**UNIT 6 PROJECT**

**THE ONLINE BOOK CLUB**

**DOCUMENTATION**

**NATHAN JAMES 2019**

**PROGRAM DISCRIPTION:**

**The Online Book club is Web application/Website that allows people to register/login/update details that allows the user to proceed to either rate a book that exists within the database and allow the user that is logged in to add their own books and rate them, NB. Every user is only allowed to rate a book once and a user can only register once with a specific email**

**If the User wishes to Register their account Details like Name/surname /email/security question/Answer and Password will be asked once the Register is Successful, the User will be Redirected to the Login Page where the user has to login with the details they’ve just registered**

**If the User runs into the situation of entering the wrong Password the user can change their password by clicking on the forgot password link that will ask the user to enter their email with the program checking If the specific user account exists if not the user has to Register but if the email does exists their security Question will be displayed asking the user to enter the Answer they used if its correct the user will be allowed to update their Password if wrong a Error will be displayed**

**Once the specific user is Logged in, they have option of viewing Personal Details with an extra detail of Updating it**

**Software Used:**

**Visual Studio 2017/2019 and Microsoft access**

**File paths that need to be changed:**

**The Database needs to be Changed from within the Webservice to find to correct file path click on solution explorer then on App-Data folder then select the Database click on properties and copy the file Path and paste it inside the webservice**

**By a variable Named: String DbPath = “file Path”;**

**//////////////////////////////////////////////////////////////////////////////////////////////**

**Stream writer file path needs to be changed aswell:**

**Select the streamWritter text file inside the project folder go to properties and copy the file path**

**Then go to the Webservice source code by the AddRating Region then add the new file path**

**NB. Note that the stream file path does not contain and end to ensure the streamwritter works**

**Add filePath/StreamWritter.txt at the end of every streamwritter file path**

**How to run the Program**

**Open the project folder**

**Click on the program file named:**

**Press F5 or click the start button to run the project the Login page is saved as the start-up options page so if you wish to run the Web Service itself**

**Go to the project tab click on start-up options and select specific page**

**Source code**

**WEBSERVICE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Services;

using System.Data;

using System.Data.OleDb;

using System.Text;

using System.Web.Services.Protocols;

using System.IO;//For streamwritter files

/// <summary>

/// Summary description for BookStoreWebService

/// </summary>

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class BookStoreWebService : System.Web.Services.WebService

{

OleDbCommand Cmd;

OleDbConnection Con;

OleDbDataReader Reader;

string dbPath = "C:/Users/f9f6qth22/Desktop/BookClubProject (2)/BookClubProject/BookClubProject/App\_Data/BookClubDB.accdb";

public BookStoreWebService()

{

//Uncomment the following line if using designed components

//InitializeComponent();

}

#region Creating Connection To DB

[WebMethod(Description = "Creating a Connection to the Database")]

public bool ConnectToDB()

{

try

{

string ConString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source =" + dbPath;

Con = new OleDbConnection(ConString);

//THE FILE PATH FOR THE DATABASE WILL NEED TO BE CHANGED WHEN USING ANOTHER PC TO WHERE IT WILL BE LOCATED

Con.Open();

return true;

//Opening the connection to the Database

}

catch (OleDbException)

{

return false;

}

}

#endregion

#region Testing Connnection

[WebMethod(Description = "Testing the Database Connection")]

public bool TestDB()

{

//Testing If the Connection Works if not An Error will pop up Stating False

bool Status;

try

{

ConnectToDB();

Status = true;

}

catch (OleDbException)

{

Status = false;

}

return Status;

}

#endregion

#region Closing Connection to DB

[WebMethod(Description = "Closing/disconnecting the Database")]

public bool DisconnectDB()

{

try

{

ConnectToDB();

Con.Close();

return true;

}

catch (Exception)

{

return false;

}

//Closes the connection to the Database

}

#endregion

#region Checks if User email Exists

[WebMethod(Description = "Checking if email exists")]

public bool CheckUserExists(string Email)

