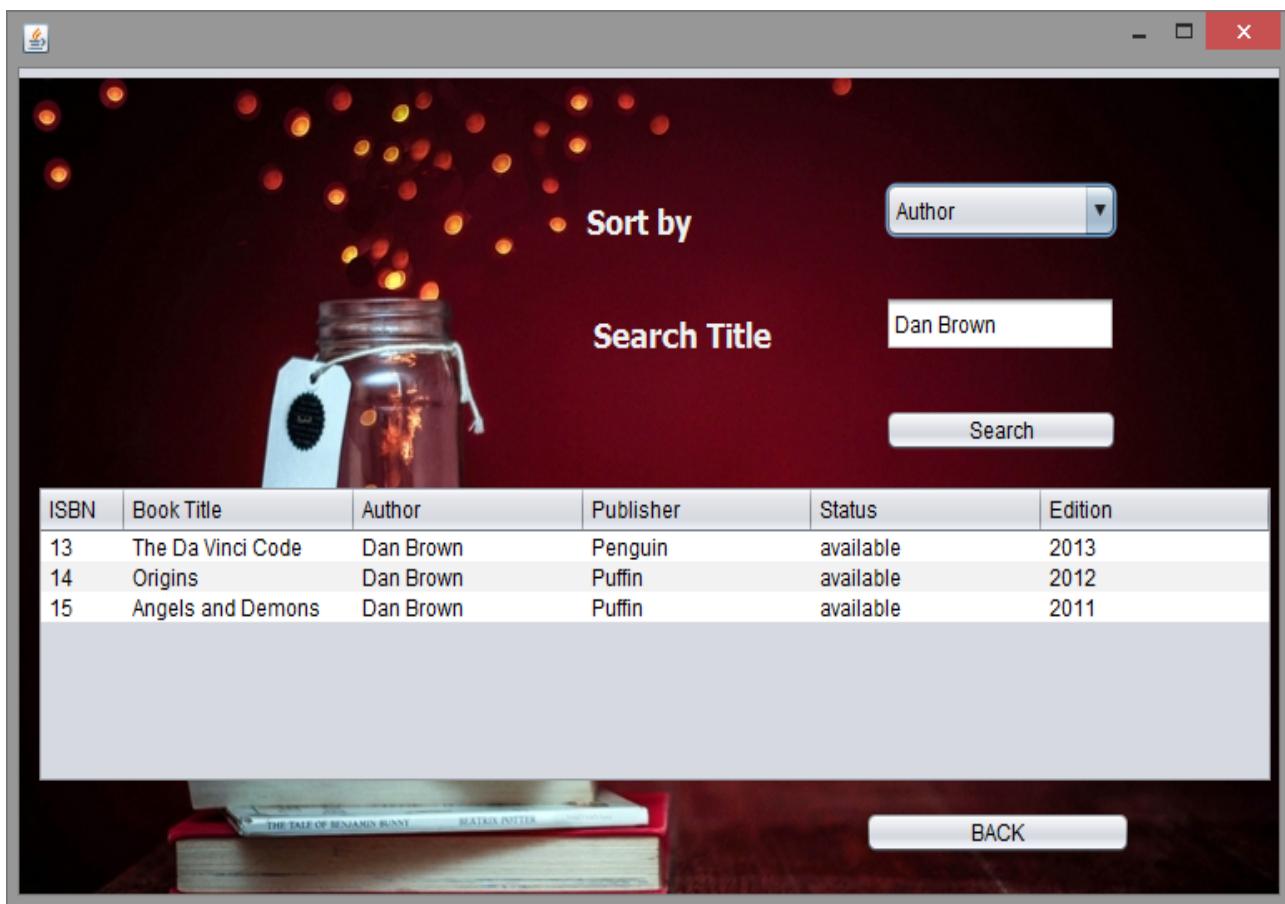
Code for back button:

```
Editprofile user = new Editprofile(uname);  
user.setVisible(false);  
IPUserData1 main3 = new IPUserData1(uname);  
main3.setVisible(true);  
this.dispose();
```

CHECK LATEST BOOKS

This portal allows the user to explore the available range of books which are shown in the jTable control.

Search can be filtered or sorted by book title, author or the publisher.



Code for search button:

```
String sort = jTextField1.getText();
String choose = jComboBox1.getSelectedItem().toString();
```

```

try{
    Class.forName("java.sql.Driver");
    Connection con=
    DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
    Statement stmt = con.createStatement();
    String query ="select ISBN,Book_Title,Author,Publisher, Status,edition from book1
where "+choose+" like '%" +sort+"%' order by "+choose+";";
    System.out.println(query);
    ResultSetrs= stmt.executeQuery(query);
    DefaultTableModel dm= (DefaultTableModel)jTable1.getModel();
    while(rs.next()){
        String isbn=rs.getString(1);
        String bTitle=rs.getString(2);
        String author=rs.getString(3);
        String publisher=rs.getString(4);
        String stat=rs.getString(5);
        String edi=rs.getString(6);
        dm.addRow(new Object[] {isbn,bTitle,author,publisher,stat,edi});
    }
}
catch(Exception e){
    System.out.println(e);
}

