ONLINE LIBRARY MANGEMENT SYSTEM

An

Object-Oriented Programming through Java Course Project Report

in partial fulfilment of the degree

Bachelor of Technology Computer Science & Engineering

in

Under the
Guidance
of
MR.N.MAHENDER SIR

By

T.Rohith	2103A51573
MD.Waheed	2103A51475

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the **Object Oriented Programming through Java - Course Project** Report entitled **" Online Libraray management system "** is a record of bonafide work carried out by the student T.Rohith,MD.Waheed bearing Roll No(s) 2103A51573,2103A51475 during the academic year 2023-24 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the SR University, Ananthasagar,Warangal.

Lab In-charge **Head of the Department**

ORGANIZATION OF REPORT

Table of Contents:

ABSTRACT:

The “Online Library Management System” is a web-based application developed using Java. This system aims to digitize the process of cataloging books and managing student records in a library.

The system provides an interface for the staff to manage books, including adding new books, updating book information, and tracking book availability. It also allows for efficient management of student records, including their issued books, due dates, and fines.

In conclusion, the “Online Library Management System” using Java is a comprehensive solution that simplifies library management, making it more efficient and user-friendly.

1.OBJECTIVE OF THE PROJECT:

The objective of the Online Library Management System using Java is to design and implement a digital platform that allows users to manage and access library resources efficiently. The system aims to:

1. **Simplify Access**: Enable users to search, reserve, and borrow books online, reducing the need for physical visits to the library.
2. **Improve Efficiency**: Automate administrative tasks such as tracking borrowed books, sending reminders for due dates, and managing book inventory.
3. **Enhance User Experience**: Provide a user-friendly interface that allows users to easily navigate and use the system.
4. **Ensure Security**: Implement secure login and data protection mechanisms to safeguard user information.
5. **Promote Learning**: Encourage reading and learning by providing easy access to a wide range of books and resources.

The system will be developed using Java, leveraging its robustness, security features, and platformindependent nature to ensure a reliable and efficient library management system.

2. DEFINITIONS OF THE ELEMENTS USED IN THE PROJECT:

The definitions of some key elements used in the Online Library Management System project:

1. **User**: A person who uses the library management system. Users can be categorized into

   different roles such as administrators, librarians, and members.

2. **Book**: An object in the library that can be borrowed by a member. Each book has attributes

   like title, author, publication year, genre, and ISBN number.

3. **Admin** (short for Administrator) is a user role with the highest level of

   access and control over the system. The Admin is responsible for

   managing the overall.

3. **Borrow**: An action performed by a member to take a book from the

   library for a certain period.

4. **Return**: An action performed by a member to give back a borrowed book to the library.

