Library Management System

Project Documentation

1. Introduction

The **Library Management System** is a Windows Forms application built using **C# (.NET)** and **Microsoft SQL Server**. It provides basic library operations such as managing members, managing books, borrowing and returning books, and tracking borrowing/returning history. Additionally, the system demonstrates **file handling** by generating and viewing return receipts as text files.

This project is designed as a **practical learning project** to cover:

- Database connectivity with SQL Server
- CRUD operations (Create, Read, Update, Delete)
- Login authentication
- Windows Forms development
- File handling (read/write operations)

2. System Requirements

Software

- Windows 10 or higher
- Visual Studio (2022 recommended)
- .NET Desktop Development workload
- SQL Server Express + SQL Server Management Studio (SSMS)

Hardware

- Minimum 4 GB RAM
- 500 MB free disk space

3. Database Design

Database: LibraryDB

Tables

1. **Users**

- UserID (int, PK)
- Username (nvarchar)
- Password (nvarchar)
- Role (nvarchar)

Column Name	Data Type	Allow Nulls
UserID	int	☐
Username	nvarchar(50)	☐
Password	nvarchar(100)	☐
Role	nvarchar(20)	☑

2. **Members**

- MemberID (int, PK)
- FullName (nvarchar)
- Email (nvarchar)
- Phone (nvarchar)

Column Name	Data Type	Allow Nulls
MemberID	int	☐
FullName	nvarchar(100)	☑
Phone	nvarchar(30)	☑
Email	nvarchar(100)	☑

3. **Books**

- BookID (int, PK)
- Title (nvarchar)
- Author (nvarchar)
- Genre (nvarchar)
- YearPublished (int)
- Quantity (int)
- Available (bit)

Column Name	Data Type	Allow Nulls
BookID	int	☐
Title	nvarchar(200)	☑
Author	nvarchar(150)	☑
YearPublished	int	☑
Available	bit	☑
Genre	nvarchar(100)	☑

4. **BorrowHistory**

- BorrowID (int, PK)
- MemberID (FK → Members)
- BookID (FK → Books)
- BorrowDate (date)
- ReturnDate (date, nullable)
- IsReturned (bit)

Column Name	Data Type	Allow Nulls
BorrowID	int	☐
MemberID	int	☐
BookID	int	☐
BorrowDate	date	☐
ReturnDate	date	☑
Returned	bit	☑

4. Project Structure

- **LoginForm** – User authentication
- **DashboardForm** – Main navigation hub
- **ManageMembersForm** – Add/Edit/Delete/View library members

- **ManageBooksForm** – Add/Edit/Delete/View books and availability
- **BorrowBookForm** – Borrow a book (updates database, decreases stock)
- **ReturnBookForm** – Return a book (updates database, increases stock, generates receipt)
- **BorrowHistoryForm** – Shows full borrowing history
- **ReturnHistoryForm** – Shows all returned books history
- **ViewReceiptsForm** – Loads saved return receipts from text file

5. Functionalities

Login System

The system begins with a **login screen** where the user must enter a valid username and password. Credentials are verified against the Users table in the database. If authentication is successful, the user is redirected to the **Dashboard**, which serves as the main navigation hub of the application. Invalid login attempts prompt an error message, ensuring only authorized users can access the system.

