﻿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 BangChungChi

{

public partial class ChungChi : 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();

private void loaddata()

{

loaddata(command);

}

void loaddata(SqlCommand command)

{

command = connection.CreateCommand();

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

adapter.SelectCommand = command;

table.Clear();

adapter.Fill(table);

dgv.DataSource = table;

}

public ChungChi()

{

InitializeComponent();

}

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

{

// Lấy dữ liệu từ các TextBox

string maChungChi = tb\_machungchi.Text.Trim();

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

// Kiểm tra dữ liệu đầu vào

if (string.IsNullOrEmpty(maChungChi))

{

MessageBox.Show("Vui lòng nhập Mã chứng chỉ để xóa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Tạo câu lệnh SQL xóa

command = connection.CreateCommand();

if (!string.IsNullOrEmpty(maNhanVien))

{

// Xóa theo cả Mã chứng chỉ và Mã nhân viên

command.CommandText = "DELETE FROM CHUNGCHI WHERE MaChungChi = @MaChungChi AND MaNhanVien = @MaNhanVien";

command.Parameters.AddWithValue("@MaChungChi", maChungChi);

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

}

else

{

// Xóa chỉ theo Mã chứng chỉ (không cần Mã nhân viên)

command.CommandText = "DELETE FROM CHUNGCHI WHERE MaChungChi = @MaChungChi";

command.Parameters.AddWithValue("@MaChungChi", maChungChi);

}

// Xác nhận xóa

DialogResult confirmResult = MessageBox.Show("Bạn có chắc chắn muốn xóa bản ghi này không?", "Xác nhận xóa", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (confirmResult == DialogResult.Yes)

{

// Thực thi lệnh SQL

int rowsAffected = command.ExecuteNonQuery();

// Kiểm tra kết quả

if (rowsAffected > 0)

{

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

loaddata();

ResetForm();

}

else

{

MessageBox.Show("Không tìm thấy bản ghi để xóa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

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

{

string tenChungChi = tb\_tenchungchi.Text.Trim();

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

// Tạo câu lệnh SQL hiển thị danh sách chứng chỉ

command = connection.CreateCommand();

string sqlQuery = "SELECT \* FROM CHUNGCHI WHERE 1=1";

// Thêm điều kiện tìm kiếm nếu người dùng nhập Tên chứng chỉ hoặc Mã nhân viên

if (!string.IsNullOrEmpty(tenChungChi))

{

sqlQuery += " AND TenChungChi LIKE @TenChungChi";

command.Parameters.AddWithValue("@TenChungChi", "%" + tenChungChi + "%");

}

if (!string.IsNullOrEmpty(maNhanVien))

{

sqlQuery += " AND MaNhanVien like @MaNhanVien";

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

}

command.CommandText = sqlQuery;

SqlDataAdapter adapter = new SqlDataAdapter(command);

DataTable table = new DataTable();

adapter.Fill(table);

dgv.DataSource = table;

ResetForm();

}

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

{

int i;

i = dgv.CurrentRow.Index;

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

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

dtime\_hansd.Text = dgv.Rows[i].Cells[2].Value.ToString();

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

tb\_machungchi.Enabled = false;

}

private void groupBox2\_Enter(object sender, EventArgs e) { }

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

{

connection = new SqlConnection(str);

connection.Open();

loaddata();

}

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

{

// Tạo MaChungChi tự động

command = connection.CreateCommand();

command.CommandText = "SELECT TOP 1 MaChungChi FROM CHUNGCHI ORDER BY MaChungChi DESC";

object result = command.ExecuteScalar();

string newMaChungChi;

if (result != null)

{

string lastMaChungChi = result.ToString();

int number = int.Parse(lastMaChungChi.Substring(2)) + 1;

newMaChungChi = "C" + number.ToString("D5");

}

else

{

newMaChungChi = "C00001";

}

// Kiểm tra nếu Mã nhân viên tồn tại

command = connection.CreateCommand();

command.CommandText = "SELECT COUNT(\*) FROM NHANVIEN WHERE MaNhanVien = @MaNhanVien";

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

int employeeExists = (int)command.ExecuteScalar();

if (employeeExists == 0)

{

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

return;

}

// Thêm chứng chỉ mới

command = connection.CreateCommand();

command.CommandText = "INSERT INTO CHUNGCHI VALUES (@MaChungChi, @TenChungChi, @HanSDChungChi, @MaNhanVien)";

command.Parameters.AddWithValue("@MaChungChi", newMaChungChi); // Sử dụng mã tự động

command.Parameters.AddWithValue("@TenChungChi", tb\_tenchungchi.Text.Trim());

command.Parameters.AddWithValue("@HanSDChungChi", dtime\_hansd.Value);

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

command.ExecuteNonQuery();

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

loaddata();

ResetForm();

}

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

{

string maChungChi = tb\_machungchi.Text.Trim();

string tenChungChi = tb\_tenchungchi.Text.Trim();

DateTime hanSD = dtime\_hansd.Value;

// Kiểm tra dữ liệu đầu vào

if (string.IsNullOrEmpty(maChungChi))

{

MessageBox.Show("Vui lòng nhập Mã chứng chỉ để cập nhật!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Tạo lệnh SQL cập nhật thông tin

command = connection.CreateCommand();

command.CommandText = "UPDATE CHUNGCHI SET TenChungChi = @TenChungChi, HanSDChungChi = @HanSD WHERE MaChungChi = @MaChungChi";

// Thêm tham số cho câu lệnh SQL

command.Parameters.AddWithValue("@TenChungChi", tenChungChi);

command.Parameters.AddWithValue("@HanSD", hanSD);

command.Parameters.AddWithValue("@MaChungChi", maChungChi);

try

{

// Thực thi lệnh SQL

int rowsAffected = command.ExecuteNonQuery();

// Kiểm tra kết quả

if (rowsAffected > 0)

{

MessageBox.Show("Cập nhật thông tin chứng chỉ thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddata();

ResetForm();

}

else

{

MessageBox.Show("Không tìm thấy Mã chứng chỉ để cập nhật!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

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

}

}

private void label7\_Click(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void tb\_machungchi\_TextChanged(object sender, EventArgs e)

{

}

private void tb\_manhanvien\_TextChanged(object sender, EventArgs e)

{

}

private void ResetForm()

{

// Xóa nội dung các TextBox

tb\_machungchi.Text = string.Empty;

tb\_tenchungchi.Text = string.Empty;

tb\_manhanvien.Text = string.Empty;

tb\_tenchungchi.Text = string.Empty;

// Đặt lại giá trị mặc định cho DateTimePicker

dtime\_hansd.Value = DateTime.Now;

// Xóa lựa chọn trên DataGridView

if (dgv.DataSource != null)

{

dgv.ClearSelection();

}

// Đặt lại tiêu điểm về TextBox đầu tiên

tb\_machungchi.Focus();

}

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

{

ResetForm();

loaddata();

}

}

}