SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Mumbai Campus)
Computer Engineering Department (B.Tech Integrated Sem VI)
Java Programming
Mini Project

# PART A

(Part A: TO BE REFFERED BY STUDENTS)

## A.1 AIM:

Develop a Mini Project in Java

## A.2 Pre requisite:

Basic Knowledge of Java, Java FX.

## A.3 Outcome:

After successful completion of this experiment students will be able to:

**1.** Develop a standalone application using Java

## A.4 Theory:

A mini project is desirable to be completed by a group of two or three students that cover following tools.
Java
Java FX

## A.5 Procedure/Task:

1. Develop an application using Java and Java FX.
2. Prepare the document. Save and close the file and name it as RollNo._MiniProject.

SVKM's NMIMS

Mukesh Patel School of Technology Management & Engineering (Mumbai Campus)

Computer Engineering Department (B.Tech Integrated Sem VI)

Java Programming

Mini Project

**PART B**

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

| | |
|---|---|
| Roll No. : C113 | Name: Vidhie Jhunjhunwala |
| Class : C | Batch : C2 |
| Date of Experiment : | Date/Time of Submission : |
| Grade : | |

**B.1 Project Details**

1. Motivation to take the proposed project: The proposed project, a Library Management System, is motivated by the need for efficient management of books and their related data in a library. Traditional manual methods of managing libraries can be time-consuming, error-prone and may not be able to handle large amounts of data. Hence, there is a need for a digital solution that can effectively manage the library's data, including books, authors, categories, etc. The proposed project aims to provide such a solution with a user-friendly interface and functionalities such as adding, editing, and deleting books from the library's database.
2. Working of project: The Library Management System project with GUI allows the user to input book details, such as book name, author name, ISBN, and category. These details are then added to a JTable that displays all the books in the library. The user can also delete or edit books from the table as required.

When the user clicks on the "Add Book" button, the data entered in the form fields is added as a new row to the JTable. Similarly, when the user clicks on the "Delete Book" button, the selected rows from the JTable are deleted, and when the user clicks on the "Edit Book" button, the selected row in the JTable is edited with the new data entered in the form fields.

3. Advantages & disadvantages of project: Advantages:

- Efficient management of books and related data in a library.
- A user-friendly interface that allows easy interaction with the system.

- The ability to handle large amounts of data, which can be difficult or impossible to manage manually.
- Time-saving, as the system performs tasks much faster than manual methods.
- Improved accuracy and reduced errors, as the system eliminates the need for manual data entry and processing.

Disadvantages:

- Lack of a backend, which means that the data is stored in memory and is lost when the program is closed.
- The lack of security measures, such as user authentication and authorization, which can result in unauthorized access to the system and data.
- Limited functionalities, as the system only allows adding, deleting, and editing books.

4. Technologies used for developing the project: The Library Management System project with GUI is developed using Java programming language and Swing GUI toolkit. The project does not use a backend, and hence, no database management system is used to store the data. The project is implemented using a simple JTable to store the book details in memory. The project can be run on any platform that supports Java.

**B.2 Output**

SVKM's NMIMS

Mukesh Patel School of Technology Management & Engineering (Mumbai Campus)

