﻿using Liaqat\_Gym\_Management\_System;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Liaqat\_Gym\_Management\_System

{

public static class Connection

{

public static SqlConnection Sql\_Connection()

{

SqlConnection sqlConnection = new SqlConnection("Data Source=DESKTOP-H03LBFQ;Initial Catalog=Gym;Integrated Security=True");

//SqlConnection sqlConnection = new SqlConnection(@"Data Source=DESKTOP-28C09H2 ;Initial Catalog=Gym;Integrated Security=True");

//SqlConnection sqlConnection = new SqlConnection(@"Data Source=DESKTOP-3B4STNK ;Initial Catalog=Gym;Integrated Security=True");

//SqlConnection sqlConnection = new SqlConnection(@"Data Source=DESKTOP-E7QJS2H\SQLEXPRESS ;Initial Catalog=Gym;Integrated Security=True");

return sqlConnection;

}

}

}

public class GymDatabase

{

public void CreateBackup()

{

SqlConnection con = Connection.Sql\_Connection();

string path = Directory.GetCurrentDirectory() + "\\Backup\\Gym.bak";

string str = "USE Gym";

string str1 = "BACKUP DATABASE Gym TO DISK ='" + path + "' WITH FORMAT,MEDIANAME = 'Z\_SQLServerBackups',NAME = 'Full Backup of Gym';";

try

{

con.Open();

using (con)

{

SqlCommand cmd1 = new SqlCommand(str, con);

SqlCommand cmd2 = new SqlCommand(str1, con);

cmd1.ExecuteNonQuery();

cmd2.ExecuteNonQuery();

}

MessageBox.Show("Backup is successfully created", "Backup", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

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

}

}

public DataTable GetMembers()

{

SqlConnection connection = Connection.Sql\_Connection();

connection.Open();

using (connection)

{

SqlDataAdapter cmd = new SqlDataAdapter("Select \* from [dbo].[user] order by id desc", connection);

DataTable dataTable = new DataTable();

cmd.Fill(dataTable);

return dataTable;

}

}

public DataTable GetJoinThisMonthMembers()

{

SqlConnection connection = Connection.Sql\_Connection();

DataTable dataTable = new DataTable();

connection.Open();

using (connection)

{

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter("Select regno as 'Reg #', name as Name , packege as Package , paid as Paid from [user] Where Month(joindate) = Month(GetDate()) And Year(joindate) = Year(GetDate())", connection);

sqlDataAdapter.Fill(dataTable);

}

return dataTable;

}

public DataTable GetRemaningFeeMembers()

{

SqlConnection connection = Connection.Sql\_Connection();

DataTable dataTable = new DataTable();

connection.Open();

using (connection)

{

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter("Select u.regno as 'Reg #' ,Name , class as Type , prevusFee as Remaning, Fee as 'Last Payment' , date as 'Last Payment Date' from [user] as u Left Join Payments as p On u.regno = p.regno Where prevusFee >0 AND prevusFee Is Not Null Order by date desc", connection);

sqlDataAdapter.Fill(dataTable);

}

return dataTable;

}

public DataTable GetFeeExpiredMembers()

{

SqlConnection connection = Connection.Sql\_Connection();

DataTable dataTable = new DataTable();

connection.Open();

using (connection)

{

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter("Select u.regno as 'Reg #',Name , Packege as Package , date as 'Last Payment Date' , NextDate as 'Fee Expired Date' from [user] as u Left Join Payments as p On u.regno = p.regno Where GETDATE() >= NextDate Order by date desc", connection);

sqlDataAdapter.Fill(dataTable);

}

return dataTable;

}

public bool AddAttendence(int Regno)

{

bool count = false;

int checkuser = 0;

bool checkpayment = false;

SqlConnection connection = Connection.Sql\_Connection();

SqlCommand sqlCommandCheck = new SqlCommand("select count(\*) present from attendence where Convert([date],date) = Convert(date,GetDate()) And regno = @regno", connection);

SqlCommand PaymentCheck = new SqlCommand("Select \* from Payments where regno = @regno", connection);

SqlCommand cmd = new SqlCommand("Insert into Attendence Values(@regno,@date,@time)", connection);

sqlCommandCheck.Parameters.AddWithValue("@regno", Regno);

PaymentCheck.Parameters.AddWithValue("@regno", Regno);

cmd.Parameters.AddWithValue("@regno", Regno);

cmd.Parameters.AddWithValue("@date", DateTime.Now);

cmd.Parameters.AddWithValue("@time", DateTime.Now.ToString("t"));

connection.Open();

using (connection)

