Faculty of Computing

Documentation For Application Development

London Metropolitan University,
Faculty of Computing
CS6004ES Application Development

Module: CS6004ES **Setter: P. K. Vimarshana**

Title: SEP 2024 - AD - CW 01

Student ID: 00238397

Introduction

A desktop program called the BookHaven Management System was created to assist sales clerks and bookstore administrators in managing users, customers, suppliers, book inventory, sales, orders, and reporting, among other areas of bookstore operations.

Key Features and Scope

Overview

The BookHaven Management System is a desktop C# application built on Windows Forms that offers a complete, effective, and expandable solution for bookstore operations. The system incorporates key features, such as user, customer, and supplier management, book inventory control, sales tracking, order processing, and reporting, and is intended for bookstore administrators and sales clerks. By simplifying these procedures, BookHaven lowers the amount of manual labor required, boosts operational effectiveness, and guarantees safe, orderly bookstore administration. Businesses can efficiently manage books, clients, and suppliers thanks to the system's smooth inventory tracking, real-time stock updates, and automated sales processing. Strong reporting and analytics features also support well-informed decision-making by offering insightful information on customer preferences, sales trends, and inventory levels. BookHaven's scalability, role-based access control, and user-friendly interface make it ideal for long-term

Objectives

- ☐ **Provide a user-friendly bookstore management system.**
- ☐ **Improve inventory tracking and order processing.**
- ☐ **Automate sales transactions and customer management.**
- ☐ **Generate detailed reports for business insights.**
- ☐ **Ensure security with role-based access and data protection.**

Functional Scope

The system includes the following core features:

- ☐ Authentication and Authorization: – Secure authentication with role based access (admin, sales clerk).
    - ☐ Users must log in to access the system.
    - ☐ Role-based access control ensures that only authorized users can perform specific actions.
- ☐ Book Inventory: – Adding, updating, and delete books information (title, author, genre, ISBN, price, stock).
    - ☐ Admins can manage the bookstore's inventory (add, update, delete books).
    - ☐ Books are categorized by genre, author, and ISBN.

- [ ] Customer Management: – Manage customer profiles, contact details, and purchase history.
    - [ ] Admins and sales clerks can manage customer information (add, update, delete).
    - [ ] Customers are associated with sales transactions.
- [ ] Sales Management: – Point-of-sale (POS) module to process purchases, apply discounts, and generate receipts.
    - [ ] Admins and sales clerks can manage sales transactions.
    - [ ] Sales Data. Details like customer, total amount, discount and sale date.
- [ ] Order Management: – Track orders.
    - [ ] Admins can manage book orders from suppliers.
    - [ ] The Orders includes order date, total amount and status (Pending, Completed).
- [ ] Supplier Management: – Maintain supplier details.
    - [ ] Admins can manage supplier details (add, update, delete).
    - [ ] Suppliers are linked to book orders.
- [ ] Dashboard: – View overall bookstore performance metrics.
    - [ ] Provides an overview of key metrics (e.g., total sales, orders, and inventory).
    - [ ] Includes charts for visualization of data (monthly sales, revenue by genre).
- [ ] Reporting & Analytics: – Generate reports on sales and orders.
    - [ ] Generate reports for sales and orders.
    - [ ] Reports can include things like monthly sales, best-selling books and genre-specific revenue.
- [ ] Security & Data Protection: Role-based access, data encryption and validation.

Out of Scope

- [ ] Online Integration:
    - [ ] The system does not include e-commerce functionality or integration with online platforms.
- [ ] Advanced Analytics:
    - [ ] system provides basic reporting, no advanced data analytics or machine learning capabilities.
- [ ] Multi-User Concurrency:
    - [ ] cannot support concurrent access by more than one user.
- [ ] Mobile Support:
    - [ ] The system is for desktop use only, and not for mobile devices.

Target Users

- [ ] Admin:

- [ ] Full access to all components (user management, customer management, supplier management, book inventory, orders, sales and reporting).
- [ ] Sales Clerk:
  - [ ] Limited access to customer management, sales management, and basic reporting.

Technology Stack

- [ ] Programming Language: C#
- [ ] Framework: .NET Windows Forms
- [ ] Database: (SQL Server)
- [ ] Charts: LiveCharts for data visualization
- [ ] Logging: Basic logging for error tracking
- [ ] Development Tools: Visual Studio 2022
- [ ] Architecture: Multi-layered (Data Layer, Business Logic, Presentation Layer)

Deliverables

- [ ] Functional Desktop Application:
  - [ ] A fully functional application with a user-friendly interface for managing bookstore operations.
- [ ] Documentation:
  - [ ] UML diagrams (class diagrams, use case diagrams).
  - [ ] User manuals for admins and sales clerks.
- [ ] Source Code:
  - [ ] Well-structured, modular, and documented code adhering to best practices.
  - [ ] Repository Link: **https://github.com/kasunvimarshana/BookHaven**

Installation Guide

System Requirements

Before installing BookHaven, ensure your system meets the following requirements:

- [ ] Operating System: Windows 10 or later (64-bit recommended)
- [ ] Processor: Intel Core i3 or higher
- [ ] RAM: 4GB minimum (8GB recommended)
- [ ] Disk Space: At least 500MB of free space
- [ ] Framework: .NET 7.0 or later
- [ ] Database: SQL Server (or compatible database)
- [ ] Screen Resolution: 1024x768 or higher

Installation Steps

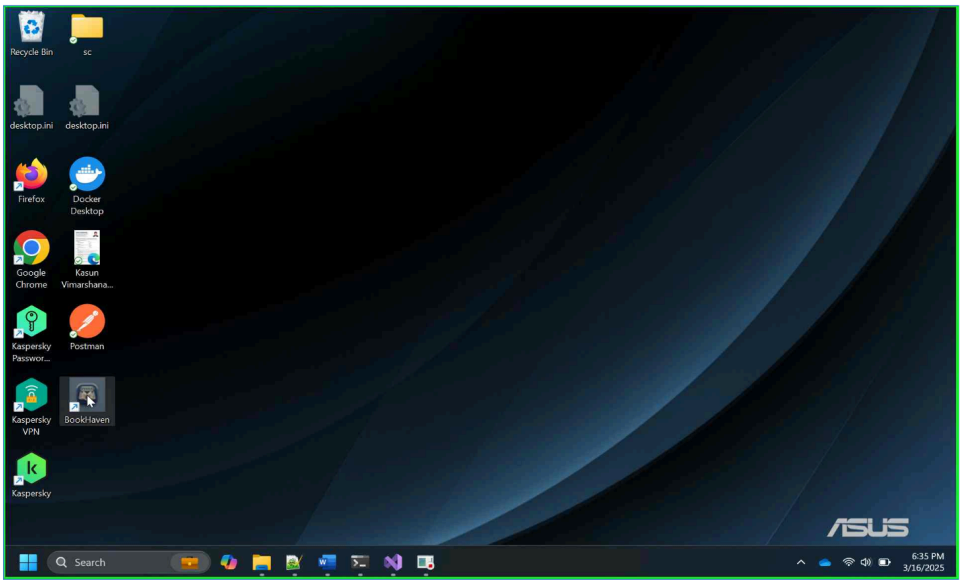
Follow these steps to install BookHaven on your computer:

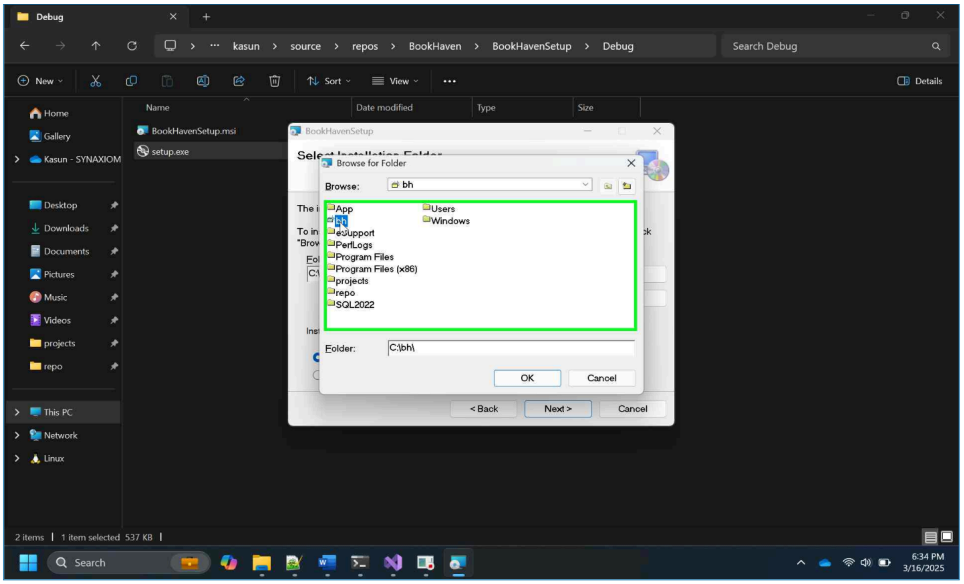
- ☐ Locate the Setup File
  - ☐ Download the BookHavenSetup.exe file from the official source.
  - ☐ Navigate to the folder where the setup file is saved.
- ☐ Start the Installation
  - ☐ Double-click on BookHavenSetup.exe to launch the installer.
  - ☐ The installer welcome screen will appear.
  - ☐ Click on Next to proceed.
- ☐ Accept the License Agreement
  - ☐ Read the end-user license agreement carefully.
  - ☐ If you agree to the terms, choose " I Agree " and " Next ".
  - ☐ If you do not agree, click Cancel to exit the installation.
- ☐ Select Installation Scope And Installation Location
  - ☐ Choose one of the following options:
    - ☐ Everyone (Install for all users on the system)
    - ☐ Just Me (Install only for the current user)
  - ☐ By default, BookHaven will install in C:\Program Files\BookHaven.
  - ☐ To select the new location for the installation place, click Browse and choose the new folder.
  - ☐ Click Next to continue.
- ☐ Begin Installation
  - ☐ Review your selected options.
  - ☐ Click Next to start the installation process.
- ☐ Complete Installation
  - ☐ Once the installation is complete, a confirmation message will appear.
  - ☐ Click Close to close the installer.
  - ☐ If prompted, restart your computer to apply changes.

