﻿using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace NhanSu

{

public partial class formDanhGiaNangLuc : Form

{

SqlConnection connection;

SqlCommand command;

string str = "Data Source=TTTT\\TTTT;Initial Catalog=NhanSu;Integrated Security=True";

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

void loaddata()

{

command = connection.CreateCommand();

command.CommandText = "select \* from DanhGiaNangLuc";

adapter.SelectCommand = command;

table.Clear();

adapter.Fill(table);

dgv\_DGNL.DataSource = table;

}

public formDanhGiaNangLuc()

{

InitializeComponent();

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void formDanhGiaNangLuc\_Load(object sender, EventArgs e)

{

connection = new SqlConnection(str);

connection.Open();

loaddata();

}

private void dgv\_DGNL\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

//tb\_MaDanhGia.ReadOnly = true; //Chỉ được đọc mã đánh giá, không được xóa

int i;

i = dgv\_DGNL.CurrentRow.Index;

tb\_MaDanhGia.Text = dgv\_DGNL.Rows[i].Cells[0].Value.ToString();

dateTime\_Ngay.Text = dgv\_DGNL.Rows[i].Cells[1].Value.ToString();

tb\_NoiDungDG.Text = dgv\_DGNL.Rows[i].Cells[2].Value.ToString();

tb\_KetQua.Text = dgv\_DGNL.Rows[i].Cells[3].Value.ToString();

tb\_MaNhanVien3.Text = dgv\_DGNL.Rows[i].Cells[4].Value.ToString();

tb\_MaDanhGia.Enabled = false;

}

/\*private void button\_Khoitao3\_Click(object sender, EventArgs e)

{

tb\_MaDanhGia.Text = string.Empty;

dateTime\_Ngay.Value = DateTime.Now;

tb\_NoiDungDG.Text= string.Empty;

tb\_KetQua.Text= string.Empty;

tb\_MaNhanVien3.Text= string.Empty;

}\*/

private void button\_TimKiem3\_Click(object sender, EventArgs e)

{

try

{

// Lấy dữ liệu từ các TextBox

string maDanhGia = tb\_MaDanhGia.Text.Trim();

string maNhanVien = tb\_MaNhanVien3.Text.Trim();

string noiDung = tb\_NoiDungDG.Text.Trim();

string ketQua = tb\_KetQua.Text.Trim();

// Kiểm tra các ký tự đặc biệt trong các ô nhập liệu để tránh SQL Injection

if (ContainsSQLInjectionKeywords(noiDung) || ContainsSQLInjectionKeywords(ketQua))

{

MessageBox.Show("Dữ liệu không hợp lệ. Vui lòng kiểm tra lại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Khởi tạo câu lệnh SQL

string query = "SELECT \* FROM DanhGiaNangLuc WHERE 1=1";

// Thêm điều kiện vào câu lệnh SQL dựa trên input (MaDanhGia, MaNhanVien, NoiDung, KetQua)

if (!string.IsNullOrEmpty(maNhanVien))

{

query += " AND MaNhanVien = @MaNhanVien";

}

if (!string.IsNullOrEmpty(maDanhGia))

{

query += " AND MaDanhGia = @MaDanhGia";

}

if (!string.IsNullOrEmpty(noiDung))

{

query += " AND NoiDung LIKE @NoiDung";

}

if (!string.IsNullOrEmpty(ketQua))

{

query += " AND KetQua LIKE @KetQua";

}

// Sử dụng SqlConnection và SqlCommand trong khối using

using (SqlConnection connection = new SqlConnection(str))

using (SqlCommand cmd = new SqlCommand(query, connection))

{

// Thêm tham số cho truy vấn

if (!string.IsNullOrEmpty(maNhanVien))

{

cmd.Parameters.AddWithValue("@MaNhanVien", maNhanVien);

}

if (!string.IsNullOrEmpty(maDanhGia))

{

cmd.Parameters.AddWithValue("@MaDanhGia", maDanhGia);

}

if (!string.IsNullOrEmpty(noiDung))

{

cmd.Parameters.AddWithValue("@NoiDung", "%" + noiDung + "%");

