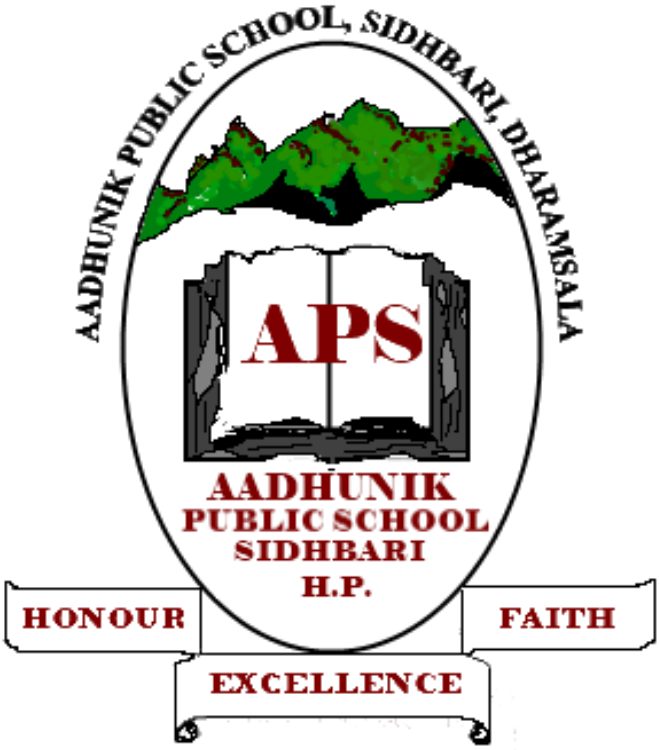
AADHUNIK PUBLIC SENIOR SECONDARY SCHOOL

SH 17, SIDHBARI, HIMACHAL PRADESH 176053




# A PROJECT ON LIBRARY MANAGEMENT

for the partial fulfillment of the AISSC Exam of CBSE of Informatics Practices (Subject Code- 065)

PRESENTED BY: VIPUL SHARMA

CLASS 12

SESSION- 2015-16

# CONTENTS

# DECLARATION

I declare that this project entitled “Library Management” done at Aadhunik Public School is a record of project work submitted by me for the partial fulfillment of the AISSC exam of CBSE under the supervision and guidance of Mrs. Vandana, our IP Teacher. The Project is genuine and not a reproduction of any project previously done project.

# ACKNOWLEDGEMENTS

I extend my sincere thanks to Aadhunik Public School which provided me with the opportunity to fulfill my wish and achieve my goal. I would like to express deep debt to Mrs. Vandana, our project guide for her vital suggestions, meticulous guidance and constant motivation which went a long way in the successful completion of this project. I cannot move on without thank beloved Principal Ms. Monika Mahajan for creating the required academic environment which made my task appreciable. On a moral personal note, my deepest appreciation and gratitude to my beloved parents, who have been an inspiration and have provided me with unrelenting encouragement and support.

# CERTIFICATE

This is to certify that the dissertation entitled, “Library Management” is a work done by _____ of class XII during the academic session 2015-2016 is partial fulfillment of CBSE’s AISSC Examination 2016 and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidate.

_____

(Signature of teacher)

# MINIMUM HARDWARE REQUIREMENTS

## PROCESSOR:

Intel Pentium IV or higher

## RAM:

1 GB or more

## HARD DISK:

20 GB or more

# MINIMUM SOFTWARE REQUIREMENTS

## OS:

Windows 7 or higher

## NETBEANS VERSION:

Netbeans 8.0

## MYSQL VERSION:

MySQL Server 5.1

## JAVA:

JRE 7 or higher

# INTRODUCTION

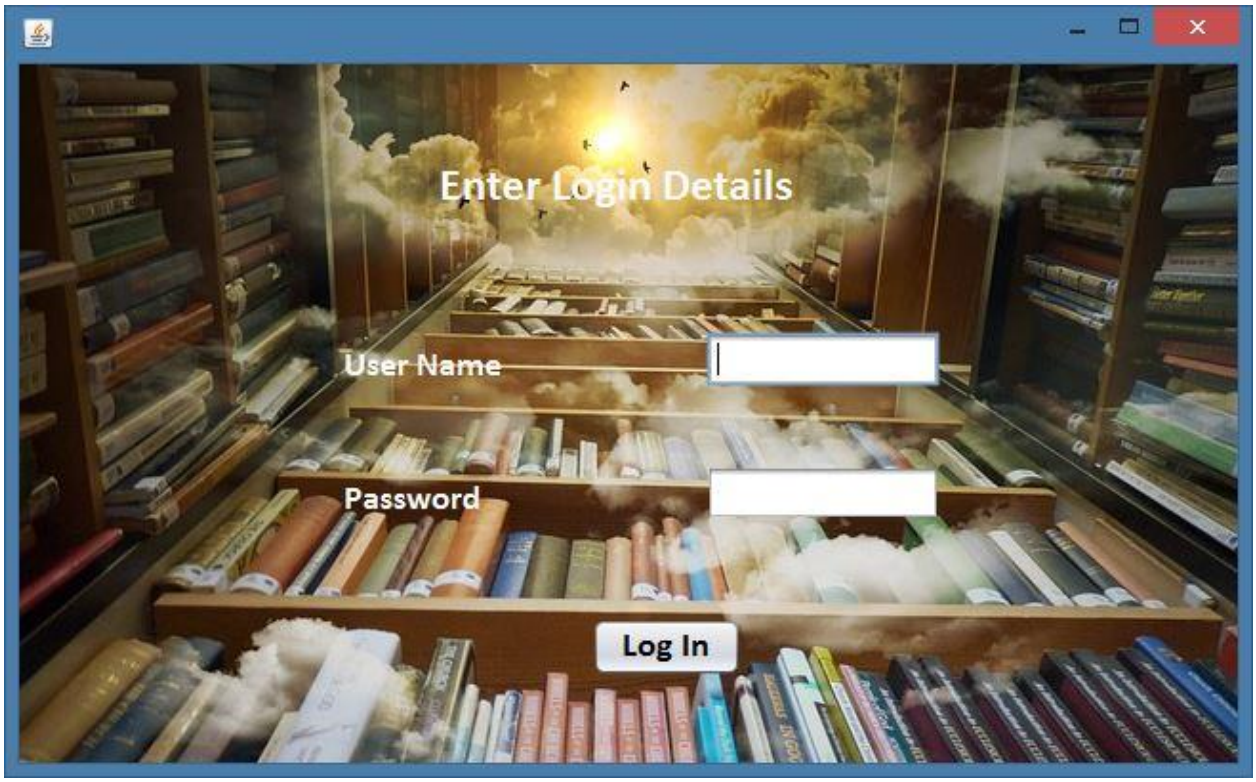
The project titled “Library Management” is Library management software for monitoring and controlling the transactions in a library. This project is designed & coded in Netbeans 8.0 and database management is handled by MySQL 5.1 . This software mainly focuses on basic operations in a library like adding new books, and updating new Information, searching books and facility to issue and return books. “Library Management” is a java Application written on 64-bit, Windows 8 OS, designed to help users maintain and organize library. My software is easy to use for both beginners and advanced users. It features a familiar and well thought-out, an attractive UI, combined with strong searching, insertion and reporting capabilities.

