CREATE procedure [dbo].[ChangePassword]

(

@Username nvarchar (50),

@OriginUser nvarchar(50),

@Password nvarchar (50)

)

as

begin

Update CurrentUsers

set Password=@PasswOrd

where Username=@Username

End

CREATE procedure [dbo].[CreateNewUser]

(

@Username nvarchar (50),

@OriginUser nvarchar(50),

@Password nvarchar(50)

)

as

begin

Insert into CurrentUsers values (@Username, @OriginUser, @Password)

End

CREATE procedure [dbo].[DeleteUser]

(

@Username nvarchar(50),

@OriginUser nvarchar(50),

@Password nvarchar(50)

)

as

begin

Delete from CurrentUsers where Username = @Username or Password = @Password

End

CREATE Procedure [dbo].[GetUserDetails]

as

begin

select \* from CurrentUsers

End

CREATE procedure [dbo].[UpdateUser]

(

@Username nvarchar (50),

@OriginUser nvarchar (50),

@Password nvarchar (50)

)

as

begin

Update CurrentUsers

set Username=@Username,

Password=@Password

where OriginUser=@OriginUser

End

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Diagnostics.Eventing.Reader;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace HealthAdviceGroup

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

//Attributes

string username;

string originUser;

string password;

bool registration;

username = txtUsername.Text;

originUser = txtUsername.Text;

password = txtPassword.Text;

registration = true;

//Checks for a null value in the username

if (username == "")

{

//Displays a message to say that the username is invalid

MessageBox.Show("Your username is invalid");

grpLogin.Visible = false;

grpRegister.Visible = true;

}

//Checks for a null value in the password.

if (password == "")

{

//Displays a message to say that the password is invalid

MessageBox.Show("Your password is invalid");

grpLogin.Visible = false;

grpRegister.Visible = true;

}

else

{

grpRegister.Visible = false;

grpLogin.Visible = true;

}

//Tells the code where the database is

string connectionString = "Data Source = (LocalDB)\\MSSQLLocalDB; AttachDbFilename = C:\\Users\\M2201992\\OneDrive - Middlesbrough College\\Year 2\\OSS (OCC)\\HealthAdviceGroup\\HAGDatabase.mdf; Integrated Security = True; Connect Timeout = 30";

SqlConnection sqlConnection = new SqlConnection(connectionString);

List<User> userList = new List<User>();

SqlCommand cmd = new SqlCommand("GetUserDetails", sqlConnection);

cmd.CommandType = CommandType.StoredProcedure;

SqlDataAdapter sd = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sqlConnection.Open();

sd.Fill(dt);

sqlConnection.Close();

foreach (DataRow dr in dt.Rows)

{

userList.Add(

new User

{

Username = Convert.ToString(dr["Username"]),

OriginUser = Convert.ToString(dr["OriginUser"]),

Password = Convert.ToString(dr["Password"])

});

}

foreach (User user in userList)

{

if (user.Username != txtUsername.Text)

{

registration = true;

break;

}

else

{

registration = false;

}

}

if (registration == false)

{

//Tells the user their password is incorrect

MessageBox.Show("This username has already been taken, please try again.");

grpRegister.Visible = true;

grpHomeDashboard.Visible = false;

grpLogin.Visible = false;

grpHAG.Visible = false;

grpLogout.Visible = false;

lblForgotPassword.Visible = true;

}

else

{

//Use stored procedure

SqlCommand command = new SqlCommand("CreateNewUser", sqlConnection);

command.CommandType = CommandType.StoredProcedure;

//Input username and password from form

string Username = txtUsername.Text;

string Password = txtPassword.Text;

string OriginUser = txtUsername.Text;

//Tells the user they've successfully created an account

MessageBox.Show("You have successfully created an account");

//Call stored procedure passing username and password as parameters

command.Parameters.AddWithValue("@Username", Username);

command.Parameters.AddWithValue("@Password", Password);

command.Parameters.AddWithValue("@OriginUser", OriginUser);

//Open connection to database, execute stored procedure and close the connectionD

sqlConnection.Open();

command.ExecuteNonQuery();

sqlConnection.Close();

txtUsername.Text = "";

txtPassword.Text = "";

}

}

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

{

//Attributes

bool login = false;

//Read a person record

List<User> userList = new List<User>();

string connectionString = "Data Source = (LocalDB)\\MSSQLLocalDB; AttachDbFilename = C:\\Users\\M2201992\\OneDrive - Middlesbrough College\\Year 2\\OSS (OCC)\\HealthAdviceGroup\\HAGDatabase.mdf; Integrated Security = True; Connect Timeout = 30";

SqlConnection sqlConnection = new SqlConnection(connectionString);

SqlCommand cmd = new SqlCommand("GetUserDetails", sqlConnection);

cmd.CommandType = CommandType.StoredProcedure;

SqlDataAdapter sd = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sqlConnection.Open();

sd.Fill(dt);

sqlConnection.Close();

foreach (DataRow dr in dt.Rows)