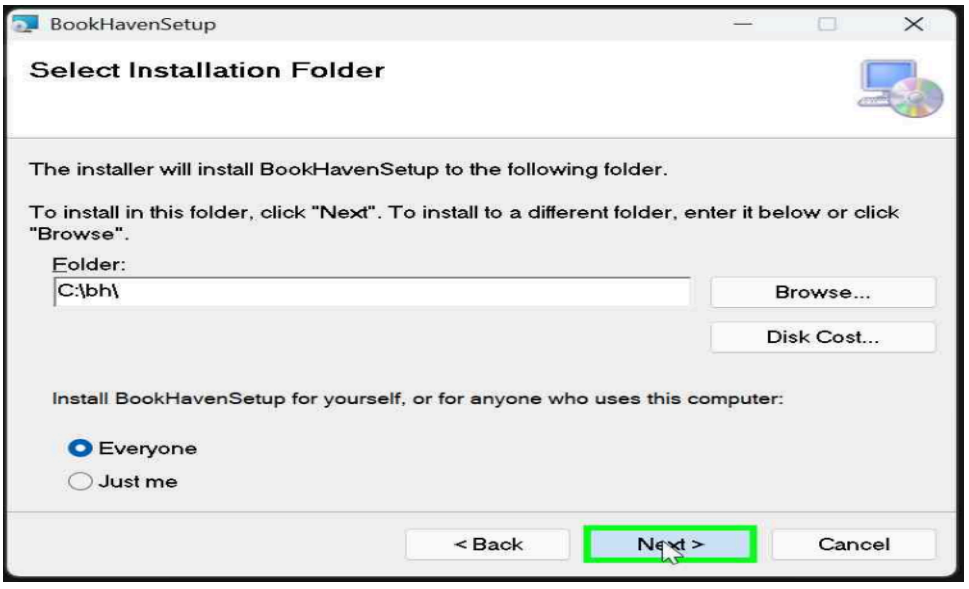
Post Installation Steps

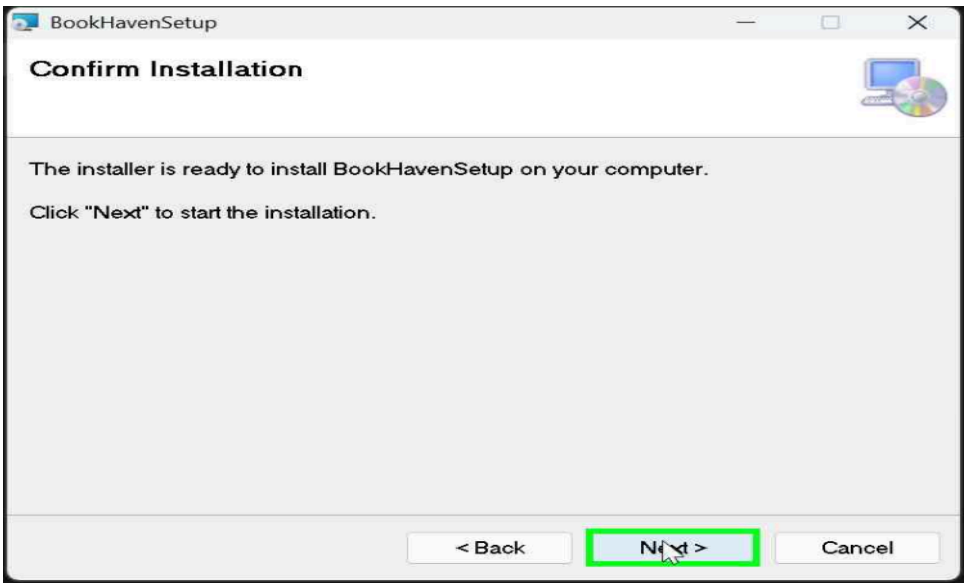
- ☐ Launch BookHaven from the Start Menu or Desktop shortcut.
- ☐ Log in using the default user credentials:
  - ☐ Username: admin
  - ☐ Password: admin

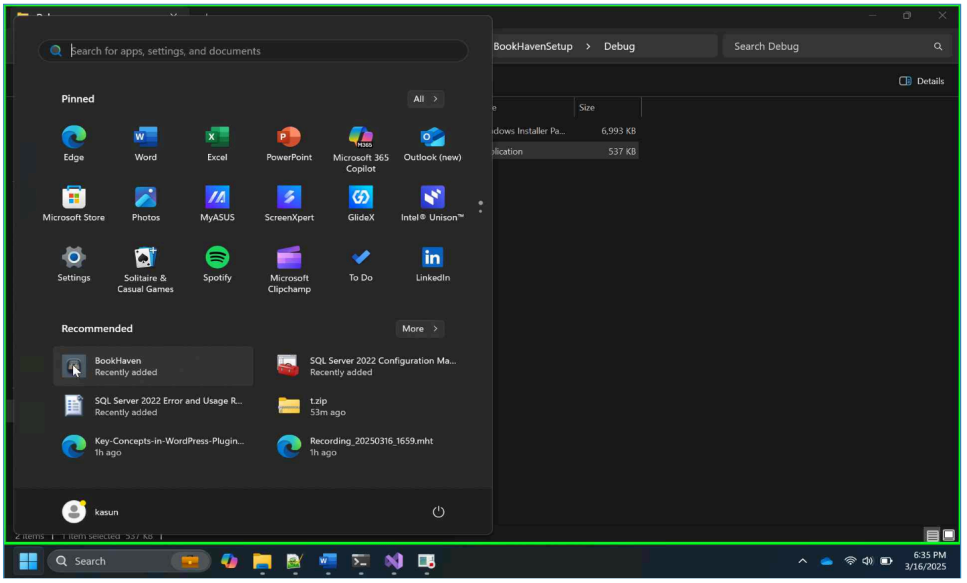
For troubleshooting or further assistance, refer to the User Manual or contact support.


Recycle Bin
sc
desktop.ini
desktop.ini
Firefox
Docker Desktop
Google Chrome
Kasun Vimarshana...
Kaspersky Passwor...
Postman
Kaspersky VPN
BookHaven
Kaspersky
Search
6:35 PM
3/16/2025


Debug
kasun
source
repos
BookHaven
BookHavenSetup
Debug
Search Debug
New
Sort
View
Details
Name
Date modified
Type
Size
Home
Gallery
Kasun - SYNAXIOM
Desktop
Downloads
Documents
Pictures
Music
Videos
projects
repo
This PC
Network
Linux
BookHavenSetup.msi
setup.exe
BookHavenSetup
Browse for Folder
Browse:
bh
App
Users
Windows
PerfLogs
Program Files
Program Files (x86)
projects
repo
SQL2022
Folder:
C:\bh\
OK
Cancel
< Back
Next >
Cancel
2 items
1 item selected 537 KB
Search
6:34 PM
3/16/2025


BookHavenSetup
Select Installation Folder
The installer will install BookHavenSetup to the following folder.
To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".
Folder:
C:\bh\
Browse...
Disk Cost...
Install BookHavenSetup for yourself, or for anyone who uses this computer:
Everyone
Just me
< Back
Next >
Cancel


BookHavenSetup
Confirm Installation
The installer is ready to install BookHavenSetup on your computer.
Click "Next" to start the installation.
< Back
Next >
Cancel

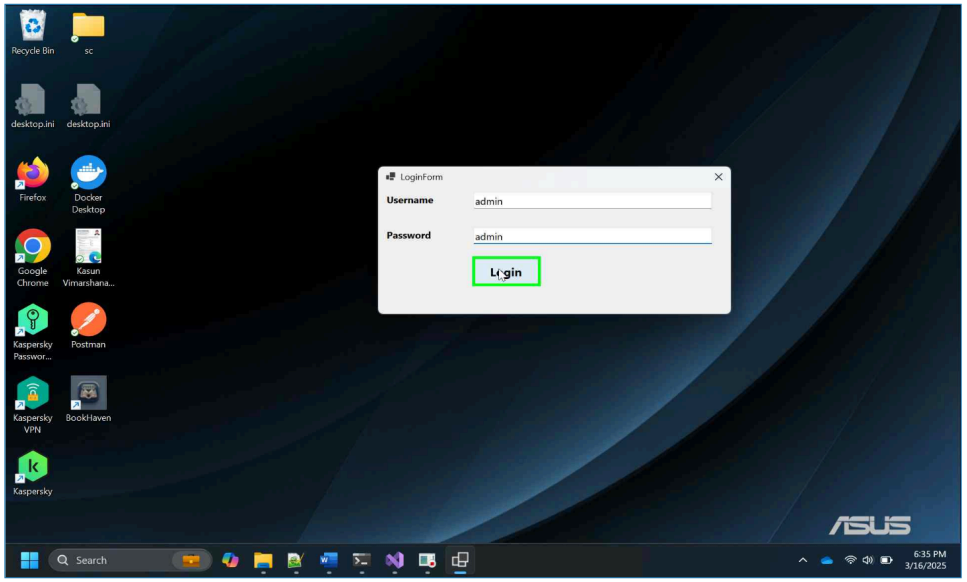

Search for apps, settings, and documents
Pinned
All
Edge
Word
Excel
PowerPoint
Microsoft 365 Copilot
Outlook (new)
Microsoft Store
Photos
MyASUS
ScreenXpert
GlideX
Intel® Unison™
Settings
Solitaire & Casual Games
Spotify
Microsoft Clipchamp
To Do
LinkedIn
Recommended
More
BookHaven
Recently added
SQL Server 2022 Configuration Ma...
Recently added
SQL Server 2022 Error and Usage R...
Recently added
t.zip
53m ago
Key-Concepts-in-WordPress-Plugin...
1h ago
Recording_20250316_1659.mht
1h ago
kasun
BookHavenSetup
Debug
Search Debug
Details
Size
6,993 KB
537 KB
Search
6:35 PM
3/16/2025

User Manual for BookHaven Management System

This user manual provides step-by-step instructions on how to use the system effectively.

Getting Started

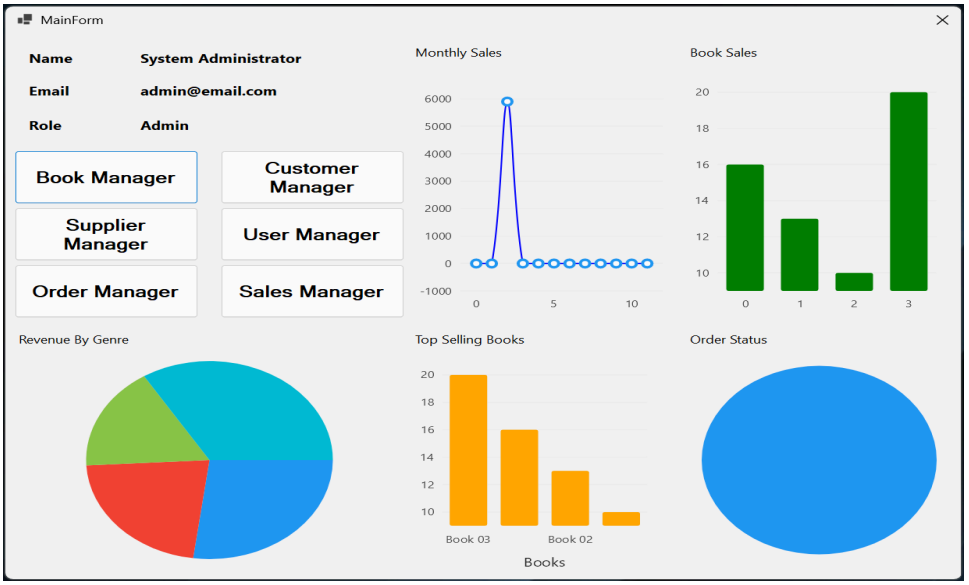
1) **Login**
   - ☐ Launch the application.
   - ☐ Enter your Username and Password.
     - ☐ Default Username: admin
     - ☐ Default Password: admin
   - ☐ Click Login.
     - ☐ Admins have access to all features.
     - ☐ Sales Clerks have limited access (customer management, sales and some reporting).