5. **Reservation**: An action performed by a member to reserve a book that

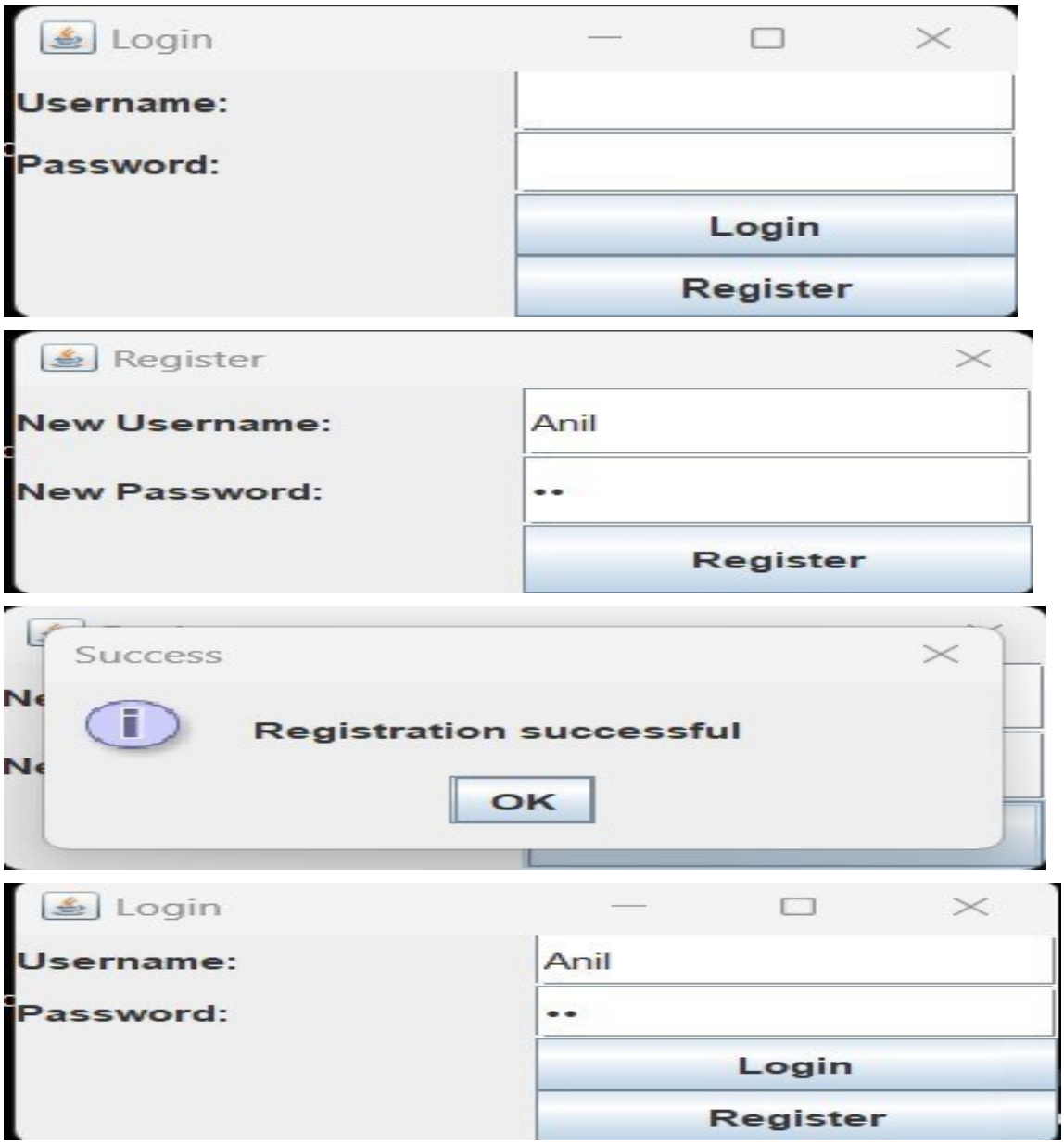
   is currently unavailable for borrowing.

6. **Client**: A device used by a user to interact with the online library management system.

These elements form the core of the Online Library Management System and

interact with each other to provide the desired functionalities

3.DESIGNING THE SCREEN:



When we click on Login

Redirected to next page

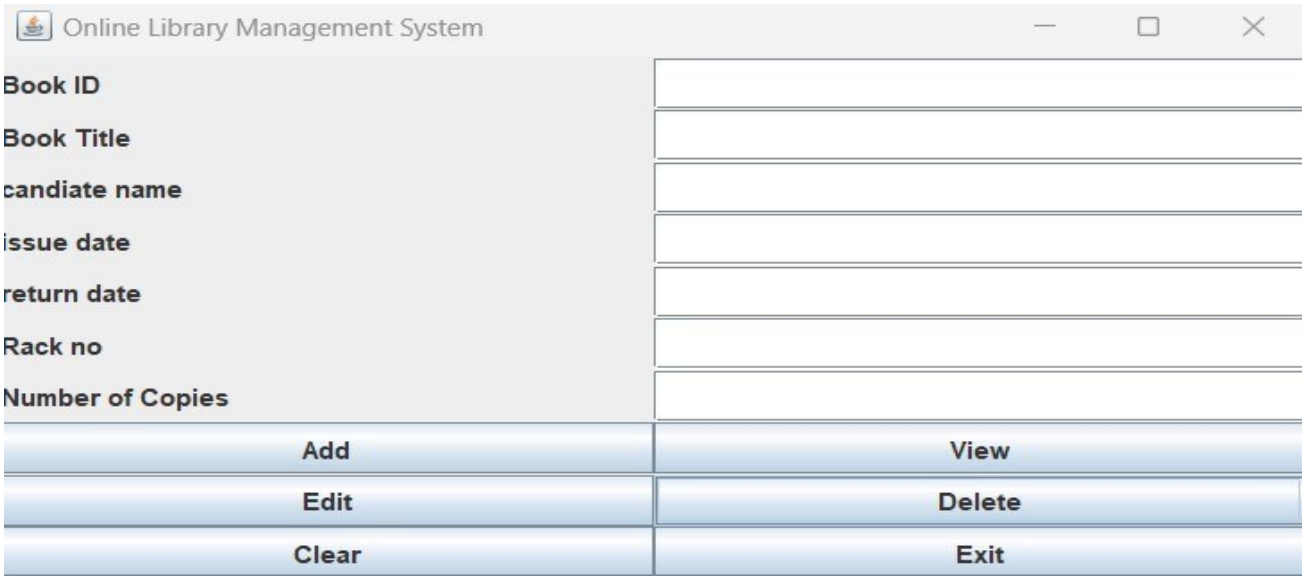


Figure:1

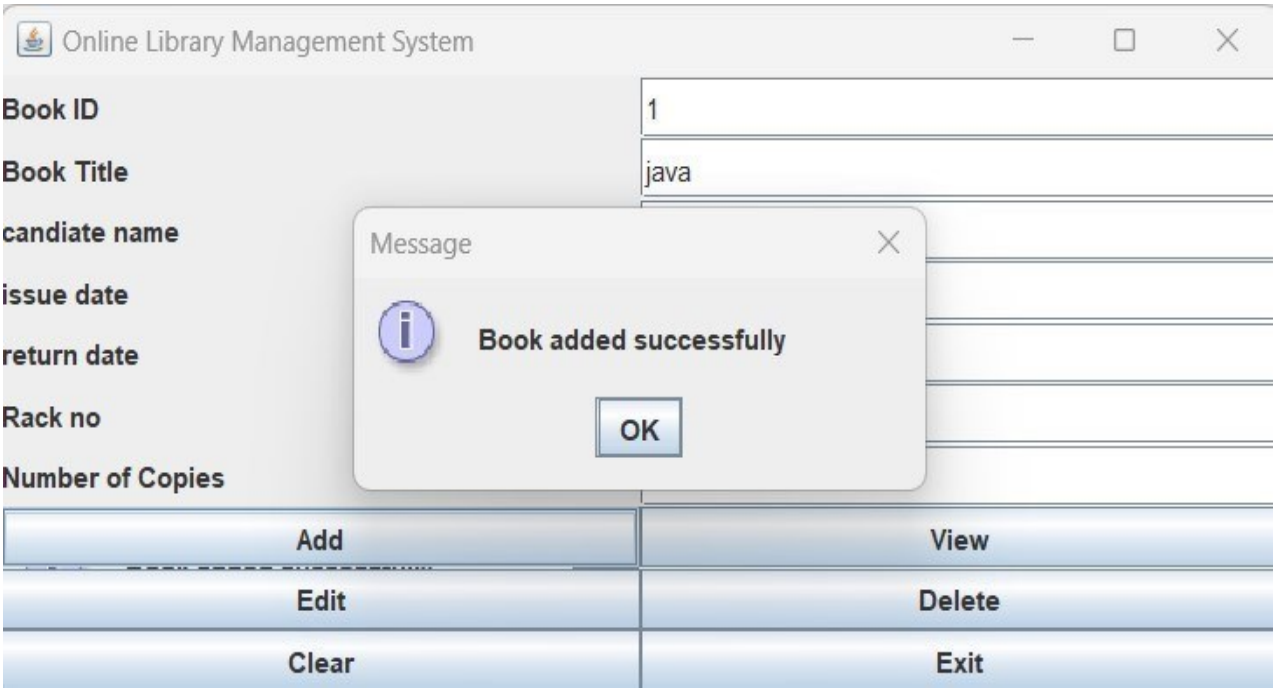


Figure:2

4. IMPLEMENTATION OF CODE:

The code of the Online Library Management System is as follows :

```
import javax.swing.*;
import java.awt.*;
import
java.awt.event.*;
import
java.util.ArrayList;
import
java.util.HashMap;

public class LibraryManagement extends JFrame implements
ActionListener {     private JLabel label1, label2, label3, label4, label5,
label6, label7;     private JTextField textField1, textField2, textField3,
textField4, textField5, textField6, textField7;     private JButton addButton,
viewButton, editButton, deleteButton, clearButton, exitButton;     private
JPanel panel;     private
HashMap<String, String> userCredentials = new HashMap<>();     private
ArrayList<String[]> books = new ArrayList<>();
```

```
   // Add login dialog
components     private
JTextField usernameField;
private JPasswordField
passwordField;

   public LibraryManagement() {
      // Initialize user credentials (for demonstration purposes)
userCredentials.put("user1", "password1");
userCredentials.put("user2", "password2");

      // Create login dialog
      JFrame loginFrame = new JFrame("Login");
usernameField = new JTextField(10);
passwordField = new
JPasswordField(10);
      JButton loginButton = new JButton("Login");
      JButton registerUserButton = new JButton("Register");

      JPanel loginPanel = new JPanel(new
GridLayout(4, 2));        loginPanel.add(new
JLabel("Username:"));
loginPanel.add(usernameField);
```

```
loginPanel.add(new JLabel("Password:"));
loginPanel.add(passwordField);
loginPanel.add(new JLabel(""));
loginPanel.add(loginButton);
loginPanel.add(new
JLabel(""));        loginPanel.add(registerUserButton);

      loginButton.addActionListener(new ActionListener() {
         @Override           public void
actionPerformed(ActionEvent e) {              // Validate the
username and password              String username =
usernameField.getText();              char[] password =
passwordField.getPassword();              boolean
authenticated = authenticate(username, new
String(password));

            if (authenticated) {                loginFrame.dispose(); // Close the
login dialog