This project has mainly been divided into 6 segments-

1. Login Screen
2. Main Window
3. Adding Books
4. Deleting Books
5. Issuing Books
6. Receiving Books
7. View Books
8. View Issuers

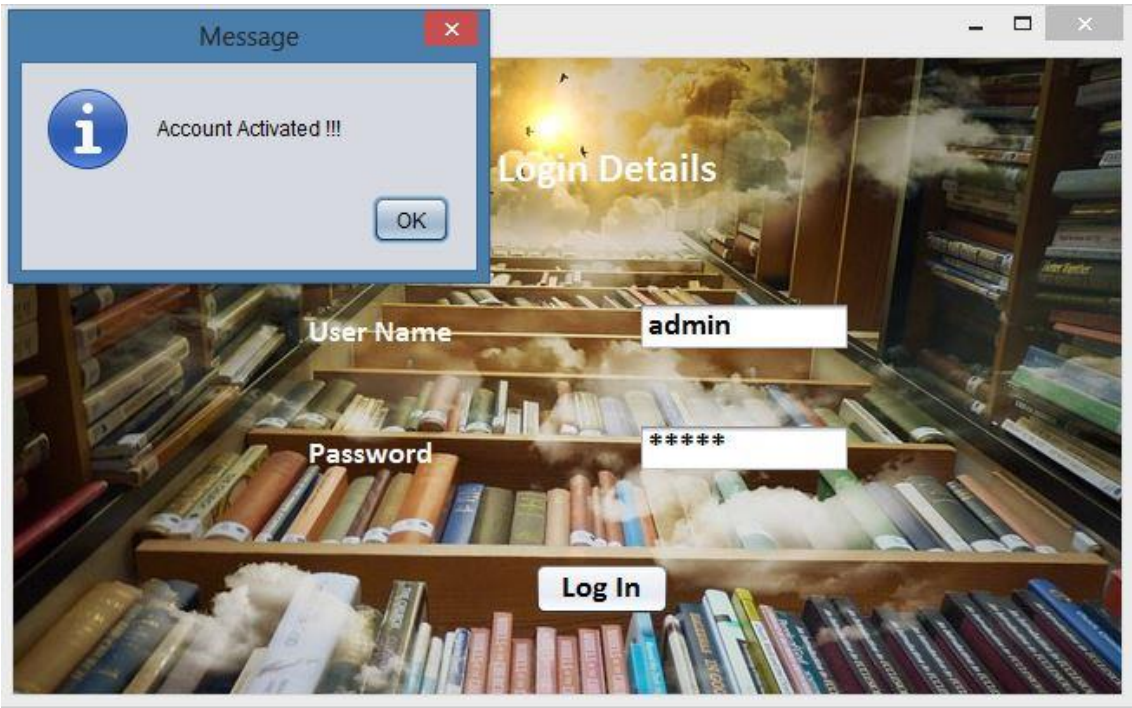
# 1. LOGIN SCREEN

As soon as you open the application, you are presented with the Login Screen with a beautiful background. The default username and password is ‘admin’ and can be changed in the code later on.



