﻿﻿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 LoiPhat

{

public partial class Form1 : 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 LoiPhat";

adapter.SelectCommand = command;

table.Clear();

adapter.Fill(table);

dgv\_loiphat.DataSource = table;

}

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

{

int i;

i = dgv\_loiphat.CurrentRow.Index;

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

tb\_TenLoi.Text = dgv\_loiphat.Rows[i].Cells[1].Value.ToString();

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

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

}

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

connection = new SqlConnection(str);

connection.Open();

loaddata();

}

private void ClearInputFields()

{

tb\_MaLoi.Clear();

tb\_TenLoi.Clear();

tb\_TienPhat.Clear();

tb\_MaLuong.Clear();

tb\_MaLoi.Focus(); // Đưa con trỏ về ô nhập "Mã Lương"

}

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

{

}

private void bt\_Them\_Click(object sender, EventArgs e)

{

try

{

// Lấy dữ liệu từ các TextBox

string tenLoi = tb\_TenLoi.Text.Trim();

string tienPhat = tb\_TienPhat.Text.Trim();

string maLuong = tb\_MaLuong.Text.Trim(); // Nhập mã lương từ người dùng

// Kiểm tra nếu các trường bắt buộc bị bỏ trống

if (string.IsNullOrEmpty(tenLoi) || string.IsNullOrEmpty(tienPhat) || string.IsNullOrEmpty(maLuong))

{

MessageBox.Show("Vui lòng nhập đầy đủ thông tin: Tên Lỗi, Tiền Phạt và Mã Lương.",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Kiểm tra định dạng số cho Tiền Phạt

if (!decimal.TryParse(tienPhat, out \_))

{

MessageBox.Show("Vui lòng nhập đúng định dạng số cho Tiền Phạt.",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Kiểm tra Mã Lương có tồn tại trong bảng 'Luong'

command = connection.CreateCommand();

command.CommandText = "SELECT COUNT(\*) FROM Luong WHERE MaLuong = @MaLuong";

command.Parameters.AddWithValue("@MaLuong", maLuong);

var count = (int)command.ExecuteScalar();

if (count == 0)

{

MessageBox.Show("Mã Lương không tồn tại.",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Tạo mã lỗi tự động (Giả sử mã lỗi có dạng L00001, L00002...)

string maLoi;

command = connection.CreateCommand();

command.CommandText = "SELECT MAX(CAST(SUBSTRING(MaLoi, 2, 5) AS INT)) FROM LoiPhat";

var result = command.ExecuteScalar();

int nextMaLoi = result == DBNull.Value ? 1 : Convert.ToInt32(result) + 1;

maLoi = "L" + nextMaLoi.ToString("D5"); // Tạo mã lỗi mới (L00001, L00002,...)

// Câu lệnh SQL với tham số hóa

string query = "INSERT INTO LoiPhat (MaLoi, TenLoi, TienPhat, MaLuong) VALUES (@MaLoi, @TenLoi, @TienPhat, @MaLuong)";

// Tạo đối tượng SqlCommand

command = connection.CreateCommand();

command.CommandText = query;

// Gắn giá trị tham số

command.Parameters.AddWithValue("@MaLoi", maLoi);

command.Parameters.AddWithValue("@TenLoi", tenLoi);

command.Parameters.AddWithValue("@TienPhat", tienPhat);

command.Parameters.AddWithValue("@MaLuong", maLuong);

// Thực thi lệnh

command.ExecuteNonQuery();

// Cập nhật lại dữ liệu hiển thị

loaddata();

// Hiển thị thông báo thành công

MessageBox.Show("Thêm bản ghi Lỗi Phạt mới thành công!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Xóa trắng các TextBox sau khi thêm

ClearInputFields();

}

catch (Exception ex)

{

// Nếu có lỗi, hiển thị thông báo lỗi

MessageBox.Show("Có lỗi xảy ra: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void bt\_Xem\_Click(object sender, EventArgs e)

{

try

{

// Lấy dữ liệu từ TextBox

string tenLoi = tb\_TenLoi.Text.Trim();

string maLuong = tb\_MaLuong.Text.Trim();

// Kiểm tra nếu không nhập thông tin Tên lỗi và Mã lương

if (string.IsNullOrEmpty(tenLoi) && string.IsNullOrEmpty(maLuong))

{

MessageBox.Show("Vui lòng nhập ít nhất Tên lỗi hoặc Mã lương để tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Kết thúc phương thức

}

// Xây dựng truy vấn SQL động với các tham số

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

if (!string.IsNullOrEmpty(tenLoi))

{

query += " AND TenLoi LIKE @TenLoi";

}

if (!string.IsNullOrEmpty(maLuong))

{

query += " AND MaLuong = @MaLuong";

}

// Tạo đối tượng SqlCommand

SqlCommand command = connection.CreateCommand();

command.CommandText = query;

// Gắn tham số nếu có

if (!string.IsNullOrEmpty(tenLoi))