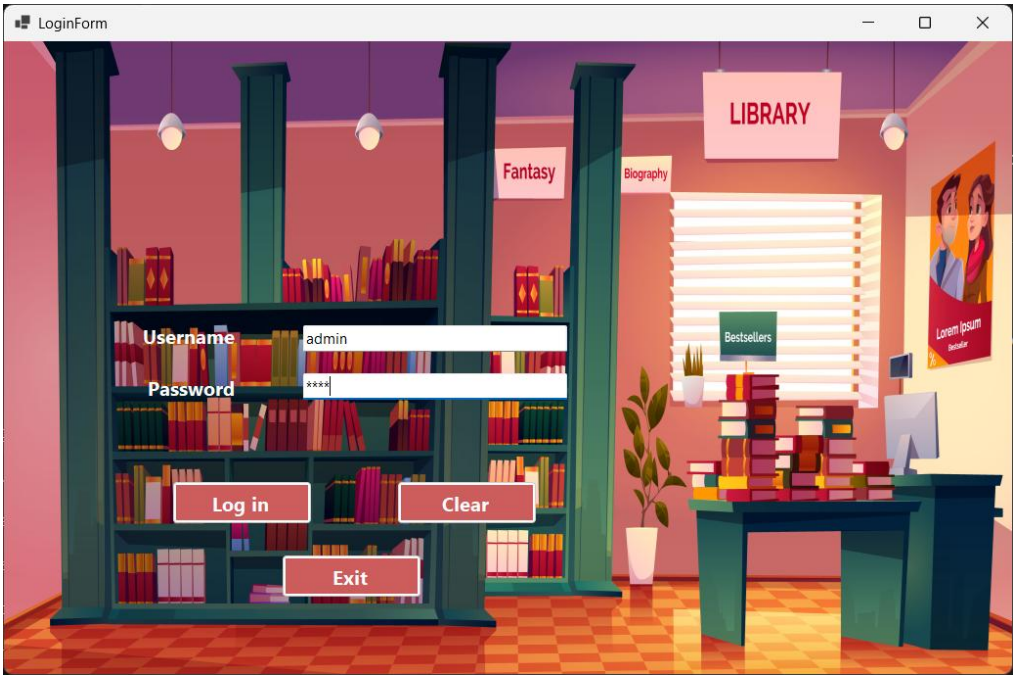

Figure 1: Login Form – User must enter valid credentials to access the dashboard.

Dashboard

The **Dashboard** acts as the central navigation hub of the Library Management System. Once a user logs in successfully, they are directed to the Dashboard. From here, the user can access all the main functionalities of the system through dedicated buttons. Options include **Manage Members**, **Manage Books**, **Borrow Books**, **Return Books**, **Borrowing History**, **Returning History**, **View Receipts**, and **Logout**. Each button opens the corresponding form while closing the dashboard, ensuring that only one form is active at a time. A **Back button** is provided in every form to return to the Dashboard.

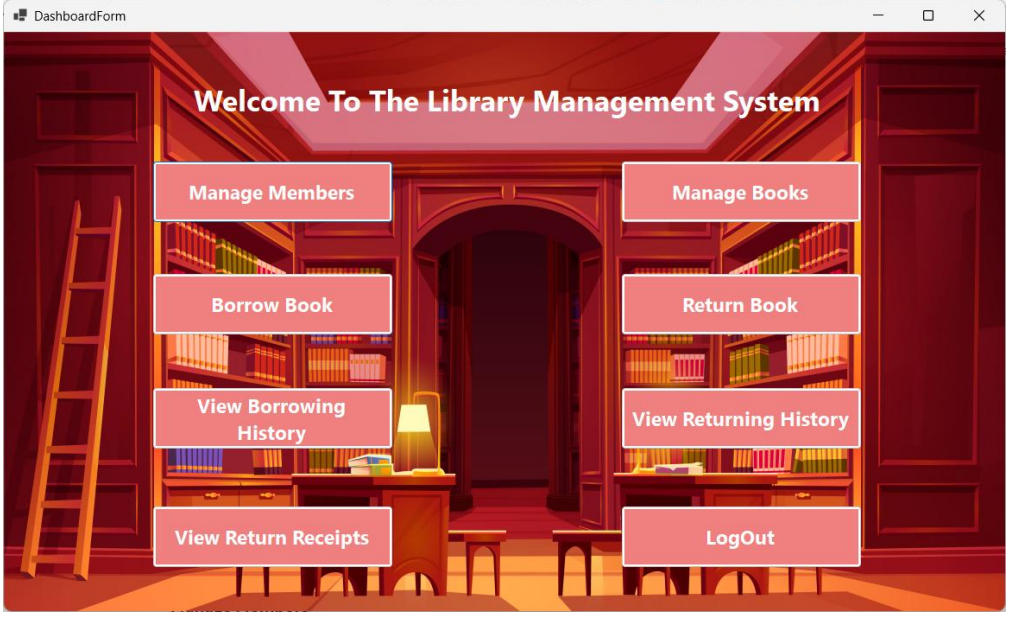

Figure 2: Dashboard Form – Main navigation hub with buttons to access different modules.

Manage Members

The **Manage Members** form allows the librarian to maintain a complete record of library members. The form provides options to **add new members** by entering details such as full name, email, and phone number. Existing member details can be **updated**, for example if a member changes their contact information. Members can also be **deleted** from the system when they are no longer active. All members are displayed in a

DataGridView, making it easy to browse and select members for editing or deletion.

