MINI PROJECT:

GUI LIBRARY MANAGEMENT SYSTEM

Ex No: 12

Date: 03/05/2023

AIM:
To develop a simple GUI based database application - Library Management System.

DESCRIPTION:
We have chosen to develop a simple GUI based application for Library Database Management. Our app provides a simple and visually appealing GUI interface for performing CRUD operations, namely inserting a new book and deleting existing books from the Library database. Moreover our application provides a visual model of the 'books' table data, dynamically fetched from the database using the API and displayed on the application table.

TECH STACK:
SceneBuilder/FXML - For designing the frontend
Java/JavaFX - For implementing the GUI elements and the backend code.
MySQL Database
Eclipse IDE

SOURCE CODE:
The project folder consists of five files -
1. Main.java file - To load the .fxml file and set up the Stage
2. LibraryUIController.java - FXML controller class
3. DButil.java file - To implement database connectivity
4. LibraryUI.fxml
5. libraryDB.sql - To create the library database and the books table

Main.java:

```
package application;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.image.Image;
import javafx.scene.Parent;
import javafx.fxml.FXMLLoader;


public class Main extends Application {
	public void start(Stage primaryStage) {
		try {
				Parent root = 
FXMLLoader.load(getClass().getResource("LibraryUI.fxml"));
				Scene scene = new Scene(root);
				primaryStage.setScene(scene);
				primaryStage.setResizable(false);
				primaryStage.setTitle("E-Libray Management");
				primaryStage.getIcons().add(new 
Image("C:\\Users\\Fathima Zulaikha\\workspace-2\\Library Management 
System - miniproject\\src\\assets\\library-logo-books.png"));

				primaryStage.show();

		}catch(Exception e) {
			e.printStackTrace();
		}
	}

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

LibraryUIController.java:

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ResourceBundle;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.AnchorPane;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import javafx.scene.text.Text;
import utilities.*;

public class LibraryUIController implements Initializable {
        Connection con = null;

        @FXML
        private TableView<Book> booksTable;

    @FXML
    private TableColumn<Book, String> Author;
```

```
@FXML
private TableColumn<Book, String> Edition;

@FXML
private TableColumn<Book, String> Name;

@FXML
private TableColumn<Book, String> Publisher;

ObservableList<Book> data = FXCollections.observableArrayList();

@FXML
private Button addBTN;

@FXML
private Text authorLabel;

@FXML
private TextField authorName;

@FXML
private Text bookLabel;

@FXML
private TextField bookName;

@FXML
private TableView<?> booksTabel;

@FXML
private Button deleteBTN;

@FXML
private Text editionLabel;
```

```
    @FXML
    private TextField editionName;

    @FXML
    private AnchorPane formSection;

    @FXML
    private Text publisherLabel;

    @FXML
    private TextField publisherName;

    @FXML
    private Label viewLabel;

    @FXML
    private AnchorPane viewSection;

    @FXML
    private Label welcomeLabel;

    @FXML
    private Font x3;

    @FXML
    private Color x4;

    @FXML
    public void addBTNOnClicked(ActionEvent event1)throws IOException {

        if(bookName.getText().isBlank() == false &&
authorName.getText().isBlank() == false &&
publisherName.getText().isBlank() == false &&
editionName.getText().isBlank() == false) {
```

```
            addbook();

            data.add(new Book(
                        bookName.getText(),
                        authorName.getText(),
                        publisherName.getText(),
                        editionName.getText()
            ));
            bookName.clear();
                  authorName.clear();
                  publisherName.clear();
                  editionName.clear();

      }else {
            System.out.println("Please fill in all the details");
      }
   }


      @FXML
   public void deleteBTNOnClicked(ActionEvent event2)throws IOException
{
            if(bookName.getText().isBlank() == false &&
authorName.getText().isBlank() == false &&
publisherName.getText().isBlank() == false &&
editionName.getText().isBlank() == false) {

            deletebook();
            bookName.clear();
                  authorName.clear();
                  publisherName.clear();
                  editionName.clear();

      }else {
```

```
                System.out.println("Please fill in all the details");
        }

    }

    //function to add a new book
    public void addbook() {
        try {
                con = DButil.getConnection();
        String bookadd = "Insert into
books(bookName,author,publisherName,edition) values(?,?,?,?)";
        PreparedStatement p=con.prepareStatement(bookadd);
                        p.setString(1,bookName.getText());
                        p.setString(2,authorName.getText());
                        p.setString(3,publisherName.getText());
                        p.setString(4,editionName.getText());
                        p.executeUpdate();
        }
        catch(Exception e) {
                e.printStackTrace();
        }

    }

    //Function to delete the book
    public void deletebook() {
        try {
                con = DButil.getConnection();
        String bookdel = "Delete from books where bookName = ? and
author = ? and publisherName = ? and edition = ?";
        PreparedStatement p=con.prepareStatement(bookdel);
                        p.setString(1,bookName.getText());
                        p.setString(2,authorName.getText());
                        p.setString(3,publisherName.getText());
                        p.setString(4,editionName.getText());
```

```
                    p.executeUpdate();

        }
        catch(Exception e) {
                e.printStackTrace();
        }
    }

    //Function to fetch the data from the Database and display it in the GUI
table
    public void fetchTable() {
        try {
                con = DButil.getConnection();
        String fetch = "Select * from books";
        PreparedStatement p=con.prepareStatement(fetch);
                    ResultSet rs = p.executeQuery();

                    while(rs.next()) {
                            data.add(new Book(
                                    rs.getString("bookName"),
                                    rs.getString("author"),
                                    rs.getString("publisherName"),
                                    rs.getString("edition")
                    ));
                    }
        }
        catch(Exception e) {
                e.printStackTrace();
        }
    }

    //Defining a data model for a book object
    public class Book {

        final String bookname;
        final String authorname;
        final String publishername;
        final String editionname;

       public Book(String bName, String aName, String pName, String
eName){
          this.bookname = bName;
          this.authorname = aName;
          this.publishername = pName;
          this.editionname = eName;
       }

        public String getBookname() {
                return bookname;
        }

        public String getAuthorname() {
                return authorname;
        }

        public String getPublishername() {
                return publishername;
        }

        public String getEditionname() {
                return editionname;
        }
    }

        @Override
        public void initialize(URL url, ResourceBundle rb) {
                fetchTable();
                Name.setCellValueFactory(new PropertyValueFactory<Book,
String>("bookname"));
```