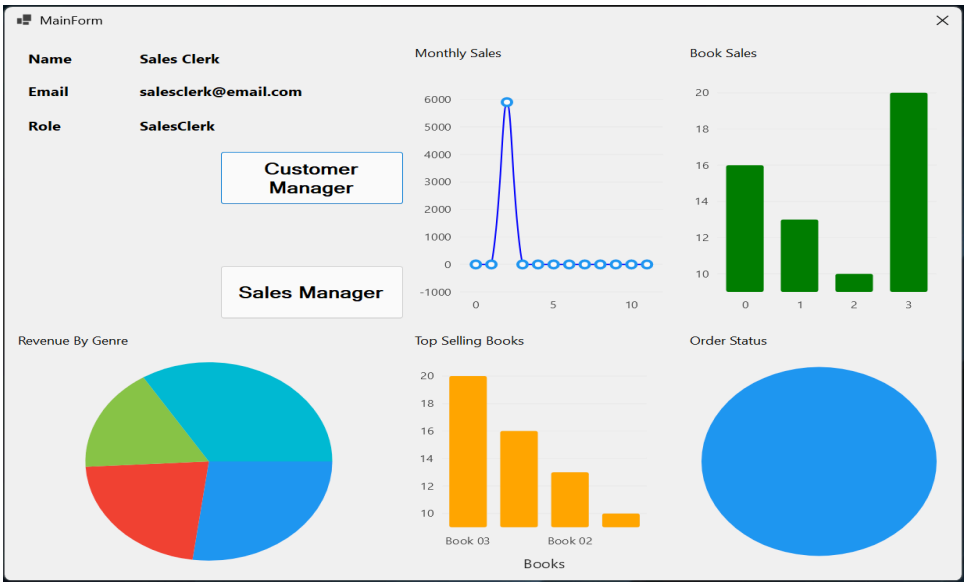


Main Features

2) Dashboard

- ☐ Purpose: Provides an overview of metric and visualizations.
- ☐ Features:
  - ☐ View total sales, orders, and inventory.
  - ☐ Visualize data using charts (e.g., monthly sales, revenue by genre).



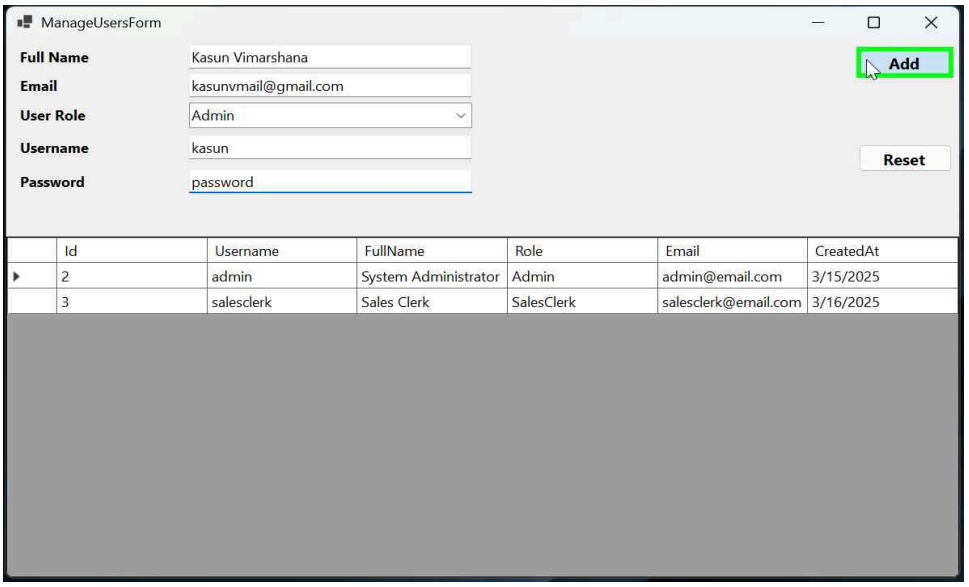


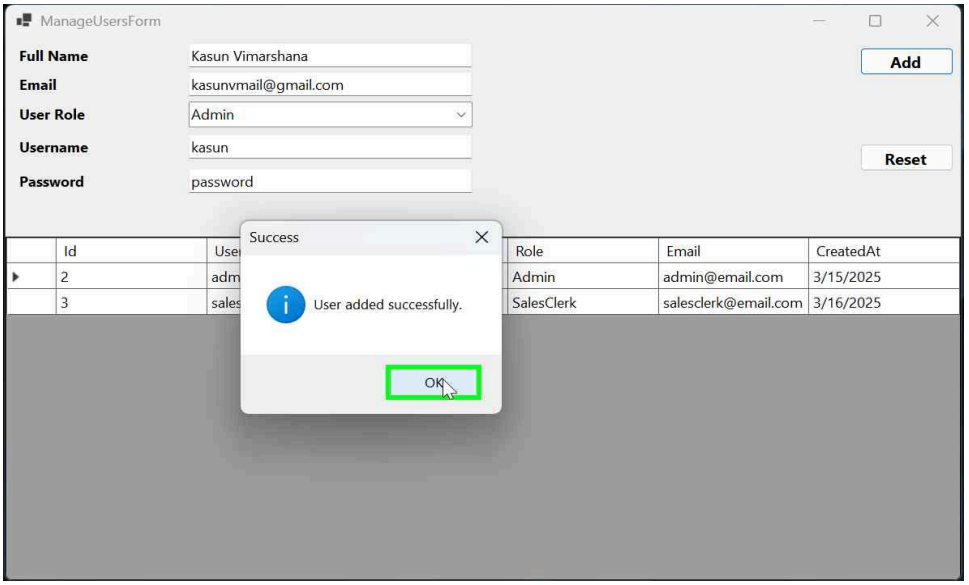
3) User Management

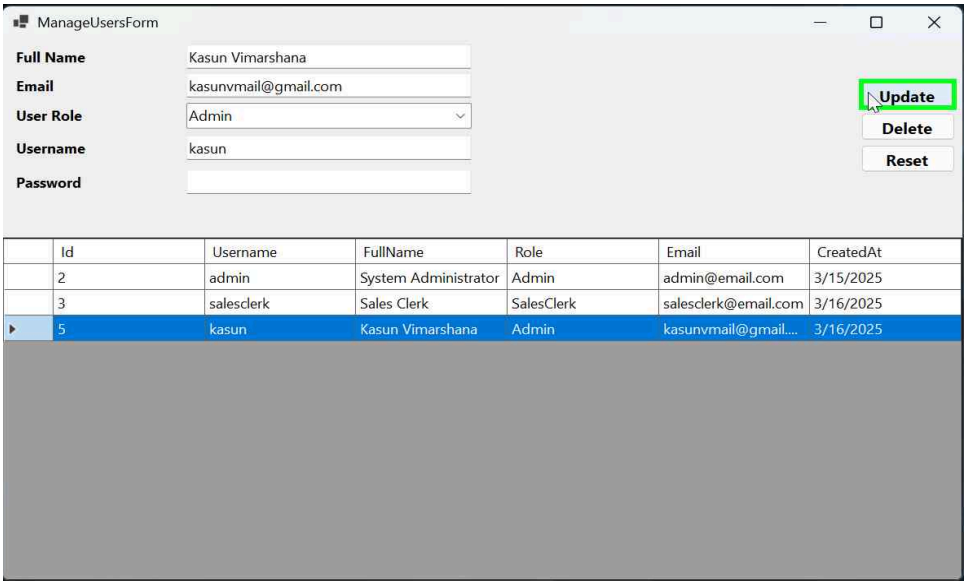
- ☐ Access: Admin only.
- ☐ Purpose: Manage user accounts (create, update, delete).
- ☐ Steps:
  - ☐ Navigate to User Management from the main menu.
  - ☐ To add a user:

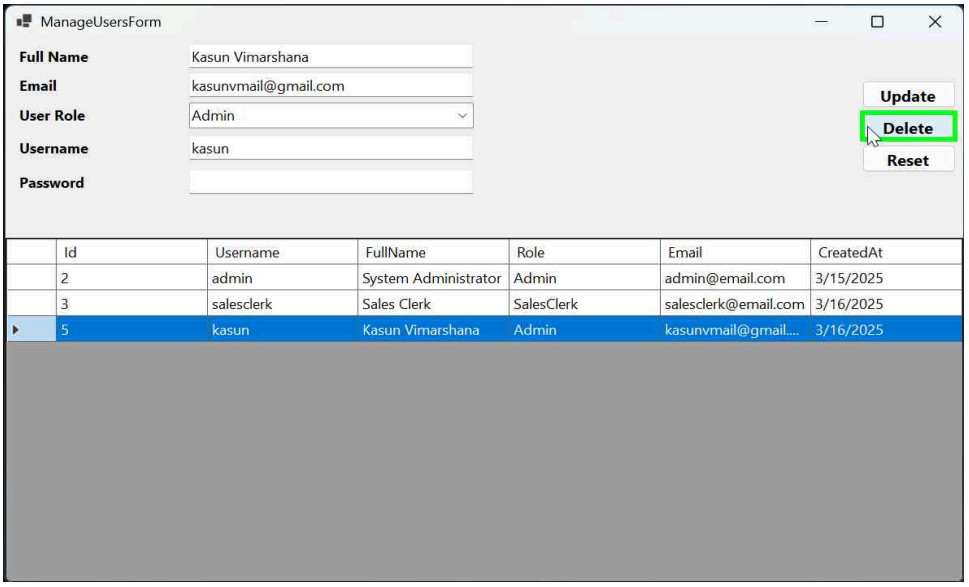
- [ ] Fill details (Username, Password, Full Name, Email, Role).
- [ ] Click Save.

