﻿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 formCongViec\_BaoCao : 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 CongViec\_BaoCao";

adapter.SelectCommand = command;

table.Clear();

adapter.Fill(table);

dgv2.DataSource = table;

}

public formCongViec\_BaoCao()

{

InitializeComponent();

}

private void CongViec\_BaoCao\_Load(object sender, EventArgs e)

{

connection = new SqlConnection(str);

connection.Open();

loaddata();

}

private void dgv2\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

//tb\_MaBaoCao.ReadOnly = true; //Chỉ được đọc mã báo cáo, không được xóa

int i;

i = dgv2.CurrentRow.Index;

tb\_MaBaoCao.Text = dgv2.Rows[i].Cells[0].Value.ToString();

tb\_MaNhanVien2.Text = dgv2.Rows[i].Cells[1].Value.ToString();

tb\_MaCongViec1.Text = dgv2.Rows[i].Cells[2].Value.ToString();

tb\_NoiDung.Text = dgv2.Rows[i].Cells[3].Value.ToString();

datetime\_ThoiGianLap.Text = dgv2.Rows[i].Cells[4].Value.ToString();

tb\_MaBaoCao.Enabled = false;

}

private void button\_Thêm2\_Click(object sender, EventArgs e)

{

string maNhanVien = tb\_MaNhanVien2.Text.Trim();

string maCongViec = tb\_MaCongViec1.Text.Trim();

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

DateTime thoiGianLap = datetime\_ThoiGianLap.Value; // DateTimePicker

if (string.IsNullOrEmpty(maNhanVien) || string.IsNullOrEmpty(maCongViec) || string.IsNullOrEmpty(noiDung))

{

MessageBox.Show("Vui lòng nhập đầy đủ thông tin!", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

using (SqlConnection con = new SqlConnection(str)) // sCon là chuỗi kết nối SQL

{

try

{

con.Open();

string newMaBaoCao = "";

string query = "SELECT TOP 1 MaBaoCao FROM CongViec\_BaoCao ORDER BY MaBaoCao DESC";

SqlCommand cmd = new SqlCommand(query, con);

// Dùng ExecuteScalar để chỉ lấy 1 giá trị

var result = cmd.ExecuteScalar();

if (result != null)

{

string lastMaNghi = result.ToString();

// Lấy số từ mã chấm công cuối cùng và tăng lên 1

int lastNumber = int.Parse(lastMaNghi.Substring(3)); // Lấy phần số sau "MC"

int newNumber = lastNumber + 1;

// Đảm bảo rằng mã mới có độ dài là 9 chữ số

newMaBaoCao = "BCA" + newNumber.ToString().PadLeft(5, '0');

}

else

{

// Nếu không có mã nào, bắt đầu từ "MC000000001"

newMaBaoCao = "BCA00001";

}

// 2. Kiểm tra mã nhân viên

string queryCheckMaNhanVien = "SELECT COUNT(\*) FROM NhanVien WHERE MaNhanVien = @MaNhanVien";

using (SqlCommand checkMaNhanVienCmd = new SqlCommand(queryCheckMaNhanVien, 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;

}

}

// 3. Kiểm tra mã công việc

string queryCheckMaCongViec = "SELECT COUNT(\*) FROM CongViec WHERE MaCongViec = @MaCongViec";

using (SqlCommand checkMaCongViecCmd = new SqlCommand(queryCheckMaCongViec, connection))

{

checkMaCongViecCmd.Parameters.AddWithValue("@MaCongViec", maCongViec);

int countMaCongViec = (int)checkMaCongViecCmd.ExecuteScalar();

if (countMaCongViec == 0)

{

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

return;

}

}

// Thêm dữ liệu vào bảng BAOCAO

string squery = @"INSERT INTO CongViec\_BaoCao (MaBaoCao, MaNhanVien, MaCongViec, NoiDung, ThoiGianLap)

VALUES (@MaBaoCao, @MaNhanVien, @MaCongViec, @NoiDung, @ThoiGianLap)";

SqlCommand command = new SqlCommand(squery, con);

command.Parameters.AddWithValue("@MaBaoCao", newMaBaoCao);

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

command.Parameters.AddWithValue("@MaCongViec", maCongViec);

command.Parameters.AddWithValue("@NoiDung", noiDung);

command.Parameters.AddWithValue("@ThoiGianLap", thoiGianLap);

int rowsAffected = command.ExecuteNonQuery();

// Kiểm tra kết quả

if (rowsAffected > 0)

{

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

loaddata(); // Tải lại dữ liệu

//ResetForm(); // Xóa form nhập liệu

}

else

{

MessageBox.Show("Thêm báo cáo thất bại!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (SqlException ex)

{

MessageBox.Show($"Lỗi SQL: {ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception ex)

{

MessageBox.Show($"Lỗi: {ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private string GenerateMaBaoCao(SqlConnection con)