{

try

{

SqlDataReader paymentcheck = PaymentCheck.ExecuteReader();

while (paymentcheck.Read())

{

if (paymentcheck.HasRows)

{

count = true;

}

}

paymentcheck.Close();

SqlDataReader dr = sqlCommandCheck.ExecuteReader();

while (dr.Read())

{

checkuser = Convert.ToInt32(dr["present"].ToString());

}

dr.Close();

if (checkuser == 0)

{

cmd.ExecuteNonQuery();

count = true;

}

else

{

MessageBox.Show("Your Attendance is already entered for today", "Already Entered", MessageBoxButtons.OK, MessageBoxIcon.Information);

count = true;

}

}

catch (Exception ex)

{

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

count = false;

}

}

return count;

}

public int TodayPresence()

{

int count = 0;

SqlConnection connection = Connection.Sql\_Connection();

SqlCommand sqlCommand = new SqlCommand("select count(\*) present from attendence where Convert([date],date) = Convert(date,GetDate())", connection);

connection.Open();

using (connection)

{

SqlDataReader dr = sqlCommand.ExecuteReader();

while (dr.Read())

{

count = Convert.ToInt32(dr["present"].ToString());

}

}

return count;

}

public GymMembersInformation AttendanceData(int Regno)

{

SqlConnection connection = Connection.Sql\_Connection();

GymMembersInformation gymMembersInformation = new GymMembersInformation();

connection.Open();

using (connection)

{

SqlCommand cmd = new SqlCommand("Select Top 1 b.regno,name,date,NextDate,class,packege,prevusFee, ReciptNo from[user] as b Left Join Payments as a On a.regno = b.regno where b.regno = @regno Order by a.date desc", connection);

cmd.Parameters.AddWithValue("@regno", Regno);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

gymMembersInformation.Name = dr["name"].ToString();

gymMembersInformation.RegNO = dr["regno"].ToString();

if (dr["date"].ToString().Trim() != "")

gymMembersInformation.FeeDate = Convert.ToDateTime(dr["date"].ToString());

if (dr["NextDate"].ToString() != "")

gymMembersInformation.ExpireDate = Convert.ToDateTime(dr["NextDate"].ToString());

gymMembersInformation.CLASS\_ = dr["class"].ToString();

gymMembersInformation.Package = dr["packege"].ToString();

if (dr["prevusFee"].ToString() != "")

gymMembersInformation.Remaning = Convert.ToInt32(dr["prevusFee"].ToString());

gymMembersInformation.RecipetNo = dr["ReciptNo"].ToString();

}

}

return gymMembersInformation;

}

public void GetPackages()

{

SqlConnection connection = Connection.Sql\_Connection();

Packages.pkg.Clear();

connection.Open();

using (connection)

{

SqlCommand cmd = new SqlCommand("Select packege from Packeges where isdefault = 1 Order by id", connection);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

string pkg = dr["packege"].ToString();

Packages.pkg.Add(pkg);

}

}

}

public DataTable GetAllPackages()

{

SqlConnection connection = Connection.Sql\_Connection();

connection.Open();

using (connection)

{

SqlDataAdapter cmd = new SqlDataAdapter("Select \* from Packeges Order by id", connection);

DataTable dataTable = new DataTable();

cmd.Fill(dataTable);

return dataTable;

}

}

public void UpdatePackage(PackagesInformation packagesInformation)

{

SqlConnection connection = Connection.Sql\_Connection();

SqlCommand cmd = new SqlCommand("Update Packeges set packege=@name,startdate = @startdate, enddate=@enddate, Months= @months ,payfor =@payfor, freemonths = @freemonths, price = @price, isdefault = @isdefault Where id = @id", connection);

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

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

cmd.Parameters.AddWithValue("@startdate", packagesInformation.StartDate);

cmd.Parameters.AddWithValue("@enddate", packagesInformation.EndDate);

cmd.Parameters.AddWithValue("@months", packagesInformation.Months);

if (packagesInformation.PayFor != null)

{

cmd.Parameters.AddWithValue("@payfor", packagesInformation.PayFor);

}

else

{

SqlParameter p = new SqlParameter("@payfor", SqlDbType.Int);

p.Value = DBNull.Value;

cmd.Parameters.Add(p);

}

if (packagesInformation.FreeMonths != null)

{

cmd.Parameters.AddWithValue("@freemonths", packagesInformation.FreeMonths);

}

else

{

SqlParameter p = new SqlParameter("@freemonths", SqlDbType.Int);

p.Value = DBNull.Value;

cmd.Parameters.Add(p);

}

if (packagesInformation.Price != null)