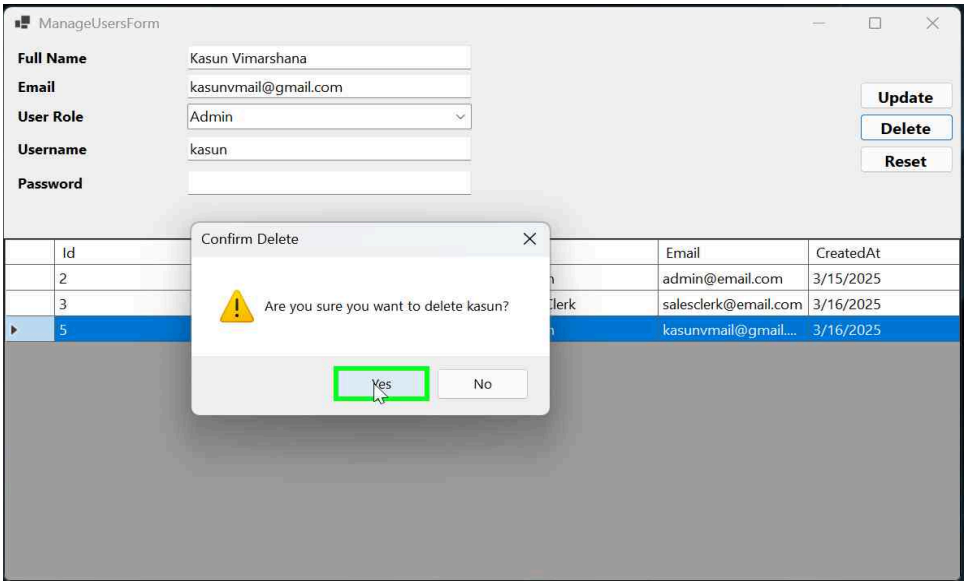
- [ ] To update a user:
  - [ ] Select user from list
  - [ ] Modify details.
  - [ ] Click Update.
- [ ] To delete a user:
  - [ ] Select user from list.
  - [ ] Click Delete.
  - [ ] Confirm deletion.

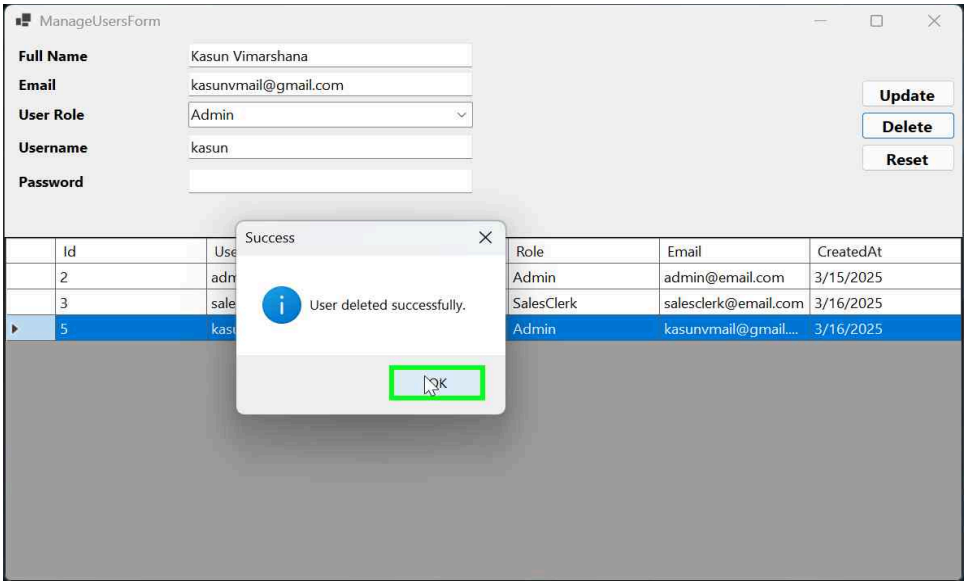



ManageUsersForm
Full Name
Kasun Vimarshana
Email
kasunvmail@gmail.com
User Role
Admin
Username
kasun
Password
password
Add
Reset
Success
User added successfully.
OK
Id
Role
Email
CreatedAt
2
Admin
admin@email.com
3/15/2025
3
SalesClerk
salesclerk@email.com
3/16/2025


ManageUsersForm
Full Name
Kasun Vimarshana
Email
kasunvmail@gmail.com
User Role
Admin
Username
kasun
Password
Update
Delete
Reset
Id
Username
FullName
Role
Email
CreatedAt
2
admin
System Administrator
Admin
admin@email.com
3/15/2025
3
salesclerk
Sales Clerk
SalesClerk
salesclerk@email.com
3/16/2025
5
kasun
Kasun Vimarshana
Admin
kasunvmail@gmail....
3/16/2025


ManageUsersForm
Full Name
Kasun Vimarshana
Email
kasunvmail@gmail.com
User Role
Admin
Username
kasun
Password
Update
Delete
Reset
Id
Username
FullName
Role
Email
CreatedAt
2
admin
System Administrator
Admin
admin@email.com
3/15/2025
3
salesclerk
Sales Clerk
SalesClerk
salesclerk@email.com
3/16/2025
5
kasun
Kasun Vimarshana
Admin
kasunvmail@gmail....
3/16/2025

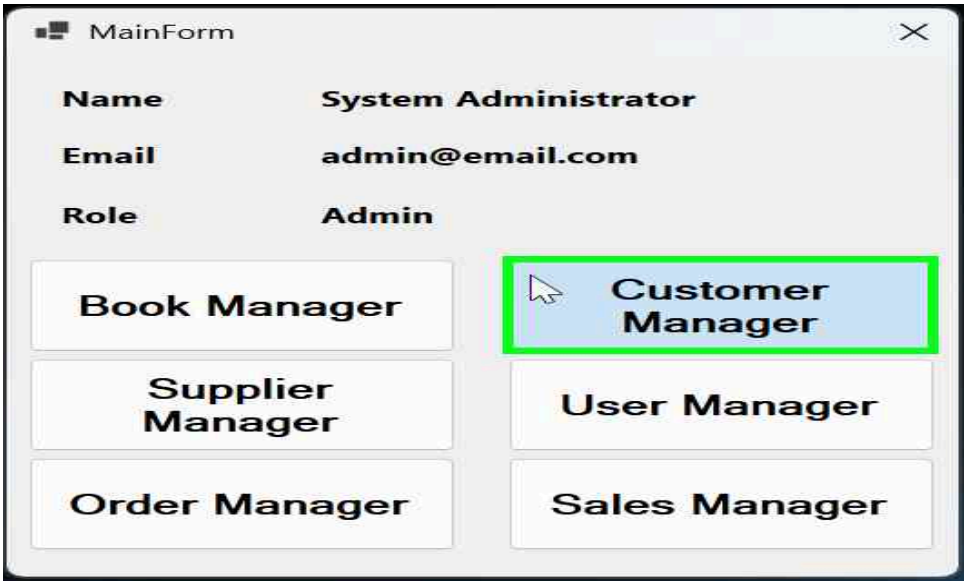


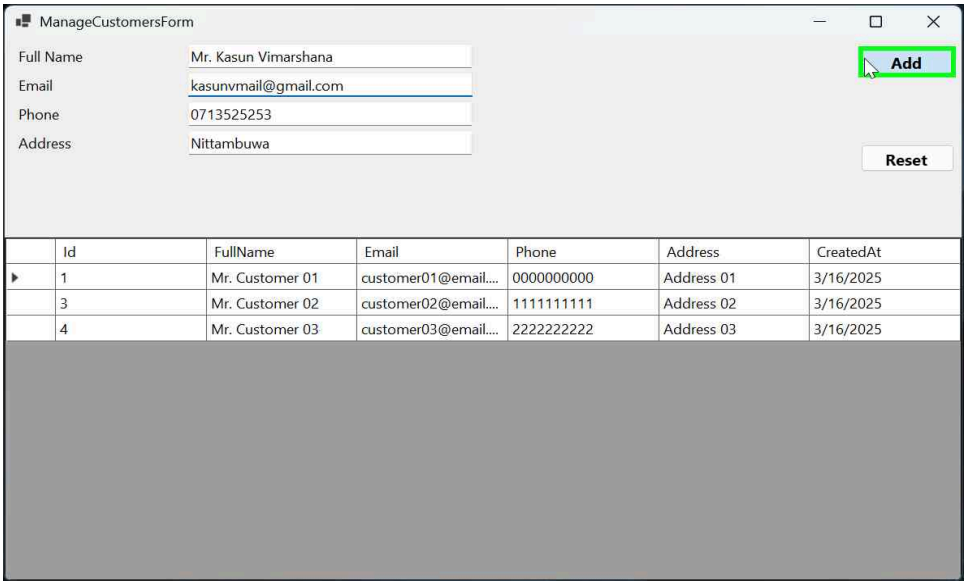


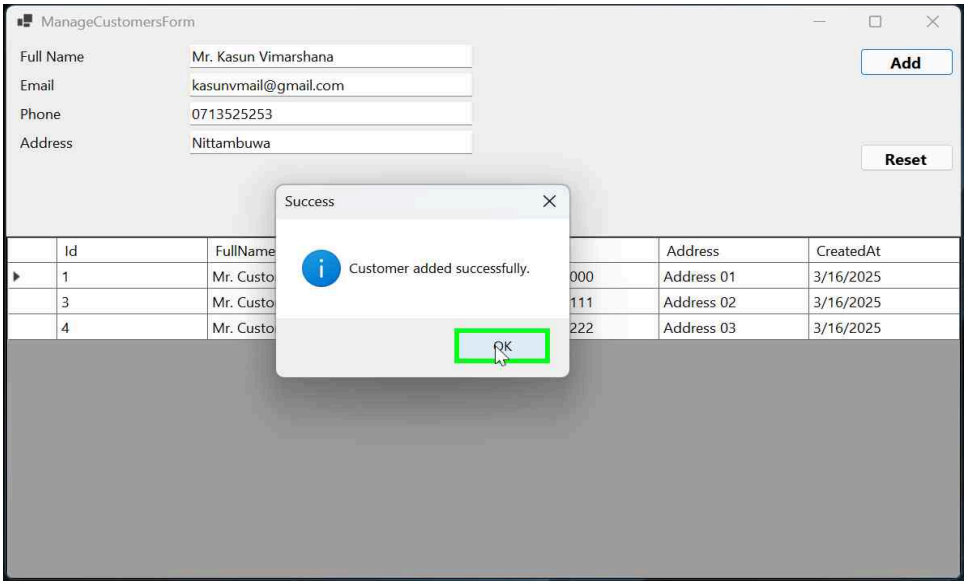
4) Customer Management

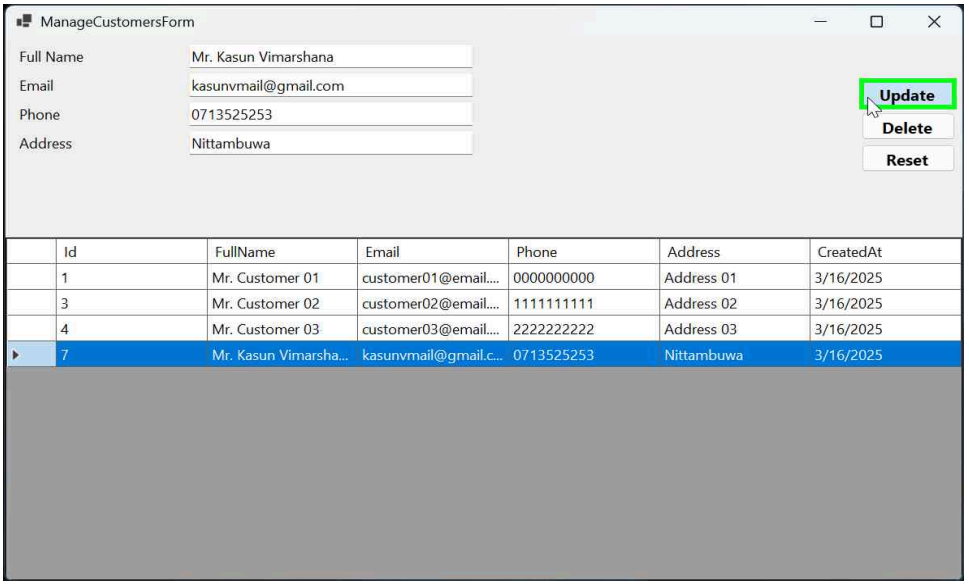
- ☐ Access: Admin and Sales Clerk.
- ☐ Purpose: Manage customer information.
- ☐ Steps:
  - ☐ Navigate to Customer Management from the main menu.
  - ☐ To add customer:
    - ☐ Fill details (Full Name, Email, Phone, Address).
    - ☐ Click Save.
  - ☐ To update a customer:
    - ☐ Select customer from list.
    - ☐ Modify details.
    - ☐ Click Update.
  - ☐ To delete a customer:
    - ☐ Select customer from list.
    - ☐ Click Delete.