{

string query = "SELECT TOP 1 MaBaoCao FROM CongViec\_BaoCao ORDER BY MaBaoCao DESC";

SqlCommand command = new SqlCommand(query, con);

object result = command.ExecuteScalar();

string newMaBaoCao;

if (result != null)

{

string lastMaBaoCao = result.ToString();

int number = int.Parse(lastMaBaoCao.Substring(3)) + 1; // Bỏ 3 ký tự đầu (BCA)

newMaBaoCao = "BCA" + number.ToString("D5"); // Format thành 'BCA00001'

}

else

{

newMaBaoCao = "BCA00001"; // Mã đầu tiên nếu chưa có dữ liệu

}

return newMaBaoCao;

}

private void button\_Xóa2\_Click(object sender, EventArgs e)

{

string maBaoCao = tb\_MaBaoCao.Text.Trim();

// Kiểm tra dữ liệu đầu vào

if (string.IsNullOrEmpty(maBaoCao))

{

MessageBox.Show("Vui lòng nhập mã báo cáo cần xóa.", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Xác nhận xóa từ người dùng

var confirmResult = MessageBox.Show($"Bạn có chắc chắn muốn xóa báo cáo có mã {maBaoCao} không?",

"Xác nhận xóa", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (confirmResult == DialogResult.No)

{

return; // Hủy thao tác xóa

}

using (SqlConnection con = new SqlConnection(str)) // sCon là chuỗi kết nối SQL

{

try

{

con.Open();

// Kiểm tra mã báo cáo có tồn tại hay không

string queryCheck = "SELECT COUNT(\*) FROM CongViec\_BaoCao WHERE MaBaoCao = @MaBaoCao";

SqlCommand cmdCheck = new SqlCommand(queryCheck, con);

cmdCheck.Parameters.AddWithValue("@MaBaoCao", maBaoCao);

int exists = (int)cmdCheck.ExecuteScalar();

if (exists == 0)

{

MessageBox.Show("Mã báo cáo không tồn tại. Vui lòng kiểm tra lại.", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Xóa báo cáo

string queryDelete = "DELETE FROM CongViec\_BaoCao WHERE MaBaoCao = @MaBaoCao";

SqlCommand cmdDelete = new SqlCommand(queryDelete, con);

cmdDelete.Parameters.AddWithValue("@MaBaoCao", maBaoCao);

int rowsAffected = cmdDelete.ExecuteNonQuery();

// Kiểm tra kết quả

if (rowsAffected > 0)

{

MessageBox.Show("Xóa báo cáo thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddata(); // Tải lại dữ liệu (nếu có)

ResetForm(); // Reset form nhập liệu

}

else

{

MessageBox.Show("Xóa báo cáo thất bại. Vui lòng thử lại.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (SqlException ex)

{

MessageBox.Show($"Lỗi SQL: {ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception ex)

{

MessageBox.Show($"Lỗi: {ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void button\_Sửa2\_Click(object sender, EventArgs e)

{

try

{

// 1. Kiểm tra dữ liệu đầu vào

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

string.IsNullOrWhiteSpace(tb\_MaNhanVien2.Text) ||

string.IsNullOrWhiteSpace(tb\_MaCongViec1.Text) ||

string.IsNullOrWhiteSpace(tb\_NoiDung.Text))

{

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

return;

}

string maBaoCao = tb\_MaBaoCao.Text.Trim();

string maNhanVien = tb\_MaNhanVien2.Text.Trim();

string maCongViec = tb\_MaCongViec1.Text.Trim();

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

DateTime thoiGianLap = datetime\_ThoiGianLap.Value;

// 2. Kiểm tra mã báo cáo có tồn tại trong bảng không

SqlCommand checkMaBaoCaoCmd = new SqlCommand("SELECT COUNT(\*) FROM CongViec\_BaoCao WHERE MaBaoCao = @MaBaoCao", connection);

checkMaBaoCaoCmd.Parameters.AddWithValue("@MaBaoCao", maBaoCao);

connection.Open();

int countMaBaoCao = (int)checkMaBaoCaoCmd.ExecuteScalar();

connection.Close();

if (countMaBaoCao == 0)

{

MessageBox.Show("Mã báo cáo không tồn tại. Vui lòng kiểm tra lại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// 3. Kiểm tra mã nhân viên có tồn tại không

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

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

connection.Open();

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

connection.Close();

if (countMaNhanVien == 0)

{

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

return;

}

// 4. Kiểm tra mã công việc có tồn tại không

SqlCommand checkMaCongViecCmd = new SqlCommand("SELECT COUNT(\*) FROM CongViec WHERE MaCongViec = @MaCongViec", connection);

checkMaCongViecCmd.Parameters.AddWithValue("@MaCongViec", maCongViec);

connection.Open();

int countMaCongViec = (int)checkMaCongViecCmd.ExecuteScalar();

connection.Close();

if (countMaCongViec == 0)

{

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

return;

