University of Engineering and Technology

New Campus, Lahore.

Department of Computer Science

Session: 2021

Project Name:

Library Management System (LMS)

Group Members:

AHMAD MURTAZA (2021-CS-629)

MUDASSAR ALI (2021-CS-635)

MASOOD ALI (2021-CS-646)

Submitted To:

Ma’am Namra Sheikh

Library Management System (LMS)

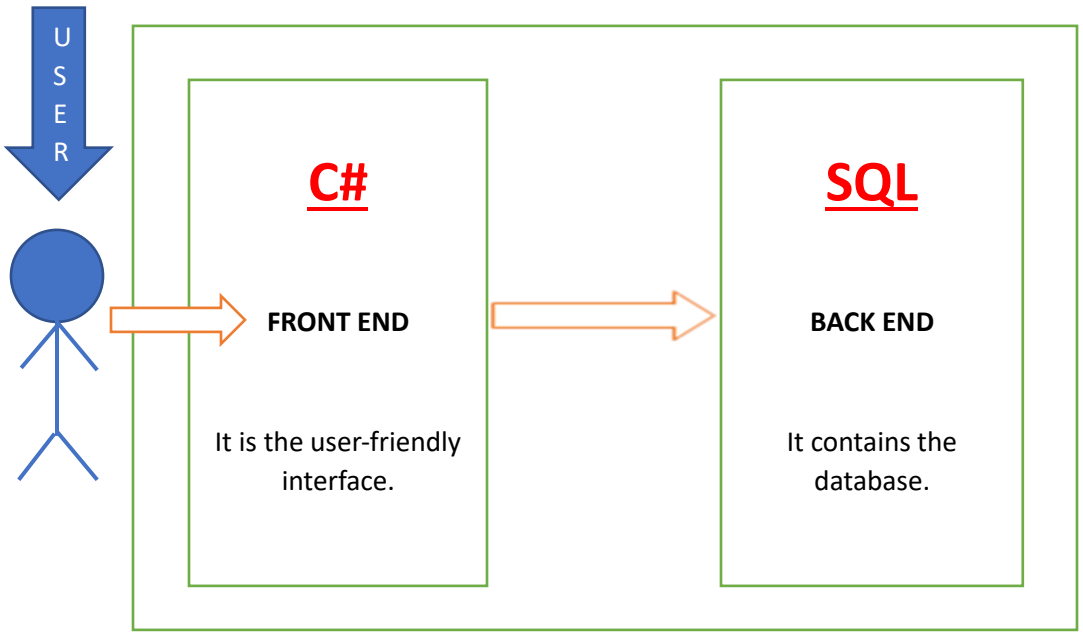
ABSTRACT:

The primary goal of the Library Management System is to simplify and enhance library operations, allowing librarians and library staff to effectively manage books, track borrowing and returning activities.

INTRODUCTION:

The project titled “Library Management System” is a Library management software for monitoring and controlling the transactions in a library.

The project “Library Management System” is developed in C#, which mainly focuses on basic operations in a library like adding new member, new books, and updating new information and members to borrow and return books.



MODULES:

The software Library Management System has four main modules.

1. Insertion to Database Module - User friendly input screen
2. Extracting from Database module – Attractive Output Screen
3. Report Generation module
    - Borrowed book list
    - Available book list
    - Returned book list

Activities:

- Add/Delete books.
- Maintain the database.
- View user details.
- Issue books.
- Return books.
- Maintain user details.

Created Tables:

1. **CREATE TABLE [dbo].[IssueTbl]**

```
(
    [INum] INT NOT NULL PRIMARY KEY IDENTITY,
    [StId] INT NOT NULL,
    [StName] VARBINARY(20) NOT NULL,
    [BookId] INT NOT NULL,
    [BookName] VARCHAR(20) NOT NULL,
    [IssueDate] DATE NOT NULL
CONSTRAINT [FK_IssueTbl_StudentTbl] FOREIGN KEY ([StId]) REFERENCES [dbo].[StudentTbl] ([StId]),
    CONSTRAINT [FK_IssueTbl_BookTbl] FOREIGN KEY ([BookId]) REFERENCES [dbo].[BookTbl] ([BId])
)
```

2. **CREATE TABLE [dbo].[BookTbl]**

```
(
    [BId] INT NOT NULL PRIMARY KEY IDENTITY,
    [BName] VARCHAR(20) NOT NULL,
    [BAuthor] VARCHAR(20) NOT NULL,
    [BPublisher] VARCHAR(100) NOT NULL,
    [BPrice] INT NOT NULL,
    [Bqty] INT NOT NULL
)
```

3. **CREATE TABLE [dbo].[LibrarianTbl]**

```
(
    [LibId] INT NOT NULL PRIMARY KEY IDENTITY(500, 1),
    [LibName] VARCHAR(20) NOT NULL,
    [LibPhone] VARCHAR(20) NOT NULL,
    [LibPass] VARCHAR(20) NOT NULL
)
```