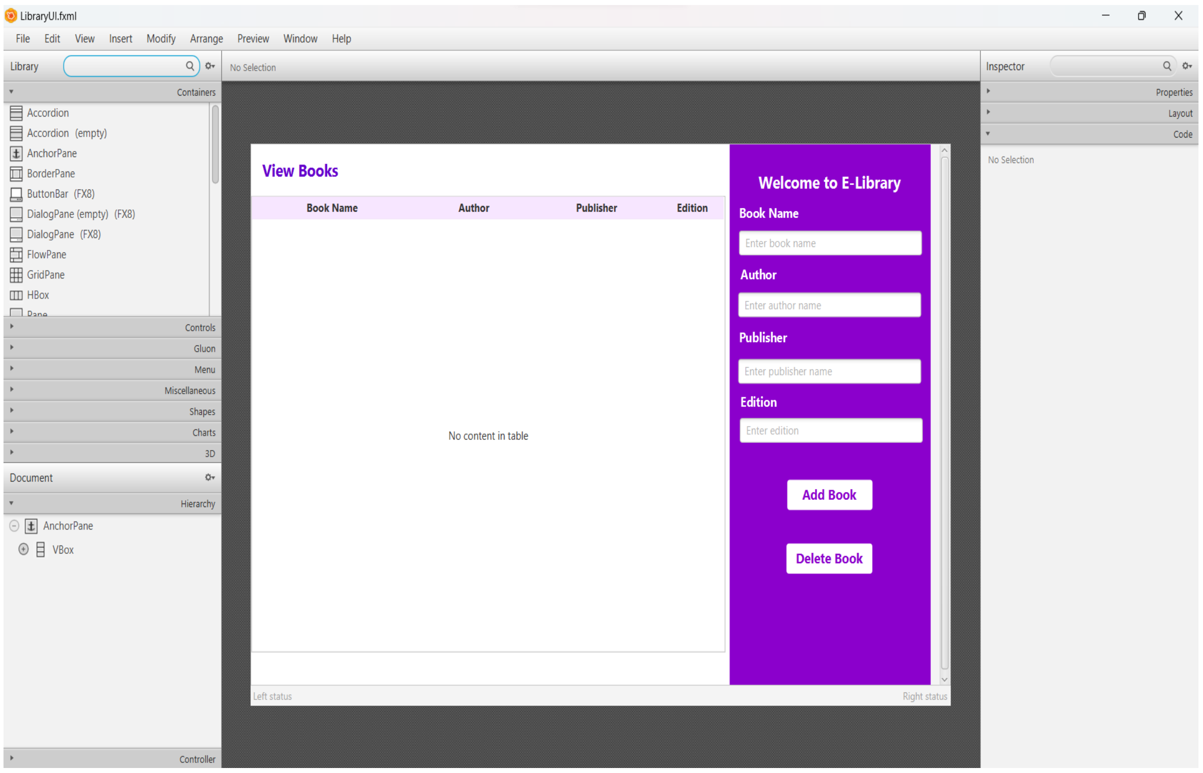
```
        Author.setCellValueFactory(new PropertyValueFactory<Book,
String>("authorname"));
        Publisher.setCellValueFactory(new PropertyValueFactory<Book,
String>("publishername"));
        Edition.setCellValueFactory(new PropertyValueFactory<Book,
String>("editionname"));
        booksTable.setItems(data);

         }

}
```