{

userList.Add(

new User

{

Username = Convert.ToString(dr["Username"]),

OriginUser = Convert.ToString(dr["OriginUser"]),

Password = Convert.ToString(dr["Password"])

});

}

foreach (User user in userList)

{

if (user.Username == txtLoginUsername.Text)

{

if (user.Password == txtLoginPassword.Text)

{

//Tells the user they've successfully logged in

MessageBox.Show("You have successfully logged in");

grpHomeDashboard.Visible = true;

grpLogin.Visible = false;

grpHAG.Visible = true;

grpLogout.Visible = true;

txtLoginUsername.Text = "";

txtLoginPassword.Text = "";

login = true;

break;

}

else

{

login = false;

}

}

else

{

login = false;

}

}

if (login == false)

{

//Tells the user their password is incorrect

MessageBox.Show("Your password is incorrect");

grpHomeDashboard.Visible = false;

grpLogin.Visible = true;

grpHAG.Visible = false;

grpLogout.Visible = false;

lblForgotPassword.Visible = true;

txtLoginPassword.Text = "";

}

}

private void btnLogout\_Click(object sender, EventArgs e)

{ //Tells the user they've successfully logged out

MessageBox.Show("You have successfully logged out.");

grpHAG.Visible = false;

grpLogout.Visible = false;

grpLogin.Visible = true;

grpEditUser.Visible = false;

}

private void btnWeather\_Click(object sender, EventArgs e)

{

}

private void lblHaveAccount\_Click(object sender, EventArgs e)

{

//Allows the user to access the login menu from the register menu

grpLogin.Visible = true;

grpRegister.Visible = false;

}

private void lblNoAccount\_Click(object sender, EventArgs e)

{

//Allows the user to access the register menu from the login menu

grpLogin.Visible = false;

grpRegister.Visible = true;

}

private void lblForgotPassword\_Click(object sender, EventArgs e)

{

grpForgotPassword.Visible = true;

}

private void btnConditionAndAllegryInfo\_Click(object sender, EventArgs e)

{

string conditionOrAllergy;

conditionOrAllergy = (string)cboConditionsAndAllergies.SelectedItem;

if (conditionOrAllergy == "Asthma")

{

rtxtConditionAndAllergyInfo.Text = ("Asthma is a common lung condition that causes occasional breathing difficulties. The main symptoms of asthma include wheezing, breathlessness, a tight chest and coughing. These symptoms can sometimes get temporarily get worse. This is known as an asthma attack. Treatments for asthma include reliever and preventer inhalers.");

}

if (conditionOrAllergy == "Frostbite")

{

rtxtConditionAndAllergyInfo.Text = ("Frostbite is damage to skin and tissue caused by exposure to freezing temperatures. The long-term side effects of frostbite are increased sensitivity to the cold, numbness in the affected body parts (most commonly in the fingers), reduced sense of touch in the affected body parts and persistent pain in the affected body parts. The treatment for frostbite depends on the severity of the victim's symptoms. Always seek medical attention if you think you or someone else has frostbite. If medical assistance isn't available, here's what you can do to treat frostbite and hypothermia. You should move the victim to a warmer place (if possible), gently remove any accessories or clothing around the affected area (such as jewellery, gloves and boots), replace wet clothing with soft and dry clothing to stop further heat loss, warm the body by wrapping it in blankets and protecting the frostbitten parts. If you think you or someone else has frostbite do not rub the affected area or apply direct heat this can cause further injury. Smoking whilst having frostbite is highly discouraged as smoking can affect blood circulation.");

}

if (conditionOrAllergy == "Hay Fever")

{

rtxtConditionAndAllergyInfo.Text = ("Hay fever is an allergic reaction to pollen, typically when it encounters your mouth, nose, eyes and throat. The symptoms of hay fever include sneezing and coughing, a runny or blocked nose, headache, earache, feeling tired, loss of smell. People with hay fever may also experience red or watery eyes, itchy eyes, throat, mouth nose and ears but also pain around their temple and forehead. Hay fever can last for weeks or months. You can ease hay fever symptoms by putting Vaseline around your nostrils to trap pollen, wear wraparound sunglasses to stop pollen from getting into your eyes, shower and change your clothes after you have been outside to wash pollen off, stay indoors whenever possible, buy a pollen filter for the air vents in your car and a vacuum cleaner with a HEPA filter and try to avoid all contact with other people by staying at home if you have a high temperature or you do not feel well enough to do your normal activities.");

}

if (conditionOrAllergy == "Heatstroke")

{

rtxtConditionAndAllergyInfo.Text = ("Heatstroke and heat exhaustion can happen if you get too hot. The signs of heat exhaustion include tiredness, dizziness, headaches, feeling or being sick, excessive sweating and skin becoming pale and clammy or getting a heat rash (a change in skin colour can be harder to see on brown and black skin), cramps in the legs and stomach, fast breathing or heartbeat, a high temperature, being very thirsty and weakness. If someone is showing any of these signs, they need to be cooled down and given fluids. Things that can be done to counter heatstroke include moving the victim to a cool place, removing all unnecessary clothing (e.g. Jackets, socks) from them, get them to drink a sports or rehydration drink cool water, cool their skin by spraying or sponging the, with cool water and fanning them. You can also place cold packs, wrapped in cloth, underneath the armpits or on the neck");