{

try

{

//Checking If user exists by Receiving Data from the Database and Matching It with the values Entered by the User

ConnectToDB();

string QueryString = "SELECT \* FROM [User] WHERE Email='" + Email + "';";

Cmd = new OleDbCommand(QueryString, Con);

Cmd.Connection = Con;

Reader = Cmd.ExecuteReader();

if (Reader.HasRows && Reader != null)

{

//Email Exists

return true;

}

else

{

return false;

}

}

catch (Exception)

{

return false;

}

}

#endregion

#region Creating Register

//Web method used to register a new user

[WebMethod(Description = "Regsitering a new user to the DB")]

//Declaring the variables used in the DB

public bool Register(string Email,string Name,string Surname,string SecurityQ,string SecurityA,string Password)

{

try

{

//The following Code allows Data to be Entered into the database also known as Registration

ConnectToDB();

Cmd = Con.CreateCommand();

//Creating a Small string Value where the insert Statment is Created

string StrCmdTxt = "INSERT INTO [User] VALUES('" + Email + "','" + Name + "','" + Surname + "','" + SecurityQ + "','" + SecurityA + "','" + Password + "')";

Cmd = new OleDbCommand(StrCmdTxt, Con);

Cmd.Connection = Con;

Cmd.CommandText = StrCmdTxt;

Cmd.ExecuteNonQuery();

return true;

}

catch(Exception)

{

return false;

}

}

#endregion

#region Checking Login

[WebMethod(Description = "Creating a Login")]

public bool Login(string Email, string Password)

{

bool Result = false;

try

{

//

int counter = 0;

ConnectToDB();

Cmd = new OleDbCommand();

Cmd.Connection = Con;

Con.CreateCommand();

//The Selecct Statement will recieve The Specfic User info and match it if true the User logs in Else an Error message will pop up

string QueryString = "SELECT \* FROM [User] WHERE Email ='";

QueryString += Email + "'AND Password ='";

QueryString += Password + "'";

Cmd.CommandText = QueryString;

Reader = Cmd.ExecuteReader();

while (Reader.Read())

{

counter++;

}

if (counter == 1)

{

Result = true;

}

else

{

Result = false;

}

Con.Close();

}

catch (OleDbException)

{

return false;

}

return Result;

}

#endregion

#region Checking if a Book Exists

[WebMethod(Description = "Checking if book exists ")]

public bool BookExists(string Name, string Author, string Genre)

{

ConnectToDB();

try

{

Cmd = new OleDbCommand();

Cmd = Con.CreateCommand();

Cmd.CommandText = "SELECT \* FROM Books WHERE BookName ='" + Name + "'AND [BookAuthor] = '" + Author + "' AND [BookGenre] = '" + Genre + "'";

Reader = Cmd.ExecuteReader();

if (Reader.Read() && Reader != null)

{

return true;

}

else

{

return false;

}

}

catch (OleDbException)

{

return false;

}

}

#endregion

#region Adding a new Book

[WebMethod(Description = "Adding a new Book")]

public bool AddBook(string Name, string Author, string Genre)

{

ConnectToDB();

try

{

Cmd = new OleDbCommand();

Cmd = Con.CreateCommand();

Cmd.CommandText = "INSERT INTO Books(BookName,BookAuthor,BookGenre) VALUES ('" + Name + "','" + Author + "','" + Genre + "')";

Cmd.ExecuteNonQuery();

Con.Close();

return true;

}

catch (OleDbException)

{

return false;

}

}

#endregion

#region Checking if BookID Exists

[WebMethod]

public bool BookIDExists(int ID)

{

ConnectToDB();

try

{

Cmd = Con.CreateCommand();

Cmd.CommandText = "SELECT \* FROM Books WHERE BookID = " + ID + "";

Reader = Cmd.ExecuteReader();

if (Reader.Read() && Reader != null)