initializeLibraryManagement(); // Initialize the library management system
            } else {
               JOptionPane.showMessageDialog(loginFrame, "Invalid
            username or password", "Login Error",
            JOptionPane.ERROR_MESSAGE);
```

```
            }

            // Clear password after authentication attempt
passwordField.setText("");
        }
    });

    registerUserButton.addActionListener(new ActionListener() {
        @Override            public void
actionPerformed(ActionEvent e) {
initializeRegistrationDialog(loginFrame);
        }
    });

    loginFrame.add(loginPanel);        loginFrame.setSize(300, 150);
loginFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
loginFrame.setVisible(true);
  }

  private boolean authenticate(String username, String
password) {        if
(userCredentials.containsKey(username)) {
```

```
return
userCredentials.get(username).equals(password);
        }
        return false;
    }

    private void initializeRegistrationDialog(JFrame loginFrame) {
        // Create registration dialog
        JDialog registrationDialog = new JDialog(loginFrame, "Register", true);
        JTextField newUsernameField = new JTextField(10);
        JPasswordField newPasswordField = new JPasswordField(10);
JButton registerButton = new JButton("Register");

        JPanel registrationPanel = new JPanel(new
GridLayout(3, 2));        registrationPanel.add(new
JLabel("New Username:"));
registrationPanel.add(newUsernameField);
registrationPanel.add(new JLabel("New Password:"));
registrationPanel.add(newPasswordField);
registrationPanel.add(new JLabel(""));
registrationPanel.add(registerButton);

        registerButton.addActionListener(new ActionListener() {
```

```
        @Override            public void
actionPerformed(ActionEvent e) {
            // Register the new user
            String newUsername =
newUsernameField.getText();                char[]
newPassword = newPasswordField.getPassword();

            if (!newUsername.isEmpty() && newPassword.length > 0) {
userCredentials.put(newUsername, new String(newPassword));
                JOptionPane.showMessageDialog(registrationDialog,
"Registration successful",
            "Success", JOptionPane.INFORMATION_MESSAGE);
                registrationDialog.dispose(); // Close the registration dialog
