

Library Management System

PROJECT REPORT

Dilawar Ali | Faizan Amir | Sabir Hussain | Sadiq Shah | Subhan Fur jam
IAD | 22 May 2024

INTRODUCTION

The project aims to develop a data-driven web application for a library management system, titled "Library Management System." This project is built using ASP.NET with VB.NET, providing functionalities for managing library resources, member activities, and administrative tasks. The application will incorporate robust state management, validation, data security measures, and role-based functionalities.

FEATURES OF THE SYSTEM

Here are some of the overarching features of the banking system:

- Book management (add, edit, delete books).
- Member management (add, edit, delete members).
- Borrowing and returning books.
- Search and filter books.
- User authentication and authorization.
- User roles (Admin, Member).

SOFTWARE REQUIREMENTS SPECIFICS

The Software Requirements Specification outlines the functional and non-functional requirements of the proposed banking system. This system aims to provide a comprehensive platform for managing various banking operations efficiently, ensuring security, reliability, and ease of use for customers and bank staff.

Functional Requirements

User Management:

- Users can register, login, and authenticate using username and password.
- Different user roles are defined: members, and Admin.
- Members can manage their account information, view borrowed books, and check due dates.
- Admin can manage member accounts, add new members, and update member details.
- Admins have full access to manage users, roles, and system settings.

Book Management:

- Admin can add new books, update book information, and delete books.
- Members can search for books by title, author, category, and availability.
- Members can borrow and return books, with real-time updates to book availability and member accounts.

Transaction Processing:

- Members can initiate transactions such as borrowing, returning books.
- Transactions are processed securely and accurately, reflecting real-time updates to the system.

Account Management:

- Members can view their borrowing history, due dates, and account information.
- Admin can manage borrowing limits, issue penalties, and update member statuses.

NON-FUNCTIONAL REQUIREMENTS

Security:

Secure password hashing and encryption methods are used for storing user credentials.

Performance:

The system should be highly available and capable of handling large number of concurrent users without significant performance degradation.

Usability:

The user interface should be intuitive and easy to navigate, with clear instructions and error messages.

Accessibility features should be considered to support users with disabilities.

Scalability:

The system should be scalable to accommodate future growth in the number of users, books, and transactions.

ANALYSIS AND DESIGN MODEL

USE CASE MODEL

Actors:

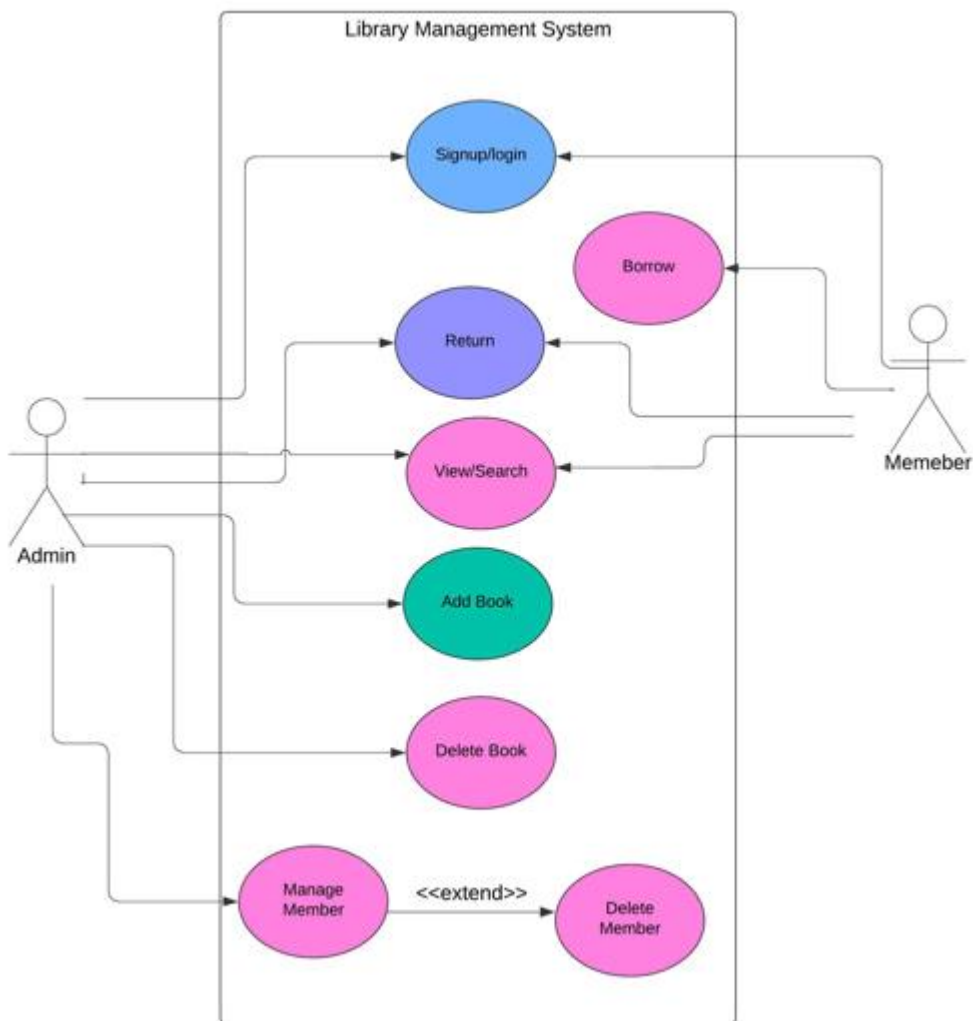
- Member: The actor who interacts with the library management system to perform various actions related to their account.
- Admin: The actor who manages the system and its users.

Use Cases for Admin:

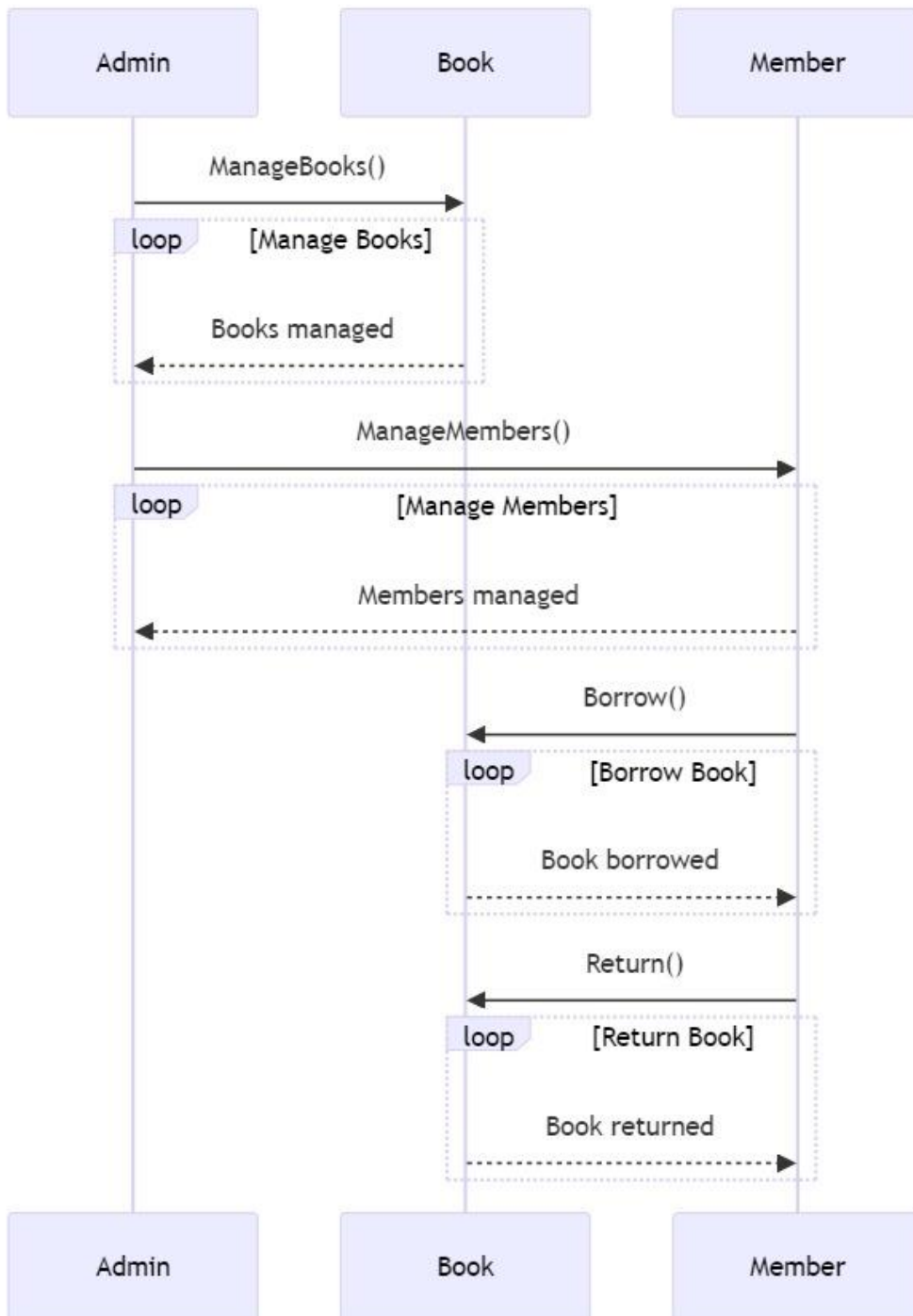
- Login: Allows user to log into their account
- Search/View: Allows admins to see books.
- Manage Member: Allows admins to manage member account.
- Return: Allows admins to return book.
- Add Book: Allows admins to add new books in library management system.
- Delete Book: Allows admins to delete books from System.