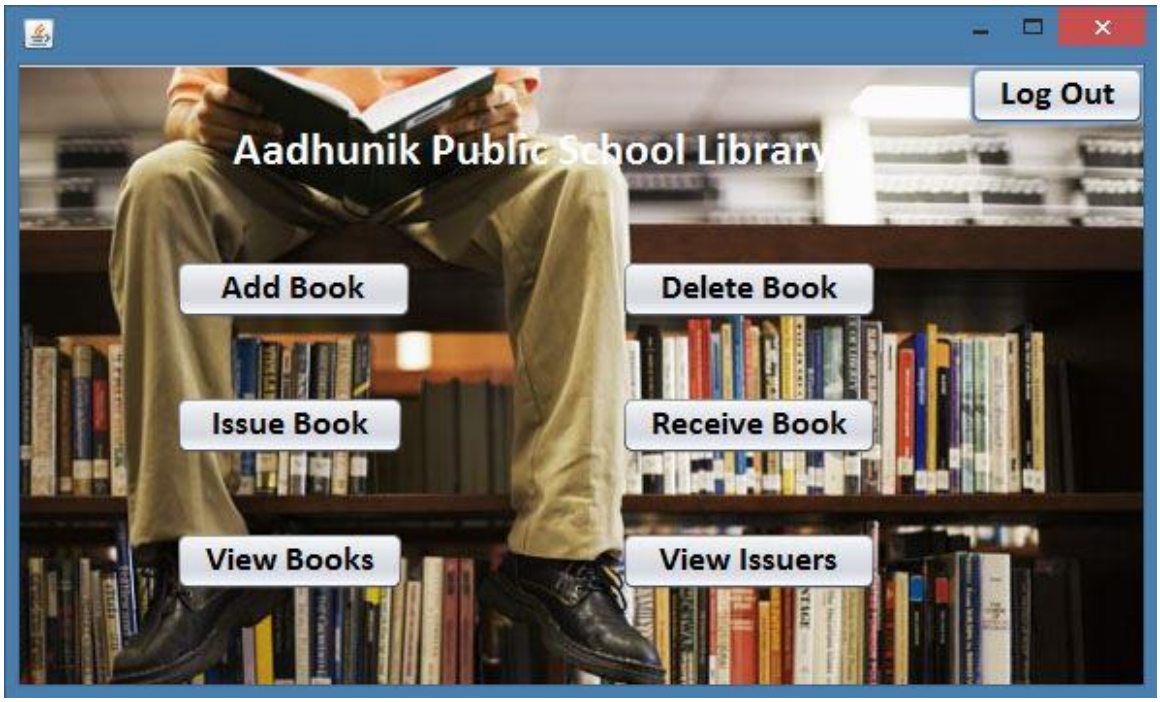

## CODE FOR THE LOGIN BUTTON-

```
String admin_name = "admin"; String admin_password="admin";
String username = jTextField1.getText();
String Password1 = jPasswordField1.getText();
 if(username.equals(admin_name)&&Password1.equals(admin_password))
 {
    JOptionPane.showMessageDialog(this,"Account Activated !!!");
    new Main().setVisible(true);
     this.setVisible(false);

 }
 else
    JOptionPane.showMessageDialog(this," Please check the Username/Password entered and try
again !!");
jTextField1.setText("");
jPasswordField1.setText("");
```




## 2. MAIN WINDOW

When the login is successful, you are presented with the main window which is divided into several segments like Add Book, Delete Book, Issue Book, Receive Book, View Issuers, View Books and the Log Out button. When the various buttons are clicked, their respective windows open and the Main Window closes.



## CODE FOR THE ADD BOOK BUTTON-

```
new AddBook().setVisible(true);
this.setVisible(false);
```

## CODE FOR THE ISSUE BOOK BUTTON-

```
new IssueBook().setVisible(true);
this.setVisible(false);
```

## CODE FOR THE RECEIVE BOOK BUTTON-

```
new ReceiveBook().setVisible(true);
this.setVisible(false);
```

## CODE FOR THE VIEW BOOKS BUTTON-

```
new ViewBooks().setVisible(true);
this.setVisible(false);
```

## CODE FOR THE VIEW ISSUERS BUTTON-

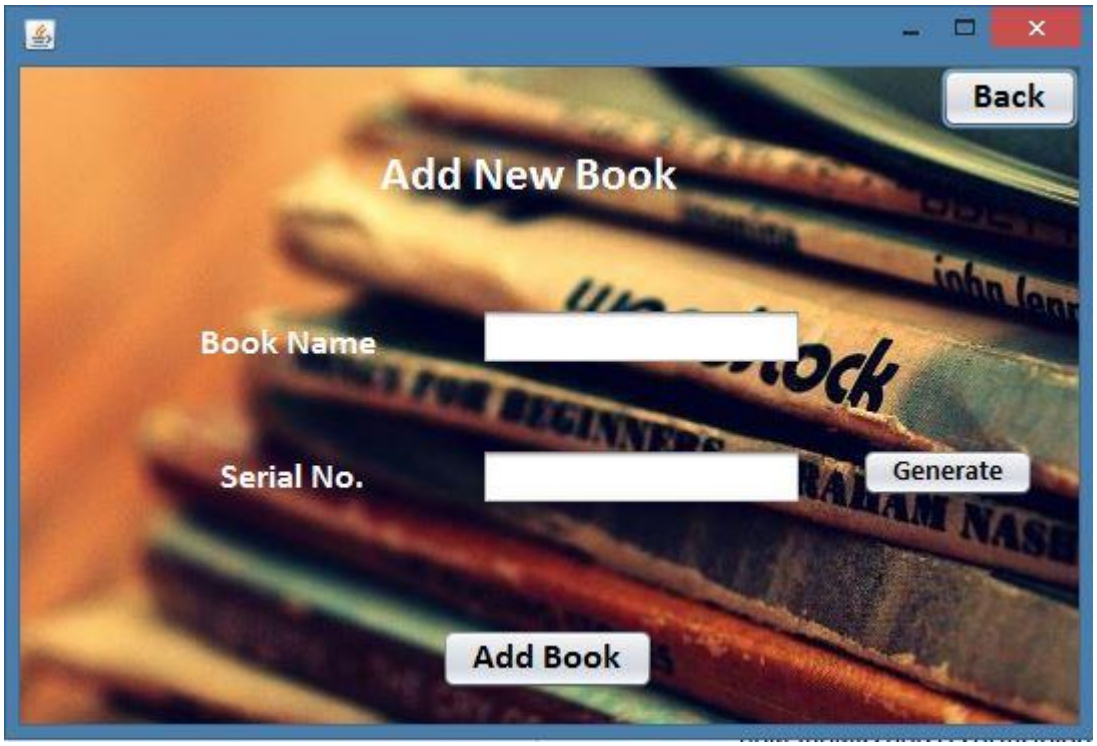
```
new ViewIssuers().setVisible(true);
this.setVisible(false);
```