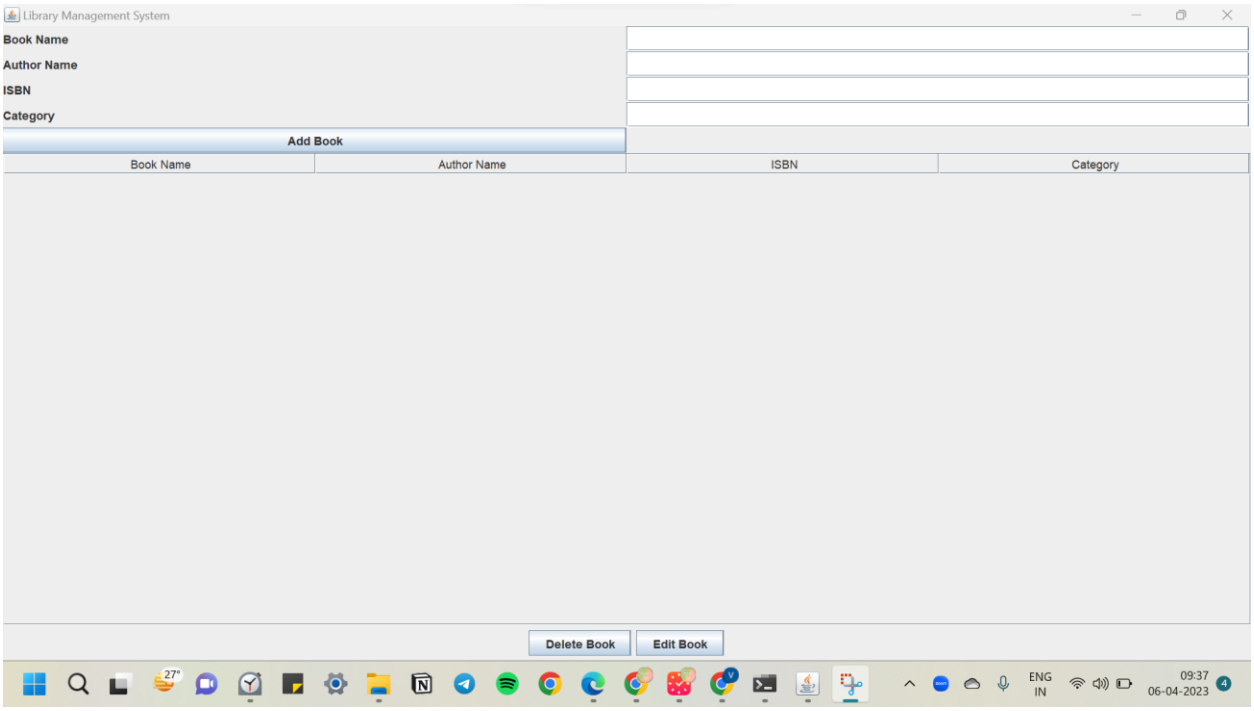
Computer Engineering Department (B.Tech Integrated Sem VI)

Java Programming

Mini Project

```
Command Prompt - java  Libr
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jiyaj>cd C:\Users\jiyaj\OneDrive\Desktop

C:\Users\jiyaj\OneDrive\Desktop>javac LibraryManagmentSystemGUI.java
error: file not found: LibraryManagmentSystemGUI.java
Usage: javac <options> <source files>
use --help for a list of possible options

C:\Users\jiyaj\OneDrive\Desktop>javac LibraryManagementSystemGUI.java

C:\Users\jiyaj\OneDrive\Desktop>java LibraryManagementSystemGUI
```