Use Cases for Member:

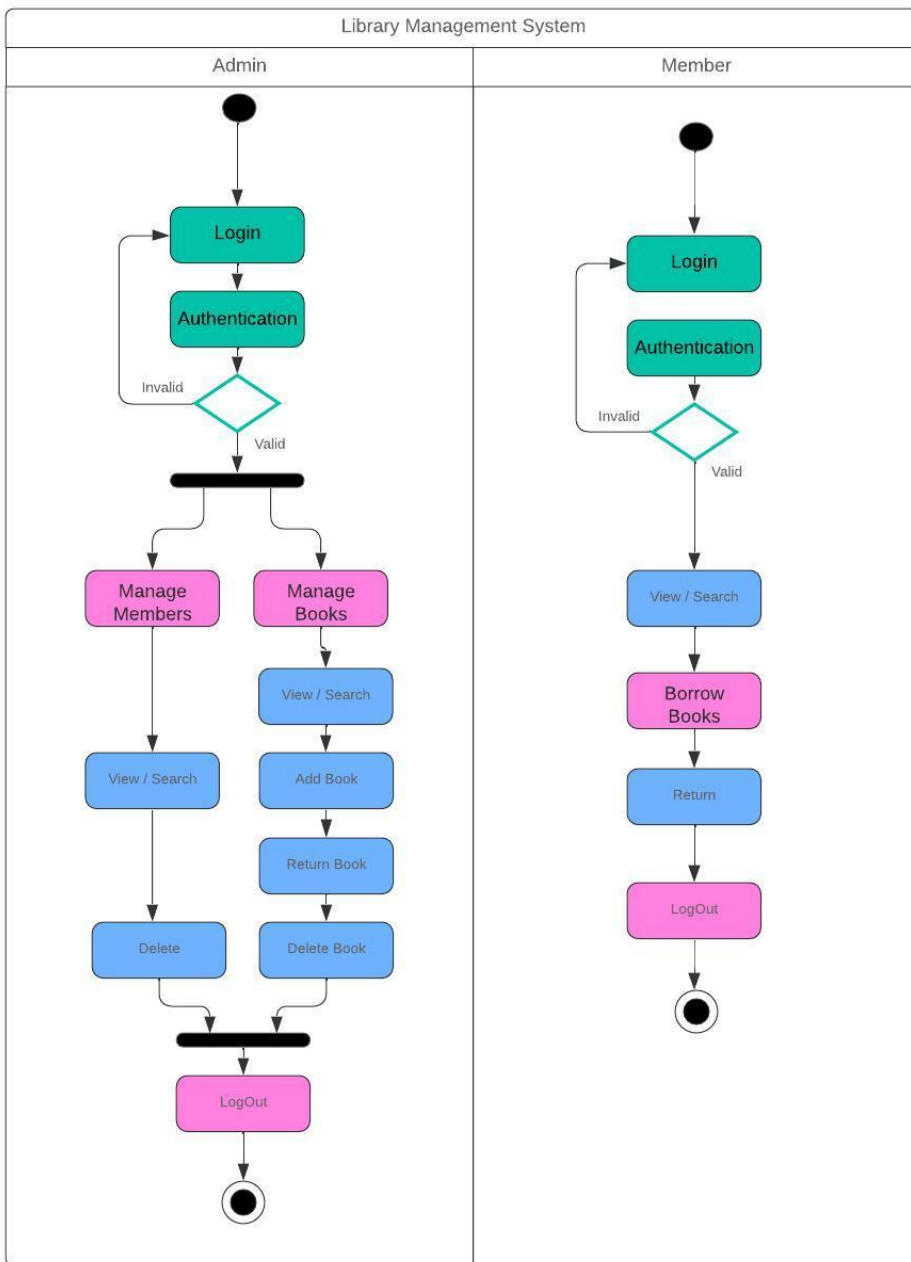
- Login: Allows member to log into their account
- Search/View: Allows member to view books.
- Borrow: Allows member to borrow books.
- Return: Allows member to return books.



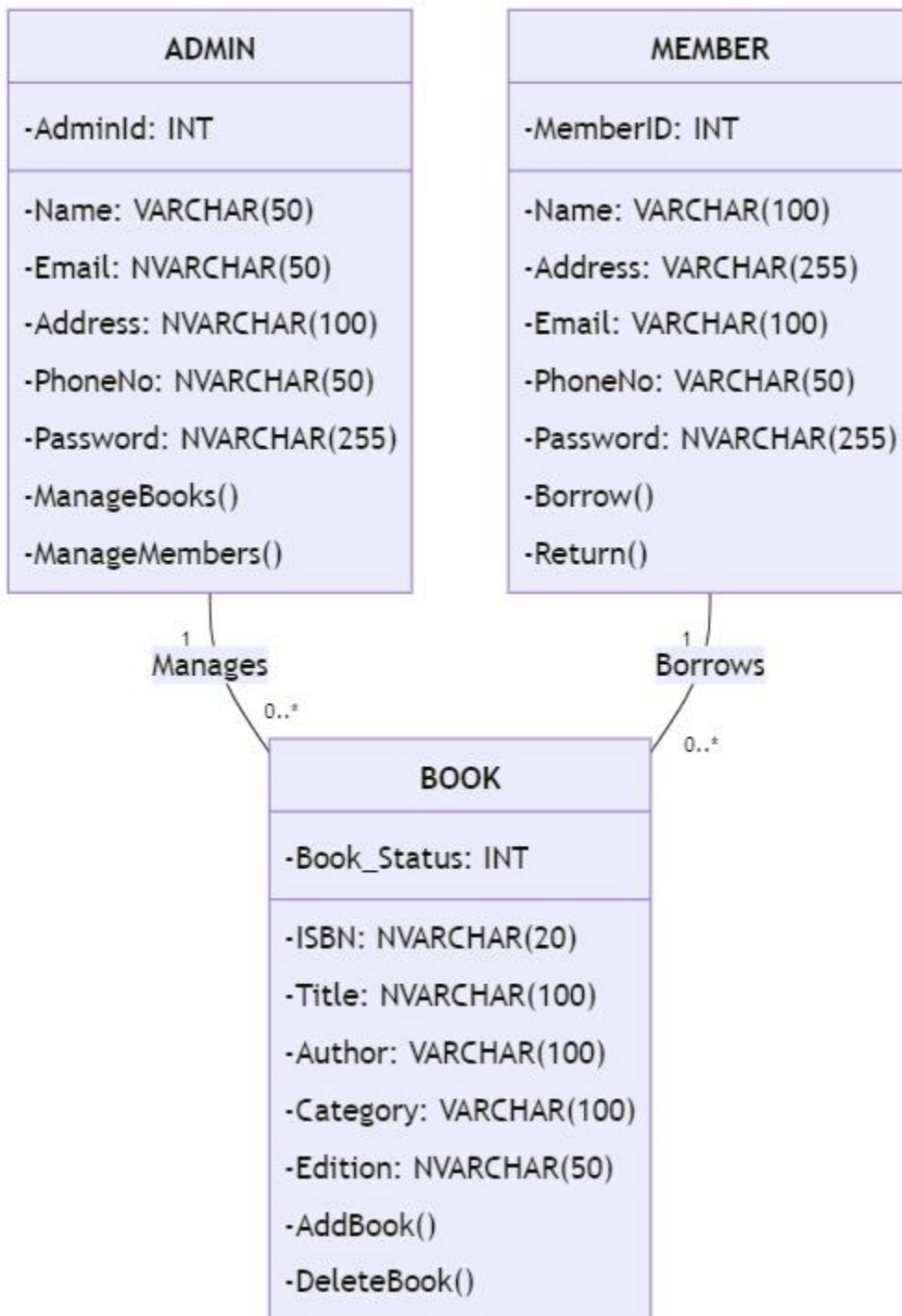
Sequence diagram:



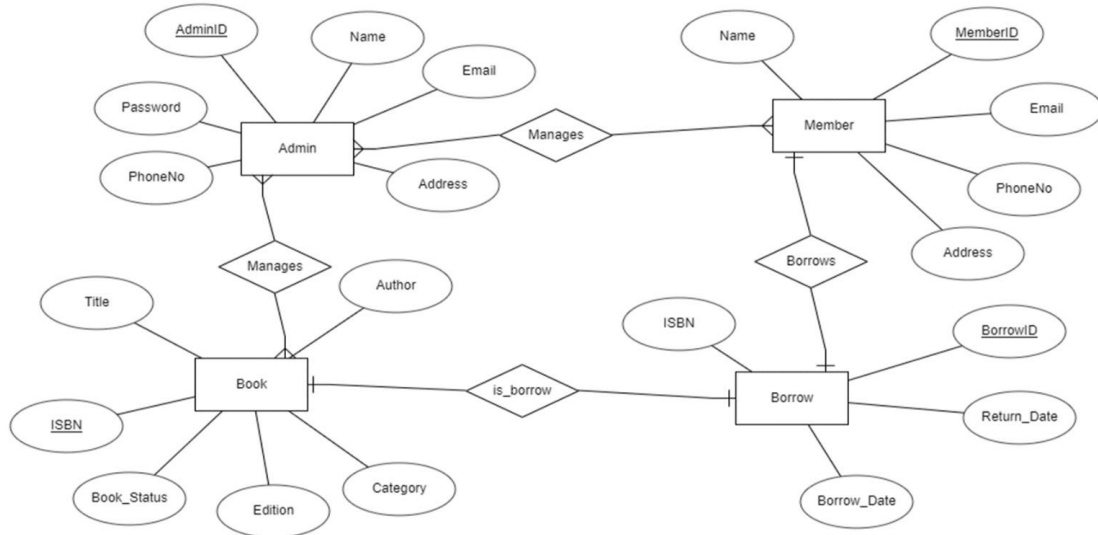
ACTIVITY DIAGRAM:



Class Diagram:



ENTITY-RELATIONSHIP DIAGRAM



RELATIONAL DIAGRAM

We use () for Primary Key and (()) for Foreign key below.

Tables and Attributes

- Admin

(AdminID) | Name | E-mail | Adress | Password | Phone No |

- Member

(MemberID) | Name | E-mail | Adress | Password | Phone No |

- Book

(ISBN) | Title | Author | Category | Edition | Book_Status |

- Borrow

(BorrowID) | ((ISBN)) | Borrowdate | Return date |

STRUCTURED QUERY SCHEMA (SQL) IMPLEMENTATION

```
CREATE TABLE ADMIN (  
    AdminId INT PRIMARY KEY IDENTITY(1,1),  
    Name VARCHAR(50),  
    Email NVARCHAR(50),  
    Address NVARCHAR(100),  
    PhoneNo NVARCHAR(50),  
    Password NVARCHAR(255)  
);  
  
CREATE TABLE BOOK (  
    ISBN NVARCHAR(20) PRIMARY KEY,  
    Title NVARCHAR(100) NOT NULL,  
    Author VARCHAR(100) NOT NULL,  
    Category VARCHAR(100) NOT NULL,  
    Edition NVARCHAR(50) NOT NULL,  
    Book_Status INT NOT NULL,  
);  
  
CREATE TABLE MEMBER (  
    MemberID INT PRIMARY KEY IDENTITY(1,1),  
    Name VARCHAR(100) NOT NULL,  
    Address VARCHAR(255),  
    Email VARCHAR(100) UNIQUE,  
    PhoneNo VARCHAR(50),  
    Password NVARCHAR(255),  
);  
  
CREATE TABLE BORROW (  
    BorrowID INT PRIMARY KEY IDENTITY(1,1),  
    ISBN NVARCHAR(20) NOT NULL,  
    BorrowDate DATETIME NOT NULL,  
    ReturnDate DATETIME,  
    CONSTRAINT FK_BORROW_BOOK FOREIGN KEY (ISBN) REFERENCES BOOK(ISBN) ON DELETE CASCADE  
);
```

DATABASE

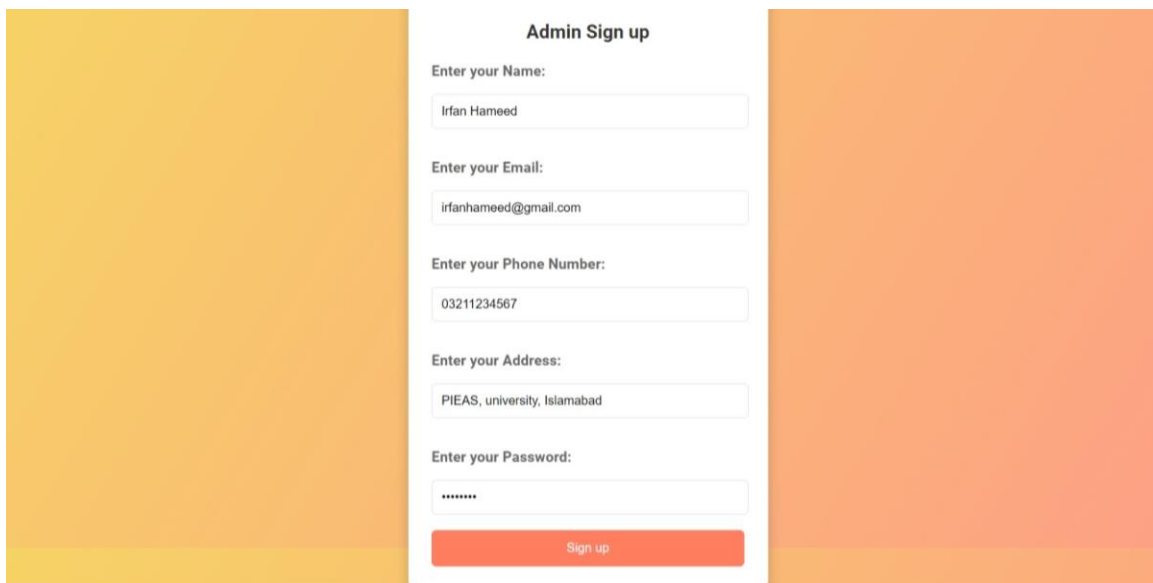
Sample Data

Discrete Mathematics and Its Applications	Kenneth H. Rosen	Mathematics	7th
Data Mining: Concepts and Techniques	Jiawei Han, Micheline Kamber, Jian Pei	Data Mining	3rd
Computer Organization and Design: The Hardware/Software Interface	David A. Patterson, John L. Hennessy	Computer Architecture	5th
The C Programming Language	Brian W. Kernighan, Dennis M. Ritchie	C Programming	2nd
Clean Code: A Handbook of Agile Software Craftsmanship	Robert C. Martin	Software Development	1st
Computer Networks	Andrew S. Tanenbaum	Networking	5th
Computer Networking: A Top-Down Approach	James F. Kurose, Keith W. Ross	Networking	7th
Computer Systems: A Programmer's Perspective	Randal E. Bryant, David R. O'Hallaron	Computer Architecture	3rd

USER INTERFACE (UI) DESIGN

BELOW ARE SNIPPETS FROM SOME PAGES.