{

return true;

}

else

{

return false;

}

}

catch (Exception)

{

return false;

}

}

#endregion//This test is All Commented out

#region Checking if User Already Rated

[WebMethod(Description = "Already Rated ")]

public bool AlreadyRated(string Email, int ID)

{

ConnectToDB();

string StrEmail = Email;

Cmd = new OleDbCommand();

Cmd = Con.CreateCommand();

string strCmdQuery = @"SELECT \* FROM [Ratings] WHERE [bookID] = " + ID + " and [Email] = '" + StrEmail + "';";

Cmd.CommandText = strCmdQuery;

Reader = Cmd.ExecuteReader();

if (Reader.Read() && Reader != null)

{

return true;

}

else

{

return false;

}

//}

//catch (Exception)

//{

// return false;

//}

}

#endregion

#region Adding a Rating

[WebMethod(Description ="Adding a Rating")]

public bool Rating(string Email,int BookID, int Rating )

{

ConnectToDB();

string StrEmail = Email;

Cmd = Con.CreateCommand();

Cmd.Connection = Con;

string strRatingText = "INSERT INTO [Ratings] VALUES('" + BookID + "','" + StrEmail+ "','" + Rating + "')";

Cmd.CommandText = strRatingText;

Cmd = new OleDbCommand(strRatingText, Con);

Cmd.ExecuteNonQuery();

Con.Close();

//if the Rating was successful it will be stored in the Database then copied into a TextFile

FileStream fs = new FileStream("C:/Users/f9f6qth22/Desktop/BookClubProject (2)/BookClubProject/SteamWritter.txt", FileMode.Append);

BufferedStream bs = new BufferedStream(fs);

fs.Close();

StreamWriter sw = new StreamWriter("C:/Users/f9f6qth22/Desktop/BookClubProject (2)/BookClubProject/StreamWritter.txt",true);

sw.WriteLine("Email:" + Session["Email"] + ",BookID : " + BookID + ", Rating : " + Rating);

sw.Close();

return true;

}

#endregion

#region Getting Ammount Rated

[WebMethod(Description = "getting Amount Rated")]

public int GetAmountRated(int ID )

{

ConnectToDB();

int AmountRated =0;

Cmd = Con.CreateCommand();

Cmd.CommandText = "SELECT AmountRated FROM [Books] WHERE BookID =" + ID;

Reader = Cmd.ExecuteReader();

if (Reader.Read() && Reader != null)

{

//try

//{

AmountRated = Reader.GetInt32(0);

//}

//catch (Exception)

//{

// throw;

//}

}

return AmountRated;

}

#endregion

#region Getting Sum Rated

[WebMethod(Description ="getting Sum of rating ")]

public int GetSum(int ID )

{

ConnectToDB();

int sum = 1;

Cmd = Con.CreateCommand();

Cmd.CommandText = @"SELECT [SumRated] FROM [Books] WHERE BookID =" + ID;

Reader = Cmd.ExecuteReader();

if(Reader.Read()&& Reader != null)

{

//try

//{

sum = Reader.GetInt32(0);

return sum++;

//}

//catch(Exception)

//{

// throw;

//}

}

return sum;

}

#endregion

#region UpdateSumRated

[WebMethod(Description ="Updating Sum")]

public bool UpdateSum(int ID, int Sum)

{

ConnectToDB();

//try

//{

Cmd = Con.CreateCommand();

Cmd.CommandText = "UPDATE [Books] SET SumRated = " + Sum + " WHERE (BookID =" + ID +")";

Cmd.ExecuteNonQuery();

Con.Close();

return true;

//}

//catch(Exception)

//{

// return false;

//}

}

#endregion

#region UpdateAmountRated

[WebMethod(Description ="Updating the Amount Rated")]

public bool UpdateAmountRated(int ID, int Amount)

{

ConnectToDB();

//try

//{

Cmd = Con.CreateCommand();