```

Code for back button:

```

IPUserLatestBooks1 user1 = new IPUserLatestBooks1();
user1.setVisible(false);
IPUserData1 main3 = new IPUserData1();
main3.setVisible(true);

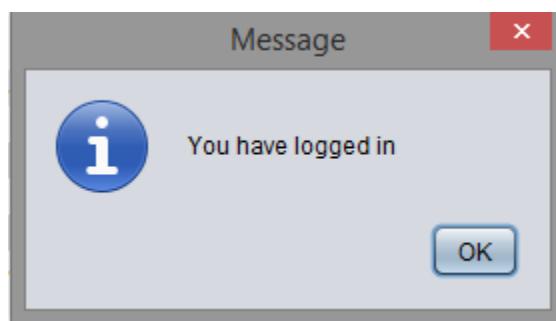
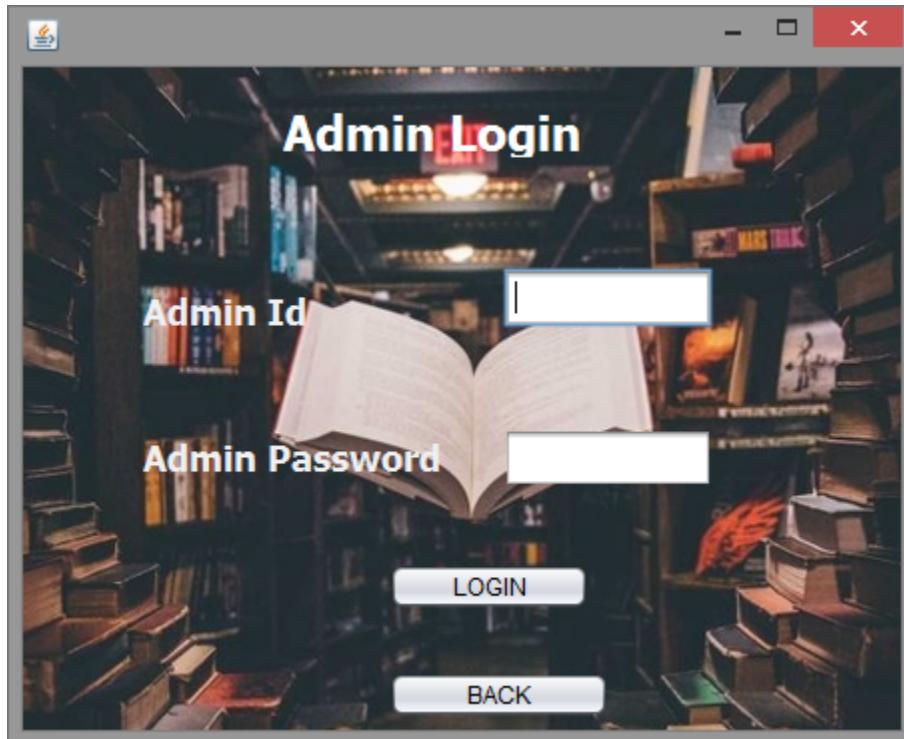
```

```
this.dispose();
```

ADMIN SECTION

This section is for the admin to monitor the activities of the user. It enables them to add books to the library, delete books from the library and help the user issue or return their books.

It can be accessed only by admin with a common username (**admin**) and common password (**admin234**).



Code for login button :

```
String str1=jTextField1.getText();
    String passad = new String(jPasswordField1.getPassword());
    if (str1.equals("admin")&&passad.equals("admin234")){
        JOptionPane.showMessageDialog(null," You have logged in");

        IP2_AdminLogin1 adm = new IP2_AdminLogin1();
        adm.setVisible(false);
        IP2LoginNext1 next = new IP2LoginNext1();
        next.setVisible(true);
        this.dispose();

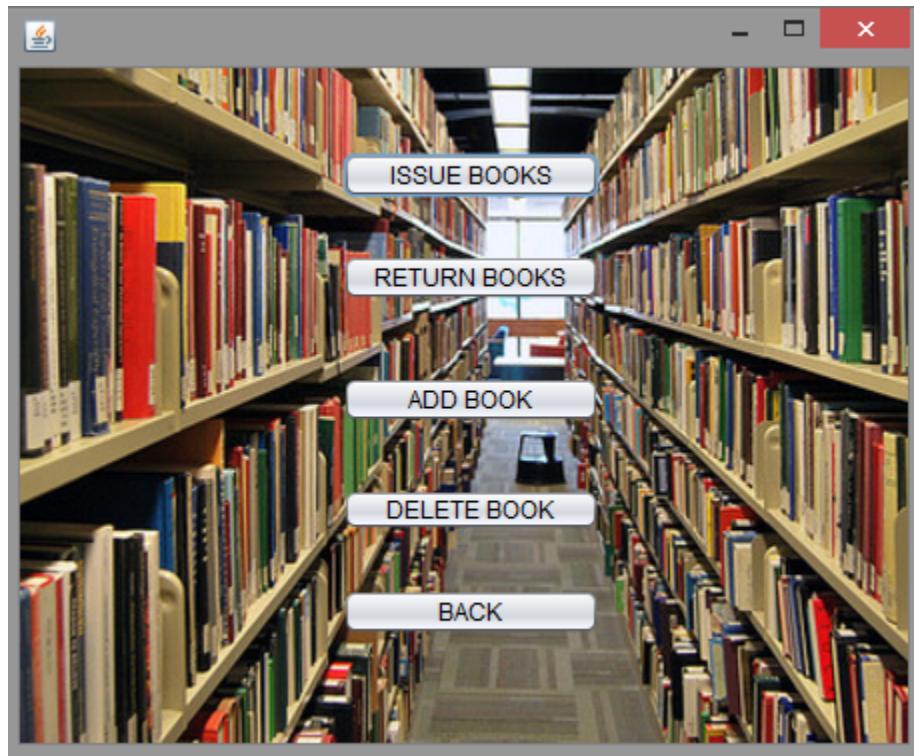
    }
    else {
        JOptionPane.showMessageDialog(null," Sorry, wrong password or username!");
    }
}
```