Admin Signup page



The image shows a snippet of an 'Admin Sign up' form. The form is centered on a white background with a light orange gradient on the sides. It contains the following fields and labels:

- Admin Sign up** (Section Header)
- Enter your Name:** Input field containing 'Irfan Hameed'
- Enter your Email:** Input field containing 'irfanhameed@gmail.com'
- Enter your Phone Number:** Input field containing '03211234567'
- Enter your Address:** Input field containing 'PIEAS, university, Islamabad'
- Enter your Password:** Input field containing '*****'
- Sign up** (Submit Button)

View Or Search page

Library Management System

Search books... All Search Add Books Delete Books

Book Title	Author	Category	Edition	Status
Discrete Mathematics and Its Applications	Kenneth H. Rosen	Mathematics	7th	Available
Data Mining: Concepts and Techniques	Jiawei Han, Micheline Kamber, Jian Pei	Data Mining	3rd	Available
Computer Organization and Design: The Hardware/Software Interface	David A. Patterson, John L. Hennessy	Computer Architecture	5th	Available
The C Programming Language	Brian W. Kernighan, Dennis M. Ritchie	C Programming	2nd	Available
Clean Code: A Handbook of Agile Software Craftsmanship	Robert C. Martin	Software Development	1st	Available
Computer Networks	Andrew S. Tanenbaum	Networking	5th	Available
Computer Networking: A Top-Down Approach	James F. Kurose, Keith W. Ross	Networking	7th	Available
Computer Systems: A Programmer's Perspective	Randal E. Bryant, David R. O'Hallaron	Computer Architecture	3rd	Available
The Pragmatic Programmer: Your Journey to Mastery	David Thomas	Software Development	2nd	Available

Add new Book.

Add New Book

Enter ISBN:
978-1234567123

Enter Title:
The Great Gatsby

Enter Author:
F. Scott Fitzgerald

Enter Category:
Fiction

Enter Edition:
2nd

Add Book

Member Signup Page

Member Signup Page

Enter your Name:

Enter your Email:

Enter your Phone Number:

Enter your Address:

Enter your Password:

IMPLEMENTATION CODE

Admin logni.aspx

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Admin-login.aspx.vb" Inherits="_Default"
%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Admin Login</title>
  <style>
    body {
      font-family: 'Roboto', sans-serif;
      background: linear-gradient(120deg, #f6d365 0%, #fda085 100%);
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    form {
      background-color: #ffffff;
      padding: 30px;
      border-radius: 12px;
      box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
      max-width: 400px;
      width: 100%;
    }
    h1, h2 {
```

```

    text-align: center;
    color: #333333;
    margin: 0;
  }
  h1 {
    font-size: 2em;
    margin-bottom: 0.5em;
  }
  h2 {
    font-size: 1.5em;
    margin-bottom: 1em;
  }
  h3 {
    margin-bottom: 10px;
    color: #666666;
  }
  input[type="email"], input[type="password"] {
    width: calc(100% - 20px);
    padding: 12px;
    margin: 8px 0 20px 0;
    border: 1px solid #ddd;
    border-radius: 6px;
    font-size: 1em;
  }
  input[type="email"]::placeholder, input[type="password"]::placeholder {
    color: #bbb;
  }
  .button {
    width: 100%;
    background-color: #ff7e5f;
    color: white;
    padding: 14px;
    border: none;
    border-radius: 6px;
    cursor: pointer;
    font-size: 1em;
    transition: background-color 0.3s ease;
  }
  .button:hover {
    background-color: #eb6b4d;
  }
  @media (max-width: 600px) {
    form {
      padding: 20px;
    }
    input[type="email"], input[type="password"] {
      padding: 10px;
    }
    .button {
      padding: 12px;
      font-size: 0.9em;
    }
  }
</style>
</head>
<body>
<form id="form1" runat="server">
  <div>

```

```

<h1><strong>Library Management System</strong></h1>
<h2>Admin Login Page</h2>
<h3>Enter your Email:</h3>
<asp:TextBox ID="email" placeholder="Email" type="email" runat="server"></asp:TextBox>
<h3>Enter your Password:</h3>
<asp:TextBox ID="password" type="password" placeholder="Password"
runat="server"></asp:TextBox>
<asp:Button ID="login" runat="server" Text="Login" CssClass="button" OnClick="login_Click" />
<br />
<br />
<br />
<asp:Button ID="Signup" runat="server" Text="Signup" CssClass="button" OnClick="Signup_Click" />
</div>
</form>
</body>
</html>

```

Admin Login.aspx.vb

```

Imports System.Data.SqlClient
Imports System.Security.Cryptography
Partial Class _Default
    Inherits System.Web.UI.Page
    Protected Sub login_Click(ByVal sender As Object, ByVal e As EventArgs) Handles login.Click
        If String.IsNullOrEmpty(email.Text) Or String.IsNullOrEmpty(password.Text) Then
            Response.Write("<script>alert('Both fields are required.');

```

```

Using sha256 As New SHA256Managed()
    Dim bytes As Byte() = Encoding.UTF8.GetBytes(password)
    Dim hash As Byte() = sha256.ComputeHash(bytes)
    Dim sb As New StringBuilder()
    For Each b As Byte In hash
        sb.Append(b.ToString("x2"))
    Next
    Return sb.ToString()
End Using
End Function
Protected Sub Signup_Click(sender As Object, e As EventArgs) Handles Signup.Click
    Response.Redirect("Admin-signup.aspx")
End Sub
End Class

```

Admin-signup.aspx:

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Admin-signup.aspx.vb" Inherits="_Default"
%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Admin Signup</title>
    <style>
    body {
        font-family: 'Roboto', sans-serif;
        background: linear-gradient(120deg, #f6d365 0%, #fda085 100%);
        margin: 0;
        padding: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
    }
    form {
        background-color: #ffffff;
        padding: 30px;
        border-radius: 12px;
        box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
        max-width: 400px;
        width: 100%;
    }
    h1, h2 {
        text-align: center;
        color: #333333;
        margin: 0;
    }
    h1 {
        font-size: 2em;
        margin-bottom: 0.5em;
    }
    h2 {
        font-size: 1.5em;
        margin-bottom: 1em;
    }
    h3 {
        margin-bottom: 10px;
        color: #666666;
    }

```

```

}
input[type="text"], input[type="password"], input[type="email"], input[type="tel"] {
    width: calc(100% - 20px);
    padding: 12px;
    margin: 8px 0 20px 0;
    border: 1px solid #ddd;
    border-radius: 6px;
    font-size: 1em;
}
input[type="text"]::placeholder, input[type="password"]::placeholder, input[type="email"]::placeholder,
input[type="tel"]::placeholder {
    color: #bbb;
}
.button {
    width: 100%;
    background-color: #ff7e5f;
    color: white;
    padding: 14px;
    border: none;
    border-radius: 6px;
    cursor: pointer;
    font-size: 1em;
    transition: background-color 0.3s ease;
}
.button:hover {
    background-color: #eb6b4d;
}
.link-button {
    background-color: transparent;
    color: #007bff;
    border: none;
    cursor: pointer;
    text-align: center;
    display: block;
    margin: 10px 0;
    text-decoration: underline;
    font-size: 1em;
}
.link-button:hover {
    color: #0056b3;
}
</style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h1><strong>Library Management System</strong></h1>
            <h2>Admin Sign up</h2>
            <h3>Enter your Name:</h3>
            <asp:TextBox ID="name" placeholder="Username" runat="server"></asp:TextBox>
            <h3>Enter your Email:</h3>
            <asp:TextBox ID="mail" placeholder="Email Address" runat="server"></asp:TextBox>
            <h3>Enter your Phone Number:</h3>
            <asp:TextBox ID="phone" placeholder="Number" Type="tel" runat="server"></asp:TextBox>
            <h3>Enter your Address:</h3>
            <asp:TextBox ID="address" placeholder="Address" Type="text" runat="server"></asp:TextBox>
            <h3>Enter your Password:</h3>
            <asp:TextBox ID="pswd" placeholder="Password" type="password" runat="server"></asp:TextBox>
        </div>
    </form>
</body>

```



```

        <br />
        <asp:Button ID="Signup" runat="server" Text="Sign up" CSSClass="button" OnClick="Signup_Click"
/>
    <br />
</div>
</form>
</body>
</html>

```

Admin-signup.aspx.vb:

```

Imports System.Data.SqlClient
Imports System.Security.Cryptography

```

```

Partial Class _Default

```

```

    Inherits System.Web.UI.Page

```

```

    Protected Sub Signup_Click(sender As Object, e As EventArgs) Handles Signup.Click

```

```

        If String.IsNullOrEmpty(name.Text) OrElse String.IsNullOrEmpty(phone.Text) OrElse
            String.IsNullOrEmpty(mail.Text) OrElse String.IsNullOrEmpty(address.Text) OrElse
            String.IsNullOrEmpty(pswd.Text) Then
            Response.Write("<script>alert('Please fill in all fields.')

```

```

        If Not IsValidPassword(pswd.Text) Then
            Response.Write("<script>alert('Password must be at least 8 characters long and include at least 1
uppercase letter, 1 lowercase letter, 1 number, and 1 special character.')

