﻿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 BangBangCap

{

public partial class BangCap : 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 BANGCAP";

adapter.SelectCommand = command;

table.Clear();

adapter.Fill(table);

dvg.DataSource = table;

}

public BangCap()

{

InitializeComponent();

}

private void textBox4\_TextChanged(object sender, EventArgs e) { }

private void textBox3\_TextChanged(object sender, EventArgs e) { }

private void textBox2\_TextChanged(object sender, EventArgs e) { }

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

private void button3\_Click(object sender, EventArgs e)

{

}

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

{

int i;

i = dvg.CurrentRow.Index;

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

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

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

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

tb\_manhanvien.Text = dvg.Rows[i].Cells[4].Value.ToString();

tb\_mabangcap.Enabled = false;

}

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

{

}

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

{

connection = new SqlConnection(str);

connection.Open();

loaddata();

}

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

{

string maBangCap = tb\_mabangcap.Text.Trim();

string tenBangCap = tb\_tenbangcap.Text.Trim();

string trinhDo = tb\_trinhdo.Text.Trim();

// Kiểm tra xem Mã bằng cấp có được nhập không (bắt buộc)

if (string.IsNullOrEmpty(maBangCap))

{

MessageBox.Show("Vui lòng nhập Mã bằng cấp!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

try

{

// Tạo lệnh SQL cập nhật

command = connection.CreateCommand();

command.CommandText = "UPDATE BANGCAP SET TenBangCap = @TenBangCap, TrinhDo = @TrinhDo WHERE MaBangCap = @MaBangCap";

// Sử dụng SqlParameter với kiểu dữ liệu cụ thể

command.Parameters.Add(new SqlParameter("@TenBangCap", SqlDbType.NVarChar, 100) { Value = tenBangCap });

command.Parameters.Add(new SqlParameter("@TrinhDo", SqlDbType.NVarChar, 50) { Value = trinhDo });

command.Parameters.Add(new SqlParameter("@MaBangCap", SqlDbType.NVarChar, 10) { Value = maBangCap });

// Thực thi lệnh SQL

int rowsAffected = command.ExecuteNonQuery();

// Kiểm tra kết quả

if (rowsAffected > 0)

{

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

loaddata();

ResetForm();

}

else

{

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

}

}

catch (Exception ex)

{

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

}

}

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

{

string tenBangCap = tb\_tenbangcap.Text.Trim();

string trinhDo = tb\_trinhdo.Text.Trim();

string donViCap = tb\_donvicap.Text.Trim();

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

// Kiểm tra dữ liệu đầu vào

if (string.IsNullOrEmpty(tenBangCap) || string.IsNullOrEmpty(trinhDo) || string.IsNullOrEmpty(donViCap) || string.IsNullOrEmpty(maNhanVien))

{

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();

// 1. Tạo MaBangCap tự động

string queryGetLastID = "SELECT TOP 1 MaBangCap FROM BANGCAP ORDER BY MaBangCap DESC";

SqlCommand cmdGetLastID = new SqlCommand(queryGetLastID, con);

object result = cmdGetLastID.ExecuteScalar();

string newMaBangCap;

if (result != null)

{

string lastMaBangCap = result.ToString();

int number = int.Parse(lastMaBangCap.Substring(1)) + 1;

newMaBangCap = "B" + number.ToString("D5");

}

else

{

newMaBangCap = "B00001"; // Nếu bảng rỗng

}

// 2. Kiểm tra MaNhanVien tồn tại

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

SqlCommand cmdCheckEmployee = new SqlCommand(queryCheckEmployee, con);

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

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

if (exists == 0)

{

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

return;

}

// 3. Thêm dữ liệu vào bảng BANGCAP

string queryInsert = @"

INSERT INTO BANGCAP (MaBangCap, TenBangCap, TrinhDo, DonViCap, MaNhanVien)

VALUES (@MaBangCap, @TenBangCap, @TrinhDo, @DonViCap, @MaNhanVien)";

SqlCommand cmdInsert = new SqlCommand(queryInsert, con);

cmdInsert.Parameters.AddWithValue("@MaBangCap", newMaBangCap);

cmdInsert.Parameters.AddWithValue("@TenBangCap", tenBangCap);

cmdInsert.Parameters.AddWithValue("@TrinhDo", trinhDo);

cmdInsert.Parameters.AddWithValue("@DonViCap", donViCap);

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

int rowsAffected = cmdInsert.ExecuteNonQuery();

// 4. Kiểm tra kết quả thêm

if (rowsAffected > 0)

{

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

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

ResetForm(); // Reset form nhập liệu

}

else

{

MessageBox.Show("Thêm bằng cấp thất bại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (FormatException)