Code for back button:

```
IP2_AdminLogin1 login2 = new IP2_AdminLogin1();
login2.setVisible(false);
IP1 main = new IP1();
main.setVisible(true);
this.dispose();
```

ADMIN'S PORTAL

This portal opens a gateway to the various frames which relate to the user's activities and choices.



Code for Issue books button :

```
IP2LoginNext1 main5 = new IP2LoginNext1();
main5.setVisible(false);
IPMemberBookIssue1 issue1 = new IPMemberBookIssue1();
issue1.setVisible(true);
this.dispose();
```

Code for Return books button :

```
IP2LoginNext1 main6 = new IP2LoginNext1();
main6.setVisible(false);
ReturnBook issue5 = new ReturnBook();
issue5.setVisible(true);
this.dispose();
```

Code for add book button ;

```
IP2LoginNext1 bc1 = new IP2LoginNext1();
bc1.setVisible(false);
BookConsole12 bc2 = new BookConsole12();
bc2.setVisible(true);
this.dispose();
```

Code for Delete book button :

```
IP2LoginNext1 bc1 = new IP2LoginNext1();
bc1.setVisible(false);
DeleteBookdb = new DeleteBook();
db.setVisible(true);
this.dispose();
```

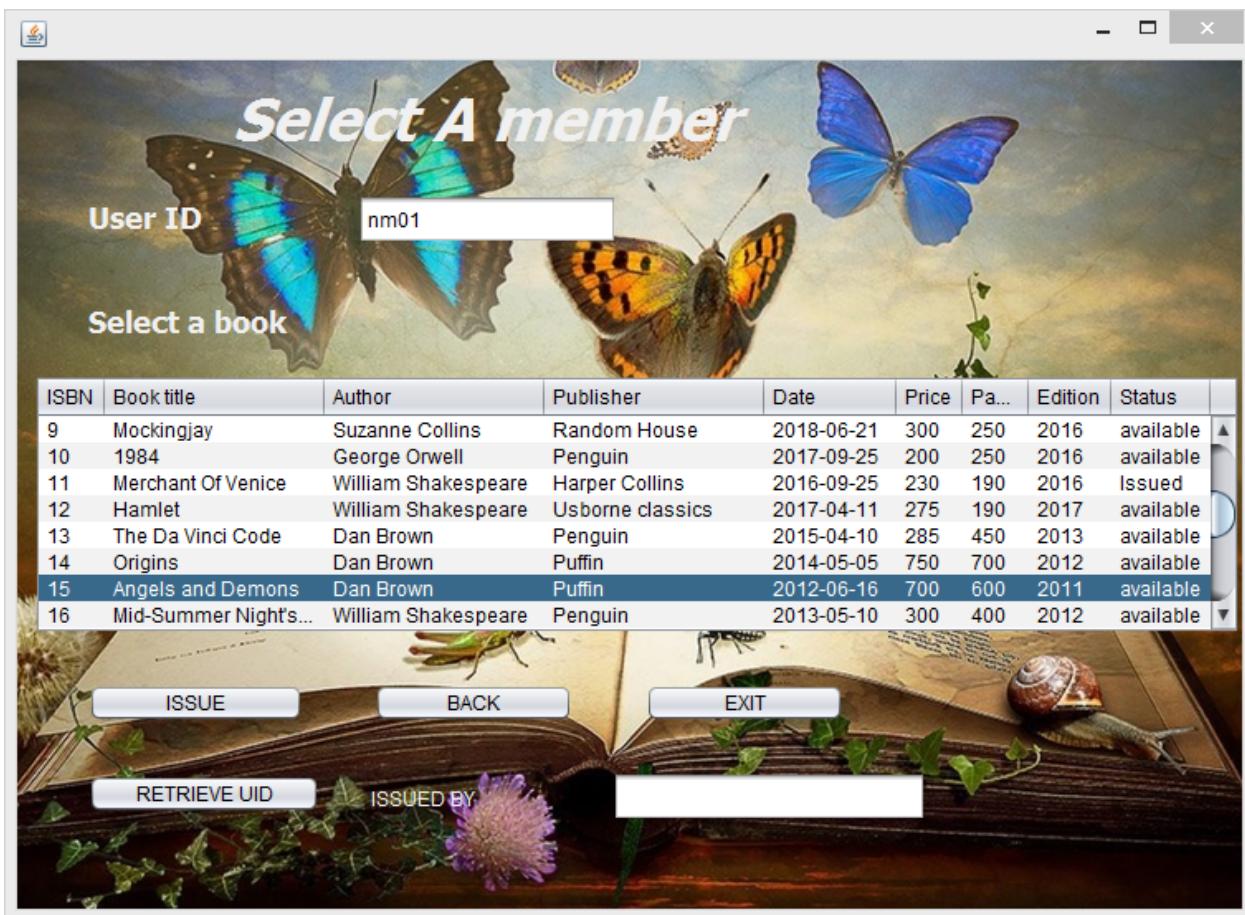
Code for back button :