}

// 5. Kiểm tra ThoiGianLap ( Trigger kiểm tra ThoiGianLap đã tự động kiểm tra rồi)

// 6. Thực hiện cập nhật báo cáo công việc

SqlCommand updateCmd = new SqlCommand(

"UPDATE CongViec\_BaoCao " +

"SET MaNhanVien = @MaNhanVien, MaCongViec = @MaCongViec, NoiDung = @NoiDung, ThoiGianLap = @ThoiGianLap " +

"WHERE MaBaoCao = @MaBaoCao", connection);

updateCmd.Parameters.AddWithValue("@MaBaoCao", maBaoCao);

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

updateCmd.Parameters.AddWithValue("@MaCongViec", maCongViec);

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

updateCmd.Parameters.AddWithValue("@ThoiGianLap", thoiGianLap);

connection.Open();

int rowsAffected = updateCmd.ExecuteNonQuery();

connection.Close();

if (rowsAffected > 0)

{

MessageBox.Show("Cập nhật báo cáo công việc thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddata();

ResetForm();

}

else

{

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

}

}

catch (SqlException sqlEx)

{

// Xử lý lỗi do trigger phát hiện( ThoiGianlap không hợp lệ)

MessageBox.Show("Lỗi từ SQL Server: " + sqlEx.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

if (connection.State == ConnectionState.Open)

{

connection.Close();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi cập nhật báo cáo công việc: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

if (connection.State == ConnectionState.Open)

{

connection.Close();

}

}

}

private void button\_Timkiem2\_Click(object sender, EventArgs e)

{

try

{

// Lấy dữ liệu từ các TextBox

string maNhanVien = tb\_MaNhanVien2.Text.Trim();

string maCongViec = tb\_MaCongViec1.Text.Trim();

string maBaoCao = tb\_MaBaoCao.Text.Trim();

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

// Kiểm tra các ký tự và từ khóa đặc biệt để tránh SQL Injection

if (ContainsSQLInjectionKeywords(maNhanVien) || ContainsSQLInjectionKeywords(maCongViec) ||

ContainsSQLInjectionKeywords(maBaoCao) || ContainsSQLInjectionKeywords(noiDung))

{

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 CongViec\_BaoCao WHERE 1=1";

// Thêm điều kiện vào câu lệnh SQL nếu có input

if (!string.IsNullOrEmpty(maNhanVien))

{

query += " AND MaNhanVien = @MaNhanVien";

}

if (!string.IsNullOrEmpty(maCongViec))

{

query += " AND MaCongViec = @MaCongViec";

}

if (!string.IsNullOrEmpty(maBaoCao))

{

query += " AND MaBaoCao = @MaBaoCao";

}

if (!string.IsNullOrEmpty(noiDung))

{

query += " AND NoiDung LIKE @NoiDung";

}

// Sử dụng SqlConnection trong khối using

using (SqlConnection connection = new SqlConnection(str))

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

{

// Thêm tham số

if (!string.IsNullOrEmpty(maNhanVien))

{

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

}

if (!string.IsNullOrEmpty(maCongViec))

{

cmd.Parameters.AddWithValue("@MaCongViec", maCongViec);

}

if (!string.IsNullOrEmpty(maBaoCao))

{

cmd.Parameters.AddWithValue("@MaBaoCao", maBaoCao);

}

if (!string.IsNullOrEmpty(noiDung))

{

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

}

// Mở kết nối

connection.Open();

// Thực thi truy vấn

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dgv2.DataSource = dt;

// Kiểm tra có kết quả hay không

if (dt.Rows.Count == 0)

{

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

}

}

}

catch (Exception ex)

{

// Xử lý lỗi

MessageBox.Show("Lỗi khi tìm kiếm báo cáo: " + 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 label\_MaBaoCao\_Click(object sender, EventArgs e)

{

}

private void label\_MaCongViecBC\_Click(object sender, EventArgs e)

{

}

private void ResetForm()

{

// Đặt lại các TextBox về rỗng

tb\_MaBaoCao.Text = string.Empty;

tb\_MaNhanVien2.Text = string.Empty;

tb\_MaCongViec1.Text = string.Empty;

tb\_NoiDung.Text = string.Empty;

// Đặt lại giá trị mặc định cho DateTimePicker

datetime\_ThoiGianLap.Value = DateTime.Now;

// Đặt focus về TextBox đầu tiên (nếu cần)

tb\_MaBaoCao.Focus();

tb\_MaBaoCao.Enabled = true;

}

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

{

loaddata();

ResetForm();

}

/\*private void button\_Khoitao2\_Click(object sender, EventArgs e)

{

tb\_MaBaoCao.Text = string.Empty;

tb\_MaNhanVien2.Text = string.Empty;

tb\_MaCongViec1.Text = string.Empty;

tb\_NoiDung.Text = string.Empty;

datetime\_ThoiGianLap.Value = DateTime.Now;

\*/

}

}