RAJALAKSHMI INSTITUTE OF TECHNOLOGY

BELIEVE IN THE POSSIBILITIES

AN AUTONOMOUS INSTITUTION

ANNA UNIVERSITY

PROGRESS THROUGH KNOWLEDGE

Book Return Reminder System

MINI PROJECT REPORT

Submitted by

Jonnalagadda Aakash (2117240030057)

in partial fulfillment for the awar of the degree of

BACHELOR OF ENGINEERING

IN

CSE(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

RAJALAKSHMI INSTITUTE OF TECHNOLOGY

KUTHAMBAKKAM, CHENNAI - 600 124

ANNA UNIVERSITY: CHENNAI 600 025

NOV/DEC - 2025

ANNA UNIVERSITY: CHENNAI - 600 025

BONAFIDE CERTIFICATE

Certified that this Mini Project report **Book Return Reminder System** is the Bonafide work **Jonnalagadda Aakash (2117240030099)** of who carried out the project work under my supervision.

SIGNATURE

HEAD OF THE DEPARTMENT

Dr. N. KANAGAVALLI, Ph.D

Assistant Professor,

CSE(AI & ML)

Rajalakshmi Institute of Technology,

Chennai – 600 124.

SIGNATURE

SUPERVISOR

Mrs. A LIZY. M.E

Assistant Professor,

CSE(AI & ML)

Rajalakshmi Institute of Technology,

Chennai – 600 124

Submitted for the Mini Project Viva-Voice held on ……

INTERNAL EXAMINER

EXTERNAL EXAMINER

PROJECT DESCRIPTION

The **Book Return Reminder System** is a desktop-based Java application designed to automate and streamline the process of tracking borrowed books and managing timely returns in libraries. Traditional manual record-keeping often leads to misplaced data, late returns, and human errors. This project introduces a digital solution that ensures accuracy, reduces workload, and provides timely reminder notifications to users before the book's due date.

The main objective of the system is to automatically monitor due dates and alert users in advance, minimizing delays and improving efficiency. It serves as a digital assistant for librarians and students by providing quick access to borrowing records and reminders, thereby promoting a paperless and efficient library environment. The system adopts a structured database-driven approach where each borrowing transaction is recorded and reminders are generated dynamically based on the number of days left before the return date.

The key features of this system include automated record management, real-time data access, smart reminder calculation using SQL date functions, and a user-friendly graphical interface built with Java Swing. It also includes data validation and error handling to prevent incorrect or duplicate entries. The system is designed to be scalable and can be extended to support multi-user access, automatic notifications, or integration with larger digital library systems.

Technically, the project uses **Java Swing** for the graphical interface, **MySQL** for backend data storage, and **JDBC** for database connectivity. It follows **Object-Oriented Programming (OOP)** principles to ensure modularity, reusability, and maintainability.

Overall, the Book Return Reminder System provides a practical solution for library management by combining efficient programming logic with database integration. It simplifies administrative tasks, enhances user convenience, and demonstrates the application of software engineering principles in real-world scenarios.

PUBLIC TRANSPORT ROUTE PLANNER

PO–PSO Mapping Table :

PO (Program Outcomes)	**PSO (Program Specific Outcomes)**	**PSO (Program Specific Outcomes)**
PO1 – Engineering Knowledge	PSO1 – Apply programming concepts to real- time problem solving	High
PO2 – Problem Analysis	PSO2 – Design & develop applications using modern tools	Medium
PO3 – Design/Development of Solutions	PSO1	High
PO5 – Modern Tool Usage	PSO2	High
PO5 – Modern Tool Usage	PSO3 – Develop eco- friendly and socially useful technological solutions	Medium
PO8 – Ethics	PSO3	Low

PO Justification :

PO No.	**Justification**
PO1 – Engineering Knowledge	Students apply programming concepts, Java Swing, and JDBC knowledge to build the system.
PO2 – Problem Analysis	Users analyze travel problems and determine the optimal path based on available routes.
PO3 – Design/Development of Solutions	Students design database schema, GUI layout, and logical flow to achieve the objective.
PO5 – Modern Tool Usage	Utilizes Java IDE, MySQL database, JDBC connector – modern development tools.
PO7 – Environment & Sustainability	Encourages the use of public transport, reducing pollution and carbon footprint.
PO8 – Ethics	Data is stored securely and used only for functional purposes.