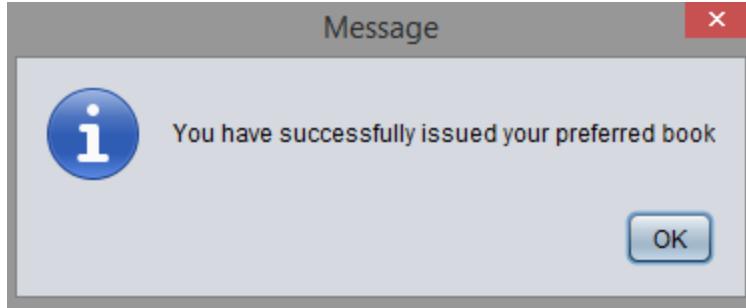
```
IP2LoginNext1 ln2 = new IP2LoginNext1();
ln2.setVisible(false);
IP2_AdminLogin1 al1 = new IP2_AdminLogin1();
al1.setVisible(true);
this.dispose();
```

ISSUE BOOKS:

This frame is used by the admin to issue a user his/her preferred book. The admin can check which book has been issued to which user by retrieving his or her uid.

This retrieved Uid can be used by the admin to return a book to the library, which that particular user has borrowed.





Code for Issue book button :

```
DefaultTableModel myTable1= ( DefaultTableModel)jTable1.getModel();
int row= jTable1.getSelectedRow();
int col=jTable1.getSelectedColumn();
String uid = jTextField1.getText().trim();
String val=(String)myTable1.getValueAt(row,0).toString();
try{
    Class.forName("java.sql.Driver");
    Connection
    con=DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
    Statement stmt=con.createStatement();
    if(uid.length()>0)
    {
        System.out.println(val);
        String sp=(String)myTable1.getValueAt(row,8);
        if(sp.equals("Issued")){
            JOptionPane.showMessageDialog(null, "Sorry, this book has already been issued ");
        }
    }
}
```

```

else{
    String str="update book1 set status= 'Issued' where ISBN='"+val+"';"
    System.out.println(str);
    intrs=stmt.executeUpdate(str);  }
}

else
{
JOptionPane.showMessageDialog(null, "Please Enter UID")
}

catch(Exception e){

}

try{
Class.forName("java.sql.Driver");
Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
String sn=(String)myTable1.getValueAt(row,1);
Statement stmt=con.createStatement();
if(uid.length()>0){
String str2="update user_records set curbook = '" +sn+"' ' where uid='"+uid+"';";
int rs2=stmt.executeUpdate(str2);

JOptionPane.showMessageDialog(null, "You have successfully issued your preferred book");
}
}

```

```
    }  
catch(Exception e){  
  
    }  

```

Code for back button:

```
IPMemberBookIssue1 issue1 = new IPMemberBookIssue1();  
issue1.setVisible(false);  
IP2LoginNext1 main4 = new IP2LoginNext1();  
main4.setVisible(true);  
this.dispose();
```