4. **CREATE TABLE [dbo].[ReturnTbl]** (

```
    [RNum] INT NOT NULL IDENTITY,
    [StId] INT NOT NULL,
    [StName] VARCHAR(20) NOT NULL,
    [BookId] INT NOT NULL,
    [BookName] VARCHAR(20) NOT NULL,
    [IssueDate] DATE NOT NULL,
    [ReturnDate] DATE NOT NULL,
    [Fine] INT NOT NULL,
    PRIMARY KEY CLUSTERED ([RNum] ASC)
CONSTRAINT [FK_ReturnTbl_BookTbl] FOREIGN KEY ([BookId]) REFERENCES [dbo].[BookTbl]
([BId]),
```

CONSTRAINT [FK_ReturnTbl_StudentTbl] FOREIGN KEY ([StId]) REFERENCES [dbo].[StudentTbl] ([StId])

);

5. **CREATE TABLE [dbo].[StudentTbl]**

```
(
    [StId] INT NOT NULL PRIMARY KEY IDENTITY(100, 1),
    [StName] VARCHAR(20) NOT NULL,
    [StDep] VARCHAR(20) NOT NULL,
    [StSem] INT NOT NULL,
    [StPhone] NCHAR(20) NOT NULL
)
```

6. **CREATE TABLE [dbo].[IssuedBooksData]** (

```
    [Id]      INT         IDENTITY (1, 1) NOT NULL,
    [Log_data] VARCHAR (100) NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

Note:

The "PRIMARY KEY" constraint indicates the primary key column(s) for each table, ensuring uniqueness and efficient retrieval of data. The "IDENTITY" attribute specifies that the column is an identity column, automatically generating unique values. The data types specified for each column define the format and size of the data stored in the table.

ENTITIES:

- [dbo].[IssueTbl]([INum] , [StId] , [StName] , [BookId] , [BookName] , [IssueDate])
- [dbo].[BookTbl]([BId] , [BName] , [BAuthor] , [BPublisher] , [BPrice] , [Bqty])
- [dbo].[LibrarianTbl]([LibId] , [LibName] , [LibPhone] , [LibPass])
- [dbo].[ReturnTbl] ([RNum] ,[StId] , [StName] , [BookId] , [BookName] , [IssueDate] , [ReturnDate] , [Fine])
- [dbo].[StudentTbl]([StId] , [StName] , [StDep] , [StSem] , [StPhone])

Tables Documentation:

1. IssueTbl:

- INum: Unique identifier for each issued book.
- StId: The ID of the student who issued the book.
- StName: The name of the student who issued the book.
- BookId: The ID of the book that was issued.
- BookName: The name of the book that was issued.
- IssueDate: The date on which the book was issued.

2. BookTbl:

- BId: Unique identifier for each book in the library.
- BName: The name of the book.
- BAuthor: The author of the book.
- BPublisher: The publisher of the book.
- BPrice: The price of the book.
- Bqty: The quantity of the book available in the library.

3. LibrarianTbl:

- LibId: Unique identifier for each librarian.
- LibName: The name of the librarian.
- LibPhone: The phone number of the librarian.
- LibPass: The password for the librarian's account.

4. ReturnTbl:

- RNum: Unique identifier for each returned book.
- StId: The ID of the student who returned the book.
- StName: The name of the student who returned the book.
- BookId: The ID of the book that was returned.
- BookName: The name of the book that was returned.
- IssueDate: The date on which the book was issued.

- ReturnDate: The date on which the book was returned.
- Fine: The amount of fine, if any, associated with the returned book.

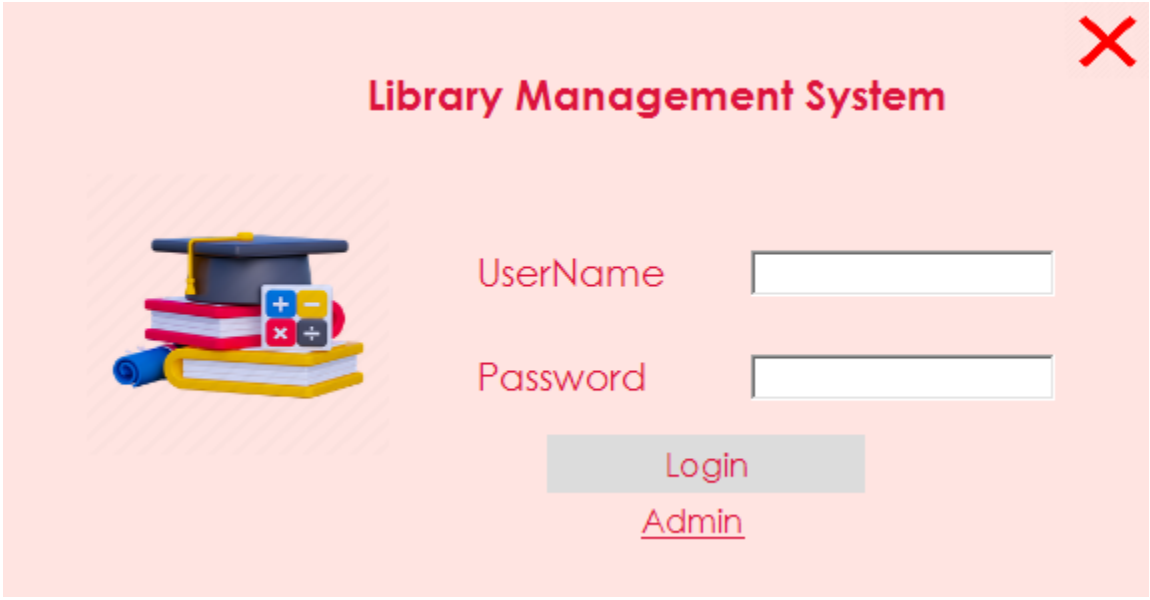
5. StudentTbl:

- StId: Unique identifier for each student.
- StName: The name of the student.
- StDep: The department of the student.