**How the system looks:**




**To add a book, you enter the details like this:**

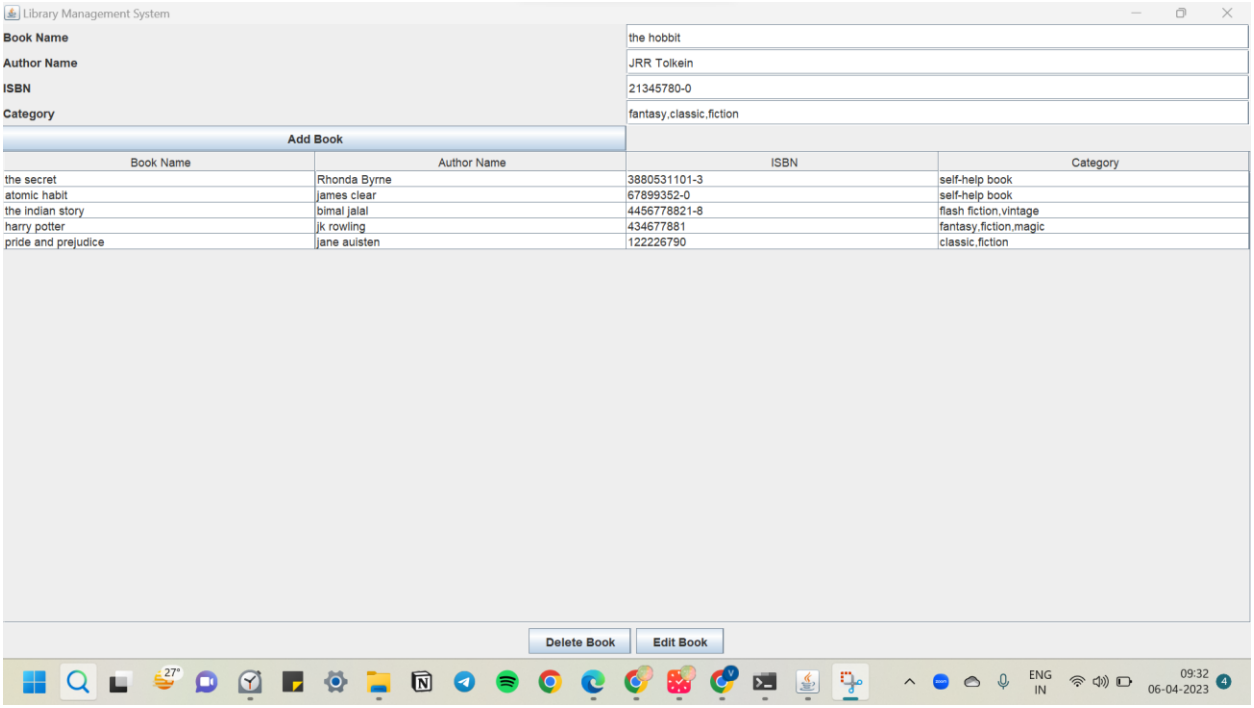
SVKM's NMIMS

Mukesh Patel School of Technology Management & Engineering (Mumbai Campus)

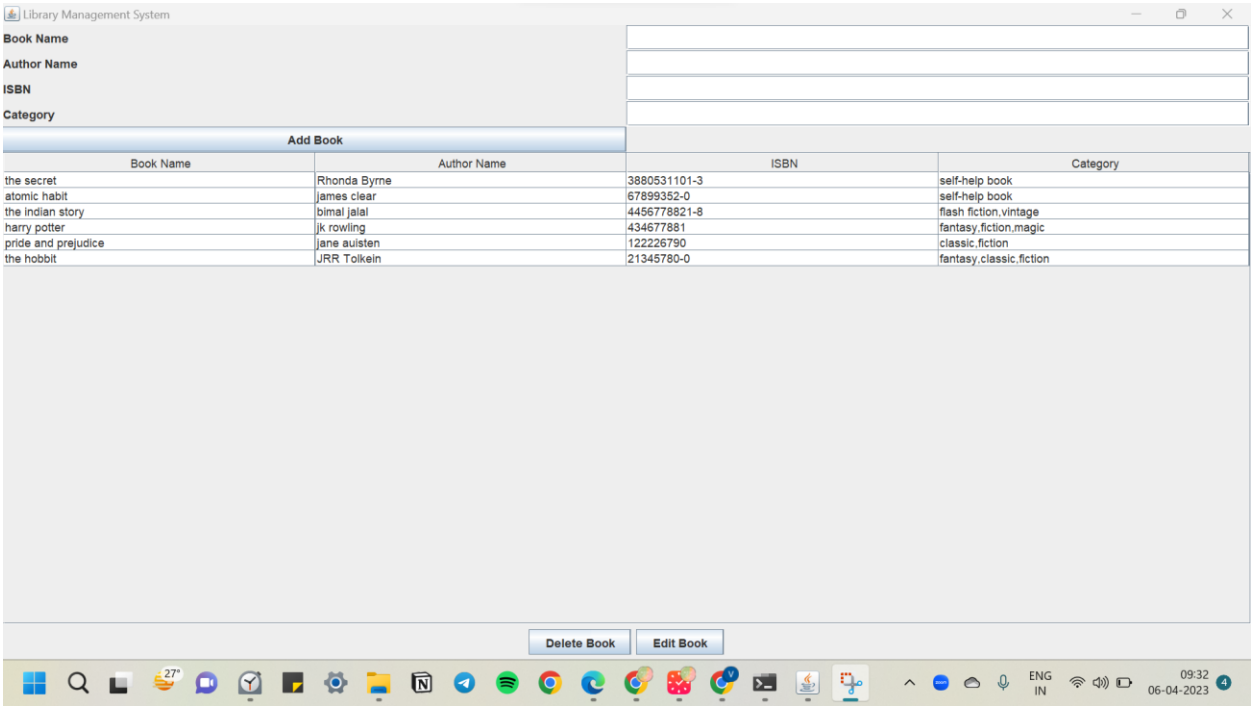
Computer Engineering Department (B.Tech Integrated Sem VI)