Code for exit button:

```
System.exit(0);
```

Code for Retrieve UID button:

```
DefaultTableModel myTable1= ( DefaultTableModel)jTable1.getModel();  
int row= jTable1.getSelectedRow();  
int col=jTable1.getSelectedColumn();  
String val=(String)myTable1.getValueAt(row,0).toString();  
try{  
  
Class.forName("java.sql.Driver");  
Connection  
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
}
```

```

Statement stmt=con.createStatement();

System.out.println(val);

String spf=(String)myTable1.getValueAt(row,1);

String shr= "Select uid from user_records where curbook= '"+ spf+"',";

ResultSet r2=stmt.executeQuery(shr);

r2.next();

System.out.println( r2.getString(1));

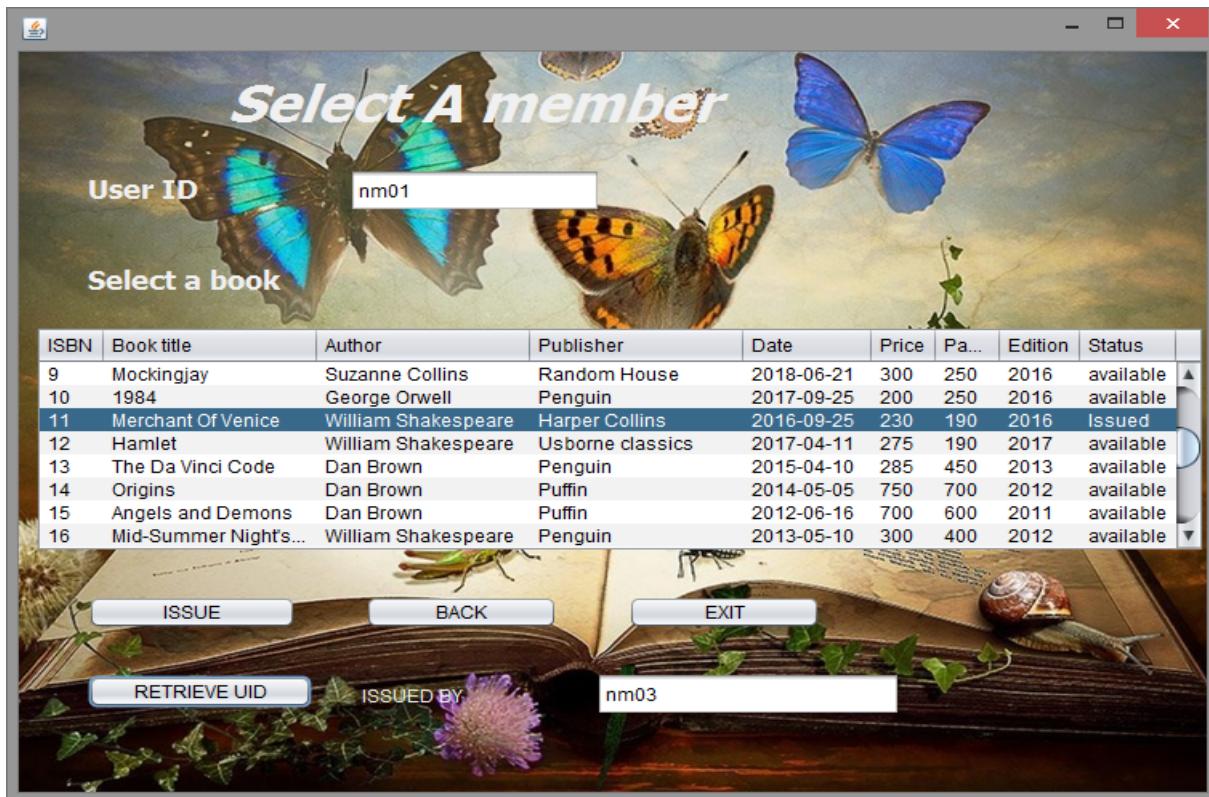
jTextField2.setText("'" + r2.getString(1));

}

catch(Exception e){

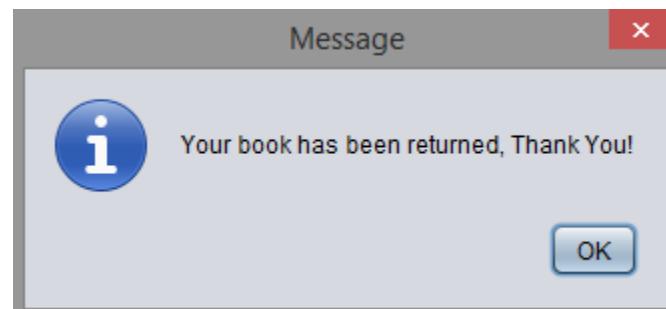
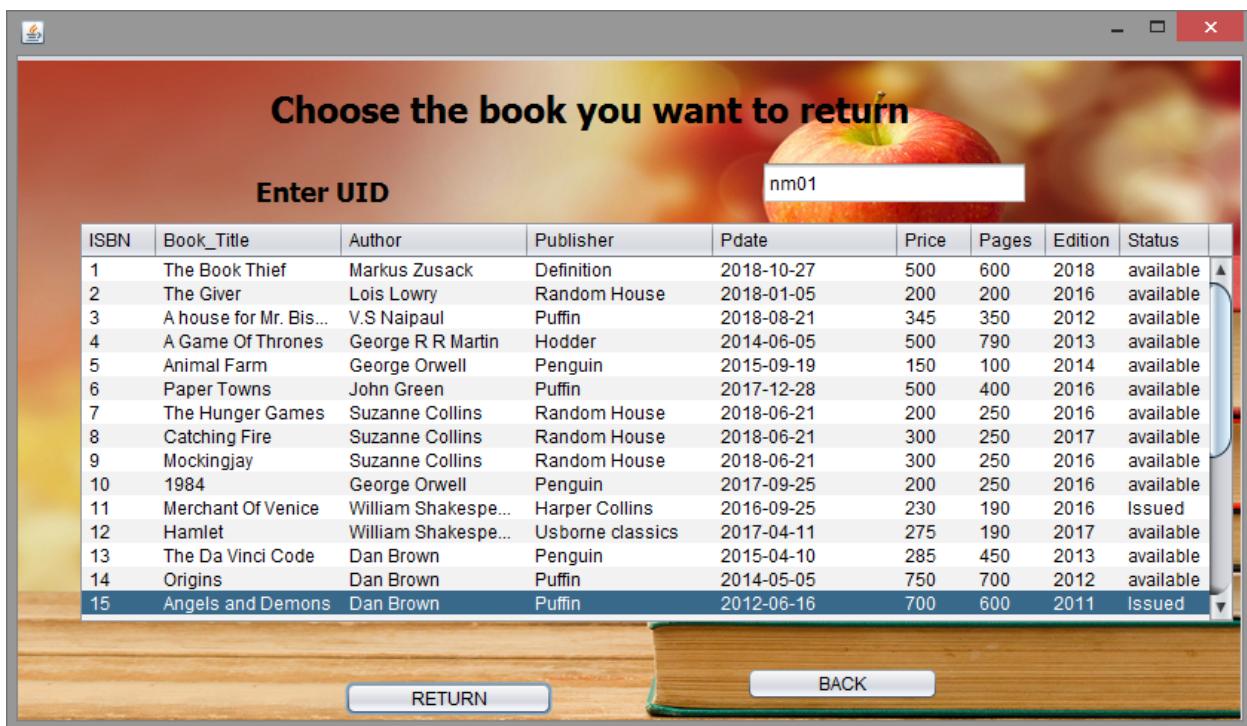
}

```



RETURN BOOKS

This section is used by the admin to return a book issued by a particular user to the library. The changes are updated in the database table.