```

```

        If CheckPhoneNumberExists(phone.Text) Then
            Response.Write("<script>alert('Phone number already in use.')

```

```

        If CheckEmailExists(mail.Text) Then
            Response.Write("<script>alert('Email already in use.')

```

```

        Dim encryptedPassword As String = EncryptPassword(pswd.Text)
        Dim connString As String = ConfigurationManager.ConnectionStrings("conn").ConnectionString

```

```

        Using conn As New SqlConnection(connString)
            Dim query As String = "INSERT INTO ADMIN (Name, Email, Address, PhoneNo, Password) " &
                "VALUES (@Name, @Email, @Address, @PhoneNo, @Password)"
            Using cmd As New SqlCommand(query, conn)
                cmd.Parameters.AddWithValue("@Name", name.Text)
                cmd.Parameters.AddWithValue("@Email", mail.Text)
                cmd.Parameters.AddWithValue("@Address", address.Text)
                cmd.Parameters.AddWithValue("@PhoneNo", phone.Text)
                cmd.Parameters.AddWithValue("@Password", encryptedPassword)
                conn.Open()
                cmd.ExecuteNonQuery()
            End Using
        End Using
        Response.Redirect("Admin.aspx")

```

End Sub

```
Private Function CheckPhoneNumberExists(phoneNumber As String) As Boolean
    Dim connString As String = ConfigurationManager.ConnectionStrings("conn").ConnectionString
    Dim query As String = "SELECT COUNT(*) FROM ADMIN WHERE PhoneNo = @PhoneNo"
    Using conn As New SqlConnection(connString)
        Using cmd As New SqlCommand(query, conn)
            cmd.Parameters.AddWithValue("@PhoneNo", phoneNumber)
            conn.Open()
            Dim count As Integer = Convert.ToInt32(cmd.ExecuteScalar())
            Return count > 0
        End Using
    End Using
End Function
```

```
Private Function CheckEmailExists(email As String) As Boolean
    Dim connString As String = ConfigurationManager.ConnectionStrings("conn").ConnectionString
    Dim query As String = "SELECT COUNT(*) FROM ADMIN WHERE Email = @Email"
    Using conn As New SqlConnection(connString)
        Using cmd As New SqlCommand(query, conn)
            cmd.Parameters.AddWithValue("@Email", email)
            conn.Open()
            Dim count As Integer = Convert.ToInt32(cmd.ExecuteScalar())
            Return count > 0
        End Using
    End Using
End Function
```

```
Private Function EncryptPassword(password As String) As String
    Using sha256 As New SHA256Managed()
        Dim bytes As Byte() = Encoding.UTF8.GetBytes(password)
        Dim hash As Byte() = sha256.ComputeHash(bytes)
        Dim sb As New StringBuilder()
        For Each b As Byte In hash
            sb.Append(b.ToString("x2"))
        Next
        Return sb.ToString()
    End Using
End Function
```

```
Private Function IsValidPassword(password As String) As Boolean
    Dim hasMinimum8Chars As New Regex(".{8,}")
    Dim hasUpperCaseLetter As New Regex("[A-Z]")
    Dim hasLowerCaseLetter As New Regex("[a-z]")
    Dim hasDecimalDigit As New Regex("[0-9]")
    Dim hasSpecialCharacter As New Regex("[^a-zA-Z0-9]")

    Return hasMinimum8Chars.IsMatch(password) AndAlso
        hasUpperCaseLetter.IsMatch(password) AndAlso
        hasLowerCaseLetter.IsMatch(password) AndAlso
        hasDecimalDigit.IsMatch(password) AndAlso
        hasSpecialCharacter.IsMatch(password)
End Function
```

End Class

Member-login.aspx:

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Member-login.aspx.vb" Inherits="_Default" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
<title>Member Login</title>
```

```
<style>
```

```
body {
```

```
font-family: 'Roboto', sans-serif;
```

```
background: linear-gradient(120deg, #f6d365 0%, #fda085 100%);
```

```
margin: 0;
```

```
padding: 0;
```

```
display: flex;
```

```
justify-content: center;
```

```
align-items: center;
```

```
height: 100vh;
```

```
}
```

```
form {
```

```
background-color: #ffffff;
```

```
padding: 30px;
```

```
border-radius: 12px;
```

```
box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
```

```
max-width: 400px;
```

```
width: 100%;
```

```
}
```

```
h1, h2 {
```

```
text-align: center;
```

```
color: #333333;
```

```
margin: 0;
```

```
}
```

```
h1 {
```

```
font-size: 2em;
```

```
margin-bottom: 0.5em;
```

```
}
```

```
h2 {
```

```
font-size: 1.5em;
```

```
margin-bottom: 1em;
```

```
}
```

```
h3 {
```

```
margin-bottom: 10px;
```

```
color: #666666;
```

```
}
```

```
input[type="text"], input[type="password"] {
```

```
width: calc(100% - 20px);
```

```
padding: 12px;
```

```
margin: 8px 0 20px 0;
```

```
border: 1px solid #ddd;
```

```
border-radius: 6px;
```

```
font-size: 1em;
```

```
}
```

```
input[type="text"]::placeholder, input[type="password"]::placeholder {
```

```
color: #bbb;
```

```
}
```

```

.button {
    width: 100%;
    background-color: #ff7e5f;
    color: white;
    padding: 14px;
    border: none;
    border-radius: 6px;
    cursor: pointer;
    font-size: 1em;
    transition: background-color 0.3s ease;
}
.button:hover {
    background-color: #eb6b4d;
}
@media (max-width: 600px) {
    form {
        padding: 20px;
    }
    input[type="text"], input[type="password"] {
        padding: 10px;
    }
    .button {
        padding: 12px;
        font-size: 0.9em;
    }
}
</style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h1><strong>Library Management System</strong></h1>
            <h2>Member Login Page</h2>
            <h3>Enter your Email:</h3>
            <asp:TextBox ID="email" placeholder="Email" runat="server"></asp:TextBox>
            <h3>Enter your Password:</h3>
            <asp:TextBox ID="password" placeholder="Password" type="password"
runat="server"></asp:TextBox>
            <asp:Button ID="login" runat="server" Text="Login" CssClass="button" OnClick="login_Click"/>
            <br />
            <br />
            <br />
            Don't have account!
            <br />
            <asp:Button ID="Signup" runat="server" Text="Signup" CssClass="button" OnClick="Signup_Click" />
        </div>
    </form>
</body>
</html>

```

Member-login.aspx.vb:

```

Imports System.Data.SqlClient
Imports System.Security.Cryptography
Partial Class _Default
    Inherits System.Web.UI.Page
    Protected Sub login_Click(ByVal sender As Object, ByVal e As EventArgs) Handles login.Click

```

```

If String.IsNullOrEmpty(email.Text) Or String.IsNullOrEmpty(password.Text) Then
    Response.Write("<script>alert('Both fields are required.');

```

Member-signup.aspx:

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Member-signup.aspx.vb"
Inherits="_Default" %>
```

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Member Signup</title>
  <style>
    body {
      font-family: 'Roboto', sans-serif;
      background: linear-gradient(120deg, #f6d365 0%, #fda085 100%);
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    form {
      background-color: #ffffff;
      padding: 30px;
      border-radius: 12px;
      box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
      max-width: 400px;
      width: 100%;
    }
    h1, h2 {
      text-align: center;
      color: #333333;
      margin: 0;
    }
    h1 {
      font-size: 2em;
      margin-bottom: 0.5em;
    }
    h2 {
      font-size: 1.5em;
      margin-bottom: 1em;
    }
    h3 {
      margin-bottom: 10px;
      color: #666666;
    }
    input[type="text"], input[type="password"] {
      width: calc(100% - 20px);
      padding: 12px;
      margin: 8px 0 20px 0;
      border: 1px solid #ddd;
      border-radius: 6px;
      font-size: 1em;
    }
    input[type="text"]::placeholder, input[type="password"]::placeholder {
      color: #bbb;
    }
    .button {
```