Cmd.CommandText = "UPDATE [Books] SET AmountRated = " + Amount + " WHERE (BookID =" + ID +")" ;

Cmd.ExecuteNonQuery();

Con.Close();

return true;

//}

//catch(Exception)

//{

// return false;

//}

}

#endregion

#region UpdateAverageRating

[WebMethod(Description = "Updating Average Rating ")]

public bool UpdateAverageRating(int ID, int Average)

{

ConnectToDB();

//Once the Server collects the total Amount Rated plus the of Ratings The Update for the Average will happen

Cmd = Con.CreateCommand();

Cmd.CommandText = "UPDATE [Books] SET AverageRating = " + Average + " WHERE (BookID =" + ID +")";

Cmd.ExecuteNonQuery();

Con.Close();

return true;

//catch(Exception)

//{

// return false;

//}

}

#endregion

#region Viewing Profile

[WebMethod(Description ="Viewing User Details")]

public DataTable Profiles(string email)

{

ConnectToDB();

try

{

//A dataSet is Created with a Datatable that recieves the User (LoggedIn) Details and displays it

string strEmail = email;

DataSet ds = new DataSet();

OleDbDataAdapter dba = new OleDbDataAdapter("SELECT \* FROM [User] WHERE Email ='" + strEmail + "'",Con);

dba.Fill(ds, "User");

DataTable dt = ds.Tables["User"];

return dt;

}

catch (Exception)

{

return null;

}

}

#endregion

#region Updating Name

[WebMethod(Description = "Updating Name")]

public bool UpdateName(string Email, string Name)

{

ConnectToDB();

try

{

Cmd = Con.CreateCommand();

Cmd.CommandText = "UPDATE [User] SET [Name] = '" + Name + "' WHERE (Email ='" + Email+"');";

Cmd.ExecuteNonQuery();

Con.Close();

return true;

}

catch(Exception)

{

return false;

}

}

#endregion

#region UpdateSurname

[WebMethod(Description = "Updating Surname")]

public bool UpdateSurname(string Email, string Surname)

{

ConnectToDB();

try

{

Cmd = Con.CreateCommand();

Cmd.CommandText = "UPDATE [User] SET [Surname] ='" + Surname + "' WHERE (Email ='" +Email+"');" ;

Cmd.ExecuteNonQuery();

Con.Close();

return true;

}

catch(Exception)

{

return false;

}

}

#endregion

#region Retrieving User Question

[WebMethod]

public string getQuestion(string Email)

{

ConnectToDB();

string question = "null";

//For the User to be able to update Their Details(password)Data will be Selected from the Database only if the User Email Exists and if The Answer Matches the Question

Cmd = Con.CreateCommand();

Cmd.CommandText = @"SELECT SecurityQ FROM [User] WHERE Email = '" + Email + "'";

Reader = Cmd.ExecuteReader();

if (Reader.Read()&&Reader!=null)

{

try

{

question = Reader.GetString(0);

}

catch (Exception)

{

throw;

}

}

return question;

}

#endregion

#region Retreiving User Answer

[WebMethod]

public bool AnswerCorrect(string Email, string answer)

{

ConnectToDB();

try

{

//Once the User Recieves there Question and gives an Answer the Select Statement will Check the Database

//Check if the Answer Matches what the user has Written Down Else an Error will Pop up

Cmd = Con.CreateCommand();

Cmd.Connection = Con;

string QueryTest = @"SELECT \* FROM [User] WHERE Email ='";

QueryTest+= Email +"'AND SecurityA ='";

QueryTest += answer + "'";

Cmd.CommandText = QueryTest;

Reader = Cmd.ExecuteReader();

if (Reader.Read()&&Reader!=null)

{

//If Password Matches the User can Change their password

return true;

}

else

{

//if Password does not Match an Error will return false

return false;

}

}

catch (SoapException)

{

return

false;

