﻿using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Liaqat\_Gym\_Management\_System

{

public static class ClubConnection

{

public static SqlConnection Sql\_Connection()

{

//SqlConnection sqlConnection = new SqlConnection("Data Source=DESKTOP-75F694L;Initial Catalog=Gym;Integrated Security=True");

//SqlConnection sqlConnection = new SqlConnection(@"Data Source=DESKTOP-ABP9NUH\SQLEXPRESS ;Initial Catalog=Gym;Integrated Security=True");

//SqlConnection sqlConnection = new SqlConnection(@"Data Source=DESKTOP-8JV4MC9 ;Initial Catalog=Gym;Integrated Security=True");

SqlConnection sqlConnection = Connection.Sql\_Connection();

return sqlConnection;

}

}

public class ClubDatabase

{

public DataTable GetCLubCertifactesDate(ClubCertificates clubCertificates)

{

SqlConnection connection = Connection.Sql\_Connection();

DataTable dataTable = new DataTable();

connection.Open();

using (connection)

{

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter("Select Id , Rank , SerialNumber ,IssueDate as 'Issue Date', Amount , c.Club\_Name as 'Club' from ClubCertification as cc Inner Join Clubs as c On c.Club\_Id = cc.Club Where cc.Club = @clubid order by cc.Id desc ", connection);

sqlDataAdapter.SelectCommand.Parameters.AddWithValue("@clubid", clubCertificates.Clubid);

sqlDataAdapter.Fill(dataTable);

}

return dataTable;

}

public void DeleteRecord(int id , string table)

{

SqlConnection connection = Connection.Sql\_Connection();

SqlCommand sqlCommand = new SqlCommand("Delete from " +table+" where Id = @Id", connection);

sqlCommand.Parameters.AddWithValue("@Id", id);

connection.Open();

using (connection)

{

try

{

sqlCommand.ExecuteNonQuery();

MessageBox.Show( " Record is Removed ", "Deleted", MessageBoxButtons.OK, MessageBoxIcon.Information); ;

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

public void UpdateCertificateRecord(ClubCertificates clubCertificates)

{

SqlConnection connection = Connection.Sql\_Connection();

SqlCommand sqlCommand = new SqlCommand("Update ClubCertification set Rank = @rank , SerialNumber =@serialnumber, IssueDate = @issuedate, Amount = @amount , Club = @clubid Where Id = @id", connection);

sqlCommand.Parameters.AddWithValue("@id", clubCertificates.Id);

sqlCommand.Parameters.AddWithValue("@clubid", clubCertificates.Clubid);

sqlCommand.Parameters.AddWithValue("@rank", clubCertificates.Rank);

sqlCommand.Parameters.AddWithValue("@serialnumber", clubCertificates.SerialNumber);

sqlCommand.Parameters.AddWithValue("@issuedate", clubCertificates.IsseDate);

if (clubCertificates.Amount == "")

{

sqlCommand.Parameters.AddWithValue("@amount", "Credit");

}

else

{

sqlCommand.Parameters.AddWithValue("@amount", clubCertificates.Amount);

}

connection.Open();

using (connection)

{

try

{

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Record is Updated Successfully ", "Record Updated", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

public void AddClubCertificationRecord(ClubCertificates clubCertificates)

{

SqlConnection connection = Connection.Sql\_Connection();

SqlCommand cmd = new SqlCommand("Select \* from ClubCertification where SerialNumber =@serialnumber", connection);

cmd.Parameters.AddWithValue("@serialnumber", clubCertificates.SerialNumber);

SqlCommand sqlCommand = new SqlCommand("Insert into ClubCertification Values(@club,@rank,@serialnumber,@amount,@issuedate)", connection);

sqlCommand.Parameters.AddWithValue("@club", clubCertificates.Clubid);

sqlCommand.Parameters.AddWithValue("@rank", clubCertificates.Rank);

sqlCommand.Parameters.AddWithValue("@serialnumber", clubCertificates.SerialNumber);

sqlCommand.Parameters.AddWithValue("@issuedate", clubCertificates.IsseDate);

if(clubCertificates.Amount == "")

{

sqlCommand.Parameters.AddWithValue("@amount", "Credit");

}

else

{

sqlCommand.Parameters.AddWithValue("@amount", clubCertificates.Amount);

}

connection.Open();

using (connection)

{

try

{

SqlDataReader dr = cmd.ExecuteReader();

if (!dr.HasRows)

{

dr.Close();

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Record Added Successfully ", "Record Added", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("This Serial Number already Exists","Record Already Exists",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

public void Club\_Names()