Code for return button:

```
DefaultTableModel myTable1= ( DefaultTableModel)jTable1.getModel();
int row=jTable1.getSelectedRow();
int col=jTable1.getSelectedColumn();
String str6=jTextField1.getText().trim();
String val5=(String)myTable1.getValueAt(row,0).toString();
try{
    Class.forName("java.sql.Driver");
    Connection
    con=DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
    Statement stmt=con.createStatement();

    System.out.println(val5);
    String sp=(String)myTable1.getValueAt(row,8);
    if(str6.length()>0)
    {
        if(sp.equals("Issued")){
            JOptionPane.showMessageDialog(null, "Your book has been returned, Thank You! ");
            String str="update book1 set status= 'available' where ISBN="+val5+";";
        }
    }
}
```

```

System.out.println(str);
intrs=stmt.executeUpdate(str);

}

}

else

{

JOptionPane.showMessageDialog(null,"Please enter your UID");

}

}

catch(Exception e){

}

try{

Class.forName("java.sql.Driver");

Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
String sn=(String)myTable1.getValueAt(row,1);

Statement stmt=con.createStatement();

if(str6.length()>0)

{

String str5="update user_records set curbook = 'Nil ' where uid='"+str6+"';";
int rs2=stmt.executeUpdate(str5); }

}

catch(Exception e){



}

```

```
}

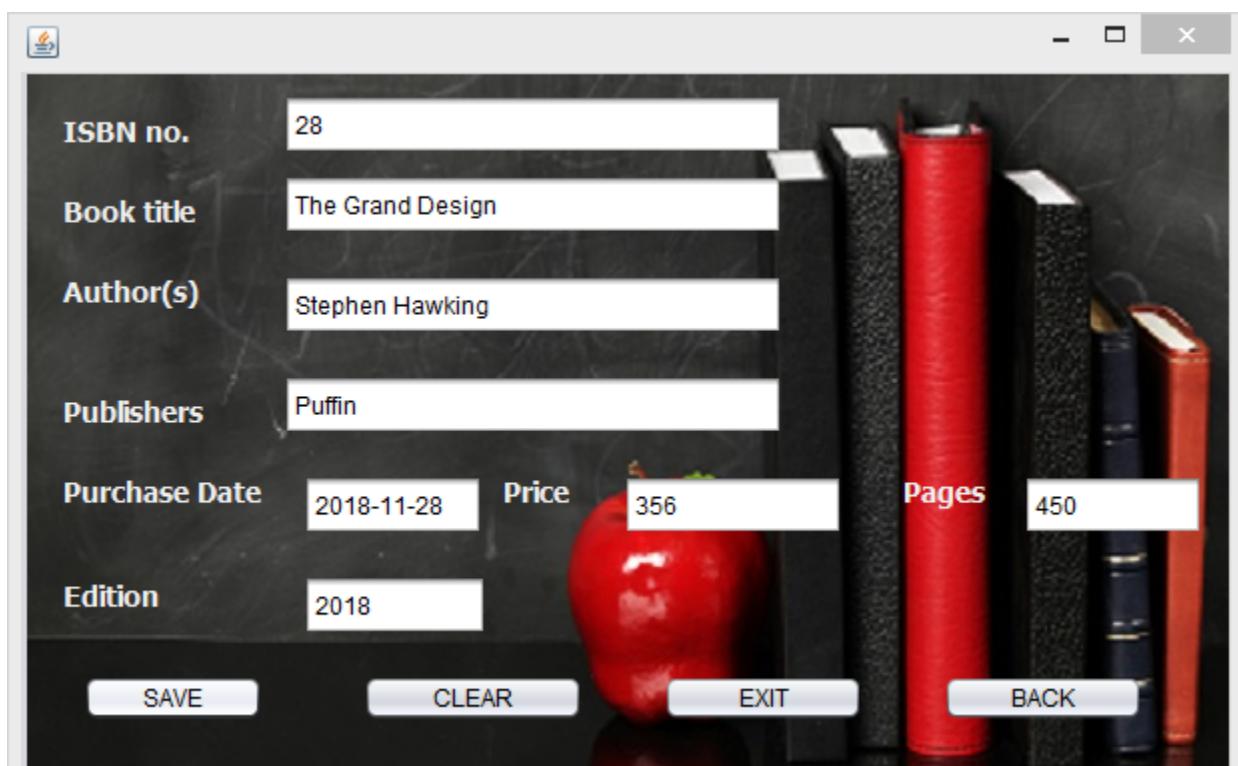
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    ReturnBook issue2 = new ReturnBook();
    issue2.setVisible(false);
    IP2LoginNext1 main5 = new IP2LoginNext1();
    main5.setVisible(true);
    this.dispose();
}
```

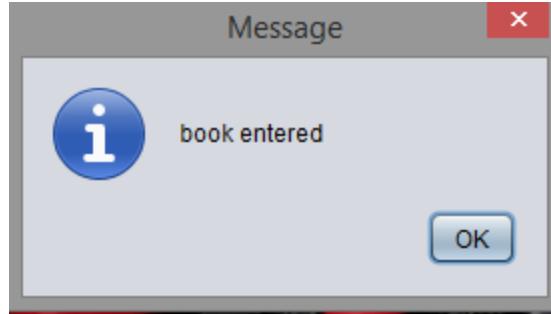
Code for back button:

```
ReturnBook issue2 = new ReturnBook();
issue2.setVisible(false);
IP2LoginNext1 main5 = new IP2LoginNext1();
main5.setVisible(true);
this.dispose();
```

ADD BOOK

This frame is used by the admin to add a book to the library.