}

}

#endregion

#region Forgot/Update Password

[WebMethod(Description = "Forgot Password")]

public bool forgotPassword(string Email, string password)

{

try

{

ConnectToDB();

//if the user wants to change /forgot their password the update statements allows them to do so and if fails an Error will pop up

Cmd = Con.CreateCommand();

Cmd.CommandText = "UPDATE [User] SET [Password] = '" + password + "' WHERE (Email = '" + Email + "');";

Cmd.ExecuteNonQuery();

Con.Close();

return true;

}

catch(Exception)

{

return false;

}

}

#endregion

}

**LOGIN PAGE**

/\*Name: Nathan James

\* BookClub Project

\* F9F6QTH22

\* Login Page

\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using AddService;

public partial class Login : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

lblLogInError.Visible = false;//Ensuring the label is hidden

}

protected void btnRegister\_Click(object sender, EventArgs e)

{

//Once the Register button is clicked the user will be redirected to the Register Page

Response.Redirect("Register.aspx");

}

protected void btnLogin\_Click(object sender, EventArgs e)

{

string Email = txtEmail.Text;

string Password = txtPassword.Text;

BookStoreWebService Connect = new BookStoreWebService();

BookStoreWebService Connection = new BookStoreWebService();

bool EmailCheck = Connection.CheckUserExists(Email);

if (EmailCheck == true)

{

bool Login = Connect.Login(Email, Password);

if (Login == true)

{

lblLogInError.Visible = false;

HttpCookie LoggedInCookie = new HttpCookie("LoggedIn", "true");

Response.Cookies.Add(LoggedInCookie);

Session["Email"] = Email;

Response.Redirect("BookPage.aspx");

}

else

{

lblLogInError.Visible = true;

lblLogInError.Text = "Password is Incorrect";

}

}

else

{

lblLogInError.Visible = true;

lblLogInError.Text = "Email Does not Exits";

}

}

protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)

{

}

protected void btnForgotPassword\_Click(object sender, EventArgs e)

{

Response.Redirect("ForgotPassword.aspx");

}

}

**REGISTRATION PAGE**

/\*Name: Nathan James

\* BookClub Project

\* F9F6QTH22

\* Register Page

\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.OleDb;

using System.Xml;

using System.Collections;

using System.Net.Mail;

using System.Text;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using AddService;

public partial class Register : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

lblRegistrationError.Visible = false;

}

protected void btnRegister\_Click(object sender, EventArgs e)

{

BookStoreWebService Connect = new BookStoreWebService();

if (Page.IsValid)

{

try

{

//Getting the data by user in the text fields

string Email = txtEmail.Text;

string Name = txtName.Text;

string Surname = txtSurname.Text;

string SecurityQ = @DropDownList1SQ.SelectedValue;

string SecurityA = txtSecurityA.Text;

string Password = txtPassword.Text;

bool Result = Connect.Register(Email, Name, Surname, SecurityQ, SecurityA, Password);

if (Result == true)

{

Response.Redirect("Login.aspx");

}

else

{

//Checks if there are any errors

lblRegistrationError.Visible = true;

lblRegistrationError.Text = "Email Already Exists";

}

}

catch (OleDbException ex)

{

lblRegistrationError.Visible = true;

lblRegistrationError.Text = ex.Message;

}

}

}

}

**FORGOT/UPDATE PASSWORD**

/\*Name: Nathan James

\* BookClub Project

\* F9F6QTH22

\* forgot/update password Page

\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using AddService;

public partial class ForgotPassword : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

EmailCheckPanel.Visible = true; ;

UpdatePasswordPanel.Visible = false;

SecurityCheckPanel.Visible = false;

lblCheckEmailError.Visible = false;

lblCheckAnswerError.Visible = false;

lblUpdatePasswordError.Visible = false;

}

protected void btnCheckEmail\_Click(object sender, EventArgs e)

{

string Email = txtCheckEmail.Text;