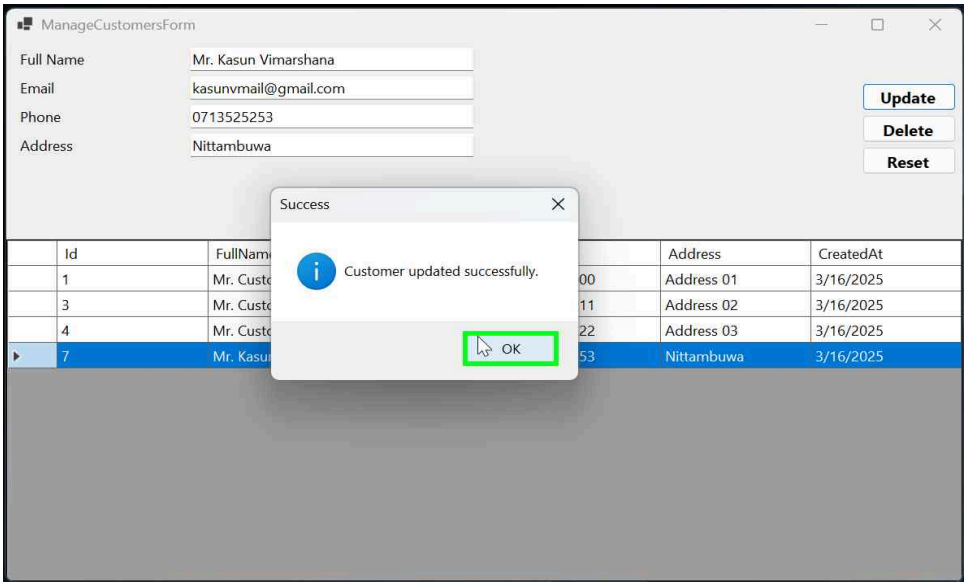
☐ Confirm deletion.

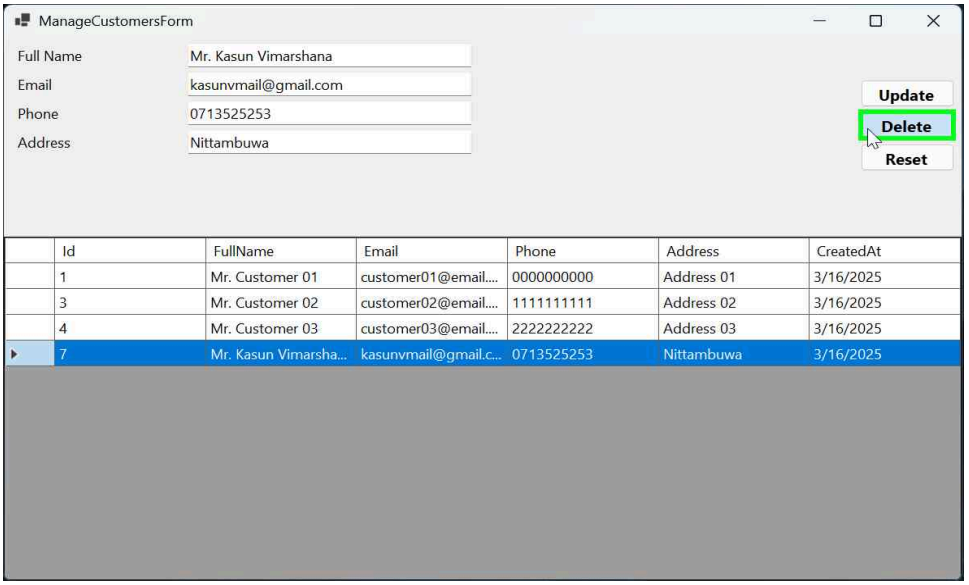


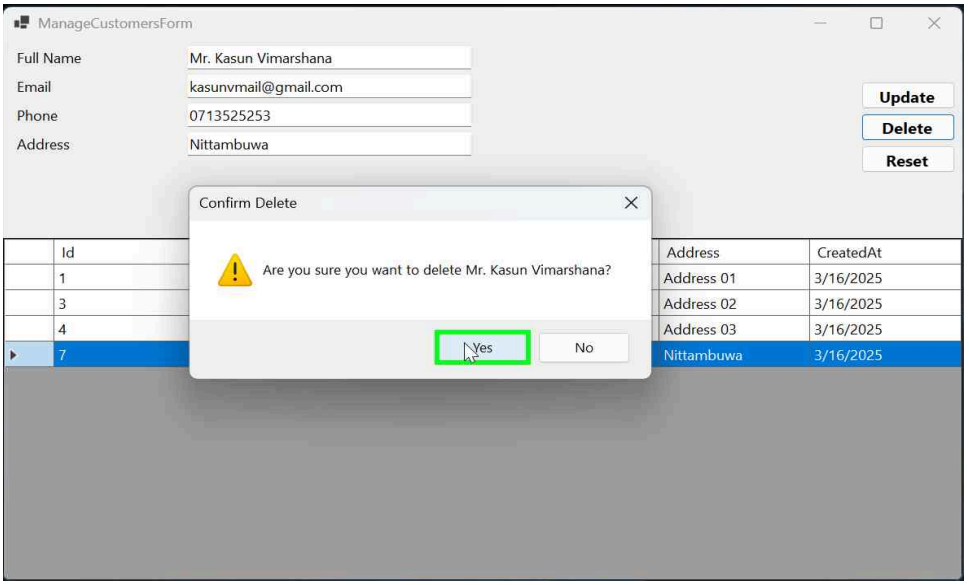


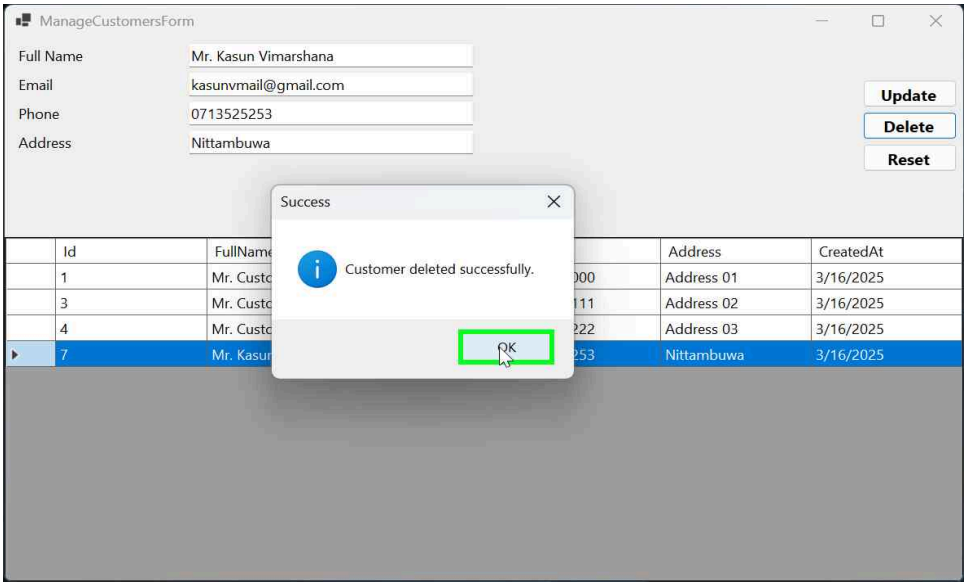

ManageCustomersForm
Full Name
Mr. Kasun Vimarshana
Email
kasunvmail@gmail.com
Phone
0713525253
Address
Nittambuwa
Add
Reset
Success
Customer added successfully.
OK
Id
FullName
Address
CreatedAt
1
Mr. Custo
000
Address 01
3/16/2025
3
Mr. Custo
111
Address 02
3/16/2025
4
Mr. Custo
222
Address 03
3/16/2025


ManageCustomersForm
Full Name
Mr. Kasun Vimarshana
Email
kasunvmail@gmail.com
Phone
0713525253
Address
Nittambuwa
Update
Delete
Reset
Id
FullName
Email
Phone
Address
CreatedAt
1
Mr. Customer 01
customer01@email....
0000000000
Address 01
3/16/2025
3
Mr. Customer 02
customer02@email....
1111111111
Address 02
3/16/2025
4
Mr. Customer 03
customer03@email....
2222222222
Address 03
3/16/2025
7
Mr. Kasun Vimarsha...
kasunvmail@gmail.c...
0713525253
Nittambuwa
3/16/2025


ManageCustomersForm
Full Name
Mr. Kasun Vimarshana
Email
kasunvmail@gmail.com
Phone
0713525253
Address
Nittambuwa
Update
Delete
Reset
Success
Customer updated successfully.
OK
Id
Address
CreatedAt
1
Address 01
3/16/2025
3
Address 02
3/16/2025
4
Address 03
3/16/2025
7
Nittambuwa
3/16/2025


ManageCustomersForm
Full Name
Mr. Kasun Vimarshana
Email
kasunvmail@gmail.com
Phone
0713525253
Address
Nittambuwa
Update
Delete
Reset
Id
FullName
Email
Phone
Address
CreatedAt
1
Mr. Customer 01
customer01@email....
0000000000
Address 01
3/16/2025
3
Mr. Customer 02
customer02@email....
1111111111
Address 02
3/16/2025
4
Mr. Customer 03
customer03@email....
2222222222
Address 03
3/16/2025
7
Mr. Kasun Vimarsha...
kasunvmail@gmail.c...
0713525253
Nittambuwa
3/16/2025


ManageCustomersForm
Full Name
Mr. Kasun Vimarshana
Email
kasunvmail@gmail.com
Phone
0713525253
Address
Nittambuwa
Update
Delete
Reset
Confirm Delete
Are you sure you want to delete Mr. Kasun Vimarshana?
Yes
No
Id
Address
CreatedAt
1
Address 01
3/16/2025
3
Address 02
3/16/2025
4
Address 03
3/16/2025
7
Nittambuwa
3/16/2025


ManageCustomersForm
Full Name
Mr. Kasun Vimarshana
Email
kasunvmail@gmail.com
Phone
0713525253
Address
Nittambuwa
Update
Delete
Reset
Success
Customer deleted successfully.
OK
Id
Address
CreatedAt
1
Address 01
3/16/2025
3
Address 02
3/16/2025
4
Address 03
3/16/2025
7
Nittambuwa
3/16/2025

5) **Supplier Management**
   - ☐ Access: Admin only.
   - ☐ Purpose: Manage supplier details.
   - ☐ Steps:
     - ☐ Navigate to Supplier Management from the main menu.
     - ☐ To add a supplier:
       - ☐ Fill details (Name, Contact Person, Phone, Email, Address).
       - ☐ Click Save.
     - ☐ To update a supplier:
       - ☐ Select supplier from list.
       - ☐ Modify details.
       - ☐ Click Update.
     - ☐ To delete a supplier:
       - ☐ Select supplier from list.
       - ☐ Click Delete.
       - ☐ Confirm deletion.