}

if (!string.IsNullOrEmpty(ketQua))

{

cmd.Parameters.AddWithValue("@KetQua", "%" + ketQua + "%");

}

// Mở kết nối

connection.Open();

// Thực thi truy vấn

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dgv\_DGNL.DataSource = dt;

ResetForm();

// Kiểm tra kết quả

if (dt.Rows.Count == 0)

{

MessageBox.Show("Không tìm thấy DGNL nào phù hợp.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tìm kiếm DGNL: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

// Hàm kiểm tra các ký tự SQL Injection

private bool ContainsSQLInjectionKeywords(string input)

{

// Kiểm tra các ký tự đặc biệt hoặc từ khóa nguy hiểm

string[] keywords = { "SELECT", "INSERT", "UPDATE", "DELETE", "DROP", "ALTER", "OR", "AND", "--", ";", "/\*", "\*/" };

foreach (string keyword in keywords)

{

if (input.IndexOf(keyword, StringComparison.OrdinalIgnoreCase) >= 0)

{

return true; //Nếu tìm thấy từ khóa, trả về True

}

}

return false; //Nếu không tìm thấy từ khóa, trả về

}

private void button\_Them3\_Click(object sender, EventArgs e)

{

try

{

// Kiểm tra dữ liệu đầu vào

if (string.IsNullOrWhiteSpace(tb\_MaDanhGia.Text) ||

string.IsNullOrWhiteSpace(tb\_NoiDungDG.Text) ||

string.IsNullOrWhiteSpace(tb\_KetQua.Text) ||

string.IsNullOrWhiteSpace(tb\_MaNhanVien3.Text))

{

MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

string maDanhGia = tb\_MaDanhGia.Text.Trim();

DateTime Ngay = dateTime\_Ngay.Value;

string noiDungDG = tb\_NoiDungDG.Text.Trim();

string ketQua = tb\_KetQua.Text.Trim();

string maNhanVien = tb\_MaNhanVien3.Text.Trim();

// Chuỗi kết nối

using (SqlConnection connection = new SqlConnection(str))

{

connection.Open();

// 1. Lấy mã báo cáo lớn nhất hiện tại và tính mã tự động

string newMaDanhGia = "DG0001"; // Mặc định nếu chưa có bản ghi nào

string queryMaxMaDanhGia = "SELECT MAX(MaDanhGia) FROM DanhGiaNangLuc";

using (SqlCommand getMaxMaDanhGiaCmd = new SqlCommand(queryMaxMaDanhGia, connection))

{

object maxMaDanhGiaObj = getMaxMaDanhGiaCmd.ExecuteScalar();

if (maxMaDanhGiaObj != DBNull.Value && maxMaDanhGiaObj != null)

{

string maxMaDanhGia = maxMaDanhGiaObj.ToString(); // Lấy mã lớn nhất

int maxNumber = int.Parse(maxMaDanhGia.Substring(2)); // Tách số và chữ ra

newMaDanhGia = "DG" + (maxNumber + 1).ToString("D4"); // Tính mã tiếp theo

}

}

// 2. Kiểm tra Ngày có hợp lý hay không (không được lớn hơn ngày hiện tại)

if (Ngay.Date > DateTime.Now.Date)

{

MessageBox.Show("Ngày đánh giá không được lớn hơn ngày hiện tại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// 3. Kiểm tra KetQua phải là số nguyên (0,1,2) --0: tốt, 1: khá, 2: chưa đạt

if (!int.TryParse(ketQua, out int ketQuaInt) || ketQuaInt < 0 || ketQuaInt > 2)

{

MessageBox.Show("Kết quả phải là số nguyên từ 0 đến 2 (0: tốt, 1: khá, 2: chưa đạt).", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// 4. Kiểm tra mã nhân viên (phải tồn tại)

using (SqlCommand checkMaNhanVienCmd = new SqlCommand("SELECT COUNT(\*) FROM NhanVien WHERE MaNhanVien = @MaNhanVien", connection))

{

checkMaNhanVienCmd.Parameters.AddWithValue("@MaNhanVien", maNhanVien);

int countMaNhanVien = (int)checkMaNhanVienCmd.ExecuteScalar();

if (countMaNhanVien == 0)