{

SqlConnection connection = ClubConnection.Sql\_Connection();

ClubList.Clubs.Clear();

connection.Open();

using (connection)

{

SqlCommand cmd = new SqlCommand("Select \* from Clubs ", connection);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

ClubNames clubs = new ClubNames();

string club = dr["Club\_Name"].ToString();

int id = Convert.ToInt32(dr["Club\_Id"].ToString());

clubs.id = id;

clubs.name = club;

ClubList.Clubs.Add(clubs);

}

}

}

public void AddClub(ClubNames clubNames)

{

SqlConnection connection = ClubConnection.Sql\_Connection();

SqlCommand cmd = new SqlCommand("Insert into Clubs Values(@clubname)", connection);

cmd.Parameters.AddWithValue("@clubname", clubNames.name);

connection.Open();

using (connection)

{

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Club :" + clubNames.name + " is Added.", "Record Added", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

public DataTable GetCludMembers(ClubNames club)

{

SqlConnection connection = ClubConnection.Sql\_Connection();

connection.Open();

using (connection)

{

SqlDataAdapter cmd = new SqlDataAdapter("Select Id, Name, JoinDate as 'Join Date' , Rank , Picture, Club\_Name as Club from Club\_Members as m Inner Join Clubs as c on m.Club\_Id = c.Club\_Id Where c.Club\_Id = @clubid order by m.Club\_Id desc ", connection);

cmd.SelectCommand.Parameters.AddWithValue("@clubid", club.id);

DataTable dataTable = new DataTable();

cmd.Fill(dataTable);

return dataTable;

}

}

public DataTable GetCludMembersSearch(string search)

{

SqlConnection connection = ClubConnection.Sql\_Connection();

connection.Open();

using (connection)

{

SqlDataAdapter cmd = new SqlDataAdapter("Select Id, Name, JoinDate as 'Join Date' , Rank , Picture, Club\_Name as Club from Club\_Members as m Inner Join Clubs as c on m.Club\_Id = c.Club\_Id Where Name Like @search order by Id desc", connection);

cmd.SelectCommand.Parameters.AddWithValue("@search", "%" + search + "%");

DataTable dataTable = new DataTable();

cmd.Fill(dataTable);

return dataTable;

}

}

public void EditClubMember(ClubMembersInformation clubMembersInformation)

{

SqlConnection connection = Connection.Sql\_Connection();

string query = "Update [dbo].[Club\_Members] set Name = @name , Rank =@rank, Club\_Id = @clubid , JoinDate = @joindate ";

if (clubMembersInformation.Picture != null)

{

query = query + " , Picture = @picture";

}

query += " Where Id = @id";

SqlCommand cmd = new SqlCommand(query, connection);

cmd.Parameters.AddWithValue("@id", clubMembersInformation.Id);

cmd.Parameters.AddWithValue("@name", clubMembersInformation.Name);

cmd.Parameters.AddWithValue("@rank", clubMembersInformation.Rank);

cmd.Parameters.AddWithValue("@clubid", clubMembersInformation.Club);

cmd.Parameters.AddWithValue("@joindate", clubMembersInformation.JoinDate);

if (clubMembersInformation.Picture != null)

{

cmd.Parameters.AddWithValue("@picture", clubMembersInformation.Picture);

}

connection.Open();

using (connection)

{

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Member : " + clubMembersInformation.Name + " record successfully Updated .", "Recod Added", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

public void AddClubMemeber(ClubMembersInformation clubMembersInformation)

{

SqlConnection connection = Connection.Sql\_Connection();

SqlCommand cmd = new SqlCommand("Insert into [dbo].[Club\_Members] Values(@name,@rank,@clubid,@joindate,@picture)", connection);

cmd.Parameters.AddWithValue("@name", clubMembersInformation.Name);

cmd.Parameters.AddWithValue("@rank", clubMembersInformation.Rank);

cmd.Parameters.AddWithValue("@clubid", clubMembersInformation.Club);

cmd.Parameters.AddWithValue("@joindate", clubMembersInformation.JoinDate);

if (clubMembersInformation.Picture != null)

{

cmd.Parameters.AddWithValue("@picture", clubMembersInformation.Picture);

}

else

{

SqlParameter p = new SqlParameter("@picture", SqlDbType.Image);

p.Value = DBNull.Value;

cmd.Parameters.Add(p);

}

connection.Open();

using (connection)

{

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Member : " + clubMembersInformation.Name + " is successfully added .", "Recod Added", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

}