}

}

private void btnDeleteUser\_Click(object sender, EventArgs e)

{

//Use stored procedure to delete record(s) containing a given name

string connectionString = "Data Source = (LocalDB)\\MSSQLLocalDB; AttachDbFilename = C:\\Users\\M2201992\\OneDrive - Middlesbrough College\\Year 2\\OSS (OCC)\\HealthAdviceGroup\\HAGDatabase.mdf; Integrated Security = True; Connect Timeout = 30";

SqlConnection sqlConnection = new SqlConnection(connectionString);

//USE STORED PROCEDURE

SqlCommand command = new SqlCommand("DeleteUser", sqlConnection);

command.CommandType = CommandType.StoredProcedure;

string Username = txtDelUserUsername.Text;

command.Parameters.AddWithValue("@Username", Username);

string Password = txtDelUserPassword.Text;

command.Parameters.AddWithValue("@Password", Password);

sqlConnection.Open();

command.ExecuteNonQuery();

sqlConnection.Close();

//Clears the text in the text box

txtDelUserPassword.Text = "";

txtDelUserUsername.Text = "";

//Hides the unnecessary group boxes

grpHAG.Visible = false;

grpLogout.Visible = false;

grpLogin.Visible = false;

grpDeleteUser.Visible = false;

grpAdvice.Visible = false;

grpInformation.Visible = false;

grpForecast.Visible = false;

grpEditUser.Visible = false;

grpRegister.Visible = true;

}

private void btnInformation\_Click(object sender, EventArgs e)

{

//Makes the group box visible

grpInformation.Visible = true;

}

private void btnDeleteAccount\_Click(object sender, EventArgs e)

{

//Makes the group box visible

grpDeleteUser.Visible = true;

}

private void btnEditUser\_Click(object sender, EventArgs e)

{

//Tells the code where the database is

string connectionString = "Data Source = (LocalDB)\\MSSQLLocalDB; AttachDbFilename = C:\\Users\\M2201992\\OneDrive - Middlesbrough College\\Year 2\\OSS (OCC)\\HealthAdviceGroup\\HAGDatabase.mdf; Integrated Security = True; Connect Timeout = 30";

SqlConnection sqlConnection = new SqlConnection(connectionString);

List<User> userList = new List<User>();

SqlCommand cmd = new SqlCommand("GetUserDetails", sqlConnection);

cmd.CommandType = CommandType.StoredProcedure;

SqlDataAdapter sd = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sqlConnection.Open();

sd.Fill(dt);

sqlConnection.Close();

foreach (DataRow dr in dt.Rows)

{

userList.Add(

new User

{

Username = Convert.ToString(dr["Username"]),

OriginUser = Convert.ToString(dr["OriginUser"]),

Password = Convert.ToString(dr["Password"])

});

}

foreach (User user in userList)

{

if (user.OriginUser == txtCurrentUsername.Text)

{

//USE STORED PROCEDURE

SqlCommand command = new SqlCommand("UpdateUser", sqlConnection);

command.CommandType = CommandType.StoredProcedure;

string Username = txtNewUsername.Text;

command.Parameters.AddWithValue("@Username", Username);

string Password = txtNewPassword.Text;

command.Parameters.AddWithValue("@Password", Password);

sqlConnection.Open();

command.ExecuteNonQuery();

sqlConnection.Close();

txtNewPassword.Text = "";

txtNewUsername.Text = "";

break;

}

else

{

MessageBox.Show("Your origin user is not correct");

}

}

}

private void btnEditAccount\_Click(object sender, EventArgs e)

{

//Makes the group box visible

grpEditUser.Visible = true;

}

private void btnChangePassword\_Click(object sender, EventArgs e)

{

//Tells the code where the database is

string connectionString = "Data Source = (LocalDB)\\MSSQLLocalDB; AttachDbFilename = C:\\Users\\M2201992\\OneDrive - Middlesbrough College\\Year 2\\OSS (OCC)\\HealthAdviceGroup\\HAGDatabase.mdf; Integrated Security = True; Connect Timeout = 30";

SqlConnection sqlConnection = new SqlConnection(connectionString);

//USE STORED PROCEDURE

SqlCommand command = new SqlCommand("ChangePassword", sqlConnection);

command.CommandType = CommandType.StoredProcedure;

string Username = txtFPUsername.Text;

command.Parameters.AddWithValue("@Username", Username);

string Password = txtFPNewPassword.Text;

command.Parameters.AddWithValue("@Password", Password);

sqlConnection.Open();

command.ExecuteNonQuery();

sqlConnection.Close();

}

}

}