BookStoreWebService Connect = new BookStoreWebService();

bool Exists = Connect.CheckUserExists(Email);

if (Exists)

{

lblCheckEmailError.Visible = false;

SecurityCheckPanel.Visible = true;

lblQuestionShow.Text = Connect.getQuestion(Email);

}

else

{

EmailCheckPanel.Visible = true;

lblCheckEmailError.Visible = true;

lblCheckEmailError.Text = "Email Does not Exists Enter a Valid Email";

}

}

protected void btnCheckAnswer\_Click(object sender, EventArgs e)

{

SecurityCheckPanel.Visible = true;

lblCheckAnswerError.Visible = false;

string Email = txtCheckEmail.Text ;

string Answer = txtSecAnswer.Text;

BookStoreWebService Connect = new BookStoreWebService();

bool CorrectAnswer = Connect.AnswerCorrect(Email, Answer);

if (CorrectAnswer)

{

lblCheckAnswerError.Visible = false;

UpdatePasswordPanel.Visible = true;

}

else

{

SecurityCheckPanel.Visible = true;

lblCheckAnswerError.Visible = true;

lblCheckAnswerError.Text = "Answer is Incorrect Please Try Again....";

UpdatePasswordPanel.Visible = false;

}

}

protected void btnUpdatePassword\_Click(object sender, EventArgs e)

{

string Email = txtCheckEmail.Text;

lblUpdatePasswordError.Visible = false;

string Password = txtNewPassword.Text;

BookStoreWebService Connect = new BookStoreWebService();

bool PasswordChange = Connect.forgotPassword(Email,Password);

if(PasswordChange)

{

HttpCookie LoggedInCookie = new HttpCookie("LoggedIn", "true");

Response.Cookies.Add(LoggedInCookie);

Session["Email"] = Email;

Response.Redirect("Login.aspx");

}

else

{

lblUpdatePasswordError.Visible = true;

UpdatePasswordPanel.Visible = true;

lblUpdatePasswordError.Text = "An Error has Occured Please Try Again...";

}

}

}

**VIEW BOOKS**

/\*NATHAN JAMES

\* BOOK CLUB PROJECT

\* BOOK PAGE

\* DATE DESIGNED 19/01/2022

\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using AddService;

public partial class BookPage : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Request.Cookies["LoggedIn"]==null||(Request.Cookies["LoggedIn"] !=null &&Request.Cookies["LoggedIn"].Value!="true"))

{

Response.Redirect("Login.aspx");

}

//All Panels will not be visible until a certain action is met

PanelCheckBookID.Visible = false;

PanelAddRating.Visible = false;

}

protected void btnAddBook\_Click(object sender, EventArgs e)

{

//the button will take the user to the add books Page

Response.Redirect("AddBook.aspx");

}

protected void btnRate\_Click(object sender, EventArgs e)

{

//Once the User chooses to Rate a Book The CheckBook ID Panel Will be Shown

PanelCheckBookID.Visible = true;

}

protected void btnBookID\_Click(object sender, EventArgs e)

{

//Checks if the TextBox is not Empty

if (txtBookID.Text != "")

{

//The following Code will Check if a Book Exists if not an Error will pop up Stating That (Book Does Not Exists)

int ID = int.Parse(txtBookID.Text);

//Connecting To the WebService To Collect The BookID

BookStoreWebService Connect = new BookStoreWebService();

bool Result = Connect.BookIDExists(ID);

if (Result == true)

{

PanelAddRating.Visible = true;

lblBookIDError.Visible = false;

}

else

{

PanelCheckBookID.Visible = true;

lblBookIDError.Visible = true;

lblBookIDError.Text = "Book ID Does Not Exists";

}

}

else

{

lblBookIDError.Visible = true;

}

}

protected void btnAddRating\_Click(object sender, EventArgs e)

{

//Once a BookID exists The Add Ratings Panel will Appear And the user will be able to add a Rating

int Rate = DropDownList1.SelectedIndex + 1;

int ID = int.Parse(txtBookID.Text);

string strEmail = Session["Email"].ToString();