DButil.java file:

```
package utilities;

import java.sql.Connection;
import java.sql.DriverManager;

public class DButil {

         public DButil() {}
                  // TODO Auto-generated constructor stub
                  public static Connection getConnection() {
                           Connection con =null;
                                             try
                           {

Class.forName("com.mysql.cj.jdbc.Driver");
                                                      con=DriverManager.getConnection(

"jdbc:mysql://localhost:3306/library","root","12345");
                           }
                                             catch(Exception e)
                                             {
                                                      System.out.println(e);
```

```
                                        }
                                        return con;
                            // TODO Auto-generated constructor stub
                    }


        }
```

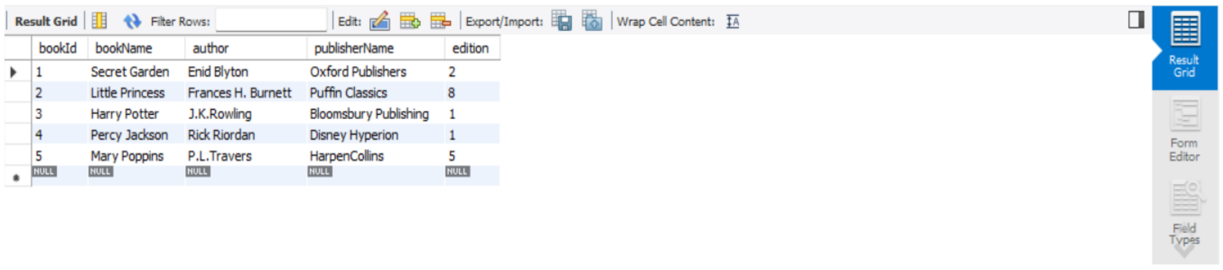
LibraryUI.fxml in SceneBuilder:



libraryDB.sql:

```
create database library;
use library;

create table books(bookId int auto_increment, bookName varchar(30),
author varchar(30), publisherName varchar(30), edition varchar(5), primary
key(bookId));
```

```
-- Initialize the books table
insert into books(bookName,author,publisherName,edition) values("Secret
Garden", "Enid Blyton", "Oxford Publishers", "2"),
                                        ("Little Princess","Frances H.
Burnett","Puffin Classics","8"),
                 ("Harry Potter","J.K.Rowling","Bloomsbury Publishing","1"),
                 ("Percy Jackson","Rick Riordan","Disney Hyperion","1"),
                 ("Mary Poppins","P.L.Travers","HarpenCollins","5");
select * from books;
```