```

width: 100%;
background-color: #ff7e5f;
color: white;
padding: 14px;
border: none;
border-radius: 6px;
cursor: pointer;
font-size: 1em;
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #eb6b4d;
}
}
.link-button {
background-color: transparent;
color: #ff7e5f;
border: none;
cursor: pointer;
text-align: center;
display: block;
margin: 15px 0;
text-decoration: underline;
font-size: 1em;
}
.link-button:hover {
color: #eb6b4d;
}
}
@media (max-width: 600px) {
form {
padding: 20px;
}
input[type="text"], input[type="password"] {
padding: 10px;
}
.button {
padding: 12px;
font-size: 0.9em;
}
}
}
</style>
</head>
<body>
<form id="form1" runat="server">
<div>
<h1><strong>Library Management System</strong></h1>
<h2><strong>Member Signup Page</strong></h2>
<h3>Enter your Name:</h3>
<asp:TextBox ID="name" placeholder="Name" runat="server"></asp:TextBox>
<h3>Enter your Email:</h3>
<asp:TextBox ID="mail" placeholder="Email" runat="server"></asp:TextBox>
<h3>Enter your Phone Number:</h3>
<asp:TextBox ID="phone" placeholder="Phone Number" runat="server"></asp:TextBox>
<h3>Enter your Address:</h3>
<asp:TextBox ID="address" placeholder="Address" runat="server"></asp:TextBox>
<h3>Enter your Password:</h3>
<asp:TextBox ID="pswd" placeholder="Password" type="password" runat="server"></asp:TextBox>
<asp:Button ID="Signup" runat="server" Text="Sign up" CssClass="button" onclick="Signup_Click"/>

```

```

    </div>
</form>
</body>
</html>

```

Member-signup.aspx.vb:

```

Imports System.Data.SqlClient
Imports System.Security.Cryptography

```

```

Partial Class _Default
    Inherits System.Web.UI.Page

```

```

    Protected Sub Signup_Click(sender As Object, e As EventArgs) Handles Signup.Click
        If String.IsNullOrEmpty(name.Text) OrElse String.IsNullOrEmpty(phone.Text) OrElse
            String.IsNullOrEmpty(mail.Text) OrElse String.IsNullOrEmpty(address.Text) OrElse
            String.IsNullOrEmpty(pswd.Text) Then
            Response.Write("<script>alert('Please fill in all fields.')</script>")
            Exit Sub
        End If
        If Not IsValidPassword(pswd.Text) Then
            Response.Write("<script>alert('Password must be at least 8 characters long and include at least 1
uppercase letter, 1 lowercase letter, 1 number, and 1 special character.')</script>")
            Exit Sub
        End If
        If CheckPhoneNumberExists(phone.Text) Then
            Response.Write("<script>alert('Phone number already in use.')</script>")
            Exit Sub
        End If

        If CheckEmailExists(mail.Text) Then
            Response.Write("<script>alert('Email already in use.')</script>")
            Exit Sub
        End If
        Dim encryptedPassword As String = EncryptPassword(pswd.Text)
        Dim connString As String = ConfigurationManager.ConnectionStrings("conn").ConnectionString
        Dim query As String = "INSERT INTO MEMBER (Name, Email, Address, PhoneNo, Password) " &
            "VALUES (@Name, @Email, @Address, @PhoneNo, @Password)"
        Using conn As New SqlConnection(connString)
            Using cmd As New SqlCommand(query, conn)
                cmd.Parameters.AddWithValue("@Name", name.Text)
                cmd.Parameters.AddWithValue("@Email", mail.Text)
                cmd.Parameters.AddWithValue("@Address", address.Text)
                cmd.Parameters.AddWithValue("@PhoneNo", phone.Text)
                cmd.Parameters.AddWithValue("@Password", encryptedPassword)
                conn.Open()
                cmd.ExecuteNonQuery()
            End Using
        End Using
        Response.Redirect("Member-login.aspx")
    End Sub

```

```

Private Function CheckPhoneNumberExists(phoneNumber As String) As Boolean
    Dim connString As String = ConfigurationManager.ConnectionStrings("conn").ConnectionString
    Dim query As String = "SELECT COUNT(*) FROM MEMBER WHERE PhoneNo = @PhoneNo"
    Using conn As New SqlConnection(connString)
        Using cmd As New SqlCommand(query, conn)

```



```

        cmd.Parameters.AddWithValue("@PhoneNo", phoneNumber)
        conn.Open()
        Dim count As Integer = Convert.ToInt32(cmd.ExecuteScalar())
        Return count > 0
    End Using
End Using
End Function

Private Function CheckEmailExists(email As String) As Boolean
    Dim connString As String = ConfigurationManager.ConnectionStrings("conn").ConnectionString
    Dim query As String = "SELECT COUNT(*) FROM MEMBER WHERE Email = @Email"
    Using conn As New SqlConnection(connString)
        Using cmd As New SqlCommand(query, conn)
            cmd.Parameters.AddWithValue("@Email", email)
            conn.Open()
            Dim count As Integer = Convert.ToInt32(cmd.ExecuteScalar())
            Return count > 0
        End Using
    End Using
End Function

Private Function EncryptPassword(password As String) As String
    Using sha256 As New SHA256Managed()
        Dim bytes As Byte() = Encoding.UTF8.GetBytes(password)
        Dim hash As Byte() = sha256.ComputeHash(bytes)
        Dim sb As New StringBuilder()
        For Each b As Byte In hash
            sb.Append(b.ToString("x2"))
        Next
        Return sb.ToString()
    End Using
End Function

Private Function IsValidPassword(password As String) As Boolean
    Dim hasMinimum8Chars As New Regex("{8,}")
    Dim hasUpperCaseLetter As New Regex("[A-Z]")
    Dim hasLowerCaseLetter As New Regex("[a-z]")
    Dim hasDecimalDigit As New Regex("[0-9]")
    Dim hasSpecialCharacter As New Regex("[^a-zA-Z0-9]")

    Return hasMinimum8Chars.IsMatch(password) AndAlso
        hasUpperCaseLetter.IsMatch(password) AndAlso
        hasLowerCaseLetter.IsMatch(password) AndAlso
        hasDecimalDigit.IsMatch(password) AndAlso
        hasSpecialCharacter.IsMatch(password)
End Function
End Class

```

Manage-book.aspx:

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Manage_Books.aspx.vb"
Inherits="Manage_Books" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">

```

```

<title>Library Management System</title>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" />
</head>
<body>
  <form id="form1" runat="server">
    <div class="container my-5">
      <h1 class="text-center mb-4">Library Management System</h1>
      <br /> <br />
      <div class="form-row align-items-center">
        <div class="col-auto">
          <asp:TextBox ID="txtSearch" runat="server" CssClass="form-control" placeholder="Search
books..."></asp:TextBox>
        </div>
        <div class="col-auto">
          <asp:DropDownList ID="ddlCategories" runat="server" CssClass="form-control">
            <asp:ListItem Value="">Filter by category</asp:ListItem>
          </asp:DropDownList>
        </div>
        <div class="col-auto">
          <asp:Button ID="btnSearch" runat="server" Text="Search" CssClass="btn btn-primary" />
          <asp:Button ID="btnaddbook" runat="server" Text="Add Books" CssClass="btn btn-secondary" />
          <asp:Button ID="btndelete" runat="server" Text="Delete Books" CssClass="btn btn-secondary" />
        </div>
      </div>

      <div class="my-4">
        <asp:GridView ID="gvBooks" runat="server" CssClass="table table-striped table-bordered"
AutoGenerateColumns="false">
          <Columns>
            <asp:BoundField DataField="Title" HeaderText="Book Title" />
            <asp:BoundField DataField="Author" HeaderText="Author" />
            <asp:BoundField DataField="Category" HeaderText="Category" />
            <asp:BoundField DataField="Edition" HeaderText="Edition" />
            <asp:TemplateField HeaderText="Status">
              <ItemTemplate>
                <%# If(Convert.ToInt32(Eval("Book_Status")) = 0, "<span style='color: green; font-weight:
bold;'>Available</span>", "<span style='color: red; font-weight: bold;'>Issued</span>") %>
              </ItemTemplate>
            </asp:TemplateField>
          </Columns>
        </asp:GridView>
      </div>
    </div>
  </form>
</body>
</html>

```

Manage-book.aspx.vb:

```

Imports System.Data
Imports System.Data.SqlClient

Public Class Manage_Books
  Inherits System.Web.UI.Page

  Private Shared ReadOnly ConnectionString As String =
  ConfigurationManager.ConnectionStrings("conn").ConnectionString

```

```
Private Shared books As DataTable
```

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load  
    If Not IsPostBack Then  
        books = LoadBooks()  
        BindBookCategories()  
        BindBookGrid()  
    End If  
End Sub
```