Code for save button:

```
int ISBN = Integer.parseInt(jTextField1.getText());
String BT = jTextField2.getText();
String author = jTextField3.getText();
String pub = jTextField4.getText();
String pdate = jTextField5.getText();
double price = Double.parseDouble(jTextField6.getText());
int pages = Integer.parseInt(jTextField7.getText());
String edition = jTextField8.getText();
if(ISBN==' '||BT.equals("")||author.equals("")||pub.equals("")||pdate.equals("")||price==' '
'||pages==' '||edition.equals("")))
{
    JOptionPane.showMessageDialog(null,"Please enter required details");
}
else{
    try{
        Class.forName("java.sql.Driver");
        Connection con=
        DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
    }
}
```

```

Statement stmt = con.createStatement();

String query = "insert into book1 values("+ISBN+","+BT+"','"+author+
"+,""+pub+"','"+pdate+"','"+price+"','"+pages+"','"+edition+"','available');";

System.out.println(query);
int succ= stmt.executeUpdate(query);

if(succ>0){

JOptionPane.showMessageDialog(null,"book entered");

}

else{

JOptionPane.showMessageDialog(null,"Please try again ");

}

catch(Exception e){

System.out.println(e);

}

}

```

Code for clear button:

```

jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField5.setText("");
jTextField6.setText("");
jTextField7.setText("");
jTextField8.setText("");

```

Code for exit button:

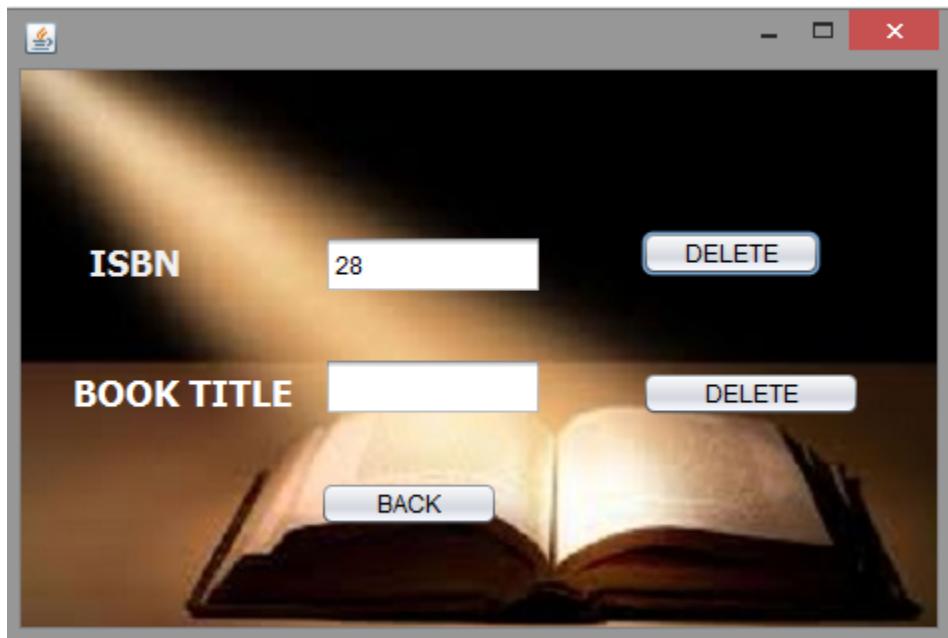
```
System.exit(0);
```

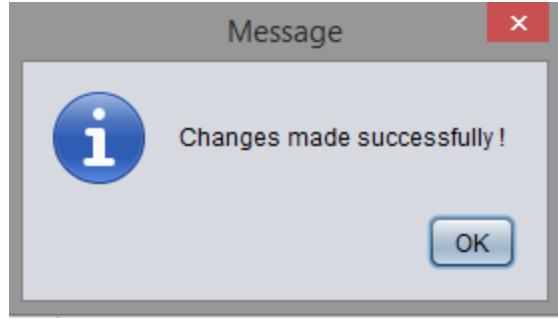
Code for back button:

```
BookConsole12 bc1 = new BookConsole12();
bc1.setVisible(false);
IP2LoginNext1 ln1 = new IP2LoginNext1();
ln1.setVisible(true)
this.dispose();
```

DELETE BOOK

This frame is used by the admin to delete a book from the library.