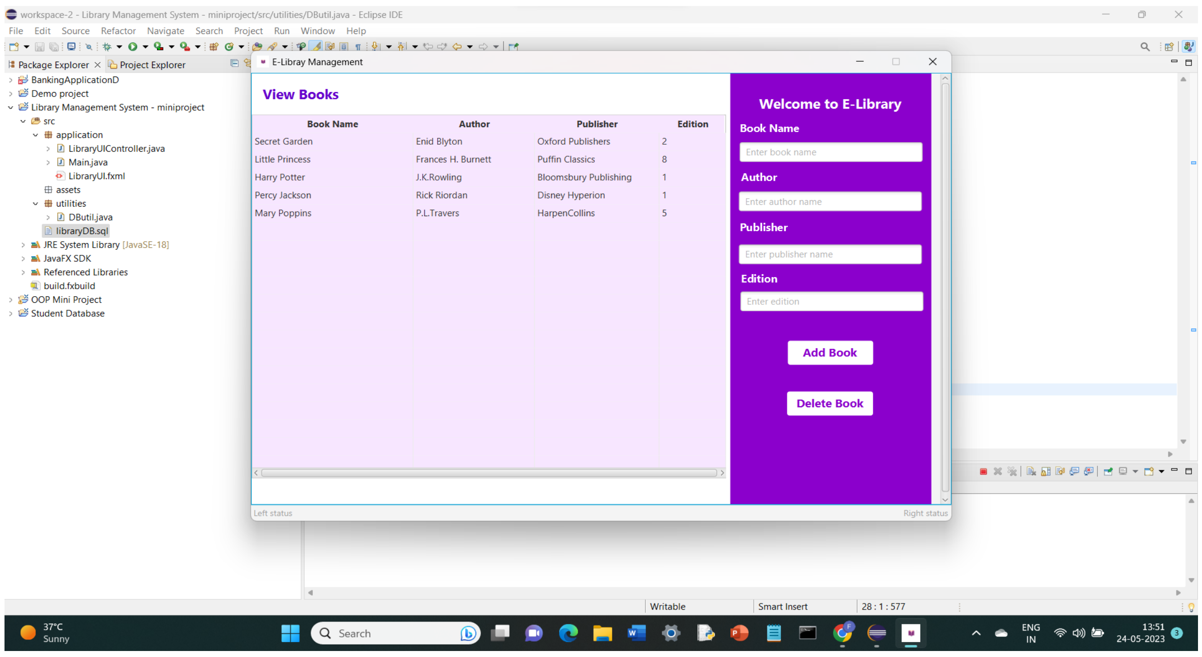


bookId	bookName	author	publisherName	edition
1	Secret Garden	Enid Blyton	Oxford Publishers	2
2	Little Princess	Frances H. Burnett	Puffin Classics	8
3	Harry Potter	J.K.Rowling	Bloomsbury Publishing	1
4	Percy Jackson	Rick Riordan	Disney Hyperion	1
5	Mary Poppins	P.L.Travers	HarpenCollins	5
NULL	NULL	NULL	NULL	NULL

```
truncate table books;
```

OUTPUT:

- Run the application on Eclipse IDE.



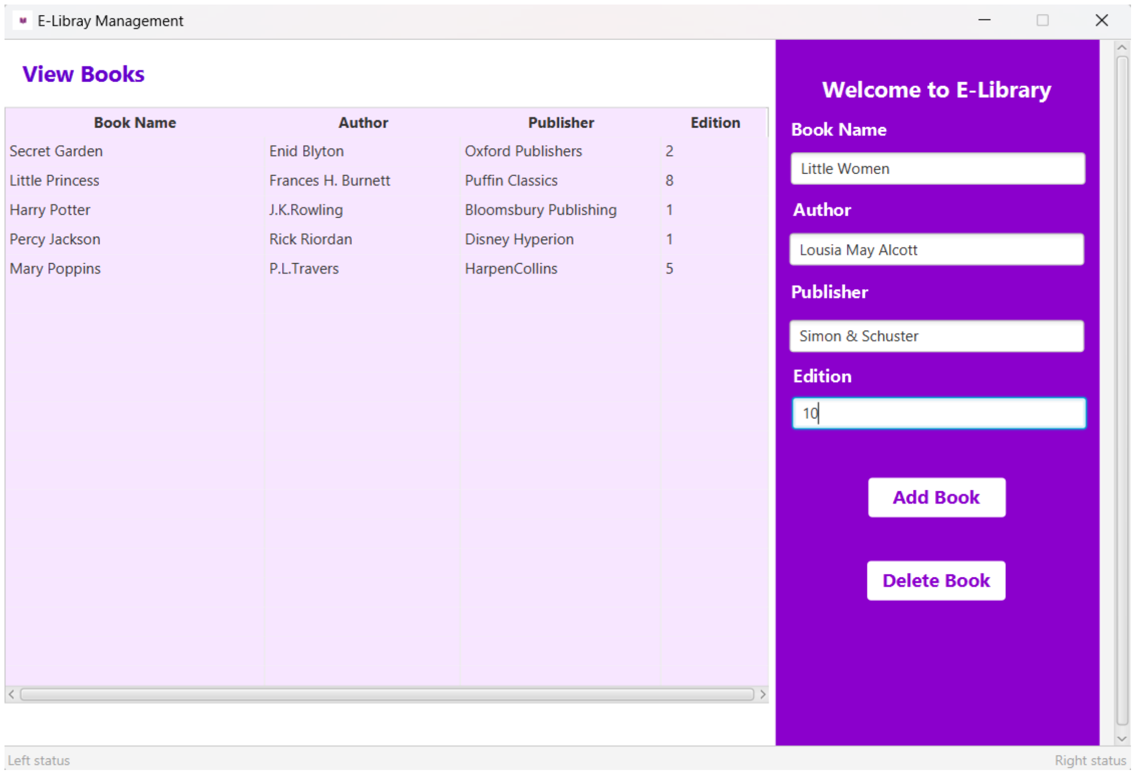
workspace-2 - Library Management System - miniproject/src/utilities/DButil.java - Eclipse IDE

E-Library Management

View Books

Book Name	Author	Publisher	Edition
Secret Garden	Enid Blyton	Oxford Publishers	2
Little Princess	Frances H. Burnett	Puffin Classics	8
Harry Potter	J.K.Rowling	Bloomsbury Publishing	1
Percy Jackson	Rick Riordan	Disney Hyperion	1
Mary Poppins	P.L.Travers	HarpenCollins	5

- Fill in the details of the book to be inserted.