- StSem: The semester in which the student is currently enrolled.
- StPhone: The phone number of the student.

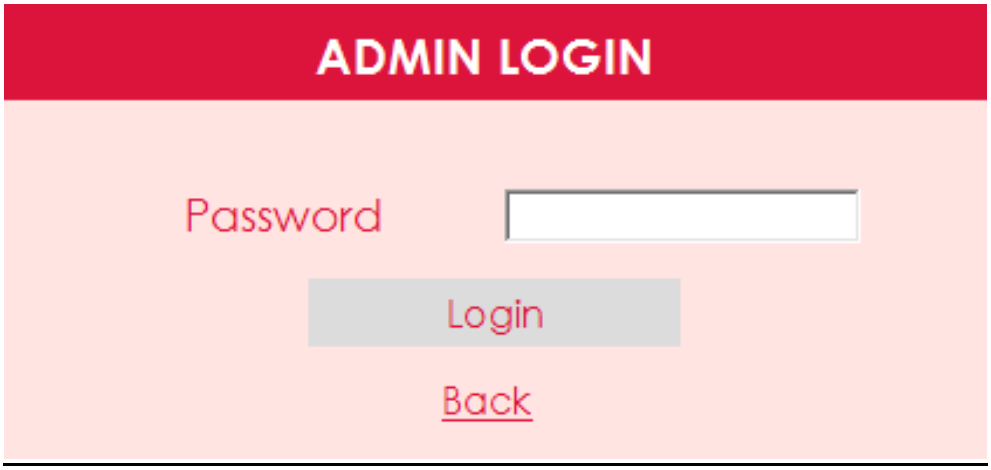
SAMPLE SCREENSHOTS

SPLASH:

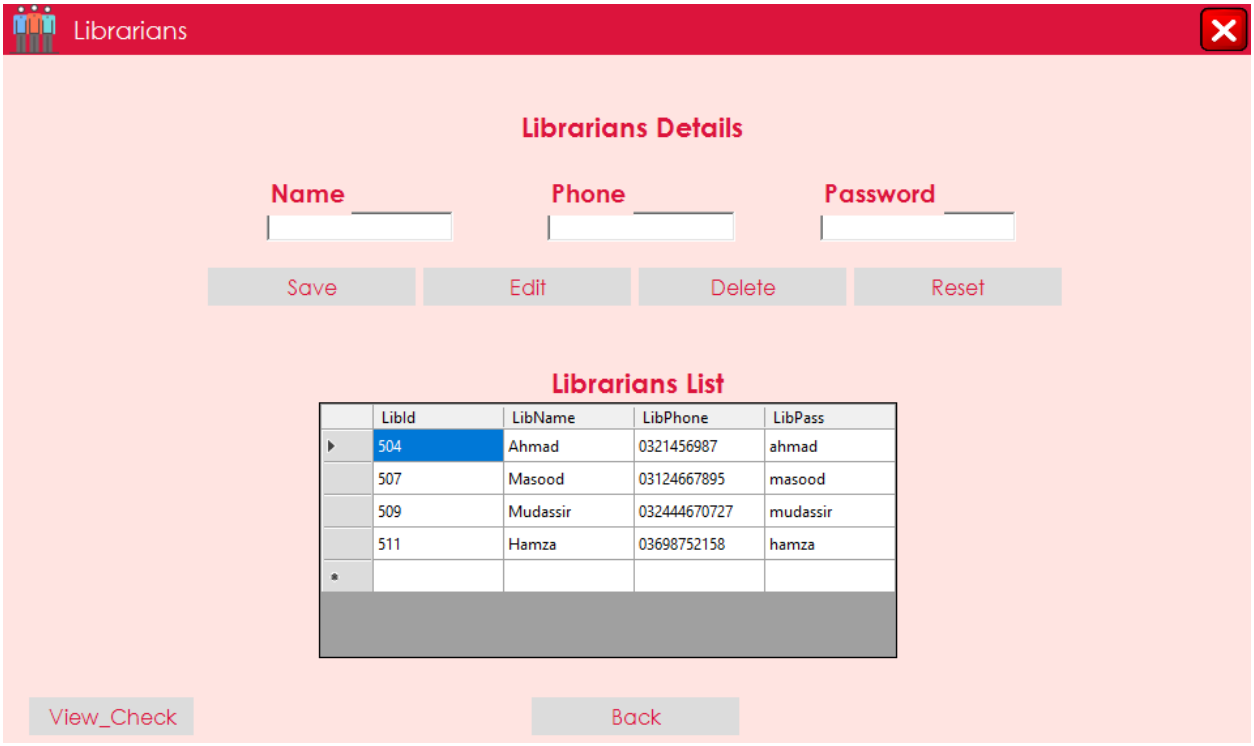
LOGIN PAGE:



ADMIN LOGIN PAGE:

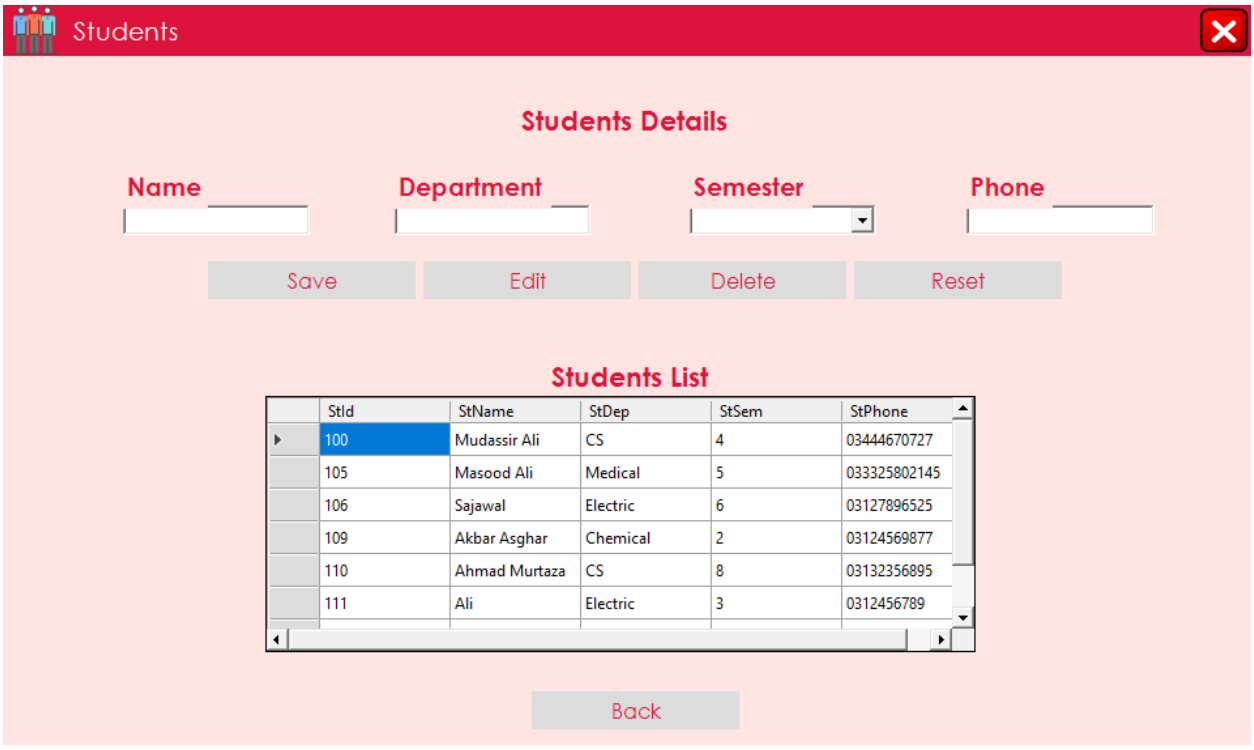


LIBRARIAN:

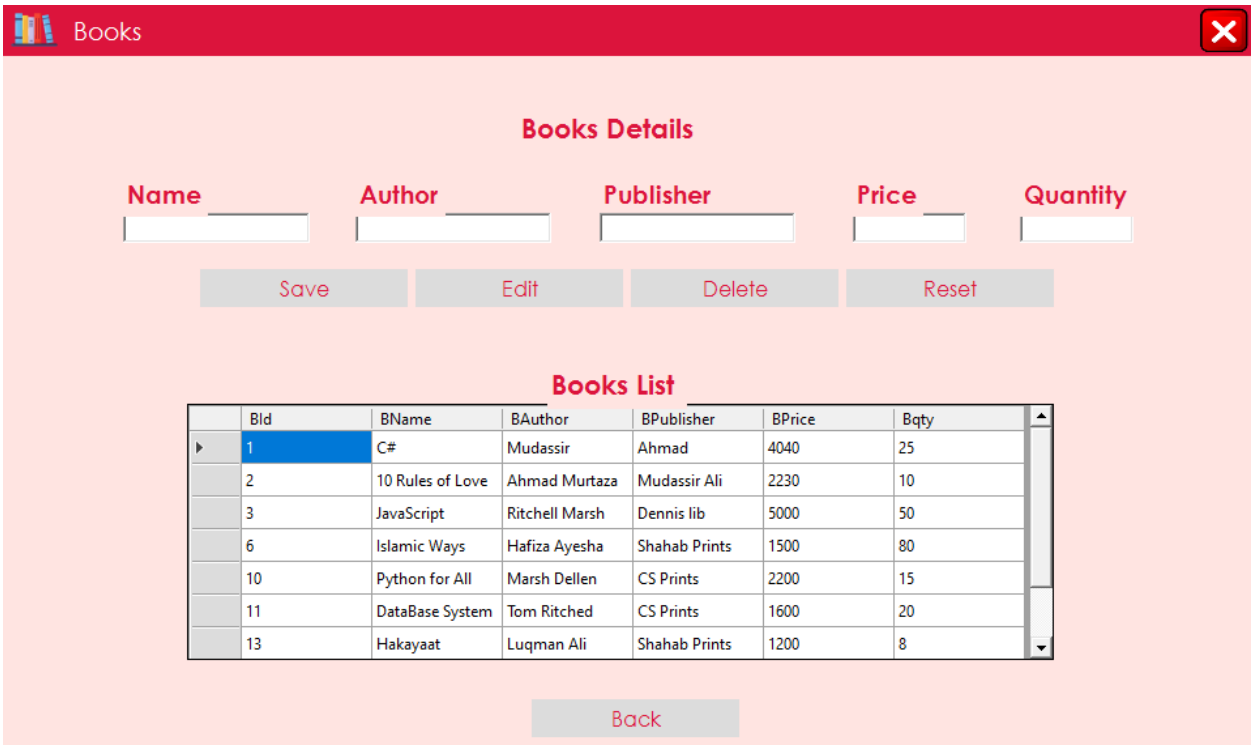


MAIN FORM:

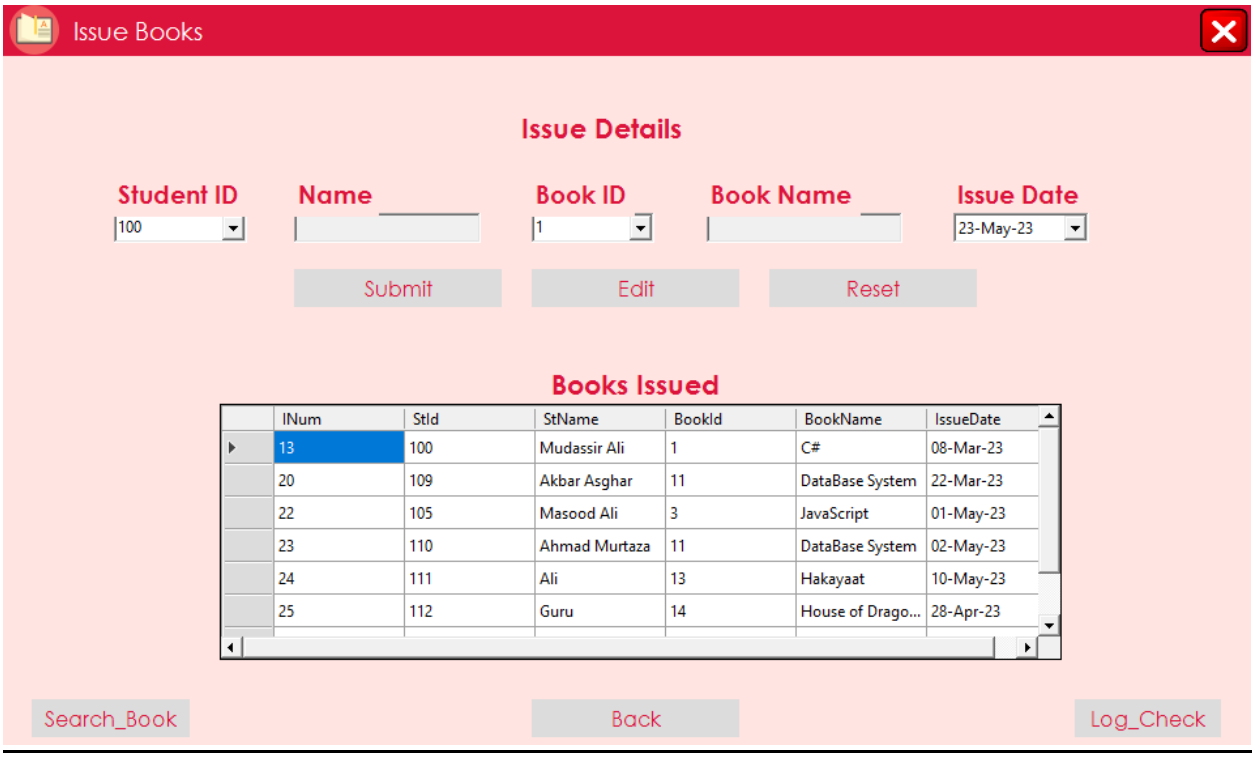
STUDENTS DETAIL:



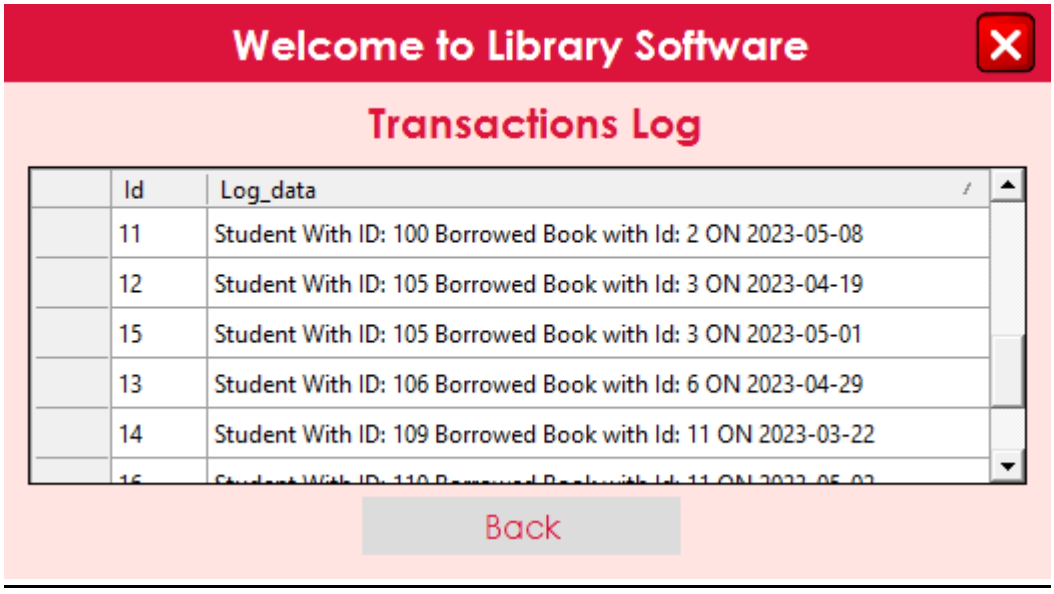
BOOKS DETAIL:



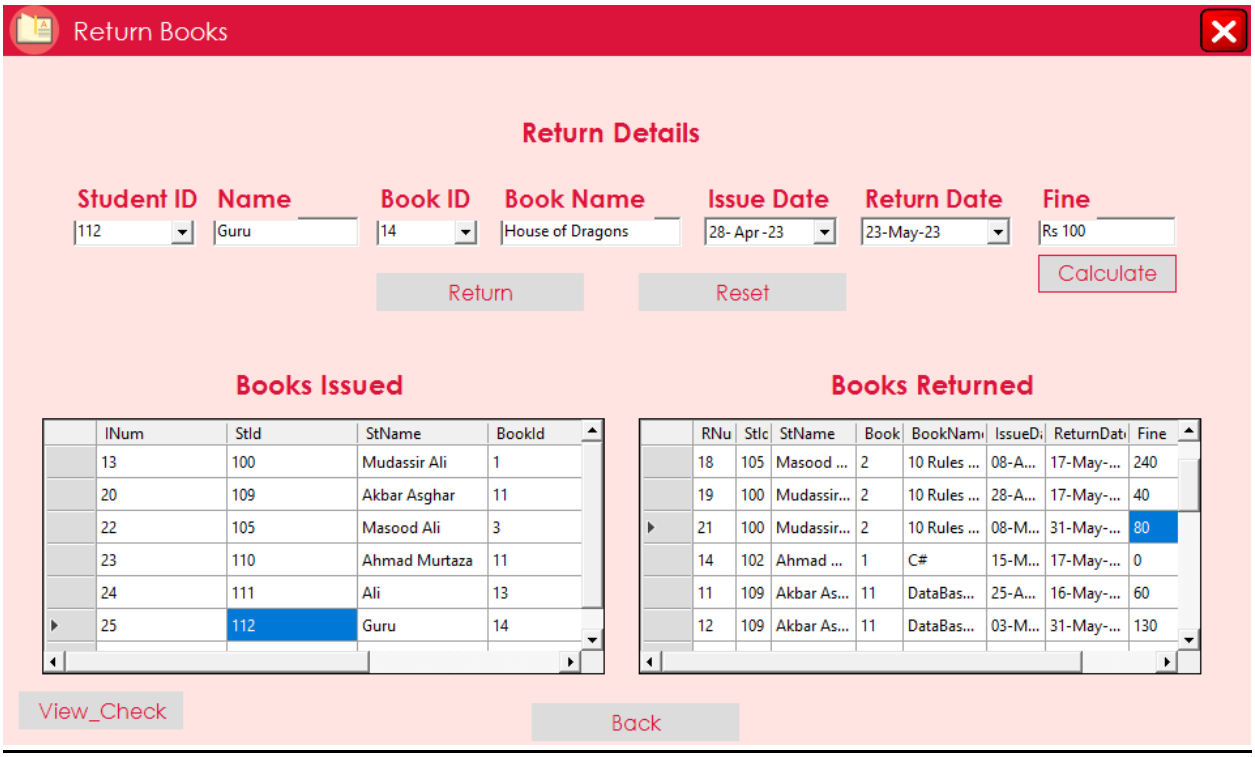
ISSUED BOOKS DETAIL:



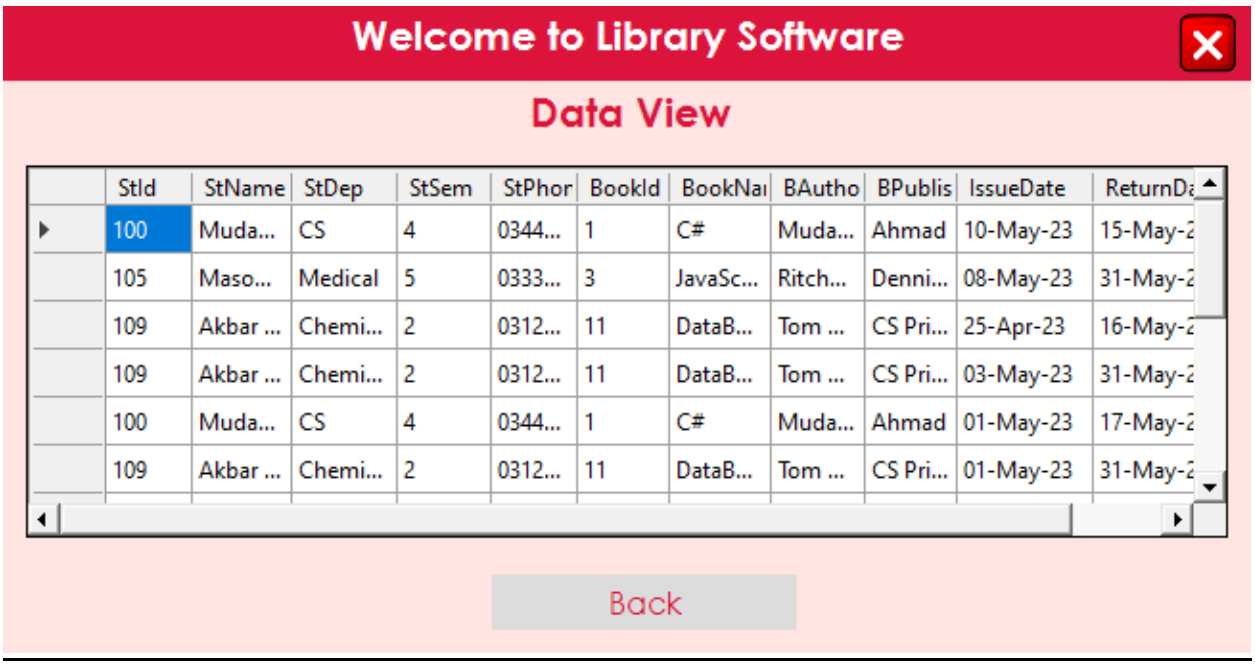
TRIGGER (LOG_CHECK_issued books history):



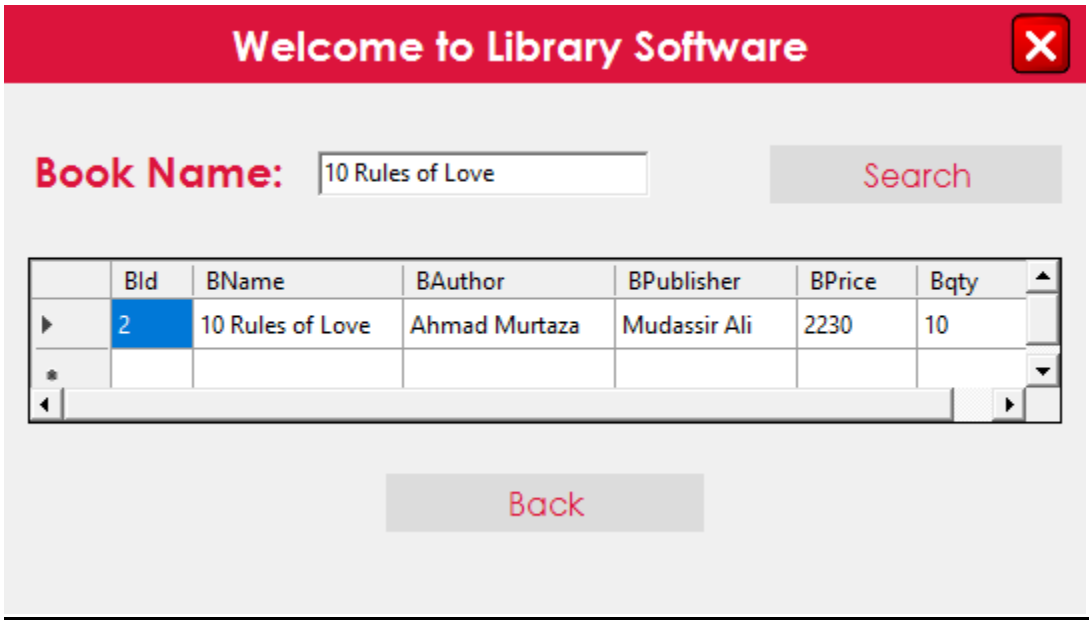
RETURNED BOOKS DETAIL:



VIEW (View_Check_Returned books history):



SEARCH BOOK:



NORMALIZATION:

Normalization is a process in database design that aims to eliminate data redundancy and improve data integrity by organizing data into well-structured relations (tables) based on a set of rules called normal forms. The goal of normalization is to minimize data duplication and ensure that each piece of data is stored in only one place, avoiding update anomalies and inconsistencies.