## CODE FOR THE LOGOUT BUTTON-

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

# 3. ADDING BOOKS

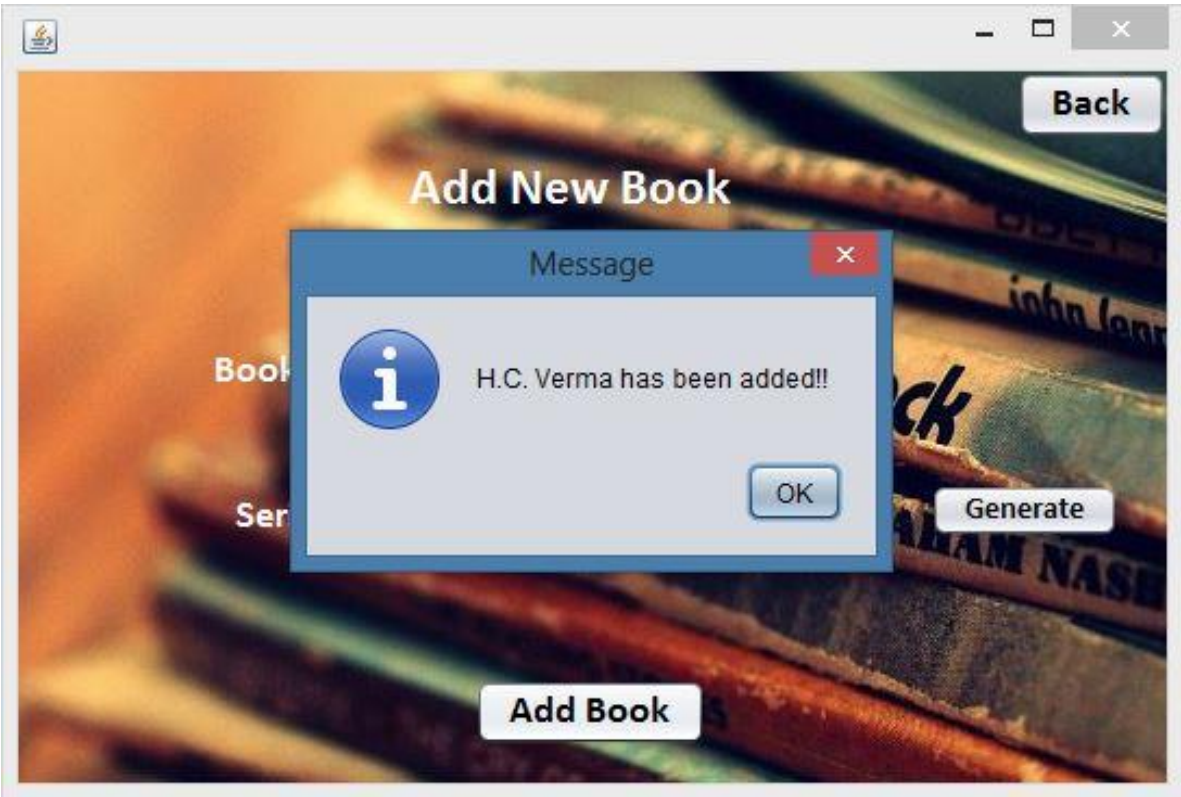
The Add Book window is used to add a new book to the Library. The next serial number, if required is automatically generated on clicking the Generate button. The Add Book button takes the book name and the serial number and saves in the back-end in a table named 'Book'.




## CODE FOR GENERATE BUTTON-

```
try
{
   Class.forName("java.sql.DriverManager");
   Connection con = (Connection)
   DriverManager.getConnection("jdbc:mysql://localhost:3306/vipul","root","apschool");
   Statement stmt = (Statement) con.createStatement();
   String query="Select Max(Serial) from book;";
   ResultSet rs=stmt.executeQuery(query);
   if (rs.next())
   {
      String Serial=rs.getString("Max(Serial)");
      jTextField2.setText(Integer.toString(Integer.parseInt(Serial)+1));
```

```
    }

}
catch (Exception e)
{
    JOptionPane.showMessageDialog(this,e.getMessage());
}
```




## CODE FOR ADD BOOK BUTTON-

```
String Name=jTextField1.getText();
String Serial=jTextField2.getText();

try
{
    Class.forName("java.sql.DriverManager");
    Connection con = (Connection)
          DriverManager.getConnection("jdbc:mysql://localhost:3306/vipul","root","apschool");
    Statement stmt = (Statement) con.createStatement();
    String query="Insert Into book VALUES('"+Name+"','"+Serial+"');";
    stmt.executeUpdate(query);
    JOptionPane.showMessageDialog(this,Name+" has been added!!");
    jTextField1.setText("");
```

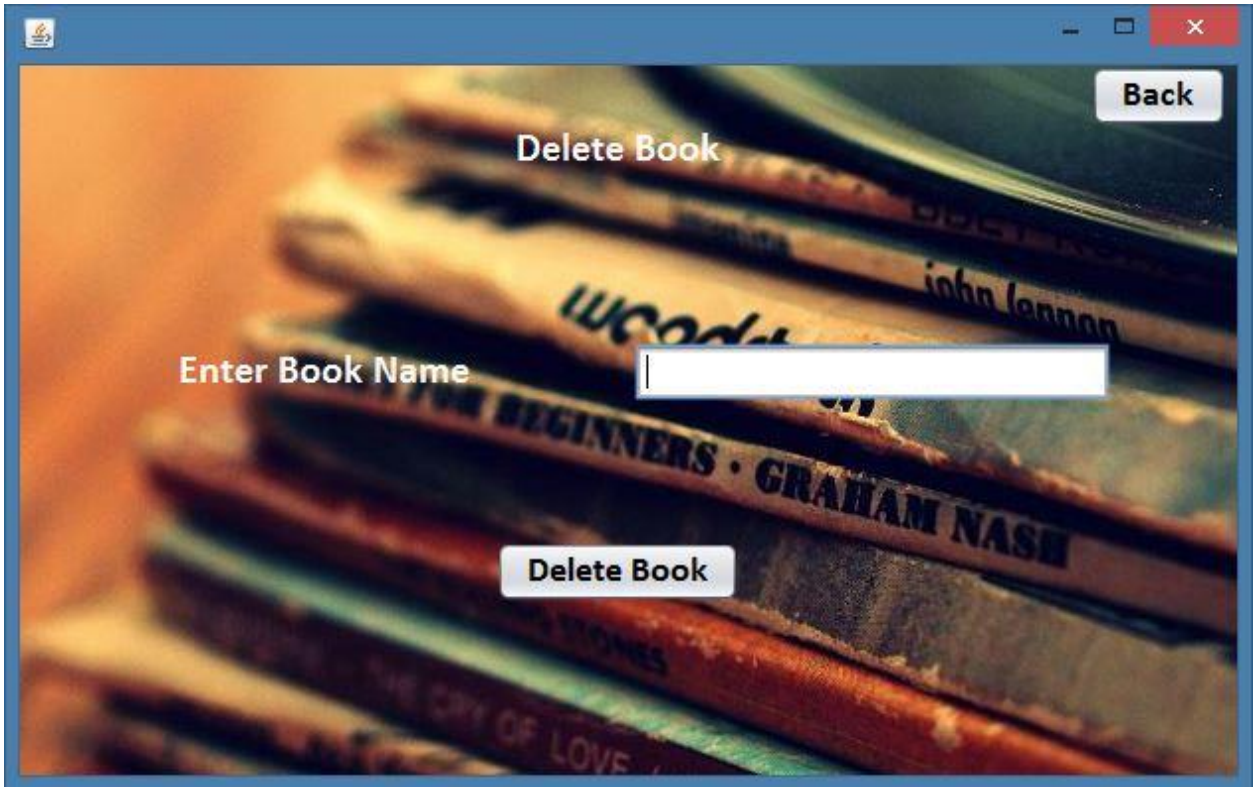
```
   jTextField2.setText("");
}
catch (Exception e)
{
   JOptionPane.showMessageDialog(this,e.getMessage());
}
```