{

MessageBox.Show("Lỗi định dạng dữ liệu. 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 string GenerateMaBangCap()

{

SqlConnection con = new SqlConnection(str);

string newMaBC = "";

try

{

con.Open();

string query = "SELECT TOP 1 MaBangCap FROM BangCap ORDER BY MaBangCap DESC";

SqlCommand cmd = new SqlCommand(query, con);

// Dùng ExecuteScalar để chỉ lấy 1 giá trị

var result = cmd.ExecuteScalar();

if (result != null)

{

string lastMaBC = result.ToString();

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

int lastNumber = int.Parse(lastMaBC.Substring(2)); // Lấy phần số sau "MC"

int newNumber = lastNumber + 1;

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

newMaBC = "B" + newNumber.ToString().PadLeft(5, '0');

}

else

{

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

newMaBC = "B00001";

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tạo mã bằng cấp!");

}

finally

{

if (con.State == ConnectionState.Open)

con.Close();

}

return newMaBC;

}

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

{

// Lấy giá trị từ TextBox

string tenBangCap = tb\_tenbangcap.Text.Trim();

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

// Tạo lệnh SQL

command = connection.CreateCommand();

if (!string.IsNullOrEmpty(tenBangCap) && !string.IsNullOrEmpty(maNhanVien))

{

command.CommandText = "SELECT \* FROM BANGCAP WHERE TenBangCap LIKE @TenBangCap AND MaNhanVien LIKE @MaNhanVien";

command.Parameters.Add(new SqlParameter("@TenBangCap", SqlDbType.NVarChar, 100) { Value = "%" + tenBangCap + "%" });

command.Parameters.Add(new SqlParameter("@MaNhanVien", SqlDbType.NVarChar, 10) { Value = "%" + maNhanVien + "%" });

}

else if (!string.IsNullOrEmpty(tenBangCap))

{

command.CommandText = "SELECT \* FROM BANGCAP WHERE TenBangCap LIKE @TenBangCap";

command.Parameters.Add(new SqlParameter("@TenBangCap", SqlDbType.NVarChar, 100) { Value = "%" + tenBangCap + "%" });

}

else if (!string.IsNullOrEmpty(maNhanVien))

{

command.CommandText = "SELECT \* FROM BANGCAP WHERE MaNhanVien LIKE @MaNhanVien";

command.Parameters.Add(new SqlParameter("@MaNhanVien", SqlDbType.NVarChar, 10) { Value = "%" + maNhanVien + "%" });

}

else

{

command.CommandText = "SELECT \* FROM BANGCAP";

}

// Liên kết command với adapter

adapter.SelectCommand = command;

// Xóa dữ liệu cũ và nạp dữ liệu mới vào bảng

table.Clear();

adapter.Fill(table);

// Hiển thị lên DataGridView

dvg.DataSource = table;

}

private void MaNhanVien\_Click(object sender, EventArgs e) { }

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

{

}

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

{

}

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

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void btXoaClick(object sender, EventArgs e)

{

try

{

// Lấy dữ liệu từ các TextBox

string maBangCap = tb\_mabangcap.Text.Trim();

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

// Kiểm tra dữ liệu đầu vào

if (string.IsNullOrEmpty(maBangCap))

{

MessageBox.Show("Vui lòng nhập Mã bằng cấp 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 bằng cấp và Mã nhân viên

command.CommandText = "DELETE FROM BANGCAP WHERE MaBangCap = @MaBangCap AND MaNhanVien = @MaNhanVien";

command.Parameters.Add(new SqlParameter("@MaBangCap", SqlDbType.NVarChar, 10) { Value = maBangCap });

command.Parameters.Add(new SqlParameter("@MaNhanVien", SqlDbType.NVarChar, 10) { Value = maNhanVien });

}

else

{

// Xóa chỉ theo Mã bằng cấp (không cần Mã nhân viên)

command.CommandText = "DELETE FROM BANGCAP WHERE MaBangCap = @MaBangCap";

command.Parameters.Add(new SqlParameter("@MaBangCap", SqlDbType.NVarChar, 10) { Value = maBangCap });

}

// 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)

{

int rowsAffected = command.ExecuteNonQuery();

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);

}

}

}

catch (Exception ex)

{

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

}

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void ResetForm()

{

// Đặt lại các TextBox về rỗng

tb\_mabangcap.Text = string.Empty;

tb\_tenbangcap.Text = string.Empty;

tb\_trinhdo.Text = string.Empty;

tb\_donvicap.Text = string.Empty;

tb\_manhanvien.Text = string.Empty;

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

tb\_mabangcap.Focus();

tb\_mabangcap.Enabled = true;

}

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

{

ResetForm();

loaddata();

}

}

}