E-Libray Management

View Books

Book Name	Author	Publisher	Edition
Secret Garden	Enid Blyton	Oxford Publishers	2
Little Princess	Frances H. Burnett	Puffin Classics	8
Harry Potter	J.K.Rowling	Bloomsbury Publishing	1
Percy Jackson	Rick Riordan	Disney Hyperion	1
Mary Poppins	P.L.Travers	HarpenCollins	5

Welcome to E-Library

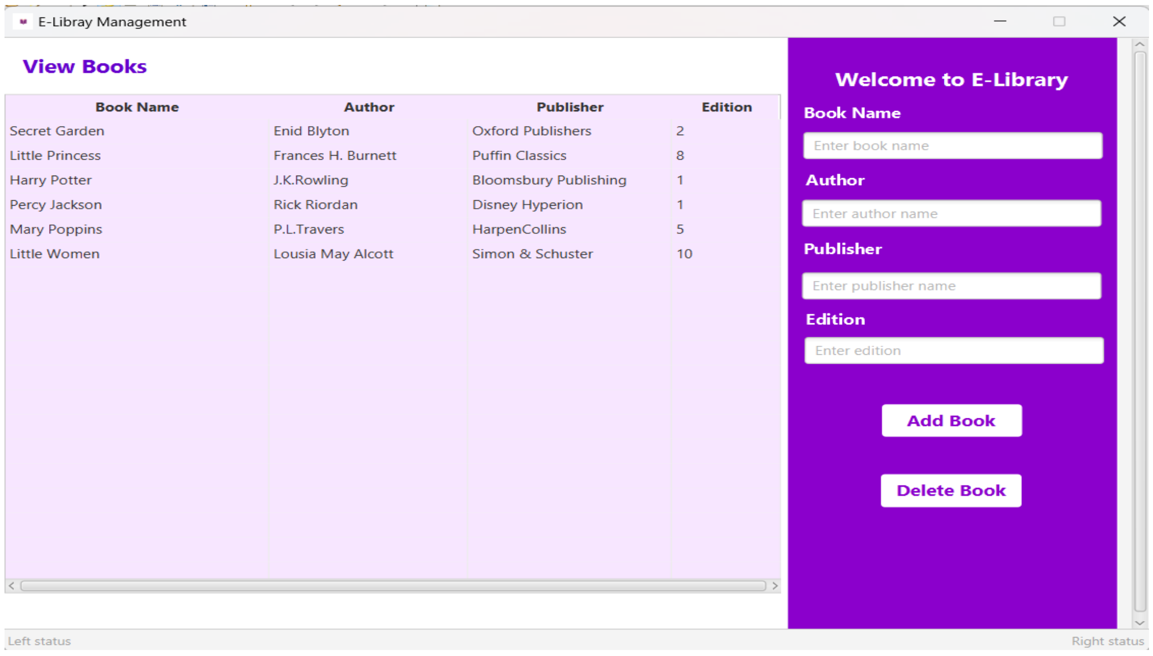
Book Name: Little Women

Author: Lousia May Alcott

Publisher: Simon & Schuster

Edition: 10

- Click the ‘Add Book’ button to insert the record.


E-Libray Management

View Books

Book Name	Author	Publisher	Edition
Secret Garden	Enid Blyton	Oxford Publishers	2
Little Princess	Frances H. Burnett	Puffin Classics	8
Harry Potter	J.K.Rowling	Bloomsbury Publishing	1
Percy Jackson	Rick Riordan	Disney Hyperion	1
Mary Poppins	P.L.Travers	HarpenCollins	5
Little Women	Lousia May Alcott	Simon & Schuster	10