Code for delete ISBN button:

```
int ISBN = Integer.parseInt(jTextField1.getText());  
  
String name = jTextField2.getText();  
  
try{  
    Class.forName("java.sql.Driver");  
    Connection con=  
    DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
    Statement stmt = con.createStatement();  
    String query4="delete from book1 where isbn =" +ISBN+ ":";  
    intrs=stmt.executeUpdate(query4);  
    JOptionPane.showMessageDialog(null,"Changes made successfully !");  
}  
catch(Exception e){  
    System.out.println(e);  
}
```

Code for delete book title button:

```
String name = jTextField2.getText();  
  
try{  
    Class.forName("java.sql.Driver");
```

```

Connection con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/library_12a","root","");
Statement stmt = con.createStatement();
String query4="delete from book1 where Book_Title = '"+name+"';";
intrs=stmt.executeUpdate(query4);
JOptionPane.showMessageDialog(null,"Changes made successfully !");

}

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

```

Code for back button:

```

IP2LoginNext1 bc1 = new IP2LoginNext1();
bc1.setVisible(true);
DeleteBookdb = new DeleteBook();
db.setVisible(false);
this.dispose();

```

MYSQL TABLES IN THE DATABASE

Book1 table:

ISBN	Book_Title	Author	Publisher	Pdate	price	pgs	edition	status
1	The Book Thief	Markus Zusack	Puffin	2018-10-27	500	600	2018	available
2	The Giver	Lois Lowry	Random House	2018-01-05	200	200	2016	available
3	A house for Mr. Biswas	V.S Naipaul	Puffin	2018-08-21	345	350	2012	available
4	A Game Of Thrones	George R R Martin	Hodder	2014-06-05	500	790	2013	available
5	Animal Farm	George Orwell	Penguin	2015-09-19	150	100	2014	available
6	Paper Towns	John Green	Puffin	2017-12-28	500	400	2016	available
7	The Hunger Games	Suzanne Collins	Random House	2018-06-21	200	250	2016	available
8	Catching Fire	Suzanne Collins	Random House	2018-06-21	300	250	2017	available
9	Mockingjay	Suzanne Collins	Random House	2018-06-21	300	250	2016	available
10	1984	George Orwell	Penguin	2017-09-25	200	250	2016	available
11	Merchant Of Venice	William Shakespeare	Harper Collins	2016-09-25	230	190	2016	available
12	Hamlet	William Shakespeare	Usborne classics	2017-04-11	275	190	2017	available
13	The Da Vinci Code	Dan Brown	Penguin	2015-04-10	285	450	2013	available
14	Origins	Dan Brown	Puffin	2014-05-05	750	700	2012	available
15	Angels and Demons	Dan Brown	Puffin	2012-06-16	700	600	2011	available
16	Mid-Summer Night's Dream	William Shakespeare	Penguin	2013-05-10	300	400	2012	available
17	Emma	Jane Austen	Harper Collins	2015-09-19	375	250	2014	available
18	Othello-The Moor Of Venice	William Shakespeare	Harper Collins	2017-09-19	195	250	2016	available
19	Pride and Prejudice	Jane Austen	Puffin	2016-09-21	205	300	2016	available
20	David Copperfield	Charles Dickens	Penguin	2018-09-01	195	175	2017	available
21	Huckleberry Finn	Mark Twain	Usborn Classics	2014-09-01	175	205	2013	available
22	A Brief History of time	Stephen Hawking	Penguin	2015-09-10	450	205	2014	available
23	I am Malala	Malala Yousafzai	Puffin	2013-09-20	375	205	2012	available
24	Diary of A Young Girl	Anne Frank	Puffin	2017-03-20	400	250	2017	available
25	Day of the Jackal	Fredrick Forsyth	Harper Collins	2017-05-09	300	350	2016	available
26	Wings of Fire	Dr . A.P.J Abdul Kalam	Puffin	2013-05-09	350	400	2012	available
27	The Godfather	Mario Puzo	Hodder	2016-09-10	800	700	2015	available

User_records table:

UID	UNAME	PSWD	EMAIL	phone	jdate	curbook
nm01	Mahati	kuram01	mahati98@gmail.com	98554734	2018-08-15	Nil
nm02	Shreya	Bhattacharyya02	shreya1504@gmail.com	905673892	2018-08-15	Nil
nm03	Rita	Abani03	rita@gmail.com	905673892	2018-08-15	Nil
nm04	Mahima	Sriram04	maxima23@gmail.com	905673892	2018-10-03	Nil
nm05	Shagufta	Anjum05	shaggychan@hotmail.com	905673892	2018-10-22	Nil
nm06	Arthi	Chandra06	arthi@gmail.com	905673892	2018-08-15	Nil
nm07	Nainika	Rao07	NULL	905673892	2018-08-15	Nil
nm08	Kavya	Narayanan08	kavya@gmail.com	905673892	2018-08-15	Nil
nm09	Anupallavi	Mogusala09	NULL	905673892	2018-08-15	Nil
nm10	Deepti	Sunkara10	deppiti@gmail.com	905673892	2018-08-15	Nil
nm11	Ishika	Gupta11	ishika@gmail.com	6544576	2018-11-03	Nil
nm12	Sameer	Maheshwari12	sameer@hotmail.com	941267823	2018-08-15	Nil
nm13	Akrita	Kaur13	akaur@gmail.com	90537455	2018-11-28	Nil
nm14	Rathnamegha	Megha14	rathnamegha@gmail.com	9966164	2018-11-28	Nil

REFERENCES:

- <https://www.scribd.com/document/252610233/IP-project-Library-Management-System>
- Projects of Samiha Antara and Hiba Fatima Of QES Indian School, Doha, Qatar.
- Reeta Sahoo, Together With Informatics Practices, Guide Book.
- <https://www.javatpoint.com>