```
Private Function LoadBooks() As DataTable  
    Dim booksTable As DataTable = New DataTable()  
    Dim query As String = "SELECT ISBN, Title, Author, Category, Edition, Book_Status FROM Book"  
    Using connection As New SqlConnection(ConnectionString)  
        Using command As New SqlCommand(query, connection)  
            Dim adapter As New SqlDataAdapter(command)  
            connection.Open()  
            adapter.Fill(booksTable)  
        End Using  
    End Using  
    Return booksTable  
End Function
```

```
Private Sub BindBookCategories()  
    Dim categoriesTable As New DataTable()  
    categoriesTable.Columns.Add("Category")  
  
    Dim uniqueCategories = books.AsEnumerable().  
        Select(Function(row) row.Field(Of String)("Category")).  
        Distinct()  
  
    For Each category As String In uniqueCategories  
        categoriesTable.Rows.Add(category)  
    Next  
  
    ddlCategories.DataSource = categoriesTable  
    ddlCategories.DataTextField = "Category"  
    ddlCategories.DataValueField = "Category"  
    ddlCategories.DataBind()  
    ddlCategories.Items.Insert(0, New ListItem("All", ""))  
End Sub
```

```
Private Sub BindBookGrid()  
    gvBooks.DataSource = books  
    gvBooks.DataBind()  
End Sub
```

```
Protected Sub btnSearch_Click(sender As Object, e As EventArgs) Handles btnSearch.Click  
    Dim searchText As String = txtSearch.Text  
    Dim selectedCategory As String = ddlCategories.SelectedValue  
  
    Dim filteredTable As DataTable = books.Clone()  
    Dim rows = books.AsEnumerable().Where(Function(row)  
        Dim matchesCategory = String.IsNullOrEmpty(selectedCategory) OrElse  
row.Field(Of String)("Category").Equals(selectedCategory, StringComparison.OrdinalIgnoreCase)  
        Dim matchesSearchText = String.IsNullOrEmpty(searchText) OrElse  
row.Field(Of String)("Title").IndexOf(searchText, StringComparison.OrdinalIgnoreCase) >= 0  
        Return matchesCategory AndAlso matchesSearchText
```

```

                End Function)
    For Each row In rows
        filteredTable.ImportRow(row)
    Next

    gvBooks.DataSource = filteredTable
    gvBooks.DataBind()
End Sub

Protected Sub btnaddbook_Click(sender As Object, e As EventArgs) Handles btnaddbook.Click
    Response.Redirect("Add_Book.aspx")
End Sub
Protected Sub btndelete_Click(sender As Object, e As EventArgs) Handles btndelete.Click
    Response.Redirect("Delete_Book.aspx")
End Sub

End Class

```

Manage Member.aspx:

```

<%@ Page Language="vb" AutoEventWireup="false" CodeFile="ManageMember.aspx.vb"
Inherits="ManageMember" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Manage Member</title>
    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet" />
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</head>
<body>
    <form id="form1" runat="server" class="container mt-5">
        <h2 class="mb-4">Manage Member</h2>

        <div class="mb-4">
            <h4>Member Information</h4>
            <asp:Label ID="lblMemberInfo" runat="server" Text="" CssClass="font-weight-bold"></asp:Label>
        </div>

        <div class="mb-4">
            <h4>Borrowed Books</h4>
            <asp:GridView ID="gvBorrowedBooks" runat="server" AutoGenerateColumns="False" CssClass="table
table-bordered table-striped">
                <Columns>
                    <asp:BoundField DataField="ISBN" HeaderText="ISBN" HeaderStyle-CssClass="bg-primary text-
white" ItemStyle-CssClass="align-middle" />
                    <asp:BoundField DataField="Title" HeaderText="Title" HeaderStyle-CssClass="bg-primary text-
white" ItemStyle-CssClass="align-middle" />
                    <asp:BoundField DataField="Author" HeaderText="Author" HeaderStyle-CssClass="bg-primary
text-white" ItemStyle-CssClass="align-middle" />
                    <asp:BoundField DataField="BorrowDate" HeaderText="Borrow Date" HeaderStyle-CssClass="bg-
primary text-white" ItemStyle-CssClass="align-middle" />
                    <asp:TemplateField HeaderText="Actions" HeaderStyle-CssClass="bg-primary text-white"
ItemStyle-CssClass="align-middle text-center">
                        <ItemTemplate>

```

```

                <asp:Button ID="btnRemoveBook" runat="server" Text="Remove" CssClass="btn btn-danger"
CommandArgument=<%# Bind("ISBN") %>' OnClick="RemoveBook" />
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
</div>

    <asp:Button ID="btnDeleteMember" runat="server" Text="Delete Member" CssClass="btn btn-danger"
OnClick="DeleteMember" />
    <asp:Label ID="lblErrorMessage" runat="server" ForeColor="Red" Visible="False"></asp:Label>
</form>
</body>
</html>

```

Manage Member .aspx.vb:

```

Imports System.Data.SqlClient
Imports System.Configuration
Imports System.Data

```

```

Partial Class ManageMember

```

```

    Inherits System.Web.UI.Page

```

```

    Private Shared ReadOnly ConnectionString As String =
ConfigurationManager.ConnectionStrings("conn").ConnectionString

```

```

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs) Handles Me.Load

```

```

        If Not IsPostBack Then

```

```

            ViewState("MemberID") = Session("MemberID")

```

```

            LoadMemberInfo()

```

```

            LoadBorrowedBooks()

```

```

        End If

```

```

    End Sub

```

```

    Private Sub LoadMemberInfo()

```

```

        Dim memberId As Integer = Convert.ToInt32(ViewState("MemberID"))

```

```

        Using conn As New SqlConnection(ConnectionString)

```

```

            Dim cmd As New SqlCommand("SELECT * FROM MEMBER WHERE MemberID = @MemberID",
conn)

```

```

            cmd.Parameters.AddWithValue("@MemberID", memberId)

```

```

            conn.Open()

```

```

            Dim rdr As SqlDataReader = cmd.ExecuteReader()

```

```

            If rdr.Read() Then

```

```

                lblMemberInfo.Text = String.Format("ID: {0}<br />Name: {1}<br />Email: {2}<br />Phone: {3}<br />Address: {4}",

```

```

                    rdr("MemberID"), rdr("Name"), rdr("Email"), rdr("PhoneNo"), rdr("Address"))

```

```

            End If

```

```

        End Using

```

```

    End Sub

```

```

    Private Sub LoadBorrowedBooks()

```

```

        Using conn As New SqlConnection(ConnectionString)

```

```

            Dim memberId As Integer = Convert.ToInt32(ViewState("MemberID"))

```

```

            Dim cmd As New SqlCommand("SELECT b.ISBN, Title, Author, Category, Edition, Book_Status,
BorrowDate FROM BOOK b JOIN BORROW br ON b.ISBN = br.ISBN JOIN MEMBER_BORROW mb ON

```

```

br.BorrowID = mb.BorrowID JOIN MEMBER m ON mb.MemberID = m.MemberID WHERE m.MemberID =
@MemberID AND br.ReturnDate IS NULL AND b.Book_Status = 1", conn)
    cmd.Parameters.AddWithValue("@MemberID", memberId)
    conn.Open()
    Dim rdr As SqlDataReader = cmd.ExecuteReader()

    Dim borrowedBooks As New DataTable()
    borrowedBooks.Load(rdr)

    gvBorrowedBooks.DataSource = borrowedBooks
    gvBorrowedBooks.DataBind()
End Using
End Sub

Protected Sub RemoveBook(ByVal sender As Object, ByVal e As EventArgs)
    Dim memberId As Integer = Convert.ToInt32(ViewState("MemberID"))
    Dim ISBN As String = (TryCast(sender, Button)).CommandArgument
    Dim returnDate As DateTime = DateTime.Now
    Return_Book(ISBN, returnDate)
    Update_Status(ISBN, 0)

    LoadBorrowedBooks()
End Sub

Private Sub Return_Book(ISBN As String, returnDate As DateTime)
    Dim memberId As Integer = Convert.ToInt32(ViewState("MemberID"))

    Dim return_query As String = "UPDATE BORROW SET ReturnDate = @returnDate FROM BORROW b
INNER JOIN MEMBER_BORROW mb ON b.BorrowID = mb.BorrowID WHERE b.ReturnDate IS NULL
AND mb.MemberID = @MemberID AND b.ISBN = @ISBN"
    Using connection As New SqlConnection(ConnectionString)
        Using command As New SqlCommand(return_query, connection)
            command.Parameters.AddWithValue("@ISBN", ISBN)
            command.Parameters.AddWithValue("@MemberID", memberId)
            command.Parameters.AddWithValue("@returnDate", returnDate)
            connection.Open()
            command.ExecuteNonQuery()
        End Using
    End Using
End Sub