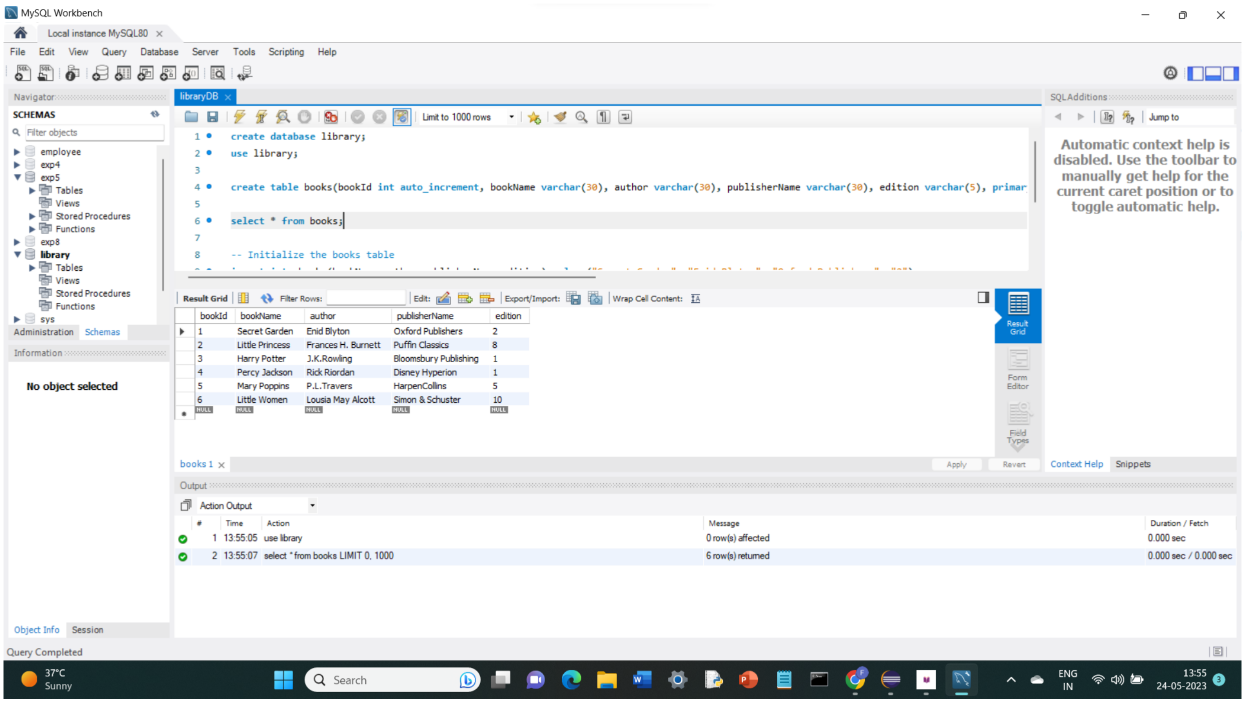
Welcome to E-Library

Book Name: Enter book name

Author: Enter author name

Publisher: Enter publisher name

Edition: Enter edition

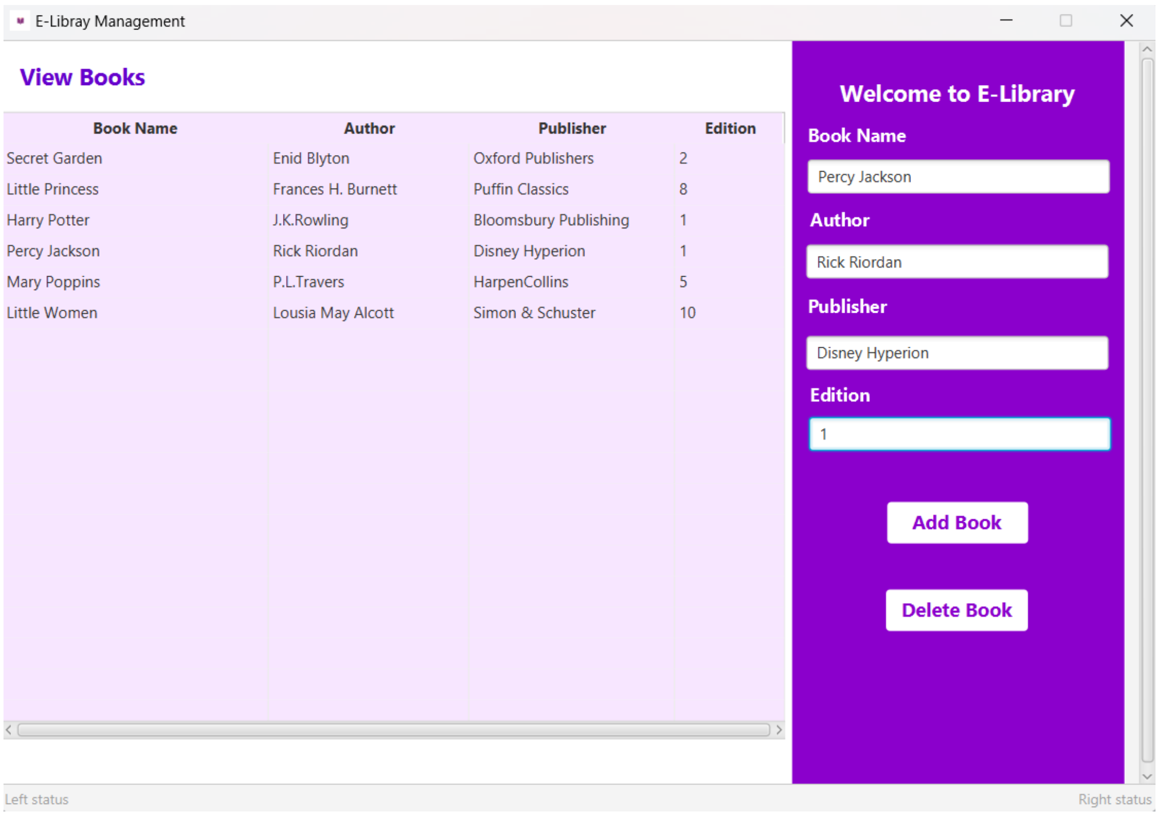


```
create database library;
use library;

create table books(bookId int auto_increment, bookName varchar(30), author varchar(30), publisherName varchar(30), edition varchar(5), primar

select * from books;

-- Initialize the books table
```