{

cmd.Parameters.AddWithValue("@price", packagesInformation.Price);

}

else

{

SqlParameter p = new SqlParameter("@price", SqlDbType.Int);

p.Value = DBNull.Value;

cmd.Parameters.Add(p);

}

cmd.Parameters.AddWithValue("@isdefault", packagesInformation.IsDefault);

connection.Open();

using (connection)

{

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Package is added", "Added", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

}

public void AddPackage(PackagesInformation packagesInformation)

{

SqlConnection connection = Connection.Sql\_Connection();

SqlCommand cmd = new SqlCommand("Insert into Packeges Values(@name,@startdate,@enddate,@months,@payfor,@freemonths,@price,@isdefault)", connection);

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

cmd.Parameters.AddWithValue("@startdate", packagesInformation.StartDate);

cmd.Parameters.AddWithValue("@enddate", packagesInformation.EndDate);

cmd.Parameters.AddWithValue("@months", packagesInformation.Months);

if (packagesInformation.PayFor != null)

{

cmd.Parameters.AddWithValue("@payfor", packagesInformation.PayFor);

}

else

{

SqlParameter p = new SqlParameter("@payfor", SqlDbType.Int);

p.Value = DBNull.Value;

cmd.Parameters.Add(p);

}

if (packagesInformation.FreeMonths != null)

{

cmd.Parameters.AddWithValue("@freemonths", packagesInformation.FreeMonths);

}

else

{

SqlParameter p = new SqlParameter("@freemonths", SqlDbType.Int);

p.Value = DBNull.Value;

cmd.Parameters.Add(p);

}

if (packagesInformation.Price != null)

{

cmd.Parameters.AddWithValue("@price", packagesInformation.Price);

}

else

{

SqlParameter p = new SqlParameter("@price", SqlDbType.Int);

p.Value = DBNull.Value;

cmd.Parameters.Add(p);

}

cmd.Parameters.AddWithValue("@isdefault", packagesInformation.IsDefault);

connection.Open();

using (connection)

{

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Package is added", "Added", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

}

public DataTable GetMember(string search)

{

SqlConnection connection = Connection.Sql\_Connection();

connection.Open();

using (connection)

{

string query = "Select \* from [dbo].[user] Where regno like @search Or name Like @search Or Father\_Name like @search Or phone like @search Or nic like @search order by id desc";

SqlDataAdapter cmd = new SqlDataAdapter(query, connection);

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

DataTable dataTable = new DataTable();

cmd.Fill(dataTable);

return dataTable;

}

}

public DataTable GetMemberFeeForm(GymMembersInformation gymMembersInformation)

{

SqlConnection connection = Connection.Sql\_Connection();

connection.Open();

using (connection)

{

SqlDataAdapter cmd = new SqlDataAdapter("Select P.id, u.regno,u.name,u.Father\_Name,u.phone, u.packege,nic,joindate,class,prevusFee as 'Remaining Fee', P.date as 'Fee Date', p.NextDate as 'Valid date' from [user] as u left Join Payments as P on u.regno = P.regno where u.regno= @regno order by P.id desc ", connection);

cmd.SelectCommand.Parameters.AddWithValue("@regno", gymMembersInformation.RegNO);

DataTable dataTable = new DataTable();

cmd.Fill(dataTable);

return dataTable;

}

}

public void RemaningFee(GymMembersInformation gymMembersInformation)

{

SqlConnection connection = Connection.Sql\_Connection();

connection.Open();

using (connection)

{

SqlCommand sqlCommand = new SqlCommand("Update Payments set prevusFee = @prevusFee Where id=@id", connection);

sqlCommand.Parameters.AddWithValue("@prevusFee", gymMembersInformation.Remaning);

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

try

{

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Fee is added for Registration No:" + gymMembersInformation.RegNO, "Fee Added", MessageBoxButtons.OK, MessageBoxIcon.Information); ;

}

catch (Exception ex)

{

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

}

}

}

public void AddMemberFee(GymMembersInformation gymMembersInformation)

{

SqlConnection connection = Connection.Sql\_Connection();

connection.Open();

using (connection)