BookStoreWebService Connect = new BookStoreWebService();

//Checks if User(Email) has already rated the Book

bool AlreadyRated = Connect.AlreadyRated(strEmail,ID);

if(AlreadyRated)

{

PanelCheckBookID.Visible = true;

PanelAddRating.Visible = false;

lblRatingError.Visible = true ;

lblRatingError.Text = "Book Has already Been Rated Please Choose another";

}

else

{

//If the Rating is Successful the database + GridView will Update displaying the Average of The Book that was Rated

lblRatingError.Visible = false;

bool Result = Connect.Rating(strEmail,ID, Rate);

int Sum = Connect.GetSum(ID);

int People = Connect.GetAmountRated(ID);

People = People + 1;

Sum = Sum += Rate;

if (Result)

{

lblRatingError.Visible = false;

int Average = Sum / People;

bool SumUpdate = Connect.UpdateSum(ID, Sum);

bool UpdatePeople = Connect.UpdateAmountRated(ID, People);

bool UpdateAverage = Connect.UpdateAverageRating(ID,Average);

Response.Redirect("BookPage.aspx");

}

else

{

//If the Rating Was Unsuccessful an Error will pop Up

lblRatingError.Visible = true;

lblRatingError.Text = "An Error Has Occured Please Try Again";

}

}

}

protected void btnViewProfile\_Click(object sender, EventArgs e)

{

//If the User wants to view their Profile the View Profile Button is Clicked

Response.Redirect("ProfilePage.aspx");

}

protected void btnLogOut\_Click(object sender, EventArgs e)

{

//If the User wishes to Log Out the Log Out Button is Clicked

if(Response.Cookies["LoggedIn"]!=null)

{

HttpCookie LoggedInCookie = new HttpCookie("LoggedInCookie", "false");

Response.SetCookie(LoggedInCookie);

Response.Redirect("Login.aspx");

}

}

}

**ADD BOOK**

/\*Name : Nathan James

\* BookClub Project

\* F9F6QTH22

\* AddBook Page

\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using AddService;

public partial class AddBook : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

lblAddBookError.Visible = false;

lblSuccess.Visible = false;

}

protected void btnAddBook\_Click(object sender, EventArgs e)

{

//Declaring the Variables

string Name = txtName.Text;

string Author = txtAuthor.Text;

string Genre = DropDownList1Genre.SelectedValue;

//Connecting To the Webserice

BookStoreWebService Connect = new BookStoreWebService();

bool Exists = Connect.BookExists(Name,Author,Genre);

if (Exists)

{

//Checks if Book Exists

lblAddBookError.Visible = true;

lblAddBookError.Text = "Book Already Exists";

}

else

{

lblAddBookError.Visible = false;

//Connecting To the Add Book Section in the Webservice

bool Result = Connect.AddBook(Name,Author,Genre);

if (Result)

{

//Adds Book Successfully

lblSuccess.Visible = true;

lblSuccess.Text = "Book Added Successfully ";

Response.Redirect("BookPage.aspx");

}

else

{

//Prints an error If book was not Added Successfully

lblAddBookError.Visible = true;

lblAddBookError.Text = "An Error Has Occured Please Try Again....";

}

}

}

}

**PROFILE PAGE**

/\*Name : Nathan James

\* F9f6QTH22

\* BookClub Project

\* Date : 19/01/2022

\* Profile Page

\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using AddService;

public partial class ProfilePage : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Request.Cookies["LoggedIn"] == null || (Request.Cookies["LoggedIn"] != null && Request.Cookies["LoggedIn"].Value != "true"))

{

Response.Redirect("Login.aspx");

}

//Ensuring that the update panel is not shown until the user clicks on the update Details Button

UpdatePanel.Visible = false;

lblEmail1.Text = "";

lblName1.Text = "";

lblSurname1.Text = "";