bookId	bookName	author	publisherName	edition
1	Secret Garden	Enid Blyton	Oxford Publishers	2
2	Little Princess	Frances H. Burnett	Puffin Classics	8
3	Harry Potter	J.K.Rowling	Bloomsbury Publishing	1
4	Percy Jackson	Rick Riordan	Disney Hyperion	1
5	Mary Poppins	P.L.Travers	HarpenCollins	5
6	Little Women	Lousia May Alcott	Simon & Schuster	10
NULL	NULL	NULL	NULL	NULL

```
#  Time      Action                           Message              Duration / Fetch
1  13:55:05  use library                      0 row(s) affected    0.000 sec
2  13:55:07  select * from books LIMIT 0, 1000  6 row(s) returned  0.000 sec / 0.000 sec
```

- Fill in the details of the book to be deleted and click 'Delete Book' .

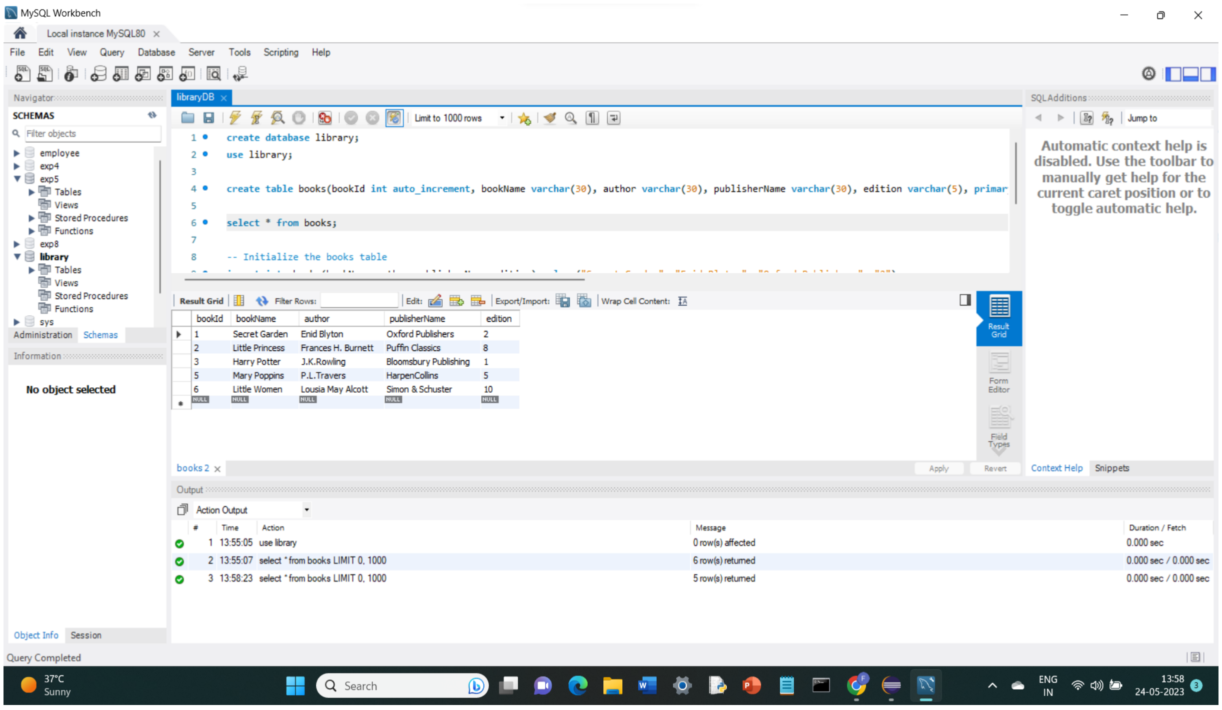


View Books

Book Name	Author	Publisher	Edition
Secret Garden	Enid Blyton	Oxford Publishers	2
Little Princess	Frances H. Burnett	Puffin Classics	8
Harry Potter	J.K.Rowling	Bloomsbury Publishing	1
Percy Jackson	Rick Riordan	Disney Hyperion	1
Mary Poppins	P.L.Travers	HarpenCollins	5
Little Women	Lousia May Alcott	Simon & Schuster	10