PSO Justification :

PSO No.	PSO No.
PSO1	Enhances ability to analyze real-world problems and convert them into a software application.
PSO2	Demonstrates proficiency in Java, JDBC, and database integration.
PSO3	Promotes development of applications beneficial for society and environmental well-being.

Introduction :

In today's digital era, libraries require automation to handle large volumes of borrowed and returned books efficiently. The **Book Return Reminder System** addresses this need through a Java-based GUI application that tracks borrow dates and calculates reminders automatically.

The system allows librarians or users to record borrowing details, set reminder days, and check due status in real-time. It integrates **Java Swing** for the interface and **MySQL** for backend storage, connected via **JDBC**.

The project emphasizes data validation, date management, and user feedback mechanisms. Through this automation, it reduces manual errors, enhances organization, and ensures timely book returns. This mini project demonstrates key software engineering practices like modularity, encapsulation, and real-world data processing.

Project Distribution :

The project consists of two main modules:

Frontend (Java Swing GUI)

Handles user input, record creation, and reminder display.

Provides forms for entering student name, book name, and date details.

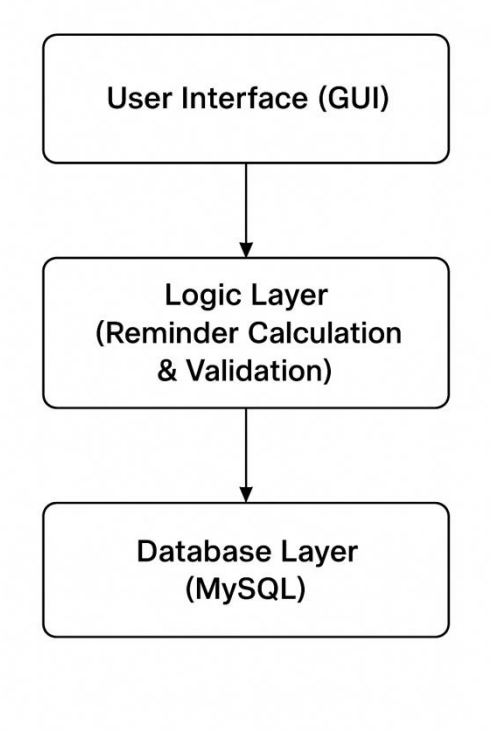
Backend (MySQL with JDBC)

Stores all borrowing and reminder records.

Executes SQL operations like insert, retrieve, and calculate date differences.

This layered structure ensures clean separation between the interface and logic, enhancing scalability and maintenance.

System Architecture :