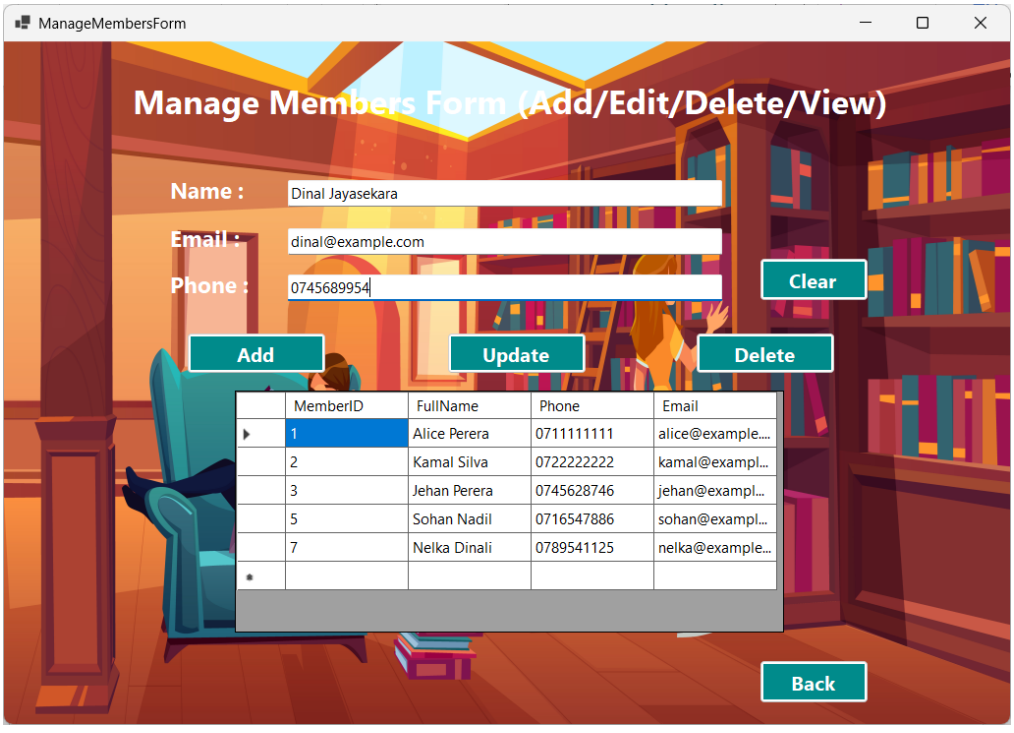

Figure 3: Manage Members Form – Members are displayed in a DataGridView with options to add, update, or delete.

Manage Books

The **Manage Books** form provides complete book management functionality. Librarians can **add new books** by specifying details such as title, author, genre, year of publication, quantity, and availability. Books can be **updated** or **deleted** from the system as needed. All books are listed in a **DataGridView**, which automatically refreshes to show the latest records. The availability of books is tracked based on the quantity in stock.