### CODE FOR THE BACK BUTTON-

```
new Main().setVisible(true);
this.setVisible(false);
```

## 4. DELETING BOOKS

The Delete Book window is used to delete a book from the library records, i.e., the 'Book' table. The Delete Book Button takes the Book name from the text field and the deletes the record with the same name as the book name from the table 'Book'.

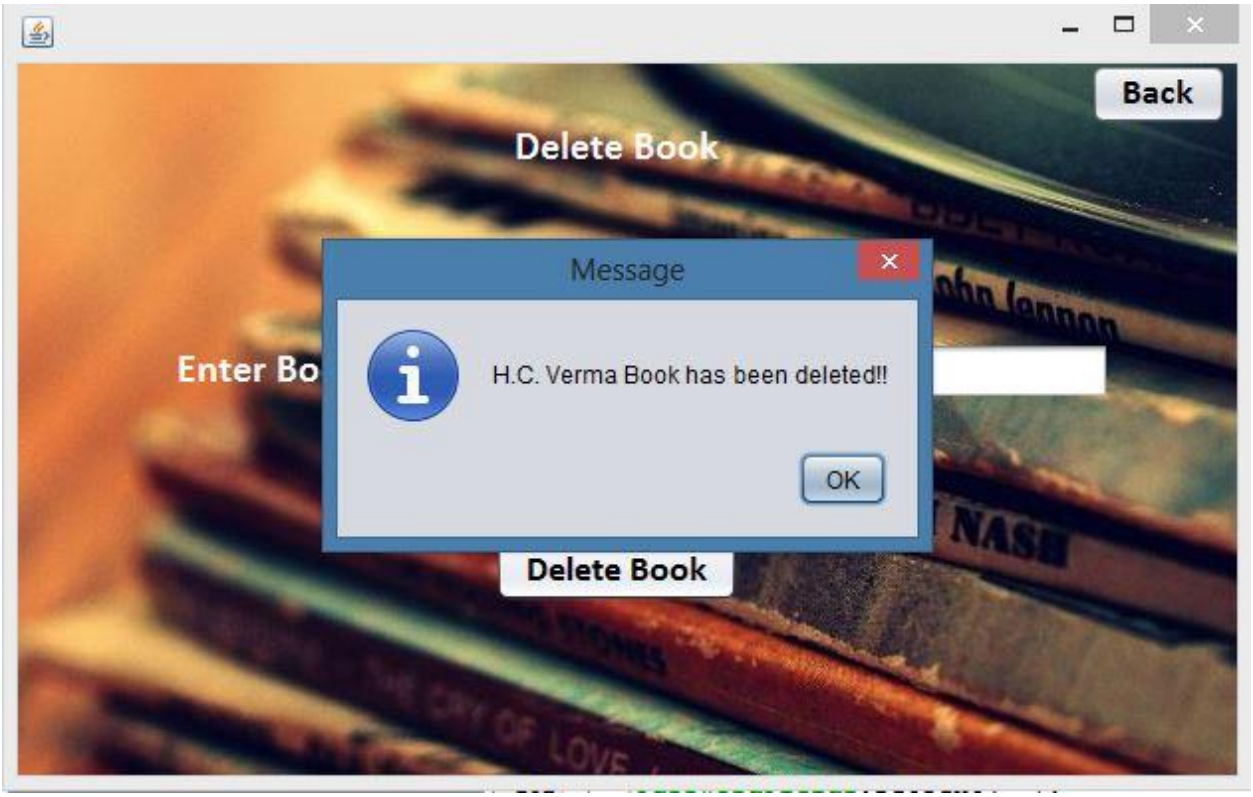


## CODE FOR DELETE BOOK BUTTON-

```
String Name=jTextField1.getText();
try
{
   Class.forName("java.sql.DriverManager");
   Connection con = (Connection)
   DriverManager.getConnection("jdbc:mysql://localhost:3306/vipul","root","apschool");
   Statement stmt = (Statement) con.createStatement();
   String query="Delete from book where Name='"+Name+"';";
   stmt.executeUpdate(query);
   JOptionPane.showMessageDialog(this,Name+" Book has been deleted!!");
   jButton2.setEnabled(false);
   jTextField1.setText("");
   }
catch(Exception e)
{
   JOptionPane.showMessageDialog(this,e.getMessage());
}
```