Java Programming

Mini Project




**And then click on add book( the data is stored in the table):**




**To edit a row, click on the edit button:** (error)

SVKM's NMIMS

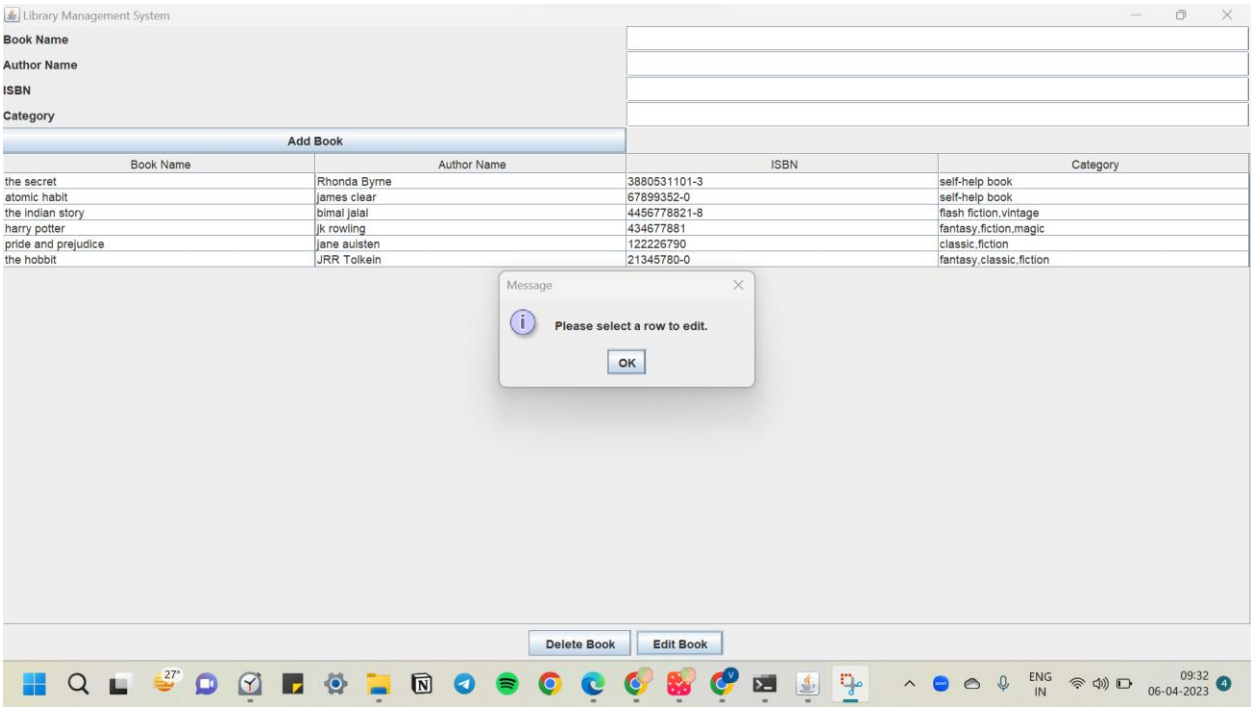
Mukesh Patel School of Technology Management & Engineering (Mumbai Campus)