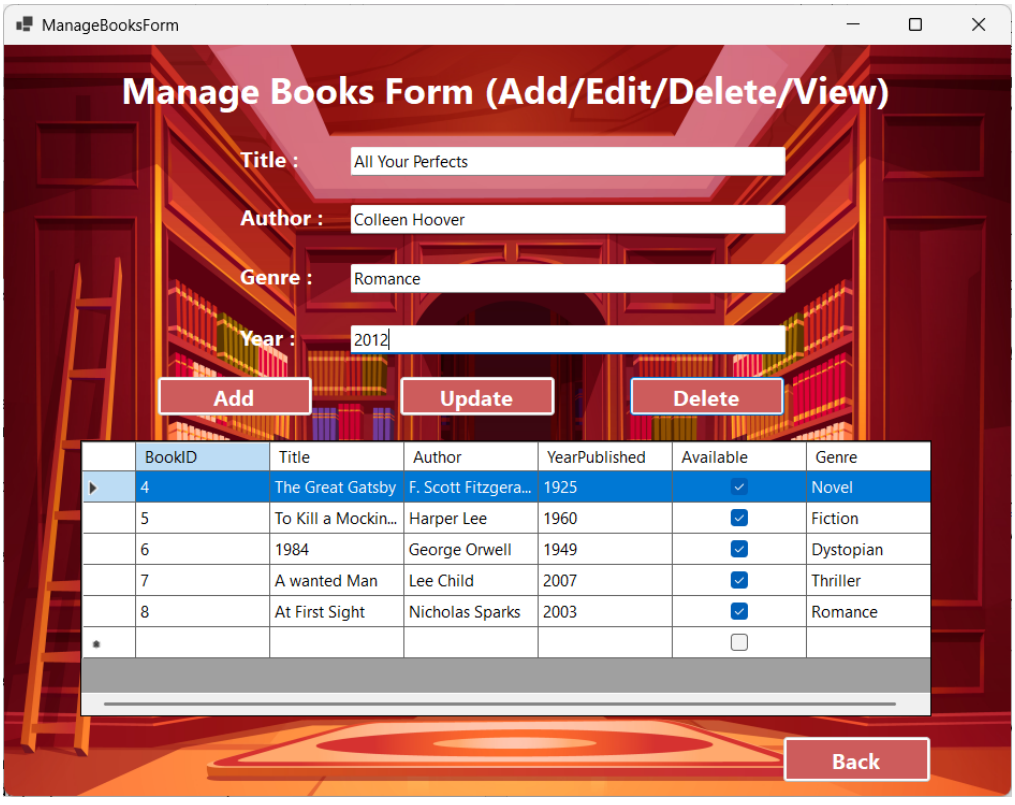

Figure 4: Manage Books Form – Books list with details and availability status.

Borrow Books

The **Borrow Books** form is used to manage lending transactions. The user can select a **member** and a **book** from dropdown lists that are automatically populated from the database. Once a book is borrowed, a record is created in the BorrowHistory table with the **borrow date**. At the same time, the quantity of the selected book in the Books table is automatically decreased. The form also displays a list of all currently borrowed books in a DataGridView for easy tracking.

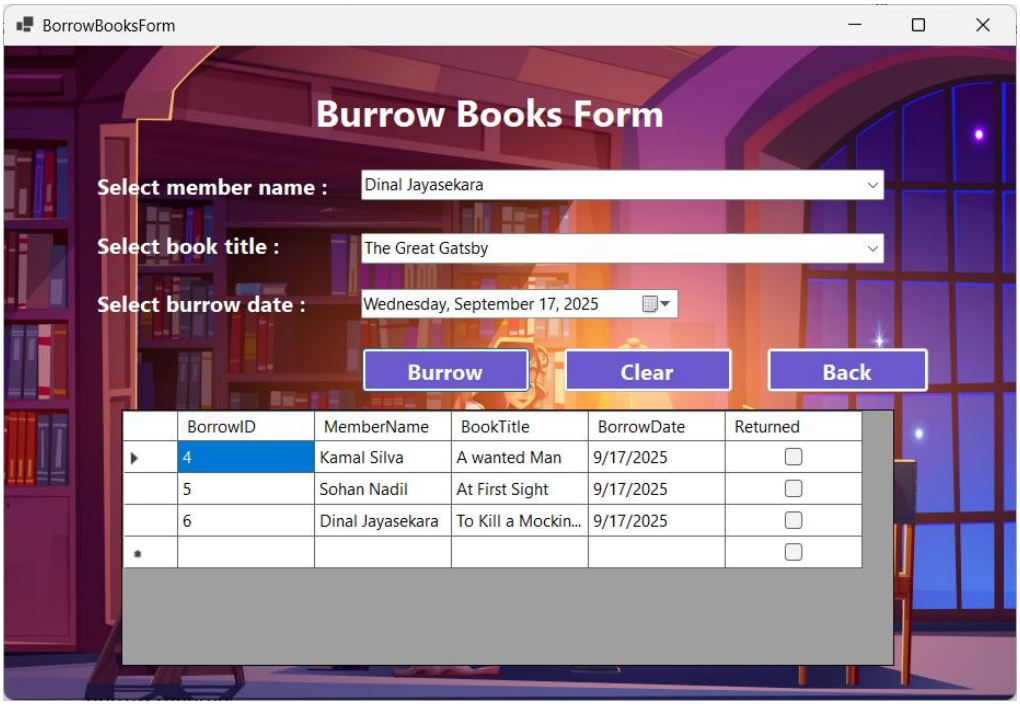

Figure 5: Borrow Books Form – Select a member and book to borrow, view ongoing borrow records.