{

MessageBox.Show("Mã nhân viên không hợp lệ hoặc không tồn tại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

}

// 5. Thêm đánh giá mới

using (SqlCommand insertCmd = new SqlCommand("INSERT INTO DanhGiaNangLuc (MaDanhGia, Ngay, NoiDung, KetQua, MaNhanVien) VALUES (@MaDanhGia, @Ngay, @NoiDung, @KetQua, @MaNhanVien)", connection))

{

insertCmd.Parameters.AddWithValue("@MaDanhGia", newMaDanhGia); // Sửa từ `maDanhGia` thành `newMaDanhGia`

insertCmd.Parameters.AddWithValue("@Ngay", Ngay);

insertCmd.Parameters.AddWithValue("@NoiDung", noiDungDG);

insertCmd.Parameters.AddWithValue("@KetQua", ketQuaInt);

insertCmd.Parameters.AddWithValue("@MaNhanVien", maNhanVien);

int rowsAffected = insertCmd.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Thêm đánh giá thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

ResetForm();

loaddata();

}

else

{

MessageBox.Show("Không thể thêm đánh giá. Vui lòng kiểm tra lại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi thêm đánh giá: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button\_Sua3\_Click(object sender, EventArgs e)

{

try

{

// 0. Vì MaDanhGia đã set ReadOnly nên không thể sửa, chỉ sửa các thuộc tính khác

// 1. Kiểm tra dữ liệu đầu vào

if (string.IsNullOrWhiteSpace(tb\_MaDanhGia.Text) ||

string.IsNullOrWhiteSpace(tb\_NoiDungDG.Text) ||

string.IsNullOrWhiteSpace(tb\_KetQua.Text) ||

string.IsNullOrWhiteSpace(tb\_MaNhanVien3.Text))

{

MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

string maDanhGia = tb\_MaDanhGia.Text.Trim();

DateTime ngay = dateTime\_Ngay.Value;

string noiDungDG = tb\_NoiDungDG.Text.Trim();

string ketQua = tb\_KetQua.Text.Trim();

string maNhanVien = tb\_MaNhanVien3.Text.Trim();

// 2. Kiểm tra ngày có hợp lý không

if (ngay.Date > DateTime.Now.Date)

{

MessageBox.Show("Ngày đánh giá không được lớn hơn ngày hiện tại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// 3. Kiểm tra kết quả (0: tốt, 1: khá, 2: chưa đạt)

if (!int.TryParse(ketQua, out int ketQuaInt) || ketQuaInt < 0 || ketQuaInt > 2)

{

MessageBox.Show("Kết quả phải là số nguyên từ 0 đến 2 (0: tốt, 1: khá, 2: chưa đạt).", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// 4. Kiểm tra mã nhân viên tồn tại

using (SqlConnection connection = new SqlConnection(str))

{

connection.Open();

using (SqlCommand checkMaNhanVienCmd = new SqlCommand("SELECT COUNT(\*) FROM NhanVien WHERE MaNhanVien = @MaNhanVien", connection))

{

checkMaNhanVienCmd.Parameters.AddWithValue("@MaNhanVien", maNhanVien);

int countMaNhanVien = (int)checkMaNhanVienCmd.ExecuteScalar();

if (countMaNhanVien == 0)

{

MessageBox.Show("Mã nhân viên không hợp lệ hoặc không tồn tại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

}

// 5. Thực hiện cập nhật đánh giá năng lực

using (SqlCommand updateCmd = new SqlCommand(

"UPDATE DanhGiaNangLuc " +

"SET Ngay = @Ngay, NoiDung = @NoiDung, KetQua = @KetQua, MaNhanVien = @MaNhanVien " +

"WHERE MaDanhGia = @MaDanhGia", connection))

{

updateCmd.Parameters.AddWithValue("@MaDanhGia", maDanhGia);

updateCmd.Parameters.AddWithValue("@Ngay", ngay);

updateCmd.Parameters.AddWithValue("@NoiDung", noiDungDG);

updateCmd.Parameters.AddWithValue("@KetQua", ketQuaInt);

updateCmd.Parameters.AddWithValue("@MaNhanVien", maNhanVien);

int rowsAffected = updateCmd.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Cập nhật đánh giá năng lực thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddata();