Welcome to E-Library

Book Name	Percy Jackson
Author	Rick Riordan
Publisher	Disney Hyperion
Edition	1



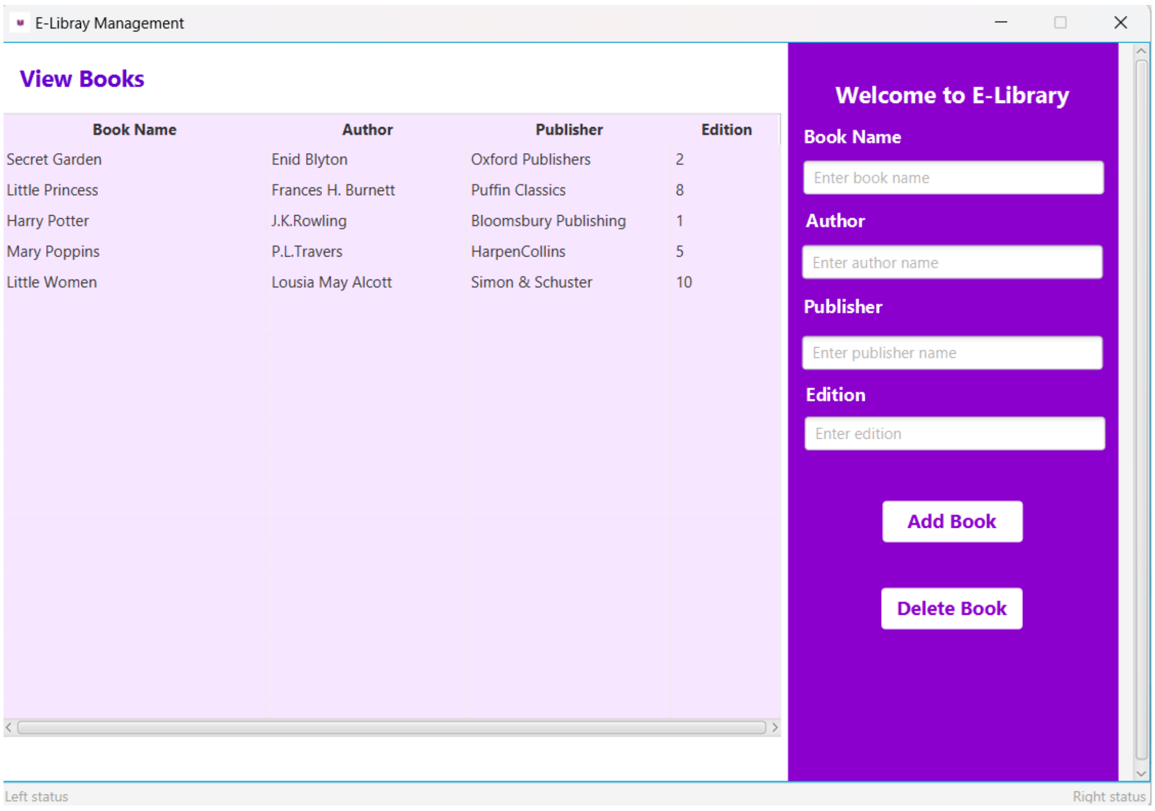
```sql
create database library;
use library;

create table books(bookId int auto_increment, bookName varchar(30), author varchar(30), publisherName varchar(30), edition varchar(5), primar

select * from books;

-- Initialize the books table
```

bookId	bookName	author	publisherName	edition
1	Secret Garden	Enid Blyton	Oxford Publishers	2
2	Little Princess	Frances H. Burnett	Puffin Classics	8
3	Harry Potter	J.K.Rowling	Bloomsbury Publishing	1
5	Mary Poppins	P.L.Travers	HarpenCollins	5
6	Little Women	Lousia May Alcott	Simon & Schuster	10
NULL	NULL	NULL	NULL	NULL

```
#  Time      Action                             Message              Duration / Fetch
1  13:55:05  use library                        0 row(s) affected    0.000 sec
2  13:55:07  select * from books LIMIT 0, 1000  6 row(s) returned    0.000 sec / 0.000 sec
3  13:58:23  select * from books LIMIT 0, 1000  5 row(s) returned    0.000 sec / 0.000 sec
```

Query Completed

- Close the window and run the application to see the updated table.



E-Libray Management

View Books

Book Name	Author	Publisher	Edition
Secret Garden	Enid Blyton	Oxford Publishers	2
Little Princess	Frances H. Burnett	Puffin Classics	8
Harry Potter	J.K.Rowling	Bloomsbury Publishing	1
Mary Poppins	P.L.Travers	HarpenCollins	5
Little Women	Lousia May Alcott	Simon & Schuster	10

Welcome to E-Library

Book Name: Enter book name

Author: Enter author name

Publisher: Enter publisher name

Edition: Enter edition

RESULT:

Hence, successfully developed a basic GUI based Library Database Management Application.