Return Books

The **Return Books** form handles the returning process. The user selects a borrow record from a DataGridView that shows all borrowed books. Once returned, the system updates the BorrowHistory table by marking the book as **returned (IsReturned = 1)** and recording the **return date**. The quantity of the returned book is automatically increased in the Books table. Additionally, a **receipt text file** is generated, which includes details such as the member name, book title, borrow date, and return date.

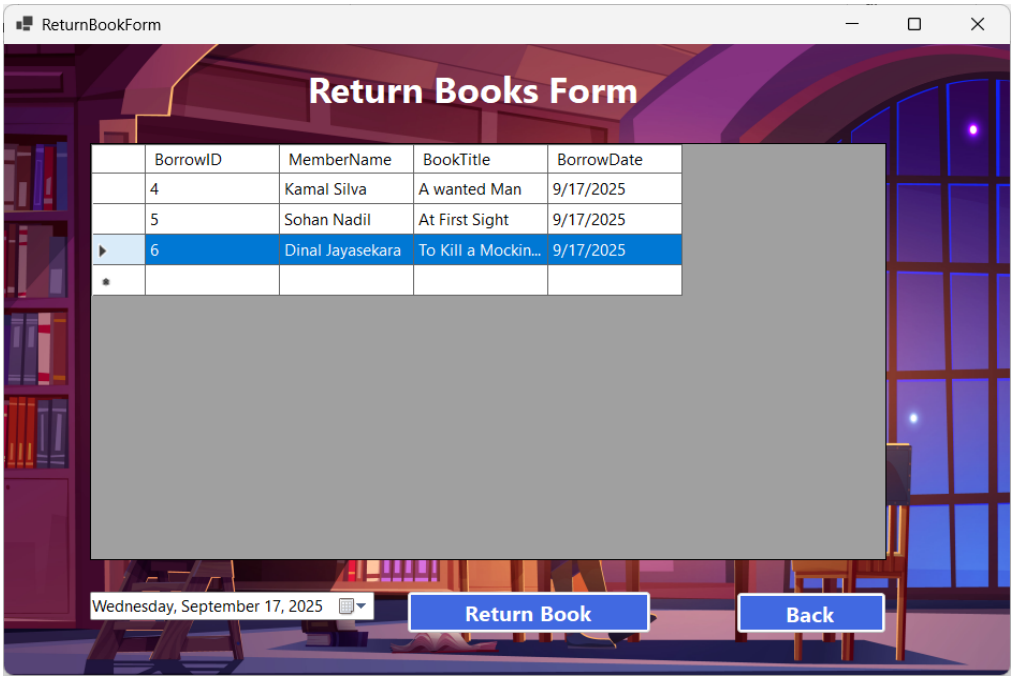

Figure 6: Return Books Form – Return borrowed books and generate text receipt.

Borrowing History

The **Borrowing History** form provides a complete record of all books borrowed. It lists details such as the member's name, book title, borrow date, and return date. The system also shows a column indicating whether the book has been returned (**Yes/No**). This helps librarians review both active and completed borrowing transactions in one place.