6) **Book Management**
   - ☐ Access: Admin only.
   - ☐ Purpose: Manage book inventory.
   - ☐ Steps:
     - ☐ Navigate to Book Management from the main menu.
     - ☐ To add a book:
       - ☐ Fill details (Title, Author, Genre, ISBN, Price, Stock Quantity).
       - ☐ Click Save.
     - ☐ To update a book:
       - ☐ Select book from list.

- ☐ Modify details.
- ☐ Click Update.

☐ To delete a book:

- ☐ Select book from list.
- ☐ Click Delete.
- ☐ Confirm deletion.

7) Order Management

☐ Access: Admin only.

☐ Purpose: Manage book orders from suppliers.

☐ Steps:

☐ Navigate to Order Management from the main menu.

☐ To add an order:

- ☐ Select supplier and fill details (Order Date, Total Amount, Status).
- ☐ Click Save.

☐ To update an order:

- ☐ Select order from list.
- ☐ Modify details.
- ☐ Click Update.

☐ To delete an order:

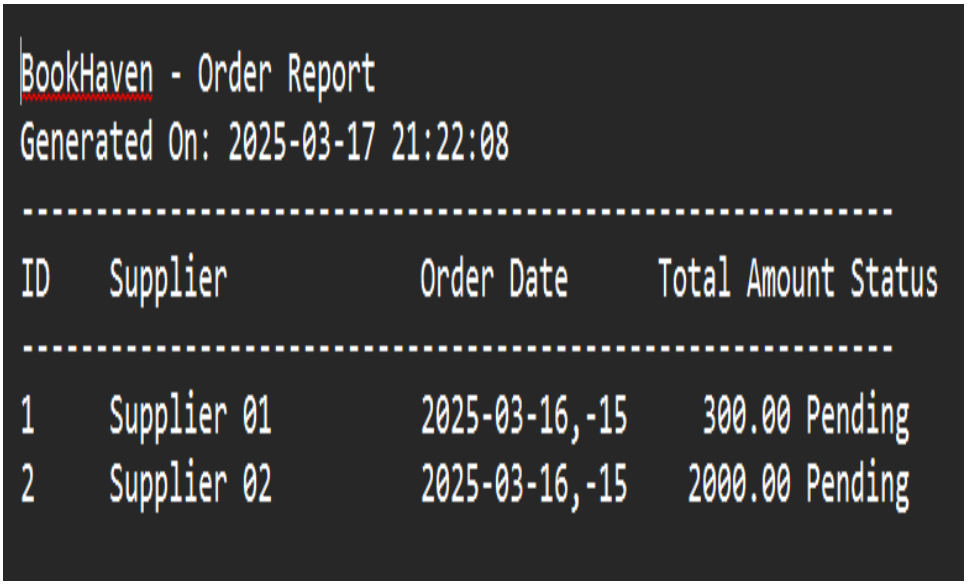
- ☐ Select order from list.
- ☐ Click Delete.
- ☐ Confirm deletion.

8) Sales Management

- ☐ Access: Admin and Sales Clerk.
- ☐ Purpose: Manage sales transactions.
- ☐ Steps:
  - ☐ Navigate to Sales Management from the main menu.
  - ☐ To add a sale:
    - ☐ Select customer and fill details (Sale Date, Total Amount, Discount).
    - ☐ Click Save.
  - ☐ To update a sale:
    - ☐ Select sale from list.
    - ☐ Modify details.
    - ☐ Click Update.
  - ☐ To delete a sale:
    - ☐ Select sale from list.
    - ☐ Click Delete.
    - ☐ Confirm deletion.

9) Reporting

- ☐ Access: Admin and Sales Clerk.
- ☐ Purpose: Generate reports for sales and orders.
- ☐ Steps:
  - ☐ Navigate to Reports from the main menu.
  - ☐ To generate sales report:
    - ☐ Click Generate Sales Report.
    - ☐ Report will be saved as a text file and opened automatically.
  - ☐ To generate an order report:
    - ☐ Click Generate Order Report.
    - ☐ Report will be saved as a text file and opened automatically.



```
********** ORDER REPORT **********
Order ID: 2
Supplier: Supplier 02
Total Amount: $2,000.00
Order Date: 3/16/2025 3:00:53 AM
-----------------------------------
Item Details:
Book     Quantity          Price
Book 01 10        $100.00
Book 03 10        $100.00
************************************
```

```
BookHaven - Sales Report
Generated On: 2025-03-17 21:23:40
-----------------------------------------------------------
ID    Customer            Sale Date       Total Amount
-----------------------------------------------------------
1     Mr. Customer 01     2025-03-16,-15     200.00
2     Mr. Customer 02     2025-03-16,-15     300.00
3     Mr. Customer 01     2025-03-16,-15    1800.00
4     Mr. Kasun Vimarshana 2025-03-16,-15    1800.00
5     Mr. Kasun Vimarshana 2025-03-16,-15    1800.00
```

```
********** SALES REPORT **********
Sale ID: 5
Customer: Mr. Kasun Vimarshana
Total Amount: $1,800.00
Discount: $0.00
Sale Date: 3/16/2025 7:29:56 PM
-----------------------------------
Item Details:
Book     Quantity          Price
Book 01 3         $100.00
Book 02 5         $100.00
Book 03 10        $100.00
************************************
```

Architecture Of System

BookHaven application has well structured multi-layer application development using MVC (Model-View-Controller) pattern. Here we explain about each layer and its responsibility and how it interacts with other layers.

```
:.
|   .gitignore
|   BookHaven.sln
|   README.md
|
+---BookHaven
|   |   App.config
|   |   BookHaven.csproj
|   |   BookHaven.csproj.user
|   |   db.mdf
|   |   db_log.ldf
|   |   Program.cs
|   |
|   +---BLL
|   |       AuthenticationService.cs
|   |       BookService.cs
|   |       CustomerService.cs
|   |       OrderDetailService.cs
|   |       OrderService.cs
|   |       SalesDetailService.cs
|   |       SalesService.cs
|   |       SupplierService.cs
|   |       UserService.cs
|   |
|   +---DAL
|   |       BookRepository.cs
|   |       CustomerRepository.cs
|   |       DatabaseHelper.cs
|   |       IDatabaseHelper.cs
|   |       OrderDetailRepository.cs
|   |       OrderRepository.cs
|   |       SalesDetailRepository.cs
|   |       SalesRepository.cs
|   |       SupplierRepository.cs
|   |       UserRepository.cs
|   |
|   +---Enums
|   |       Month.cs
|   |       OrderStatus.cs
|   |       UserRole.cs
|   |
|   +---Helpers
|   |       LoggedInUserSession.cs
|   |
|   +---Models
|   |       Book.cs
|   |       Customer.cs
|   |       Order.cs
|   |       OrderDetail.cs
|   |       Sale.cs
|   |       SalesDetail.cs
|   |       Supplier.cs
|   |       User.cs
|   |
|   +---Properties
|   |   \---PublishProfiles
|   +---Reports
|   |       OrderDetailReportService.cs
```

```
|   |       OrderReportService.cs
|   |       SalesDetailReportService.cs
|   |       SalesReportService.cs
|   |
|   +---Seeders
|   |       UserSeeder.cs
|   |
|   +---UI
|   |   \---Forms
|   |       |   LoginForm.cs
|   |       |   LoginForm.Designer.cs
|   |       |   LoginForm.resx
|   |       |   MainForm.cs
|   |       |   MainForm.Designer.cs
|   |       |   MainForm.resx
|   |       |
|   |       +---Book
|   |       |       ManageBooksForm.cs
|   |       |       ManageBooksForm.Designer.cs
|   |       |       ManageBooksForm.resx
|   |       |
|   |       +---Customer
|   |       |       ManageCustomersForm.cs
|   |       |       ManageCustomersForm.Designer.cs
|   |       |       ManageCustomersForm.resx
|   |       |
|   |       +---Order
|   |       |       ManageOrdersForm.cs
|   |       |       ManageOrdersForm.Designer.cs
|   |       |       ManageOrdersForm.resx
|   |       |
|   |       +---OrderDetail
|   |       |       ManageOrderDetailsForm.cs
|   |       |       ManageOrderDetailsForm.Designer.cs
|   |       |       ManageOrderDetailsForm.resx
|   |       |
|   |       +---Sale
|   |       |       ManageSalesForm.cs
|   |       |       ManageSalesForm.Designer.cs
|   |       |       ManageSalesForm.resx
|   |       |
|   |       +---SalesDetail
|   |       |       ManageSalesDetailsForm.cs
|   |       |       ManageSalesDetailsForm.Designer.cs
|   |       |       ManageSalesDetailsForm.resx
|   |       |
|   |       +---Supplier
|   |       |       ManageSuppliersForm.cs
|   |       |       ManageSuppliersForm.Designer.cs
|   |       |       ManageSuppliersForm.resx
|   |       |
|   |       \---User
|   |               ManageUsersForm.cs
|   |               ManageUsersForm.Designer.cs
|   |               ManageUsersForm.resx
|   |
|   \---Utilities
|           Logger.cs
|           PasswordHelper.cs
|
```

Models Layer (Entities)

- ☐ Directory: BookHaven/Models/
- ☐ Purpose: Defines data structure used in the application, representing real-world entities like books, customers, orders, etc.
- ☐ Key Characteristics:
  - ☐ Classes in this layer define the properties of entities.
  - ☐ Used throughout the application in Business Logic (BLL) and Data Access Layer (DAL).

```
namespace BookHaven.Models
{
   class Book
   {
      public int Id { get; set; }
      public string Title { get; set; }
      public string Author { get; set; }
      public string Genre { get; set; }
      public string ISBN { get; set; }
      public decimal Price { get; set; }
      public int StockQuantity { get; set; }
      public int? SupplierId { get; set; }
      public virtual Supplier Supplier { get; set; }
      public DateTime CreatedAt { get; set; } = DateTime.Now;
   }
}
```