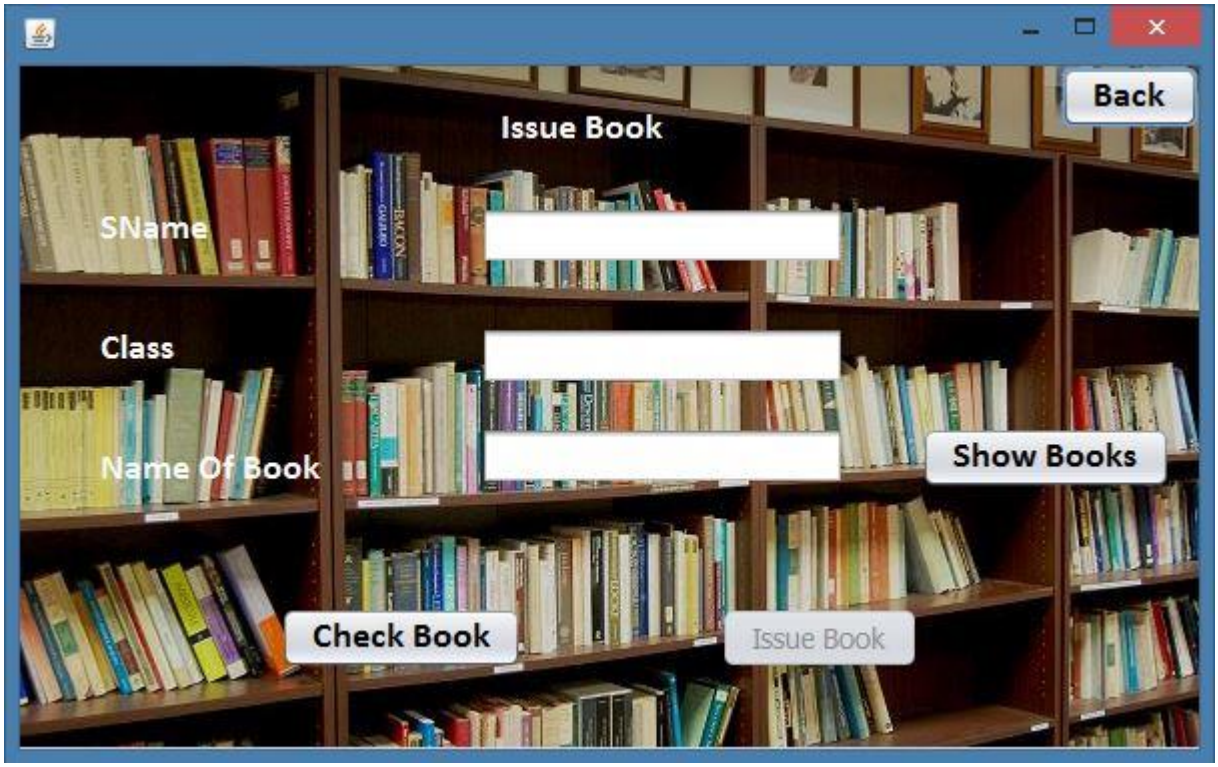
## CODE FOR THE BACK BUTTON-

```
new Main().setVisible(true);
this.setVisible(false);
```



# 5. ISSUING BOOKS

The Issue Book window is used to issue a book to a student. The Show Book button can be used to see the list of books available. Moreover, the Check Book button is used to check if the book is available or not. Finally, the Issue Book button can be used to issue the book. Also, the date of issuing is logged in the 'Student' table in the back end.




## CODE FOR SHOW BOOKS BUTTON-

```
new ShowBooks().setVisible(true);
```

## CODE FOR CHECK BOOK BUTTON-

```
String SName=jTextField1.getText();
String Name=jTextField2.getText();
String Standard=jTextField3.getText();
if (Name.isEmpty())
   JOptionPane.showMessageDialog(this,"Name not Entered.!");
else
{
   try
   {
```

```
      Class.forName("java.sql.DriverManager");
      Connection con =(Connection)
      DriverManager.getConnection("jdbc:mysql://localhost:3306/vipul","root","apschool");
      Statement stmt = (Statement) con.createStatement();
      String query="Select Name From book Where Name ='"+Name+"';";
      ResultSet rs=stmt.executeQuery(query);
      int Found=0;
      while (rs.next())
      {
         String name=rs.getString("Name");
         Found++;
         jButton2.setEnabled(true);
      }
      if (Found==0)
         JOptionPane.showMessageDialog(this,"Sorry!! No such book exists in records!!");
      else
         JOptionPane.showMessageDialog(this,"Book Exists! Please Issue now!");

   }
   catch (Exception e)
   {
      JOptionPane.showMessageDialog(this,e.getMessage());
   }
}
```

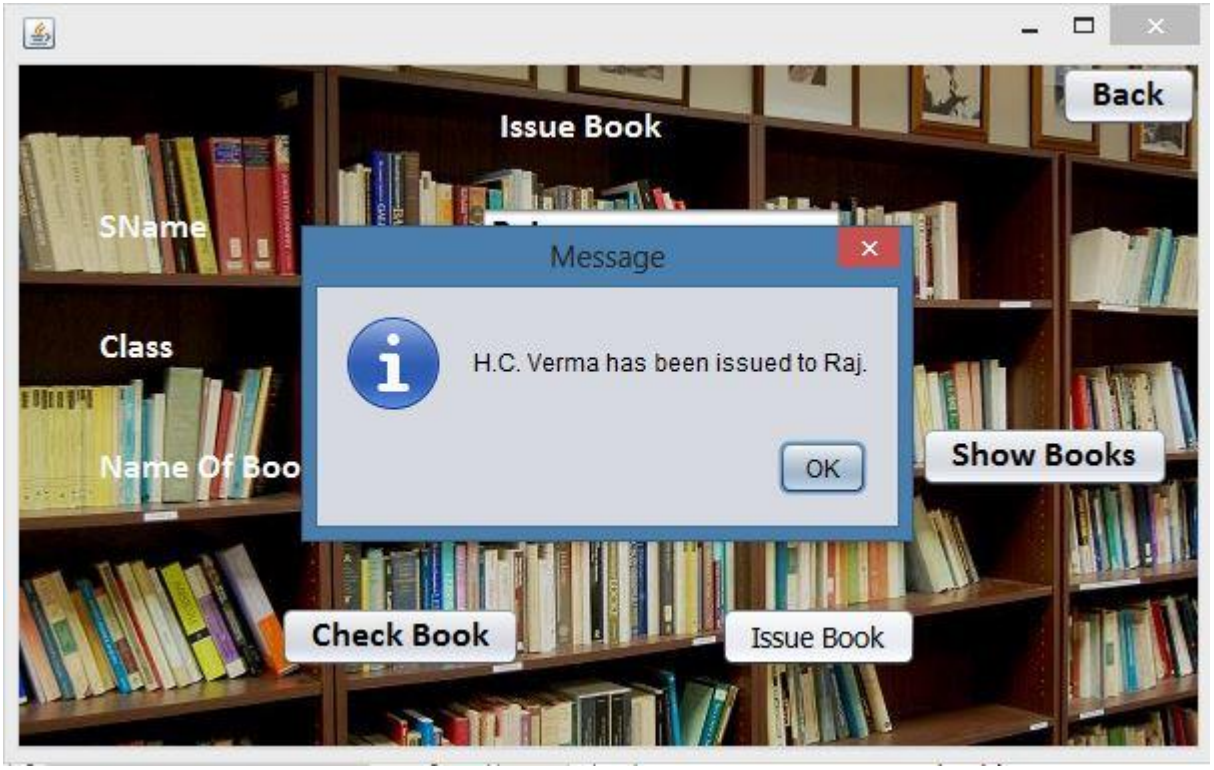
## CODE FOR ISSUE BOOK BUTTON-

```
String SName=jTextField1.getText();
String Name=jTextField2.getText();
String Standard=jTextField3.getText();
try
{
   Class.forName("java.sql.DriverManager");
   Connection con = (Connection)
   DriverManager.getConnection("jdbc:mysql://localhost:3306/vipul","root","apschool");
   Statement stmt = (Statement) con.createStatement();
   String query="Insert Into student values('"+SName+"','"+Name+"','"+Standard+"',CURDATE());";
   stmt.executeUpdate(query);
   JOptionPane.showMessageDialog(this,""+Name+" has been issued to "+SName+".");
   jButton2.setEnabled(false);
   jTextField1.setText("");
   jTextField2.setText("");
   jTextField3.setText("");

}
catch(Exception e)
      {
      JOptionPane.showMessageDialog(this,e.getMessage());
```