Private Sub Update_Status(ISBN As String, status As Integer)
    Dim memberId As Integer = Convert.ToInt32(ViewState("MemberID"))

    Dim updatequery As String = "UPDATE BOOK SET Book_Status = @status WHERE ISBN = @ISBN;"
    Using connection As New SqlConnection(ConnectionString)
        Using command As New SqlCommand(updatequery, connection)
            command.Parameters.AddWithValue("@ISBN", ISBN)
            command.Parameters.AddWithValue("@status", status)
            connection.Open()
            command.ExecuteNonQuery()
        End Using
    End Using
End Sub

Protected Sub DeleteMember(ByVal sender As Object, ByVal e As EventArgs)
    Dim memberId As Integer = Convert.ToInt32(ViewState("MemberID"))
    Using conn As New SqlConnection(ConnectionString)

```

```

conn.Open()

' Check if there are issued books for the member
Dim checkCmd As New SqlCommand("SELECT COUNT(b.BorrowID) AS BooksIssued FROM
MEMBER m LEFT JOIN MEMBER_BORROW mb ON m.MemberID = mb.MemberID LEFT JOIN
BORROW b ON mb.BorrowID = b.BorrowID AND b.ReturnDate IS NULL WHERE m.MemberID =
@MemberID GROUP BY m.MemberID, m.Name, m.Email, m.PhoneNo ORDER BY m.MemberID", conn)
checkCmd.Parameters.AddWithValue("@MemberID", memberId)
Dim issuedBooksCount As Integer = Convert.ToInt32(checkCmd.ExecuteScalar())

If issuedBooksCount > 0 Then
' Show error message if there are issued books
lblErrorMessage.Text = "This member has issued books and cannot be deleted."
lblErrorMessage.Visible = True
Else
' Delete member if no issued books
Dim deleteCmd As New SqlCommand("DELETE FROM MEMBER WHERE MemberID =
@MemberID", conn)
deleteCmd.Parameters.AddWithValue("@MemberID", memberId)
deleteCmd.ExecuteNonQuery()
Response.Redirect("Manage_Members.aspx")
End If
End Using
End Sub

End Class

```

Delete Book .aspx:

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Delete_Book.aspx.vb"
Inherits="Delete_Book" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Library Management System</title>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" />
</head>
<body>
<form id="form1" runat="server">
<div class="container my-5">
<h1 class="text-center mb-4">Library Management System</h1>
<br /> <br />
<div class="form-row align-items-center">
<div class="col-auto">
<asp:TextBox ID="txtSearch" runat="server" CssClass="form-control" placeholder="Search
books..."></asp:TextBox>
</div>
<div class="col-auto">
<asp:DropDownList ID="ddlCategories" runat="server" CssClass="form-control">
<asp:ListItem Value="">Filter by category</asp:ListItem>
</asp:DropDownList>
</div>
<div class="col-auto">

```

```

        <asp:Button ID="btnSearch" runat="server" Text="Search" CssClass="btn btn-primary" />
        <asp:Button ID="back" runat="server" Text="Return to Dashboard" CssClass="btn btn-secondary"
/>
    </div>
</div>

<div class="my-4">
    <asp:GridView ID="gvBooks" runat="server" CssClass="table table-striped table-bordered"
AutoGenerateColumns="false">
    <Columns>
        <asp:BoundField DataField="Title" HeaderText="Book Title" />
        <asp:BoundField DataField="Author" HeaderText="Author" />
        <asp:BoundField DataField="Category" HeaderText="Category" />
        <asp:BoundField DataField="Edition" HeaderText="Edition" />
        <asp:TemplateField HeaderText="Status">
            <ItemTemplate>
                <asp:Label ID="lblStatus" runat="server" />
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Actions">
            <ItemTemplate>
                <asp:Button ID="btnDelete" runat="server" Text="Delete" CssClass="btn btn-danger btn-
sm" CommandArgument="<%= # Bind("ISBN") %>" OnClick="btnDelete_Click" />
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
</div>
</div>
</form>
</body>
</html>

```

Delete book .aspx.vb:

```

Imports System.Data
Imports System.Data.SqlClient

Public Class Delete_Book
    Inherits System.Web.UI.Page

    Private Shared ReadOnly ConnectionString As String =
ConfigurationManager.ConnectionStrings("conn").ConnectionString
    Private Shared books As DataTable
    Dim Member_Id As Integer = Convert.ToInt32(2) 'Session("MemberId")

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        If Not IsPostBack Then
            books = LoadBooks()
            BindBookCategories()
            BindBookGrid()
        End If
    End Sub

    Private Function LoadBooks() As DataTable
        Dim booksTable As DataTable = New DataTable()
        Dim query As String = "SELECT ISBN, Title, Author, Category, Edition, Book_Status FROM Book"

```



```

    Using connection As New SqlConnection(ConnectionString)
    Using command As New SqlCommand(query, connection)
        Dim adapter As New SqlDataAdapter(command)
        connection.Open()
        adapter.Fill(booksTable)
    End Using
End Using
Return booksTable
End Function

Private Sub BindBookCategories()
    Dim categoriesTable As New DataTable()
    categoriesTable.Columns.Add("Category")

    Dim uniqueCategories = books.AsEnumerable().
        Select(Function(row) row.Field(Of String)("Category")).
        Distinct()

    For Each category As String In uniqueCategories
        categoriesTable.Rows.Add(category)
    Next

    ddlCategories.DataSource = categoriesTable
    ddlCategories.DataTextField = "Category"
    ddlCategories.DataValueField = "Category"
    ddlCategories.DataBind()
    ddlCategories.Items.Insert(0, New ListItem("All", ""))
End Sub

Private Sub BindBookGrid()
    gvBooks.DataSource = books
    gvBooks.DataBind()
End Sub

Protected Sub btnDelete_Click(sender As Object, e As EventArgs)
    Dim ISBN As String = (TryCast(sender, Button)).CommandArgument
    Delete_Book(ISBN)
    books = LoadBooks()
    BindBookCategories()
    BindBookGrid()
End Sub

Private Sub Delete_Book(ISBN As String)
    Dim query As String = "DELETE FROM BOOK WHERE ISBN = @ISBN AND Book_Status = 0"
    Using connection As New SqlConnection(ConnectionString)
        Using command As New SqlCommand(query, connection)
            command.Parameters.AddWithValue("@ISBN", ISBN)
            connection.Open()
            command.ExecuteNonQuery()
        End Using
    End Using
End Sub

Protected Sub btnSearch_Click(sender As Object, e As EventArgs) Handles btnSearch.Click
    Dim searchText As String = txtSearch.Text
    Dim selectedCategory As String = ddlCategories.SelectedValue

    Dim filteredTable As DataTable = books.Clone()
    Dim rows = books.AsEnumerable().Where(Function(row)

```

```

        Dim matchesCategory = String.IsNullOrEmpty(selectedCategory) OrElse
row.Field(Of String)("Category").Equals(selectedCategory, StringComparison.OrdinalIgnoreCase)
        Dim matchesSearchText = String.IsNullOrEmpty(searchText) OrElse
row.Field(Of String)("Title").IndexOf(searchText, StringComparison.OrdinalIgnoreCase) >= 0
        Return matchesCategory AndAlso matchesSearchText
    End Function)
    For Each row In rows
        filteredTable.ImportRow(row)
    Next

    gvBooks.DataSource = filteredTable
    gvBooks.DataBind()
End Sub

Protected Sub back_Click(sender As Object, e As EventArgs) Handles back.Click
    Response.Redirect("Admin.aspx")
End Sub

Protected Sub gvBooks_RowDataBound(sender As Object, e As GridViewRowEventArgs) Handles
gvBooks.RowDataBound
    If e.Row.RowType = DataControlRowType.DataRow Then
        Dim lblStatus As Label = TryCast(e.Row.FindControl("lblStatus"), Label)
        Dim btnDelete As Button = TryCast(e.Row.FindControl("btnDelete"), Button)
        Dim isBorrowed As Boolean = Convert.ToBoolean(DataBinder.Eval(e.Row.DataItem, "Book_Status"))

        If isBorrowed Then
            btnDelete.Visible = False
            lblStatus.Text = "<span style='color: red; font-weight: bold;'>Issued</span>"
        Else
            lblStatus.Text = "<span style='color: green; font-weight: bold;'>Available</span>"
        End If
    End If
End Sub
End Class

```