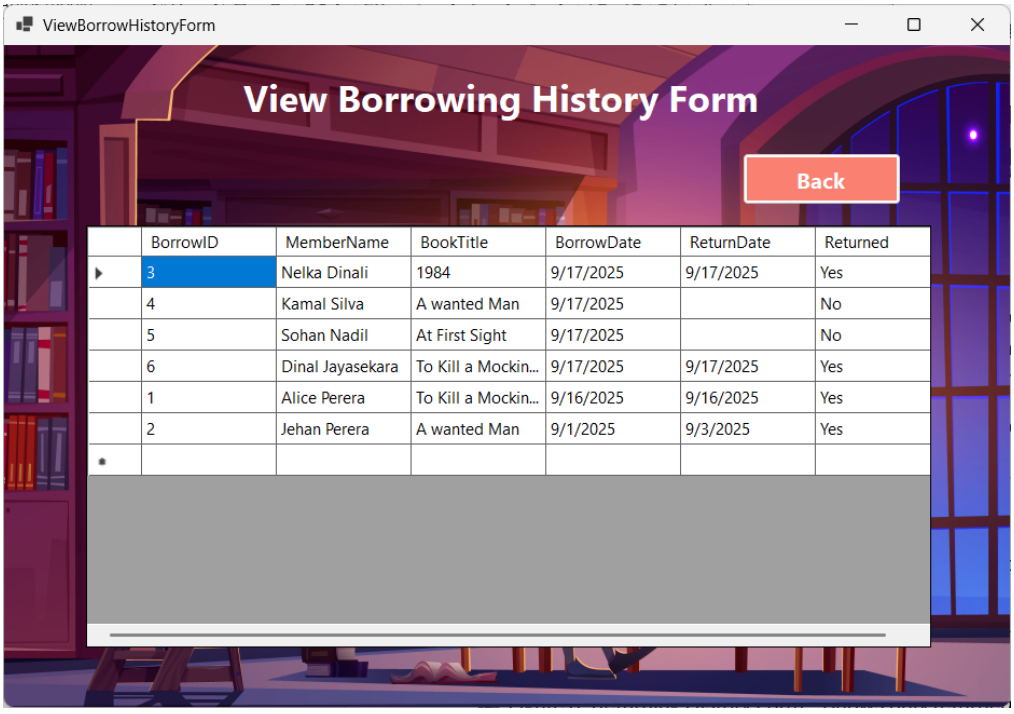

Figure 7: Borrowing History Form – Shows all borrow records with return status.

Returning History

The **Returning History** form focuses exclusively on records where books have been returned. It displays details such as the member name, book title, borrow date, and return date. This allows librarians to quickly review completed transactions and ensure all borrowed items were properly returned.

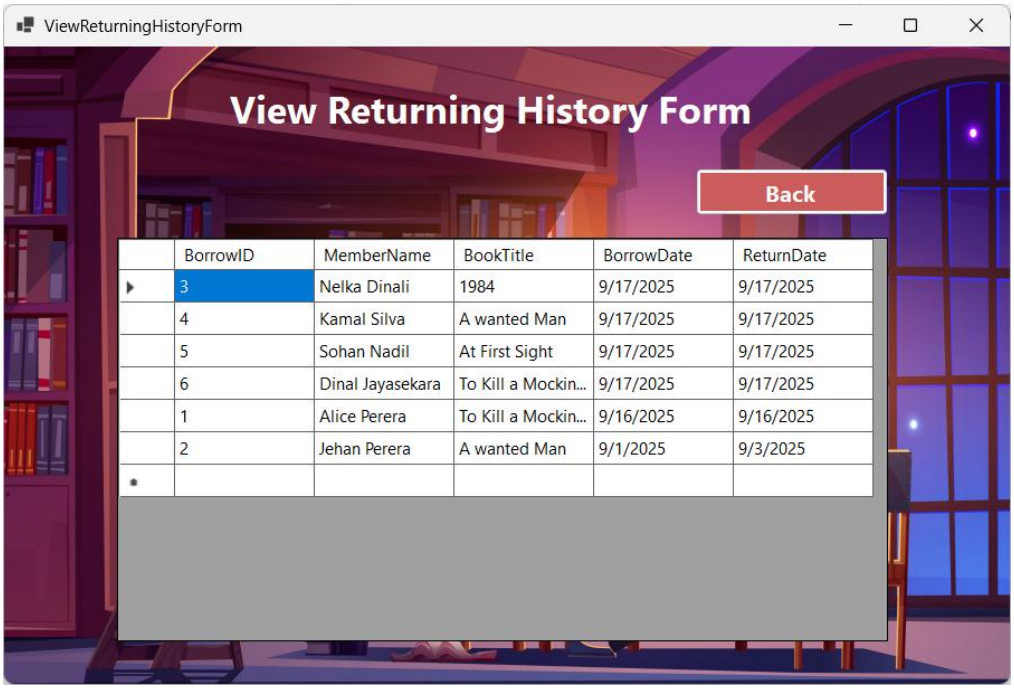

Figure 8: Returning History Form – Shows only returned books with return dates.

View Receipts

The **View Receipts** form allows librarians to view all return receipts generated by the system. Each receipt is stored as a **text file** in the receipts folder when a book is returned. The form reads the text files and displays their contents in a text box, making it easy to check or print the records.

