package Test.com;

import java.sql.\*;

import java.util.Scanner;

public class lib {

static Connection conn;

static Statement stmt;

static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

try {

Class.forName("com.mysql.cj.jdbc.Driver");

conn = DriverManager.getConnection("jdbc:mysql://localhost/projects", "root", "root");

stmt = conn.createStatement();

} catch (Exception e) {

e.printStackTrace();

}

while (true) {

System.out.println("\nLibrary Management System");

System.out.println("1. Register");

System.out.println("2. Login");

System.out.println("3. Addbook");

System.out.println("4. Updatebook");

System.out.println("5. Deletebook");

System.out.println("6. Issuebook");

System.out.println("7. Returnbook");

System.out.println("8. displaybooks");

System.out.println("9. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

scanner.nextLine();

switch (choice) {

case 1:

registerUser();

break;

case 2:

loginUser();

break;

case 3:

addBook();

break;

case 4:

updateBook();

break;

case 5:

deleteBook();

break;

case 6:

issueBook();

break;

case 7:

returnBook();

break;

case 8:

displayBooks();

break;

case 9:

exit();

break;

default:

System.out.println("Invalid choice. Please try again.");

}

}

}

static void registerUser() {

try {

System.out.println("User Registration");

System.out.print("Enter username: ");

String username = scanner.nextLine();

System.out.print("Enter password: ");

String password = scanner.nextLine();

// You should add more fields like name, email, etc., and validate user inputs.

String insertQuery = "INSERT INTO users (username, password) VALUES (?, ?)";

PreparedStatement preparedStatement = conn.prepareStatement(insertQuery);

preparedStatement.setString(1, username);

preparedStatement.setString(2, password);

int rowsInserted = preparedStatement.executeUpdate();

if (rowsInserted > 0) {

System.out.println("User registered successfully!");

} else {

System.out.println("User registration failed.");

}

} catch (SQLException e) {

e.printStackTrace();

}

}

static void loginUser() {

try {

System.out.println("User Login");

System.out.print("Enter username: ");

String username = scanner.nextLine();

System.out.print("Enter password: ");

String password = scanner.nextLine();

String selectQuery = "SELECT \* FROM users WHERE username = ? AND password = ?";

PreparedStatement preparedStatement = conn.prepareStatement(selectQuery);

preparedStatement.setString(1, username);

preparedStatement.setString(2, password);

ResultSet resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

System.out.println("Login successful!"+username);

// Implement actions after successful login, e.g., display user options.

// You can call other functions like addBook(), updateBook(), etc., here.

} else {

System.out.println("Login failed. Invalid username or password.");

}

} catch (SQLException e) {

e.printStackTrace();

}

}

static void addBook() {

try {

System.out.println("Add a Book");

System.out.print("Enter book title: ");

String title = scanner.nextLine();

System.out.print("Enter author: ");

String author = scanner.nextLine();

System.out.print("Enter ISBN: ");

String isbn = scanner.nextLine();

// You can add more fields like publication year, genre, etc.

String insertQuery = "INSERT INTO books (title, author, isbn) VALUES (?, ?, ?)";

PreparedStatement preparedStatement = conn.prepareStatement(insertQuery);

preparedStatement.setString(1, title);

preparedStatement.setString(2, author);

preparedStatement.setString(3, isbn);

int rowsInserted = preparedStatement.executeUpdate();

if (rowsInserted > 0) {

System.out.println("Book added successfully!");

} else {

System.out.println("Failed to add the book.");

}

} catch (SQLException e) {

e.printStackTrace();

}

}

static void updateBook() {

try {

System.out.println("Update a Book");

System.out.print("Enter book ISBN to update: ");

String isbn = scanner.nextLine();

// Check if the book with the provided ISBN exists

String selectQuery = "SELECT \* FROM books WHERE isbn = ?";

PreparedStatement selectStatement = conn.prepareStatement(selectQuery);

selectStatement.setString(1, isbn);

ResultSet resultSet = selectStatement.executeQuery();

if (resultSet.next()) {

// Book exists, allow updates

System.out.print("Enter new book title (leave empty to keep existing): ");

String newTitle = scanner.nextLine();

System.out.print("Enter new author (leave empty to keep existing): ");

String newAuthor = scanner.nextLine();

// You can add fields like publication year, genre, etc.