Coding :

```
package aakash;

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

public class BookReminderSystem extends JFrame {

    JTextField tStudent, tBook, tBorrow, tReturn, tReminder;
    JTextArea displayArea;
    JButton btnAdd, btnCheck;

    static final String URL = "jdbc:mysql://localhost:3306/library_reminder";
    static final String USER = "root";
```

```
static final String PASS = "Saicharan@06";

public BookReminderSystem() {
    setTitle("Book Return Reminder System");
    setSize(400, 500);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new FlowLayout());

    add(new JLabel("Student Name:"));
    tStudent = new JTextField(20);
    add(tStudent);

    add(new JLabel("Book Name:"));
    tBook = new JTextField(20);
    add(tBook);

    add(new JLabel("Borrow Date (YYYY-MM-DD):"));
    tBorrow = new JTextField(20);
    add(tBorrow);

    add(new JLabel("Return Date (YYYY-MM-DD):"));
    tReturn = new JTextField(20);
    add(tReturn);

    add(new JLabel("Reminder Days:"));
    tReminder = new JTextField(20);
    add(tReminder);

    btnAdd = new JButton("Add Record");
    btnCheck = new JButton("Check Reminder");
```

```
        add(btnAdd);
        add(btnCheck);

        displayArea = new JTextArea(10, 30);
        displayArea.setEditable(false);
        add(new JScrollPane(displayArea));

        btnAdd.addActionListener(e -> addRecord());