{

SqlCommand sqlCommand = new SqlCommand("Insert into Payments Values( @regno, @fee, @prevusFee , @date , @NextDate , @Type , @reciptNo)", connection);

sqlCommand.Parameters.AddWithValue("@regno", gymMembersInformation.RegNO);

sqlCommand.Parameters.AddWithValue("@fee", gymMembersInformation.Fee);

sqlCommand.Parameters.AddWithValue("@prevusFee", gymMembersInformation.Remaning);

sqlCommand.Parameters.AddWithValue("@date", gymMembersInformation.FeeDate);

sqlCommand.Parameters.AddWithValue("@NextDate", gymMembersInformation.ExpireDate);

sqlCommand.Parameters.AddWithValue("@Type", gymMembersInformation.CLASS\_);

sqlCommand.Parameters.AddWithValue("@reciptNo", gymMembersInformation.RecipetNo);

try

{

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Fee is added for Registration No:" + gymMembersInformation.RegNO, "Fee Added", MessageBoxButtons.OK, MessageBoxIcon.Information); ;

}

catch (Exception ex)

{

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

}

}

}

public void UpdateMember(GymMembersInformation gymMembersInformation)

{

SqlConnection connection = Connection.Sql\_Connection();

SqlCommand sqlCommand = new SqlCommand("Update [dbo].[user] set name = @name, Father\_Name= @Father\_Name,phone = @phone, nic = @cnic,joindate = @joindate,class = @class\_, packege = @package,chest = @chest,hips = @hips,weight = @weight where regno = @regno", connection);

sqlCommand.Parameters.AddWithValue("@regno", gymMembersInformation.RegNO);

sqlCommand.Parameters.AddWithValue("@name", gymMembersInformation.Name);

sqlCommand.Parameters.AddWithValue("@Father\_Name", gymMembersInformation.Father\_Name);

sqlCommand.Parameters.AddWithValue("@phone", gymMembersInformation.Phone);

sqlCommand.Parameters.AddWithValue("@cnic", gymMembersInformation.CNIC);

sqlCommand.Parameters.AddWithValue("@joindate", gymMembersInformation.DateTime);

sqlCommand.Parameters.AddWithValue("@class\_", gymMembersInformation.CLASS\_);

sqlCommand.Parameters.AddWithValue("@package", gymMembersInformation.Package);

sqlCommand.Parameters.AddWithValue("@chest", gymMembersInformation.CHEST);

sqlCommand.Parameters.AddWithValue("@hips", gymMembersInformation.HIPS);

sqlCommand.Parameters.AddWithValue("@weight", gymMembersInformation.WEIGHT);

connection.Open();

using (connection)

{

try

{

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Member :" + gymMembersInformation.Name + " record is Updated Successfully ", "Updated", MessageBoxButtons.OK, MessageBoxIcon.Information); ;

}

catch (Exception ex)

{

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

}

}

}

public void DeleteMember(GymMembersInformation gymMembersInformation)

{

SqlConnection connection = Connection.Sql\_Connection();

SqlCommand sqlCommand = new SqlCommand("Delete from [dbo].[user] where regno = @regno", connection);

sqlCommand.Parameters.AddWithValue("@regno", gymMembersInformation.RegNO);

connection.Open();

using (connection)

{

try

{

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Member :" + gymMembersInformation.Name + " record is Removed ", "Deleted", MessageBoxButtons.OK, MessageBoxIcon.Information); ;

}

catch (Exception ex)

{

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

}

}

}

public void AddNewMember(string regno, string name, string Father\_Name, string phone, string cnic, DateTime joindate, string class\_, string package, bool paid, float? chest, float? hips, float? weight)

{

SqlConnection sqlConnection = Connection.Sql\_Connection();

SqlCommand sqlCommand = new SqlCommand("INSERT INTO [dbo].[user] Values(@regno,@name ,@Father\_Name,@phone,@cnic,@joindate,@class\_, @package, @paid,@chest,@hips,@weight)", sqlConnection);

sqlCommand.Parameters.AddWithValue("@regno", regno);

sqlCommand.Parameters.AddWithValue("@name", name);

sqlCommand.Parameters.AddWithValue("@Father\_Name", Father\_Name);

sqlCommand.Parameters.AddWithValue("@phone", phone);

sqlCommand.Parameters.AddWithValue("@cnic", cnic);

sqlCommand.Parameters.AddWithValue("@joindate", joindate);

sqlCommand.Parameters.AddWithValue("@class\_", class\_);

sqlCommand.Parameters.AddWithValue("@package", package);

sqlCommand.Parameters.AddWithValue("@paid", paid);

sqlCommand.Parameters.AddWithValue("@chest", chest);

sqlCommand.Parameters.AddWithValue("@hips", hips);

sqlCommand.Parameters.AddWithValue("@weight", weight);

sqlConnection.Open();

using (sqlConnection)

{

try

{

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Member :" + name + " is added in the system");

}

catch (SqlException ex)

{

if (ex.Number == 2627)

{

MessageBox.Show("This Reg No already Exists , Please Try another", "Reg No Already Exists", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else throw;

}

}

}

}