}

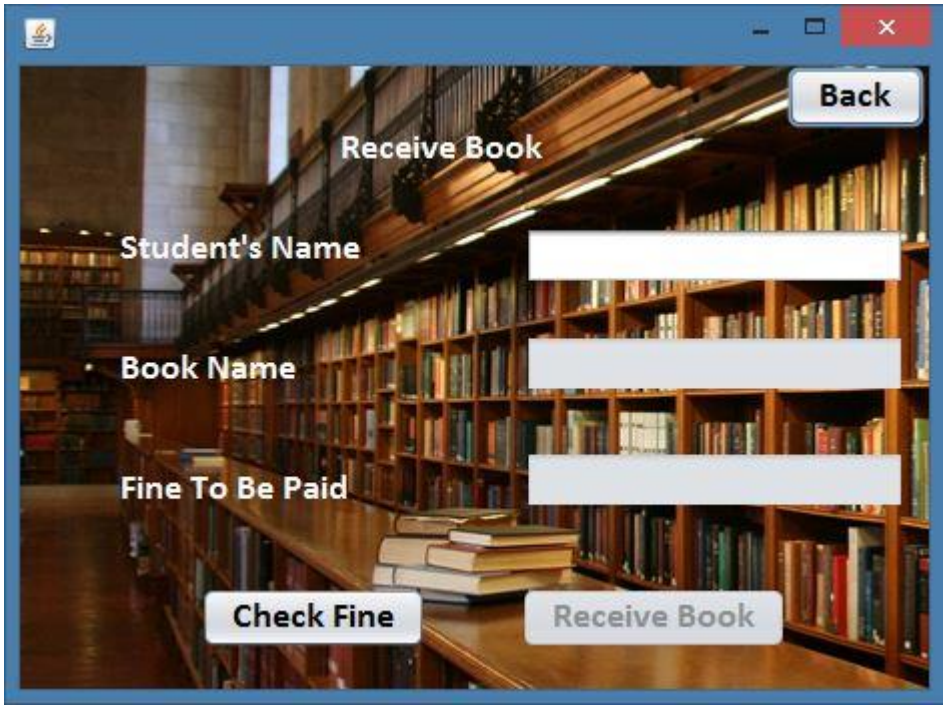
## CODE FOR BACK BUTTON-

```
new Main().setVisible(true);
this.setVisible(false);
```



# 6. RECEIVING BOOKS

The Receive Book Window can be used to receive the book returned by the student with their calculated fines. The student's name is entered and then the Check Fine button is clicked, the book name and the fine is displayed and then when the fine is paid, the librarian can click on the Receive Book Button which will remove the record of the student from the 'Student' table.




## CODE FOR CHECK FINE BUTTON-

```
String SName=jTextField1.getText();
try
{
   Class.forName("java.sql.DriverManager");
   Connection con = (Connection)
   DriverManager.getConnection("jdbc:mysql://localhost:3306/vipul","root","apschool");
   Statement stmt = (Statement) con.createStatement();
   String query=("SELECT Name,DateDiff(Curdate(),DateOfIssue) From Student where
SName='"+SName+"';");
   ResultSet rs=stmt.executeQuery(query);
   if (rs.next())
```

```
   {
   String days=rs.getString("DateDiff(Curdate(),DateOfIssue)");
   String Name=rs.getString("Name");
   jTextField2.setText(Name);
   int fine=(Integer.parseInt(days)-7);
   if (fine>=0)
   jTextField3.setText(Integer.toString(fine));
   else
   jTextField3.setText("0");
   jButton2.setEnabled(true);
   }
   else
      JOptionPane.showMessageDialog(this,"Sorry! No Book issued to "+SName);
}
catch(Exception e)
{
   JOptionPane.showMessageDialog(this,e.getMessage());
}
```


## CODE FOR RECEIVE BOOK BUTTON-

```
String SName=jTextField1.getText();
try
{
   Class.forName("java.sql.DriverManager");
   Connection con = (Connection)
```

```
   DriverManager.getConnection("jdbc:mysql://localhost:3306/vipul","root","apschool");
   Statement stmt = (Statement) con.createStatement();
   String query="Delete from STUDENT where Sname='"+SName+"';";
   stmt.executeUpdate(query);
   JOptionPane.showMessageDialog(this,SName+"'s Book has been received!!");
   jButton2.setEnabled(false);
   jTextField1.setText("");
   jTextField2.setText("");
   jTextField3.setText("");
   }
catch(Exception e)
{
   JOptionPane.showMessageDialog(this,e.getMessage());
}
```

## CODE FOR BACK BUTTON-

```
new Main().setVisible(true);
this.setVisible(false);
```

# 7. VIEW BOOKS

The View Books window is to view the list of books in the library with their serial numbers.