        btnCheck.addActionListener(e -> checkReminders());

        setVisible(true);
    }

    private Connection connect() throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        return DriverManager.getConnection(URL, USER, PASS);
    }

    private void addRecord() {
        try (Connection con = connect()) {
            String student = tStudent.getText();
            String book = tBook.getText();
            String bDate = tBorrow.getText();
            String rDate = tReturn.getText();
            int reminder = Integer.parseInt(tReminder.getText());

            String sql = "INSERT INTO borrow(student_name, book_name, borrow_date,
return_date, reminder_days) VALUES (?,?,?,?,?)";
            PreparedStatement ps = con.prepareStatement(sql);
```

```
            ps.setString(1, student);
            ps.setString(2, book);
            ps.setDate(3, Date.valueOf(bDate));
            ps.setDate(4, Date.valueOf(rDate));
            ps.setInt(5, reminder);
            ps.executeUpdate();

            JOptionPane.showMessageDialog(this, "✅ Record Added Successfully!");
        } catch (IllegalArgumentException ex) {
            JOptionPane.showMessageDialog(this, "❌ Date must be YYYY-MM-DD");
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(this, "❌ Error: " + ex.getMessage());
        }
    }

    private void checkReminders() {
        try (Connection con = connect()) {
            String sql = "SELECT student_name, book_name, return_date, reminder_days, " +
                         "DATEDIFF(return_date, CURDATE()) AS days_left FROM borrow";
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery(sql);
            displayArea.setText("Reminder Status:\n\n");

            while (rs.next()) {
                String name = rs.getString("student_name");
                String book = rs.getString("book_name");
                int daysLeft = rs.getInt("days_left");
                int reminder = rs.getInt("reminder_days");

                displayArea.append("Student: " + name + "\n");
```

```
            displayArea.append("Book: " + book + "\n");
            displayArea.append("Days Left: " + daysLeft + "\n");

            if (daysLeft <= reminder && daysLeft >= 0)
               displayArea.append("⚠ Reminder: Return soon!\n");
            else if (daysLeft < 0)
               displayArea.append("❌ Overdue!\n");
            else
               displayArea.append("✅ No reminder needed\n");

            displayArea.append("-------------------------\n");
         }
      } catch (Exception ex) {
         JOptionPane.showMessageDialog(this, "❌ Error: " + ex.getMessage());
      }
   }

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

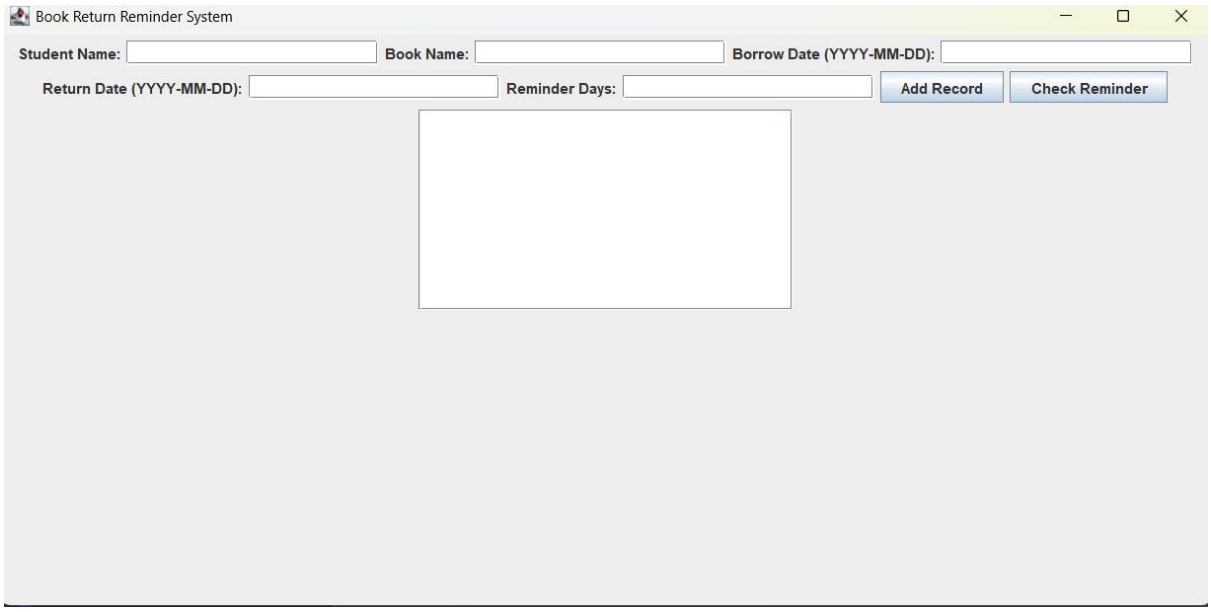
GitHub Link for Full Code :

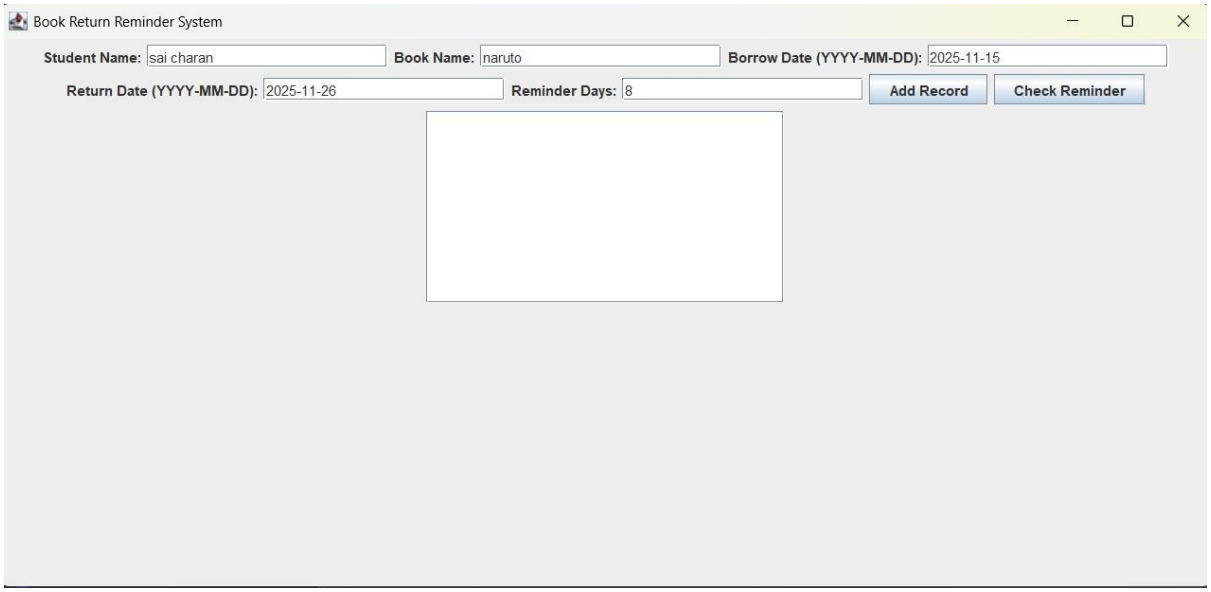