            } else {
                JOptionPane.showMessageDialog(registrationDialog, "Invalid
            username or password", "Registration Error",
            JOptionPane.ERROR_MESSAGE);
            }

            // Clear password after registration attempt
newPasswordField.setText("");
        }
    });
```

```
        registrationDialog.add(registrationPanel);
registrationDialog.setSize(300, 150);
registrationDialog.setDefaultCloseOperation(JDialog.DISPOS
E_ON_CLOSE);        registrationDialog.setVisible(true);
    }

    private void initializeLibraryManagement() {
setTitle("Online Library Management System");
setSize(600, 300);
setDefaultCloseOperation(EXIT_ON_CLOSE);

        label1 = new JLabel("Book ID");        label2
= new JLabel("Book Title");        label3 = new
JLabel("Author");        label4 = new
JLabel("Publisher");        label5 = new
JLabel("Year of Publication");        label6 = new
JLabel("Rack no");        label7 = new
JLabel("Number of Copies");

        textField1 = new JTextField(10);
textField2 = new JTextField(20);
textField3
= new JTextField(20);        textField4 = new
```

```
JTextField(20);        textField5 = new
JTextField(10);        textField6 = new
JTextField(20);        textField7 = new
JTextField(10);

        addButton = new JButton("Add");
viewButton = new JButton("View");
editButton = new JButton("Edit");
deleteButton = new JButton("Delete");
clearButton = new JButton("Clear");
exitButton = new JButton("Exit");

        addButton.addActionListener(this);
viewButton.addActionListener(this);
editButton.addActionListener(this);
deleteButton.addActionListener(this);
clearButton.addActionListener(this);
exitButton.addActionListener(this);

        panel    =    new    JPanel(new
GridLayout(10,                    2));
panel.add(label1);
panel.add(textField1);
```

```
panel.add(label2);
panel.add(textField2);
panel.add(label3);
panel.add(textField3);
panel.add(label4);
panel.add(textField4);
panel.add(label5);
panel.add(textField5);
panel.add(label6);
panel.add(textField6);
panel.add(label7);
panel.add(textField7);
panel.add(addButton);
panel.add(viewButton);
panel.add(editButton);
panel.add(deleteButton);
panel.add(clearButton);
panel.add(exitButton);

      add(panel);
setVisible(true);
   }
```