Computer Engineering Department (B.Tech Integrated Sem VI)

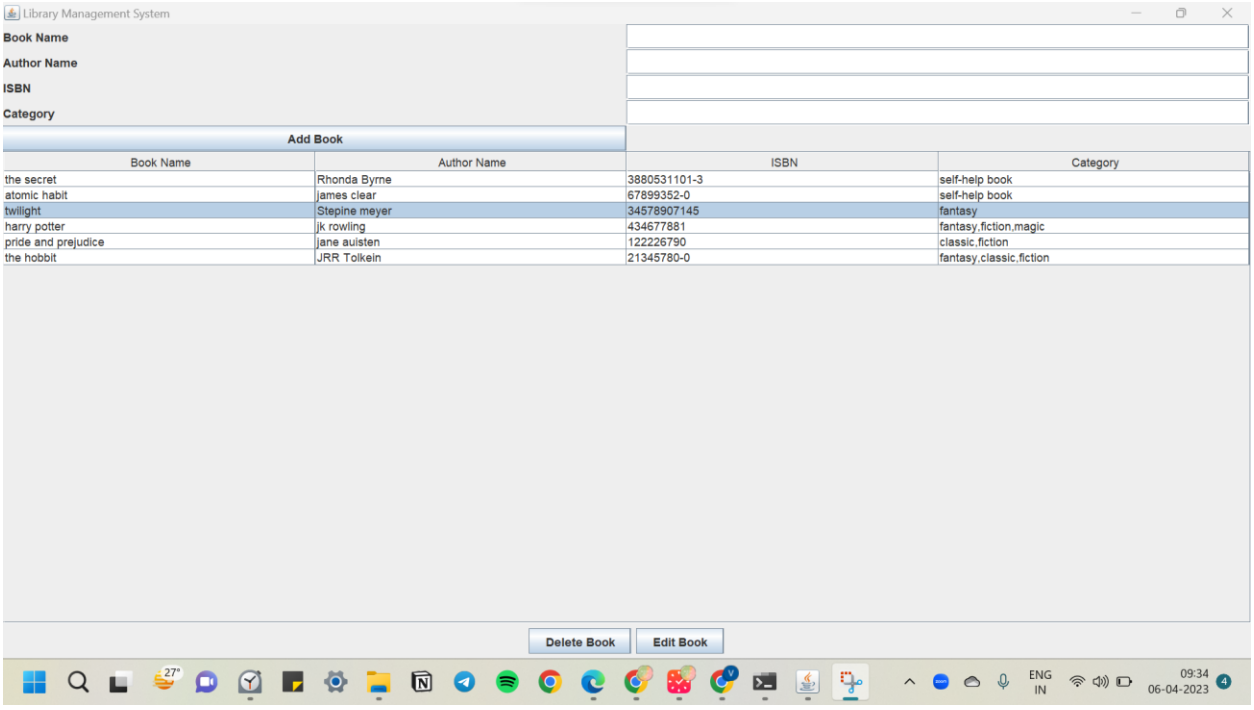
Java Programming

Mini Project

## Asks to select a row




## Select a row and edit:



SVKM's NMIMS