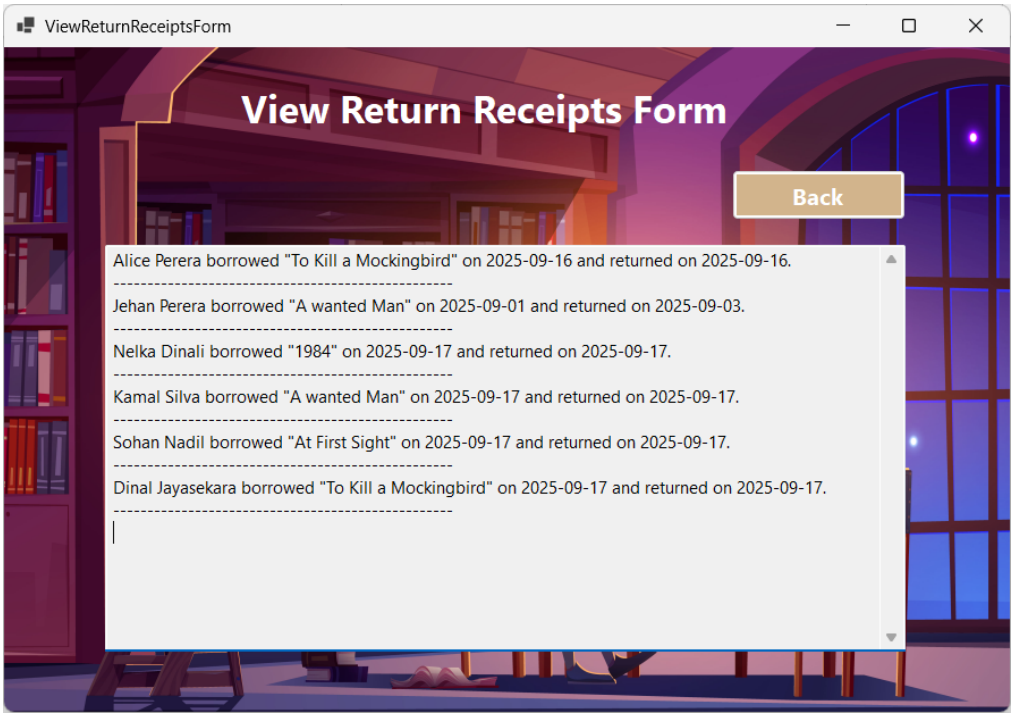

Figure 9: View Receipts Form – Displays all receipt text files saved during returns.

6. File Handling

- **Write Operation:**
  When a book is returned, a text file receipt is generated.
  Example content:

  ```
  John Smith borrowed "The Great Gatsby" on 2025-08-01 and returned on 2025-08-11.
  ```

- **Read Operation:**
  Receipts are later loaded and displayed in the **ViewReceiptsForm**.

- **Location:**
  All receipts are stored in:

  C:\Users\User\Desktop\LibraryManagement\Receipts\

7. Code Walkthrough

- **Database Connection**

```
string connectionString =
@"DataSource=localhost\SQLEXPRESS;InitialCatalog=Li
braryDB;Integrated Security=True";
```

- **Login Validation**

```
string query = "SELECT COUNT(*) FROM Users WHERE
Username=@u AND Password=@p";
```

- **Loading DataGridView**

```
SqlDataAdapter da = new SqlDataAdapter("SELECT *
FROM Members", con);

DataTable dt = new DataTable();

da.Fill(dt);

dgvMembers.DataSource = dt;
```

- **Receipt Generation (Return Book)**

```
string receiptText = $"{memberName} borrowed
\"{bookTitle}\" on {borrowDate:yyyy-MM-dd} and
returned on {returnDate:yyyy-MM-dd}.";

File.WriteAllText(filePath, receiptText);
```

8. Testing & Validation

- **Login Form:** tested with admin account
- **Members & Books:** verified CRUD operations
- **Borrow/Return Books:** tested with multiple records
- **Histories:** confirmed filtering works correctly
- **Receipts:** confirmed text file is generated and readable

Common Issues Solved:

- Wrong column names in SQL queries → fixed
- DataGridView not loading → ensured Form_Load events were wired properly
- File handling errors → used correct folder path

9. Conclusion

This Library Management System demonstrates how to:

- Connect C# WinForms to SQL Server
- Perform CRUD operations
- Manage relationships between members, books, and borrow history
- Implement authentication
- Use file handling for receipts

Future Improvements:

- Search and filter options
- User role management (Admin vs. Librarian)
- Reports with charts (borrow trends)
- Export to PDF/Excel