{

command.Parameters.AddWithValue("@TenLoi", "%" + tenLoi + "%");

}

if (!string.IsNullOrEmpty(maLuong))

{

command.Parameters.AddWithValue("@MaLuong", maLuong);

}

// Thực thi truy vấn và hiển thị kết quả

SqlDataAdapter adapter = new SqlDataAdapter(command);

DataTable table = new DataTable();

adapter.Fill(table);

// Kiểm tra nếu không có dữ liệu trả về

if (table.Rows.Count == 0)

{

MessageBox.Show("Không tìm thấy dữ liệu nào khớp với điều kiện tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

dgv\_loiphat.DataSource = table;

MessageBox.Show("Hiển thị dữ liệu thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

// Nếu có lỗi, hiển thị thông báo lỗi

MessageBox.Show("Có lỗi xảy ra: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void bt\_Sua\_Click(object sender, EventArgs e)

{

try

{

// Lấy dữ liệu từ các TextBox

string maLoi = tb\_MaLoi.Text.Trim();

string tenLoi = tb\_TenLoi.Text.Trim();

string tienPhat = tb\_TienPhat.Text.Trim();

// Kiểm tra nếu Mã Lỗi không được nhập

if (string.IsNullOrEmpty(maLoi))

{

MessageBox.Show("Vui lòng nhập Mã Lỗi để cập nhật thông tin!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Kiểm tra nếu không có gì để cập nhật

if (string.IsNullOrEmpty(tenLoi) && string.IsNullOrEmpty(tienPhat))

{

MessageBox.Show("Vui lòng nhập ít nhất một thông tin (Tên Lỗi hoặc Tiền Phạt) để cập nhật!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Kiểm tra định dạng số cho Tiền Phạt (nếu có nhập)

if (!string.IsNullOrEmpty(tienPhat) && !decimal.TryParse(tienPhat, out \_))

{

MessageBox.Show("Vui lòng nhập đúng định dạng số cho Tiền Phạt.",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Tạo câu lệnh SQL động với tham số hóa

string query = "UPDATE LoiPhat SET";

List<string> updates = new List<string>();

if (!string.IsNullOrEmpty(tenLoi))

{

updates.Add(" TenLoi = @TenLoi");

}

if (!string.IsNullOrEmpty(tienPhat))

{

updates.Add(" TienPhat = @TienPhat");

}

query += string.Join(",", updates) + " WHERE MaLoi = @MaLoi";

// Tạo đối tượng SqlCommand

SqlCommand command = connection.CreateCommand();

command.CommandText = query;

// Gắn giá trị tham số

command.Parameters.AddWithValue("@MaLoi", maLoi);

if (!string.IsNullOrEmpty(tenLoi))

{

command.Parameters.AddWithValue("@TenLoi", tenLoi);

}

if (!string.IsNullOrEmpty(tienPhat))

{

command.Parameters.AddWithValue("@TienPhat", tienPhat);

}

// Thực thi lệnh

int rowsAffected = command.ExecuteNonQuery();

// Hiển thị thông báo

if (rowsAffected > 0)

{

MessageBox.Show("Cập nhật thông tin lỗi phạt thành công!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddata(); // Làm mới danh sách

// Xóa trắng các TextBox

ClearInputFields();

}

else

{

MessageBox.Show("Không tìm thấy Mã Lỗi để cập nhật!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

// Nếu có lỗi, hiển thị thông báo lỗi

MessageBox.Show("Có lỗi xảy ra: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void bt\_Xoa\_Click(object sender, EventArgs e)

{

try

{

// Lấy Mã Lỗi từ TextBox

string maLoi = tb\_MaLoi.Text.Trim();

// Kiểm tra nếu Mã Lỗi không được nhập

if (string.IsNullOrEmpty(maLoi))

{

MessageBox.Show("Vui lòng nhập Mã Lỗi để xóa!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Câu lệnh SQL xóa bản ghi theo Mã Lỗi

string query = "DELETE FROM LoiPhat WHERE MaLoi = @MaLoi";

// Tạo đối tượng SqlCommand

SqlCommand command = connection.CreateCommand();

command.CommandText = query;

// Gắn giá trị tham số

command.Parameters.AddWithValue("@MaLoi", maLoi);

// Thực thi lệnh

int rowsAffected = command.ExecuteNonQuery();

// Kiểm tra số lượng bản ghi bị xóa

if (rowsAffected > 0)

{

MessageBox.Show("Xóa bản ghi thành công!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddata(); // Làm mới danh sách sau khi xóa

// Xóa trắng các TextBox

ClearInputFields();

}

else

{

MessageBox.Show("Không tìm thấy Mã Lỗi để xóa!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

// Nếu có lỗi, hiển thị thông báo lỗi

MessageBox.Show("Có lỗi xảy ra: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void gr\_Thaotac\_Enter(object sender, EventArgs e)

{

}

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

{

loaddata();

ClearInputFields();

}

}

}