BookStoreWebService Connect = new BookStoreWebService();

DataTable dt = new DataTable();

string emailProfile = Session["Email"].ToString();

dt = Connect.Profiles(emailProfile);

foreach (DataRow User in dt.Rows)

{

lblEmail1.Text = lblEmail1.Text + User["Email"];

lblName1.Text = lblName1.Text + User["Name"];

lblSurname1.Text = lblSurname1.Text + User["Surname"];

}

}

protected void btnUpdateDetails\_Click(object sender, EventArgs e)

{

UpdatePanel.Visible = true;//Ensures that the next Panel is Shown when the user clicks the button

}

protected void btnUpdate\_Click(object sender, EventArgs e)

{

string fName = txtName.Text;

string SName = txtSurname.Text;

string Email = lblEmail1.Text;

if (fName!="")

{

//Updating an Individual Section for the Name

BookStoreWebService connection = new BookStoreWebService();

bool Result = connection.UpdateName(Email,fName);

if(Result)

{

if (Response.Cookies["LoggedIn"]==null)

{

BookStoreWebService Connect = new BookStoreWebService();

DataTable dt = new DataTable();

string emailProfile = Session["Email"].ToString();

dt = Connect.Profiles(emailProfile);

foreach (DataRow User in dt.Rows)

{

lblEmail1.Text = lblEmail1.Text + User["Email"];

lblName1.Text = lblName1.Text + User["Name"];

lblSurname1.Text = lblSurname1.Text + User["Surname"];

}

}

else//Basically if the user decides to Cheat the System by typing in a specific Page while not being Loggedin the Program will kick them out by sending them

//back to the Log in Page

{

Response.Redirect("Login.aspx");

}

}

if (SName != "")

{

//Updating individual Section for the Surname

//Calling the Webservice

if (Response.Cookies["LoggedIn"] == null)

{

BookStoreWebService Connection = new BookStoreWebService();

//Calling The Update Section from the webService

bool Result1 = Connection.UpdateSurname(Email, SName);

if (Result1)

{

BookStoreWebService Connect = new BookStoreWebService();

DataTable dt = new DataTable();

string emailProfile = Session["Email"].ToString();

dt = Connect.Profiles(emailProfile);

foreach (DataRow User in dt.Rows)

{

lblEmail1.Text = lblEmail1.Text + "<br>" + User["Email"];

lblName1.Text = lblName1.Text + "<br>" + User["Name"];

lblSurname1.Text = lblSurname1.Text + "<br>" + User["Surname"];

}

}

else

{

Response.Redirect("Login.aspx");

}

}

//Added this because there was a glitch so that the Page refreshes every time the user Updates

Response.Redirect("ProfilePage.aspx");

}

else

{

UpdatePanel.Visible = true;

lblUpdateError.Visible = true;

lblUpdateError.Text = "Unable to update";

}

}

}

protected void btnLogOut\_Click(object sender, EventArgs e)

{

if (Response.Cookies["LoggedIn"] != null)

{

//if the user chooses to log out after either viewing or updating their details

HttpCookie LoggedInCookie = new HttpCookie("LoggedIn", "false");

Response.SetCookie(LoggedInCookie);

Response.Redirect("Login.aspx");

}

}

protected void btnViewBooks\_Click(object sender, EventArgs e)

{

//once User view Profile and updates any details has the option to either log out or go back to viewing the Books

Response.Redirect("BookPage.aspx");

}

}

LOGIN PAGE

A screenshot of a computer

Description automatically generated with medium confidence

REGISTER PAGE

A screenshot of a computer

Description automatically generated

FORGOT /UPDATE PASSWORD PAGE

Graphical user interface, website

Description automatically generated

BOOKS /RATING PAGE

A screenshot of a computer

Description automatically generated

ADD BOOKS

A screenshot of a computer

Description automatically generated

VIEW PROFILE

A screenshot of a computer

Description automatically generated