Frontend – https://github.com/aakashjonnalagadda/Book-return-reminder-system/blob/main/book-return.java

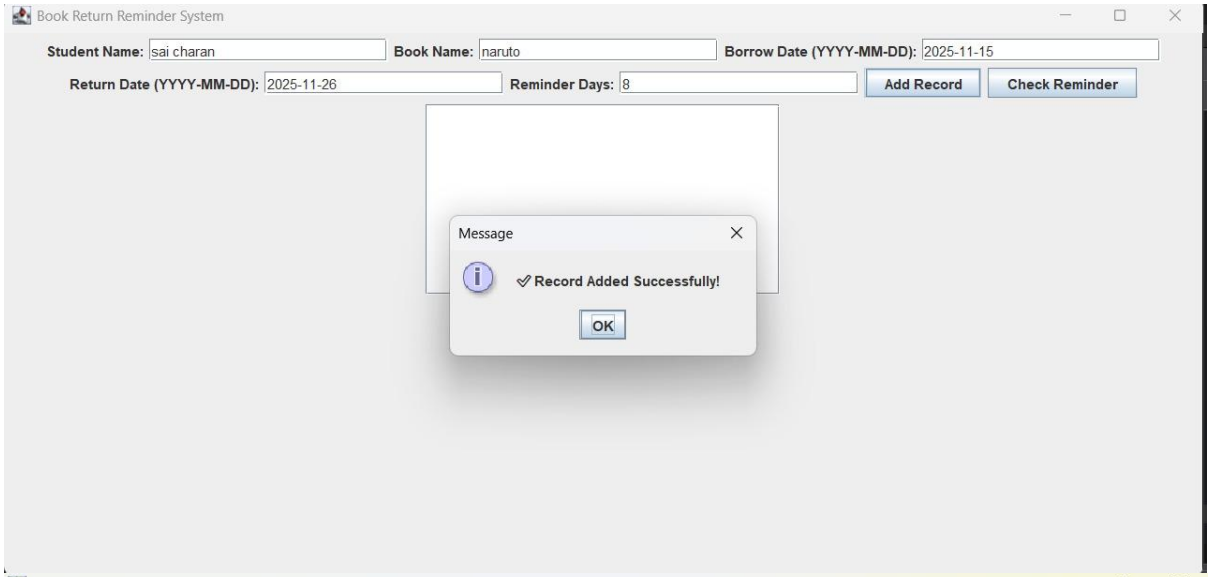
Backend – https://github.com/aakashjonnalagadda/Book-return-reminder-system/blob/main/report

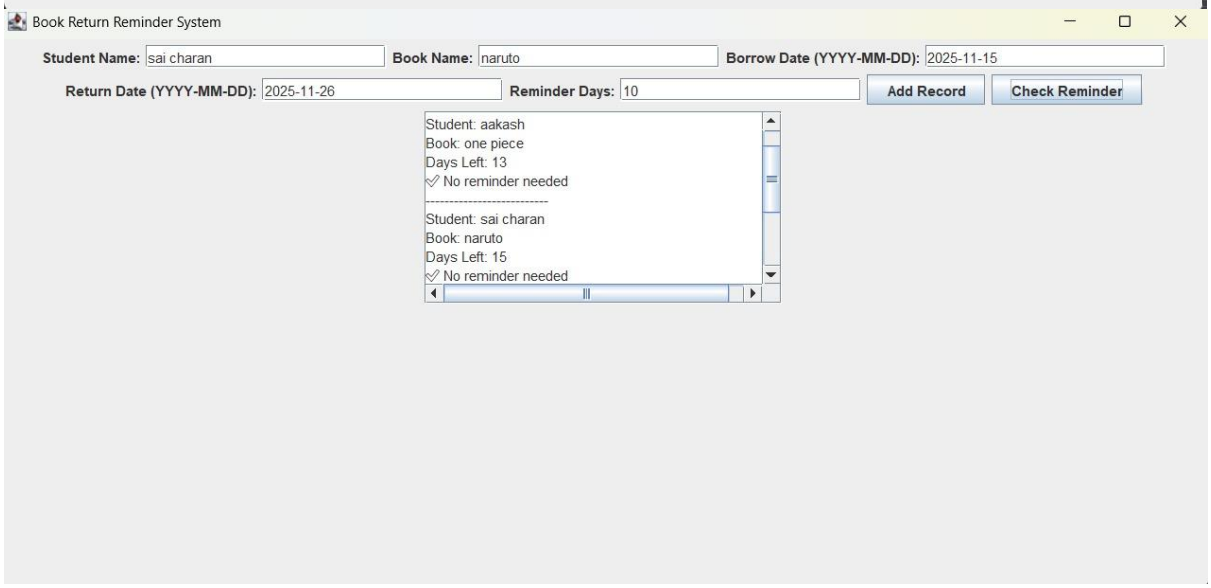
Screenshots :

Frontend Structure :

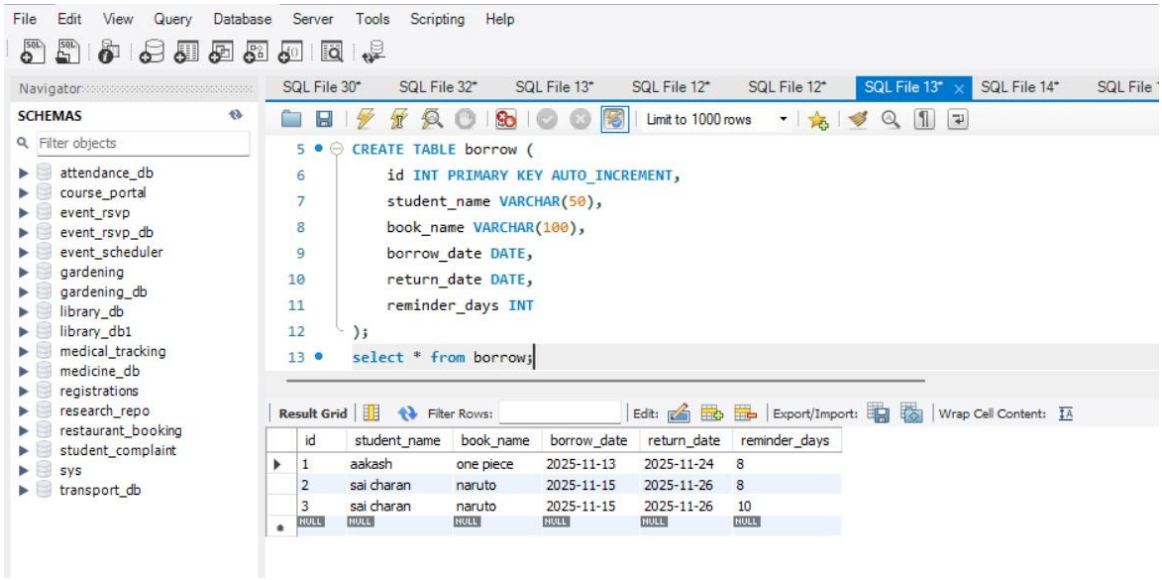








Backend Structure :



CONCLUSION

The Book Return Reminder System efficiently automates the library's book return process by managing due dates and sending timely reminders. The integration of Java Swing, MySQL, and JDBC ensures robust performance and smooth data handling.

This project minimizes human effort, reduces late returns, and demonstrates the application of programming and database management skills in real-world use cases. Its modular design ensures scalability and easy maintenance.

Overall, the system reflects the core principles of modern software engineering, combining clarity, usability, and automation.

FUTURE WORK

- Add admin login and authentication.
- Enable automated email/SMS notifications.
- Integrate barcode scanning for book entries.
- Include book renewal and fine calculation modules.
- Implement analytics dashboard for borrowed book trends.
- Create a cloud-based multi-user version.
- Develop Android/iOS app versions.
- Add dark/light theme and accessibility support.
- Use AI/ML models for predicting late returns.
- Deploy with CI/CD and Docker for scalability.