Data Access Layer (DAL)

- ☐ Directory: BookHaven/DAL/
- ☐ Purpose:Handles direct interaction with database.
- ☐ Key Characteristics:
  - ☐ Provides CRUD (Create, Read, Update, Delete) operations.
  - ☐ Encapsulates all database logic.
  - ☐ Each repository handles database operations for specific entity.

```
namespace BookHaven.DAL
{
   class BookRepository
   {
      private readonly DatabaseHelper _dbHelper = new DatabaseHelper();
      public int CreateBook(Book book, SqlTransaction transaction = null)
      {
         try
         {
            string query = @"
```

```
                INSERT INTO Books (Title, Author, Genre, ISBN, Price,
StockQuantity, SupplierId, CreatedAt)
                OUTPUT INSERTED.Id
                VALUES (@Title, @Author, @Genre, @ISBN, @Price,
@StockQuantity, @SupplierId, @CreatedAt)";

            SqlParameter[] parameters = {
                new SqlParameter("@Title", SqlDbType.NVarChar) { Value =
book.Title },
                new SqlParameter("@Author", SqlDbType.NVarChar) { Value =
book.Author },
                new SqlParameter("@Genre", SqlDbType.NVarChar) { Value =
book.Genre ?? (object)DBNull.Value },
                new SqlParameter("@ISBN", SqlDbType.NVarChar) { Value =
book.ISBN },
                new SqlParameter("@Price", SqlDbType.Decimal) { Value =
book.Price },
                new SqlParameter("@StockQuantity", SqlDbType.Int) { Value =
book.StockQuantity },
                new SqlParameter("@SupplierId", SqlDbType.Int) { Value =
book.SupplierId ?? (object)DBNull.Value },
                new SqlParameter("@CreatedAt", SqlDbType.DateTime) { Value =
book.CreatedAt }
            };

            // Execute the query and return the inserted ID
            object result = _dbHelper.ExecuteScalar(query, parameters,
transaction);
            return result != null ? Convert.ToInt32(result) : -1;
        }
        catch (Exception ex)
        {
            Logger.LogError("CreateBook failed: " + ex.Message);
            return -1;
        }
    }

  }
}
```

Business Logic Layer (BLL)

- ☐ Directory: BookHaven/BLL/
- ☐ Purpose: Contains core business logic and acts as an intermediary between the UI and the Data Access Layer.
- ☐ Key Characteristics:
  - ☐ Calls DAL methods and applies necessary validations, transformations, and business rules.

- ☐ Encapsulates business-related operations to ensure maintainability.
- ☐ Calls corresponding repository methods from the DAL.
- ☐ Ensures business rules like "Stock cannot be negative" are applied.
- ☐ Handles transactions when multiple operations need to be executed together.

```
namespace BookHaven.BLL
{
   class BookService
   {
      public int CreateBook(Book supplier, SqlTransaction transaction = null)
         => _bookRepo.CreateBook(supplier, transaction);
   }
}
```

User Interface Layer (UI)

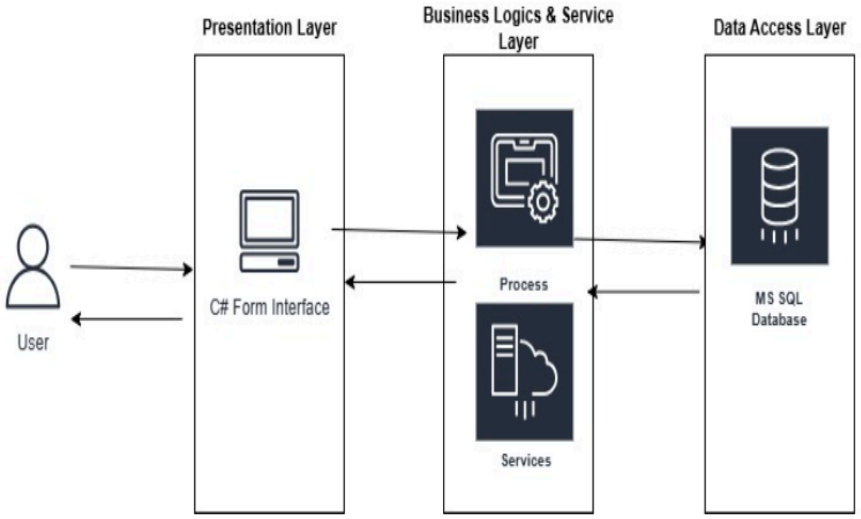
- ☐ Directory: BookHaven/UI/
- ☐ Purpose: Contains Windows Forms (.cs files) for interacting with users.
- ☐ Key Characteristics:
  - ☐ Displays data using DataGridView and other UI components.
  - ☐ Captures user input and sends it to the BLL for processing.
  - ☐ Calls BLL services and updates UI elements accordingly.
  - ☐ Uses Events and Event Handlers for user interactions.

Breakdown of the Architecture

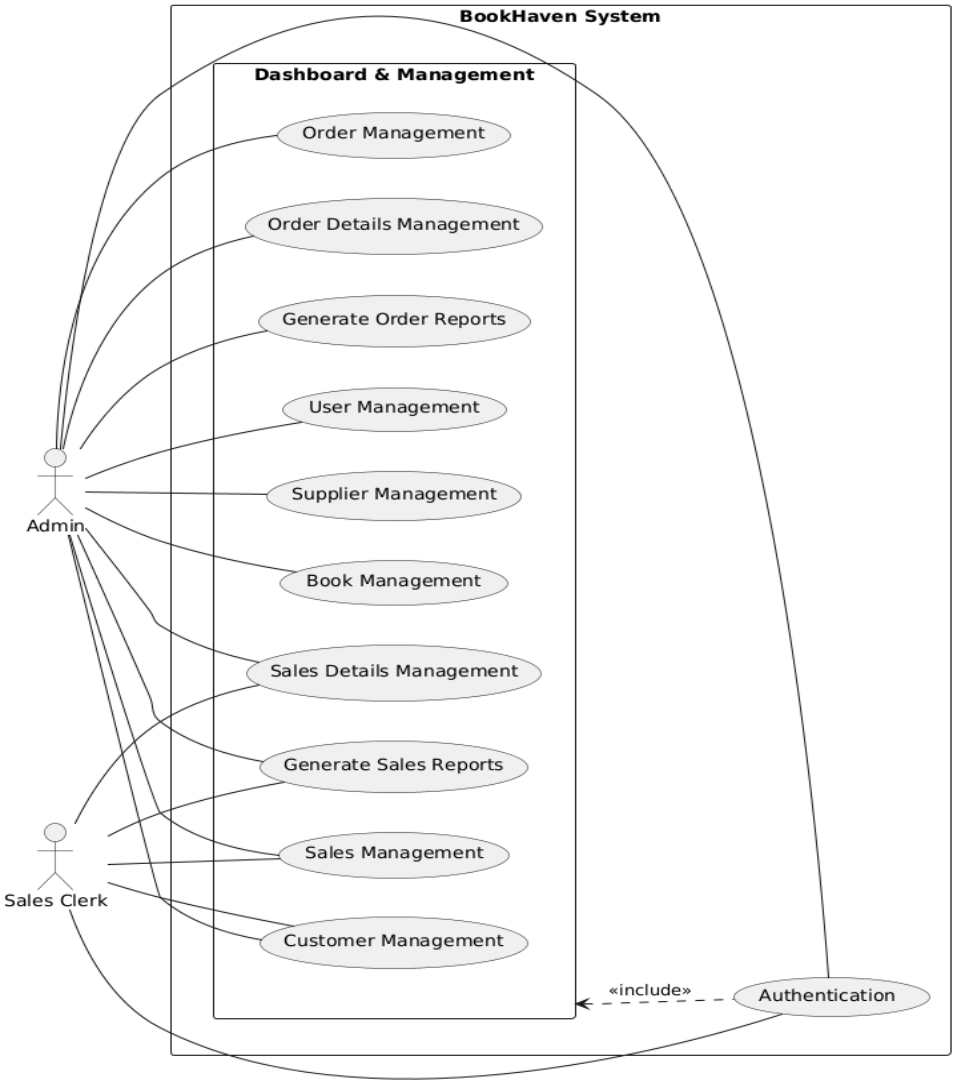
Layer	Responsibilities
Models (Entities)	Defines data structures
Data Access Layer (DAL)	Handles database queries and transactions
Business Logic Layer (BLL)	Implements business rules and validations
User Interface Layer (UI)	Manages user interactions (Windows Forms)
Reports Layer	Generates structured reports
Utilities Layer	Provides common utilities (logging, encryption)
Seeders Layer	Initializes database with default data
Enums Layer	Defines constants and enumerations