ResetForm();

}

else

{

MessageBox.Show("Không thể cập nhật đánh giá năng lực. Vui lòng kiểm tra lại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi cập nhật đánh giá năng lực: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button\_Xoa3\_Click(object sender, EventArgs e)

{

try

{

// Kiểm tra dữ liệu đầu vào (MaDanhGia/MaNhanVien)

if (string.IsNullOrWhiteSpace(tb\_MaDanhGia.Text.Trim()) &&

string.IsNullOrWhiteSpace(tb\_MaNhanVien3.Text.Trim()))

{

MessageBox.Show("Vui lòng nhập ít nhất một trong hai thông tin: MaDanhGia hoặc MaNhanVien ");

return;

}

// Kiểm tra các trường đầu vào để đảm bảo không chứa từ khóa SQL độc hại

string[] sqlKeywords = { "DROP", "DELETE", "OR", "AND", "TRUNCATE", "UPDATE", "ALTER", "INSERT", "EXEC", "--", ";", "/\*", "\*/" };

foreach (var keyword in sqlKeywords)

{

if (tb\_MaDanhGia.Text.IndexOf(keyword, StringComparison.OrdinalIgnoreCase) >= 0 ||

tb\_MaNhanVien3.Text.IndexOf(keyword, StringComparison.OrdinalIgnoreCase) >= 0 ||

tb\_NoiDungDG.Text.IndexOf(keyword, StringComparison.OrdinalIgnoreCase) >= 0 ||

tb\_KetQua.Text.IndexOf(keyword, StringComparison.OrdinalIgnoreCase) >= 0)

{

MessageBox.Show("Dữ liệu chứa từ khóa SQL không hợp lệ.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

}

// Tạo câu lệnh SQL động

string query = "DELETE FROM DanhGiaNangLuc WHERE 1=1";

// Sử dụng SqlCommand và SqlConnection trong khối using để tự động đóng kết nối

using (SqlConnection connection = new SqlConnection(str))

using (SqlCommand command = connection.CreateCommand())

{

if (!string.IsNullOrWhiteSpace(tb\_MaDanhGia.Text.Trim()))

{

query += " AND MaDanhGia = @MaDanhGia";

command.Parameters.AddWithValue("@MaDanhGia", tb\_MaDanhGia.Text.Trim());

}

if (!string.IsNullOrWhiteSpace(tb\_MaNhanVien3.Text.Trim()))

{

query += " AND MaNhanVien = @MaNhanVien";

command.Parameters.AddWithValue("@MaNhanVien", tb\_MaNhanVien3.Text.Trim());

}

if (!string.IsNullOrWhiteSpace(tb\_NoiDungDG.Text.Trim()))

{

query += " AND NoiDung = @NoiDung";

command.Parameters.AddWithValue("@NoiDung", tb\_NoiDungDG.Text.Trim());

}

if (!string.IsNullOrWhiteSpace(tb\_KetQua.Text.Trim()))

{

query += " AND KetQua = @KetQua";

command.Parameters.AddWithValue("@KetQua", tb\_KetQua.Text.Trim());

}

command.CommandText = query;

// Mở kết nối và thực thi truy vấn

connection.Open();

int rowsAffected = command.ExecuteNonQuery();

// Phản hồi cho người dùng

if (rowsAffected > 0)

{

loaddata();

ResetForm();

MessageBox.Show("Bản ghi đã được xóa thành công.");

}

else

{

MessageBox.Show("Không tìm thấy bản ghi phù hợp để xóa.");

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi xóa bản ghi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void ResetForm()

{

tb\_MaDanhGia.Clear();

tb\_NoiDungDG.Clear();

tb\_KetQua.Clear();

tb\_MaNhanVien3.Clear();

dateTime\_Ngay.Value = DateTime.Now;

dgv\_DGNL.ClearSelection();

}

private void Reset\_click(object sender, EventArgs e)

{

loaddata();

ResetForm();

}

}

}