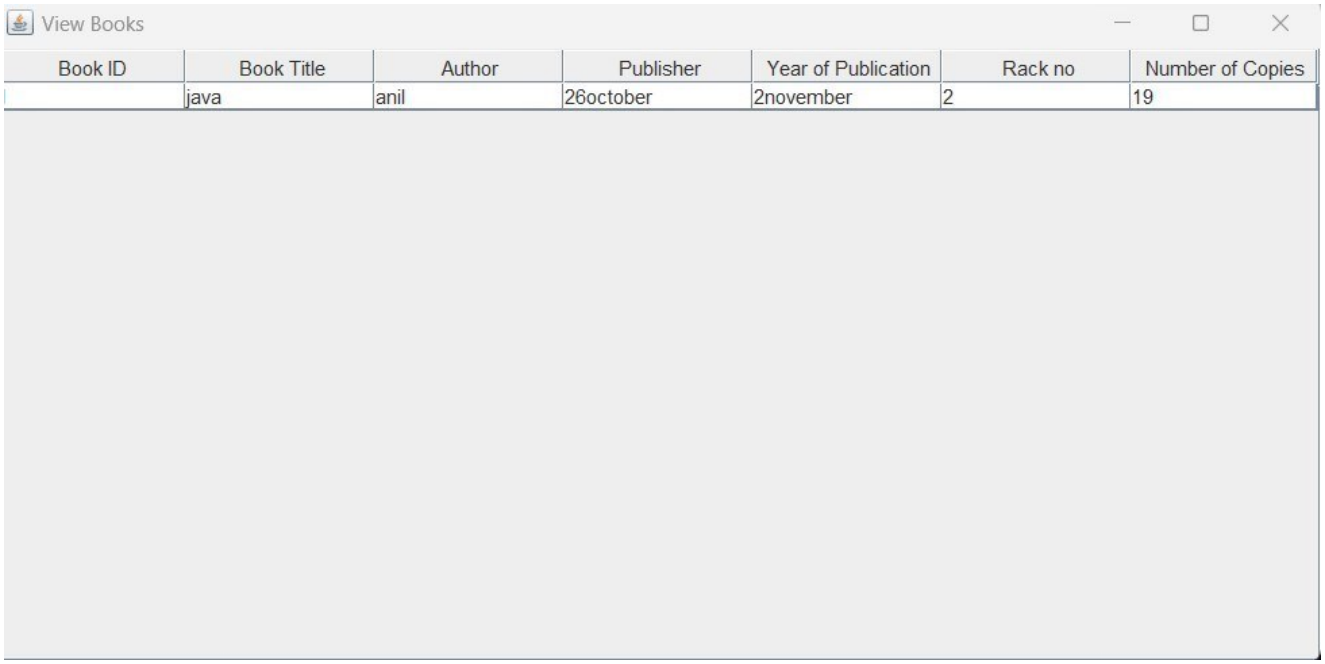
```
    public void
actionPerformed(ActionEvent e) {
if (e.getSource() == addButton) {
String[] book = new String[7];
book[0] = textField1.getText();
book[1] = textField2.getText();
book[2] = textField3.getText();
book[3] = textField4.getText();
book[4] = textField5.getText();
book[5] = textField6.getText();
book[6] = textField7.getText();
books.add(book);
JOptionPane.showMessageDialog(this,
"Book added successfully");

            clearFields();
        } else if (e.getSource() == viewButton) {
            String[] columns = {"Book ID", "Book Title", "Author", "Publisher",
              "Year of Publication", "Rack no", "Number of Copies"};
            Object[][] data = new
Object[books.size()][7];            for (int i
= 0; i < books.size(); i++) {
```

```
data[i][0] = books.get(i)[0];
data[i][1] = books.get(i)[1];
data[i][2] = books.get(i)[2];
data[i][3] = books.get(i)[3];
data[i][4] = books.get(i)[4];
data[i][5] = books.get(i)[5];
data[i][6] = books.get(i)[6];
        }
        JTable table = new JTable(data, columns);
        JScrollPane scrollPane = new JScrollPane(table);
JFrame frame = new JFrame("View Books");
frame.add(scrollPane);          frame.setSize(800, 400);
frame.setVisible(true);
      } else if (e.getSource() == editButton) {
        String bookID = JOptionPane.showInputDialog(this, "Enter book ID
to edit:");
        for (int i = 0; i <
books.size(); i++) {                if
(books.get(i)[0].equals(bookID))
{                String[] book = new
String[7];                  book[0] =
bookID;                     book[1] =
textField2.getText();
```

```
book[2] = textField3.getText();

book[3] = textField4.getText();

book[4] = textField5.getText();

book[5] = textField6.getText();

book[6] = textField7.getText();

books.set(i, book);

            JOptionPane.showMessageDialog(this, "Book edited
successfully");

            clearFields();

return;

          }

        }

        JOptionPane.showMessageDialog(this, "Book not found");

    } else if (e.getSource() == deleteButton) {

        String bookID = JOptionPane.showInputDialog(this, "Enter book ID
to delete:");

        for (int i = 0; i < books.size();

i++) {            if

(books.get(i)[0].equals(bookID)) {

books.remove(i);

            JOptionPane.showMessageDialog(this, "Book deleted
successfully");

            clearFields();

return;
```

```
            }
        }
        JOptionPane.showMessageDialog(this, "Book not found");
    } else if (e.getSource() == clearButton) {
clearFields();
    } else if (e.getSource() == exitButton) {
        System.exit(0);
    }
  }

  private void clearFields() {
textField1.setText("");
textField2.setText("");
textField3.setText("");
textField4.setText("");
textField5.setText("");
textField6.setText("");
textField7.setText("");
  }

  public static void main(String[] args) {
new LibraryManagement();
  } }
```

5.RESULT SCREENSHORT:



6.CONCLUSION:

The Online Library Management System project, developed using Java, aims to digitize and streamline the operations of a traditional library. By providing functionalities such as book borrowing, returning, reservation, and fine imposition, it enhances the user experience and increases the efficiency of library management. The system also ensures data security and provides robust features for administrators, including user and book management, system configuration, report generation, and data backup and recovery.

In conclusion, this project leverages modern technology to transform the way libraries function, making them more accessible and user-friendly. It serves as a testament to the potential of digital transformation in traditional sectors.