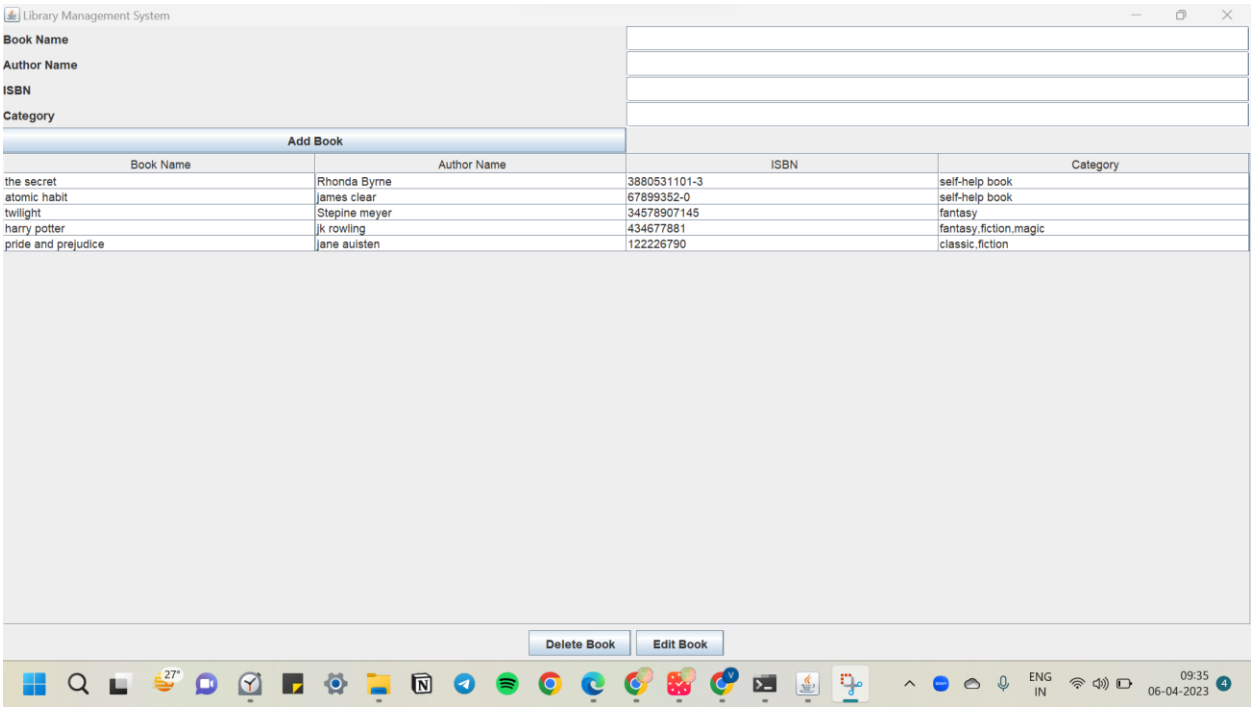
Mukesh Patel School of Technology Management & Engineering (Mumbai Campus)

Computer Engineering Department (B.Tech Integrated Sem VI)