## CODE FOR VIEW BOOKS BUTTON-

```
DefaultTableModel model = (DefaultTableModel)
jTable1.getModel();
try
{
   Class.forName("java.sql.DriverManager");
   Connection con=(Connection)
   DriverManager.getConnection("jdbc:mysql://localhost:3306/vipul","root","apschool");
   Statement stmt=(Statement) con.createStatement();
   String Tname;
   Tname="Book";
   String query="SELECT *FROM "+Tname+";";
   ResultSet rs=stmt.executeQuery(query);
   while (rs.next())
   {
```

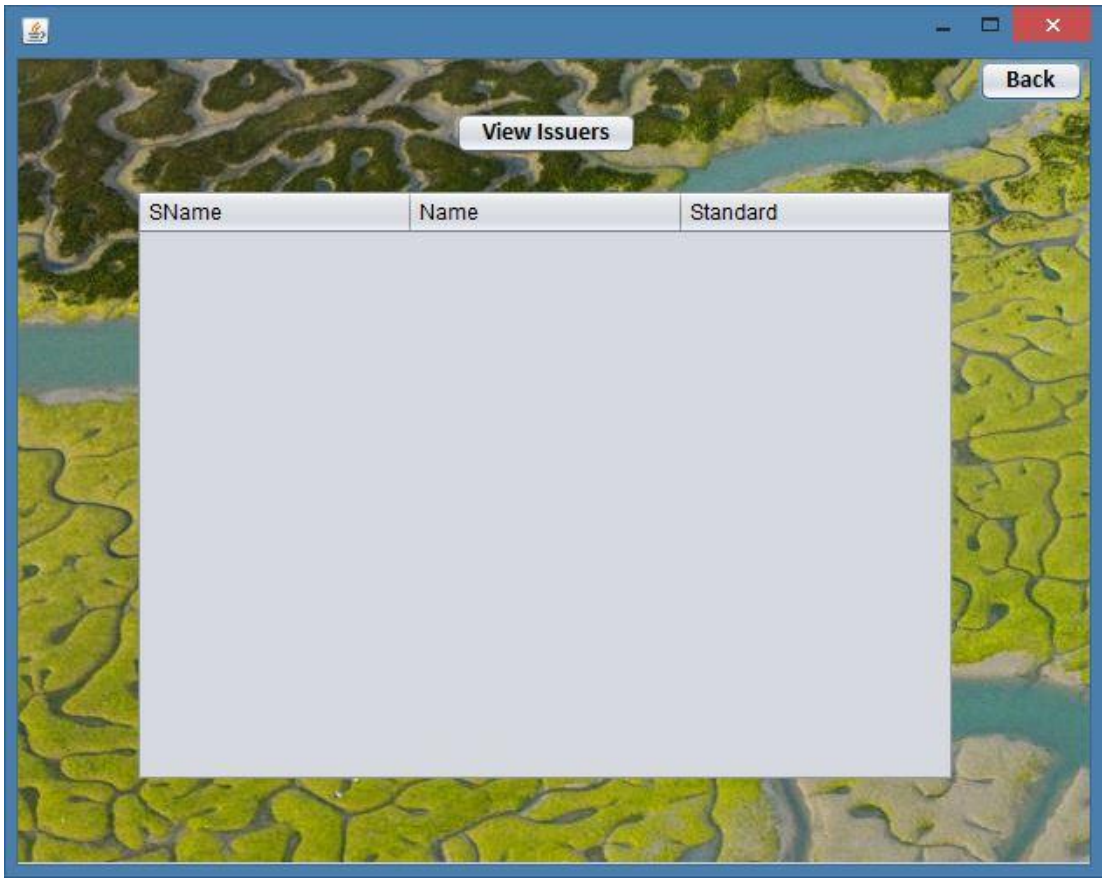
```
        String Name=rs.getString("Name");
        String Serial=rs.getString("Serial");
        model.addRow(new Object[] {Name,Serial});

    }

}
catch (Exception e)
{
JOptionPane.showMessageDialog(this,e.getMessage());
}
```

## CODE FOR THE BACK BUTTON-

```
new Main().setVisible(true);
this.setVisible(false);
```

# 8. VIEW ISSUERS

The View Issuers Window is used to view the list of all the students who have borrowed a book from the library with the book name and their class.




## CODE FOR THE VIEW ISSUERS BUTTON-

```
DefaultTableModel model = (DefaultTableModel)
jTable1.getModel();
try
{
   Class.forName("java.sql.DriverManager");
   Connection con=(Connection)
   DriverManager.getConnection("jdbc:mysql://localhost:3306/vipul","root","apschool");
   Statement stmt=(Statement) con.createStatement();
   String Tname;
   Tname="Student";
   String query="SELECT *FROM "+Tname+";";
```

```
   ResultSet rs=stmt.executeQuery(query);
   while (rs.next())
   {
      String SName=rs.getString("SName");
      String Name=rs.getString("Name");
      String Standard=rs.getString("Standard");
      model.addRow(new Object[] {SName,Name,Standard});
   }
}
catch (Exception e)
{
JOptionPane.showMessageDialog(this,e.getMessage());
}
```

### CODE FOR THE BACK BUTTON-

```
new Main().setVisible(true);
this.setVisible(false);
```

## 9. TABLES USED

Two table have been used in this project- 'Book' table and 'Student' table.

### STRUCTURE OF BOOK TABLE-

```
mysql> desc book;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| Name   | varchar(40) | YES  |     | NULL    |       |
| Serial | int(5)      | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
2 rows in set (0.14 sec)
```

STRUCTURE OF STUDENT TABLE-

```
mysql> desc student;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| SName       | varchar(40) | YES  |     | NULL    |       |
| Name        | varchar(40) | YES  |     | NULL    |       |
| Standard    | int(2)      | YES  |     | NULL    |       |
| DateOfIssue | date        | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

# 10. CONCLUSION

This was an effort to develop a simple Library Management System which may be useful in a Library to insert, store, handle and retrieve information about books and students who have issued a book.

# 11. BIBLIOGRAPHY

There are many sources from which some elements of the code and ideas have been inspired from. Some of them are listed below.


- CBSE Class 12 Informatics Practices
- www.scribd.com
- www.netbeans.org
- www.mysql.com
- www.stackoverflow.com