UML Diagrams

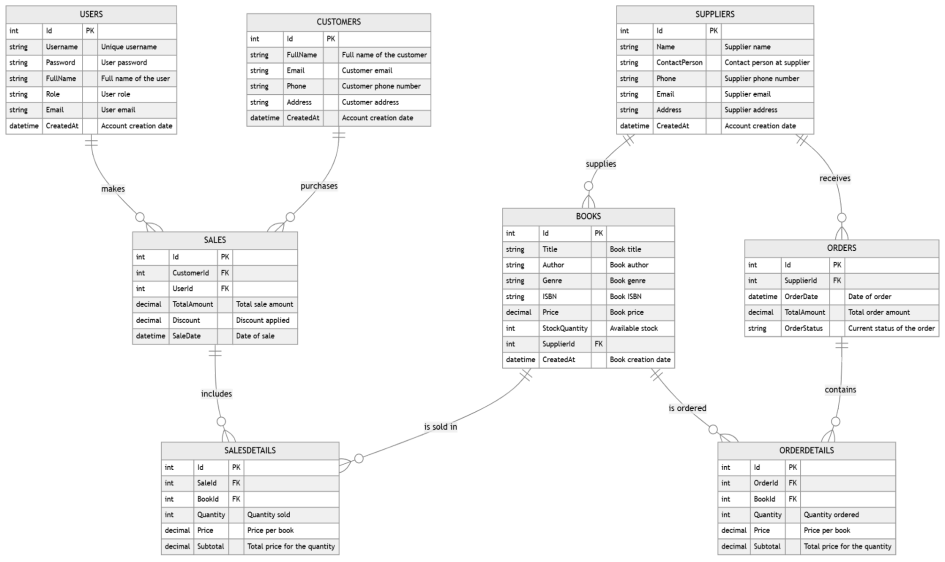
System Architecture



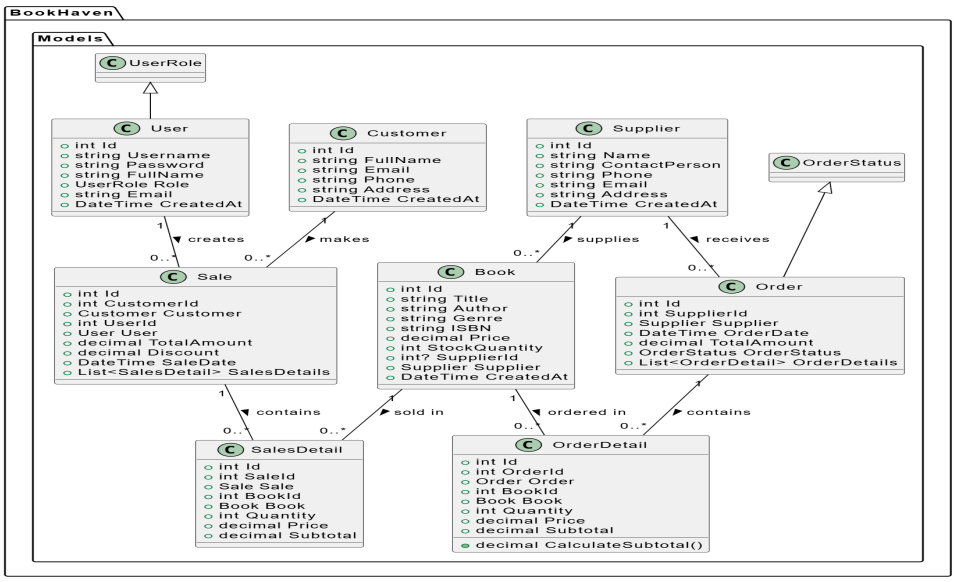
Use Case Diagram



ER Diagram

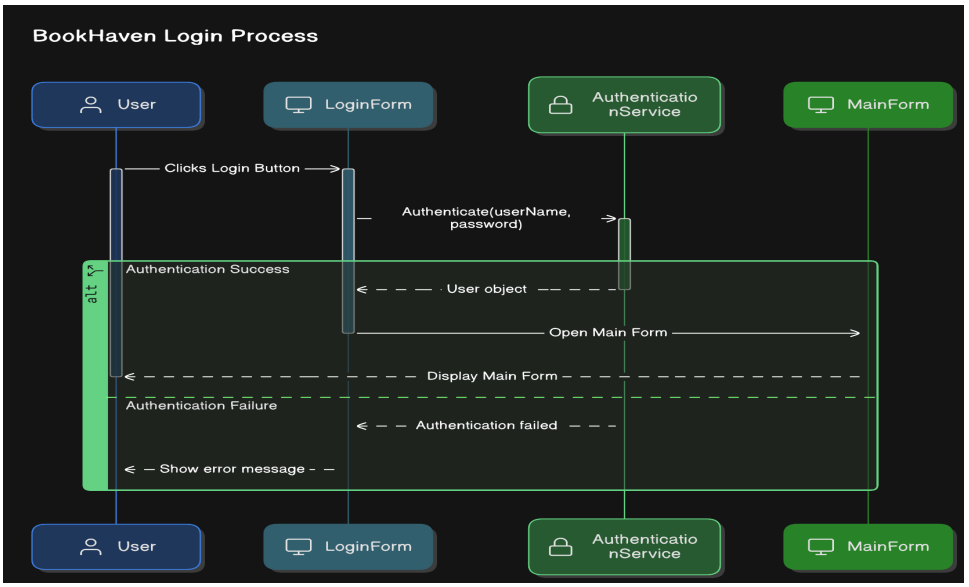


Class Diagram

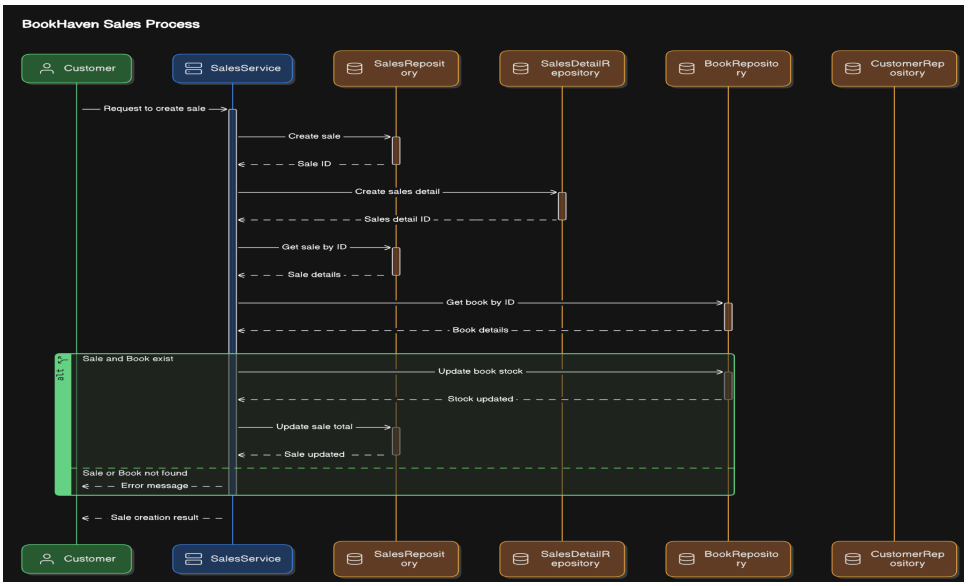


Sequence Diagram

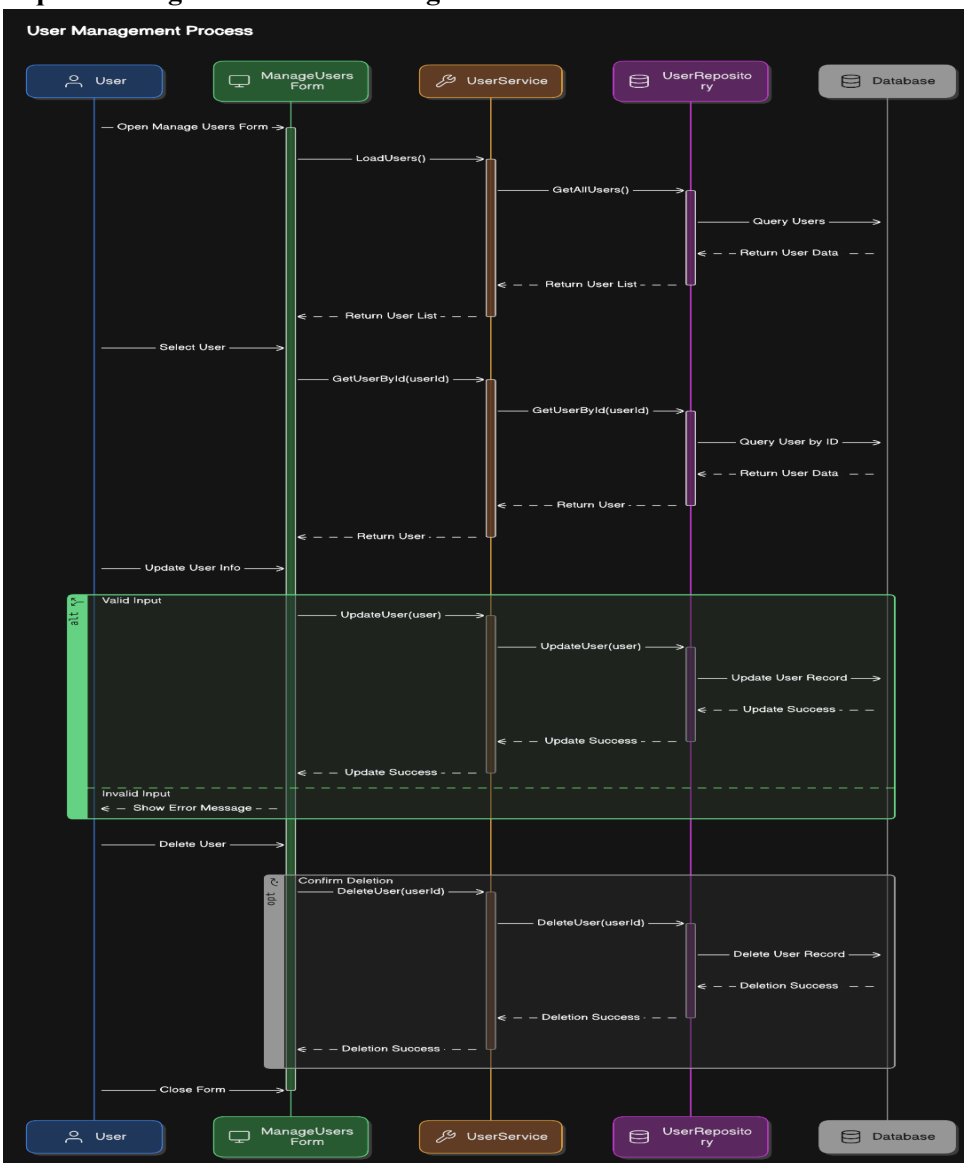
Sequence Diagram for Login



Sequence Diagram for Sales Management



Sequence Diagram for User Management



Own Reflection of My Experience

Developing the BookHaven bookstore management system was both a challenging and rewarding experience. As a beginner-level C# developer, this project helped me deepen my understanding of Windows Forms development, C# programming, and the MVC architecture.

Key Learnings

- ☐ Understanding MVC Architecture
  - ☐ Initially, structuring the project using the Model-View-Controller (MVC) pattern was challenging, but as I progressed, I realized the importance of separating concerns for maintainability and scalability.
- ☐ Database Design and SQL Integration
  - ☐ Designing a relational database and writing SQL queries were essential skills I developed.
- ☐ Authentication and Role-Based Access Control
  - ☐ Implementing user authentication and role-based access for Admin and SalesClerk roles gave me hands-on experience with secure login systems, password hashing, and user role management.
- ☐ Error Handling and Debugging
  - ☐ Debugging errors, especially in database connections and UI interactions, was a crucial learning experience. Using try-catch blocks, logging errors, and debugging tools in Visual Studio improved my troubleshooting skills.
- ☐ Windows Forms UI Design
  - ☐ Designing a user-friendly interface using Windows Forms was a challenge. I learned how to work with DataGridView, Forms, and Charts.

Challenges Faced

- ☐ Handling Database Transactions: Ensuring that multiple operations (such as adding a sale and updating inventory) were correctly executed required learning about transactions.
- ☐ Implementing Role-Based Access Control: Restricting features based on user roles was a new concept, requiring proper authorization logic.

Future Improvements

If I had more time, I would like to:

- ☐ Migrate the application to a web-based solution using ASP.NET Core.
- ☐ Implement an API layer to make it extendable.
- ☐ Enhance security measures with OAuth or JWT authentication.

- [ ] Integrate barcode scanning for faster book searching.

Final Thoughts

This project has been a wonderful learning experience in C# programming, database management, and software development best practices. I felt increasingly confident that I am capable of developing a working application independently, and I look forward to continuing to improve my skills for future projects.

Additional Resources

For access to the source code and further details about the project, please visit the GitHub repository:

Repository Link: **https://github.com/kasunvimarshana/BookHaven**