Java Programming

Mini Project

**To delete, simple select the row and click on the delete button: (we deleted the last row)**




**B.3 Project Code :**

```
import javax.swing.*;

import javax.swing.table.DefaultTableModel;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Vector;


public class LibraryManagementSystemGUI extends JFrame implements ActionListener {

    private JButton addButton, deleteButton, editButton;

    private JLabel bookNameLabel, authorNameLabel, isbnLabel, categoryLabel;
```

```
    private JTextField bookNameTextField, authorNameTextField, isbnTextField,
categoryTextField;

    private JTable booksTable;


    public LibraryManagementSystemGUI() {

        setTitle("Library Management System");

        setSize(800, 600);

        setLocationRelativeTo(null);

        setDefaultCloseOperation(EXIT_ON_CLOSE);


        // Add Book Panel

        JPanel addBookPanel = new JPanel(new GridLayout(5, 2));

        bookNameLabel = new JLabel("Book Name");

        bookNameTextField = new JTextField();

        authorNameLabel = new JLabel("Author Name");

        authorNameTextField = new JTextField();

        isbnLabel = new JLabel("ISBN");

        isbnTextField = new JTextField();

        categoryLabel = new JLabel("Category");

        categoryTextField = new JTextField();

        addBookPanel.add(bookNameLabel);

        addBookPanel.add(bookNameTextField);

        addBookPanel.add(authorNameLabel);

        addBookPanel.add(authorNameTextField);

        addBookPanel.add(isbnLabel);
```

```
addBookPanel.add(isbnTextField);

addBookPanel.add(categoryLabel);

addBookPanel.add(categoryTextField);

addButton = new JButton("Add Book");

addButton.addActionListener(this);

addBookPanel.add(addButton);


// Books Table Panel

JPanel booksTablePanel = new JPanel(new BorderLayout());

booksTable = new JTable(new DefaultTableModel(new Object[]{"Book Name", "Author Name", "ISBN", "Category"}, 0));

JScrollPane scrollPane = new JScrollPane(booksTable);

booksTablePanel.add(scrollPane);


// Delete and Edit Buttons Panel

JPanel buttonsPanel = new JPanel(new FlowLayout());

deleteButton = new JButton("Delete Book");

deleteButton.addActionListener(this);

buttonsPanel.add(deleteButton);

editButton = new JButton("Edit Book");

editButton.addActionListener(this);

buttonsPanel.add(editButton);


// Main Panel

JPanel mainPanel = new JPanel(new BorderLayout());
```

```
    mainPanel.add(addBookPanel, BorderLayout.NORTH);
    mainPanel.add(booksTablePanel, BorderLayout.CENTER);
    mainPanel.add(buttonsPanel, BorderLayout.SOUTH);

    add(mainPanel);
    setVisible(true);
}

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

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == addButton) {
        String bookName = bookNameTextField.getText();
        String authorName = authorNameTextField.getText();
        String isbn = isbnTextField.getText();
        String category = categoryTextField.getText();
        DefaultTableModel model = (DefaultTableModel) booksTable.getModel();
        model.addRow(new Object[]{bookName, authorName, isbn, category});
        bookNameTextField.setText("");
        authorNameTextField.setText("");
        isbnTextField.setText("");
```

```
            categoryTextField.setText("");
        } else if (e.getSource() == deleteButton) {
            int[] rows = booksTable.getSelectedRows();
            DefaultTableModel model = (DefaultTableModel) booksTable.getModel();
            for (int i = 0; i < rows.length; i++) {
                model.removeRow(rows[i] - i);
            }
        } else if (e.getSource() == editButton) {
            int row = booksTable.getSelectedRow();
            if (row == -1) {
                JOptionPane.showMessageDialog(this, "Please select a row to edit.");
            } else {


String bookName = bookNameTextField.getText();
                String authorName = authorNameTextField.getText();
                String isbn = isbnTextField.getText();
                String category = categoryTextField.getText();
                DefaultTableModel model = (DefaultTableModel) booksTable.getModel();
                model.setValueAt(bookName, row, 0);
                model.setValueAt(authorName, row, 1);
                model.setValueAt(isbn, row, 2);
                model.setValueAt(category, row, 3);
                bookNameTextField.setText("");
                authorNameTextField.setText("");
```

```
            isbnTextField.setText("");

            categoryTextField.setText("");

        }

    }

}
```

**B.4 Conclusion:**

The Library Management System project with GUI provides an efficient solution for managing books and their related data in a library. The project is developed using Java programming language and Swing GUI toolkit, and it allows the user to add, delete, and edit books in a JTable. The project has several advantages, including efficient data management, time-saving, and reduced errors. However, the lack of a backend and security measures can be considered as a disadvantage.

**B.5 Observations and Learning:**

The Library Management System project with GUI provides a good understanding of Java programming language and the Swing GUI toolkit. The project involves several key concepts, such as event-driven programming, JTable, and user interface design, which are essential for developing GUI-based applications. Additionally, the project highlights the importance of data management and processing, which are essential for any software application. Overall, this project provides a great opportunity to learn and apply Java programming skills in a practical and meaningful way.