There are several normal forms in database normalization, each with its own set of rules. The commonly known normal forms are:

1. **First Normal Form (1NF):** Ensures that each column in a table contains only atomic (indivisible) values and there are no repeating groups of columns.
2. **Second Normal Form (2NF):** Builds upon 1NF and ensures that all non-key attributes in a table are functionally dependent on the entire primary key. It eliminates partial dependencies.
3. **Third Normal Form (3NF):** Builds upon 2NF and ensures that there are no transitive dependencies, where a non-key attribute is functionally dependent on another non-key attribute.

There are higher normal forms like Boyce-Codd Normal Form (BCNF), Fourth Normal Form (4NF), and Fifth Normal Form (5NF) that address more complex dependencies and further eliminate anomalies.

Normalization helps in organizing data efficiently, reducing data redundancy, maintaining data consistency, and improving database performance. It is an iterative process that involves analyzing the relationships and dependencies among data attributes to determine the appropriate normalization level for a given database schema.

ANALYZATION OF TABLES FOR NORMALIZATON:

1. **IssueTbl**:
    - The table has a primary key (**INum**) that uniquely identifies each row.
    - The columns (**StId**, **BookId**) are foreign keys referencing the primary keys of other tables (**StudentTbl** and **BookTbl**).
    - All the non-key attributes (**StName**, **BookName**, **IssueDate**) are directly dependent on the primary key.
    - Therefore, **IssueTbl** satisfies 3NF.
2. **BookTbl**:
    - The table has a primary key (**BId**) that uniquely identifies each row.
    - All the attributes (**BName**, **BAuthor**, **BPublisher**, **BPrice**, **Bqty**) are directly dependent on the primary key.
    - Therefore, **BookTbl** satisfies 3NF.
3. **LibrarianTbl**:
    - The table has a primary key (**LibId**) that uniquely identifies each row.
    - All the attributes (**LibName**, **LibPhone**, **LibPass**) are directly dependent on the primary key.
    - Therefore, **LibrarianTbl** satisfies 3NF.
4. **ReturnTbl**:
    - The table has a primary key (**RNum**) that uniquely identifies each row.
    - The columns (**StId**, **BookId**) are foreign keys referencing the primary keys of other tables (**StudentTbl** and **BookTbl**).
    - All the non-key attributes (**StName**, **BookName**, **IssueDate**, **ReturnDate**, **Fine**) are directly dependent on the primary key.
    - Therefore, **ReturnTbl** satisfies 3NF.
5. **StudentTbl**:
    - The table has a primary key (**StId**) that uniquely identifies each row.
    - All the attributes (**StName**, **StDep**, **StSem**, **StPhone**) are directly dependent on the primary key.
    - Therefore, **StudentTbl** satisfies 3NF.

In summary, all the given tables (**IssueTbl**, **BookTbl**, **LibrarianTbl**, **ReturnTbl**, **StudentTbl**) satisfy the requirements of the Third Normal Form (3NF).

ENTITY – RELATIONSHIP DIAGRAM