// Build the update query dynamically based on user input

StringBuilder updateQuery = new StringBuilder("UPDATE books SET ");

boolean needsComma = false;

if (!newTitle.isEmpty()) {

updateQuery.append("title = ?");

needsComma = true;

}

if (!newAuthor.isEmpty()) {

if (needsComma) {

updateQuery.append(", ");

}

updateQuery.append("author = ?");

needsComma = true;

}

// Add more fields to update as needed

updateQuery.append(" WHERE isbn = ?");

PreparedStatement updateStatement = conn.prepareStatement(updateQuery.toString());

int parameterIndex = 1;

if (!newTitle.isEmpty()) {

updateStatement.setString(parameterIndex++, newTitle);

}

if (!newAuthor.isEmpty()) {

updateStatement.setString(parameterIndex++, newAuthor);

}

// Add parameters for additional fields to update

updateStatement.setString(parameterIndex, isbn);

int rowsUpdated = updateStatement.executeUpdate();

if (rowsUpdated > 0) {

System.out.println("Book updated successfully!");

} else {

System.out.println("No changes made to the book.");

}

} else {

System.out.println("Book with ISBN " + isbn + " does not exist.");

}

} catch (SQLException e) {

e.printStackTrace();

}

}

static void deleteBook() {

try {

System.out.println("Delete a Book");

System.out.print("Enter book ISBN to delete: ");

String isbn = scanner.nextLine();

// Check if the book with the provided ISBN exists

String selectQuery = "SELECT \* FROM books WHERE isbn = ?";

PreparedStatement selectStatement = conn.prepareStatement(selectQuery);

selectStatement.setString(1, isbn);

ResultSet resultSet = selectStatement.executeQuery();

if (resultSet.next()) {

// Book exists, proceed with deletion

String deleteQuery = "DELETE FROM books WHERE isbn = ?";

PreparedStatement deleteStatement = conn.prepareStatement(deleteQuery);

deleteStatement.setString(1, isbn);

int rowsDeleted = deleteStatement.executeUpdate();

if (rowsDeleted > 0) {

System.out.println("Book deleted successfully!");

} else {

System.out.println("Failed to delete the book.");

}

} else {

System.out.println("Book with ISBN " + isbn + " does not exist.");

}

} catch (SQLException e) {

e.printStackTrace();

}

}

static void issueBook() {

try {

System.out.println("Issue a Book");

System.out.print("Enter book ISBN to issue: ");

String isbn = scanner.nextLine();

// Check if the book with the provided ISBN is available for issuance

String selectQuery = "SELECT \* FROM books WHERE isbn = ? AND available = 1";

PreparedStatement selectStatement = conn.prepareStatement(selectQuery);

selectStatement.setString(1, isbn);

ResultSet resultSet = selectStatement.executeQuery();

if (resultSet.next()) {

// Book is available for issuance, proceed with issuing

System.out.print("Enter user ID: "); // You may have a separate user registration and management system.

int userId = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

// Update the book's availability status and record the issuance

String updateQuery = "UPDATE books SET available = 0, issued\_to = ? WHERE isbn = ?";

PreparedStatement updateStatement = conn.prepareStatement(updateQuery);

updateStatement.setInt(1, userId);

updateStatement.setString(2, isbn);

int rowsUpdated = updateStatement.executeUpdate();

if (rowsUpdated > 0) {

System.out.println("Book issued successfully!");

} else {

System.out.println("Failed to issue the book.");

}

} else {

System.out.println("Book with ISBN " + isbn + " is not available for issuance.");

}

} catch (SQLException e) {

e.printStackTrace();

}

}

static void returnBook() {

try {

System.out.println("Return a Book");

System.out.print("Enter book ISBN to return: ");

String isbn = scanner.nextLine();

// Check if the book with the provided ISBN is currently issued

String selectQuery = "SELECT \* FROM books WHERE isbn = ? AND available = 0";

PreparedStatement selectStatement = conn.prepareStatement(selectQuery);

selectStatement.setString(1, isbn);

ResultSet resultSet = selectStatement.executeQuery();

if (resultSet.next()) {

// Book is currently issued, proceed with return

String updateQuery = "UPDATE books SET available = 1, issued\_to = NULL WHERE isbn = ?";

PreparedStatement updateStatement = conn.prepareStatement(updateQuery);

updateStatement.setString(1, isbn);

int rowsUpdated = updateStatement.executeUpdate();

if (rowsUpdated > 0) {

System.out.println("Book returned successfully!");

} else {

System.out.println("Failed to return the book.");

}

} else {

System.out.println("Book with ISBN " + isbn + " is not currently issued.");

}

} catch (SQLException e) {

e.printStackTrace();

}

}

static void displayBooks() {

try {

System.out.println("List of Books:");

// SQL query to retrieve all books from the 'books' table

String selectQuery = "SELECT \* FROM books";

PreparedStatement preparedStatement = conn.prepareStatement(selectQuery);

ResultSet resultSet = preparedStatement.executeQuery();

// Check if there are any books in the database

boolean hasBooks = false;

while (resultSet.next()) {

hasBooks = true;

String title = resultSet.getString("title");

String author = resultSet.getString("author");

String isbn = resultSet.getString("isbn");

// Display book information

System.out.println("Title: " + title);

System.out.println("Author: " + author);

System.out.println("ISBN: " + isbn);

System.out.println("-------------------");

}

if (!hasBooks) {

System.out.println("No books found in the library.");

}

} catch (SQLException e) {

e.printStackTrace();

}

}

static void exit() {

try {

conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

System.exit(0);

}

}