﻿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;

using System.Data.SqlClient;

namespace frmChucVu

{

public partial class frmChucVu : Form

{

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

public frmChucVu()

{

InitializeComponent();

}

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối DB", "Thông báo");

}

string sQuery = "select \* from ChucVu";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

// Đổ dữ liệu vào DataTable

adapter.Fill(ds, "ChucVu");

// Hiển thị kết quả trên DataGridView

dgvChucVu.DataSource = ds.Tables["ChucVu"];

con.Close();

}

// Hàm ResetForm để làm trống các trường nhập liệu

private void ResetForm()

{

txtMaCVu.Text = "";

txtTenCVu.Text = "";

numHesoLuong.Value = 0;

numPhuCap.Value = 0;

txtMaCVu.Enabled = true;

}

// Hàm LoadDataChucVu (giả sử bạn có một DataGridView hiển thị danh sách chức vụ)

private void LoadDataChucVu()

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối DB", "Thông báo");

}

string sQuery = "select \* from ChucVu";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

// Đổ dữ liệu vào DataTable

adapter.Fill(ds, "ChucVu");

// Hiển thị kết quả trên DataGridView

dgvChucVu.DataSource = ds.Tables["ChucVu"];

con.Close();

}

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

{

// Lấy thông tin từ các control trên form

string maChucVu = GenerateMaChucVu();

string tenChucVu = txtTenCVu.Text.Trim();

decimal heSoLuong = numHesoLuong.Value;

decimal phuCap = numPhuCap.Value;

// Kiểm tra kết nối SQL

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

{

try

{

con.Open();

// Lệnh SQL thêm mới

string query = @"INSERT INTO ChucVu (MaChucVu, TenChucVu, HeSoLuong, PhuCap)

VALUES (@MaChucVu, @TenChucVu, @HeSoLuong, @PhuCap)";

// Tạo SqlCommand

SqlCommand command = new SqlCommand(query, con);

// Thêm tham số vào câu lệnh SQL

command.Parameters.AddWithValue("@MaChucVu", maChucVu);

command.Parameters.AddWithValue("@TenChucVu", tenChucVu);

command.Parameters.AddWithValue("@HeSoLuong", heSoLuong);

command.Parameters.AddWithValue("@PhuCap", phuCap);

// Thực thi lệnh

int rowsAffected = command.ExecuteNonQuery();

// Kiểm tra kết quả

if (rowsAffected > 0)

{

MessageBox.Show("Thêm chức vụ thành công!", "Thông báo");

LoadDataChucVu();

}

else

{

MessageBox.Show("Thêm chức vụ thất bại.", "Thông báo");

}

// Cập nhật lại dữ liệu (nếu cần) và reset form

ResetForm();

}

catch (SqlException ex)

{

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

}

catch (Exception ex)

{

MessageBox.Show($"Lỗi: {ex.Message}", "Lỗi");

}

}

}

private void dgvChucVu\_CellClick(object sender, DataGridViewCellEventArgs e)

{

// Kiểm tra nếu click vào tiêu đề cột thì không thực hiện

if (e.RowIndex < 0)

{

return;

}

// Lấy hàng được chọn

DataGridViewRow selectedRow = dgvChucVu.Rows[e.RowIndex];

try

{

// Gán giá trị từ các ô (cells) vào các control trên form

txtMaCVu.Text = selectedRow.Cells["MaChucVu"].Value?.ToString().Trim();

txtTenCVu.Text = selectedRow.Cells["TenChucVu"].Value?.ToString().Trim();

numHesoLuong.Value = Convert.ToDecimal(selectedRow.Cells["HeSoLuong"].Value ?? 0.00m);

numPhuCap.Value = Convert.ToDecimal(selectedRow.Cells["PhuCap"].Value ?? 0.000m);

txtMaCVu.Enabled = false;

}

catch (Exception ex)

{

MessageBox.Show($"Lỗi khi hiển thị dữ liệu: {ex.Message}", "Lỗi");

}

}

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

{

// Lấy dữ liệu từ các control trên form

string maChucVu = txtMaCVu.Text.Trim();

string tenChucVu = txtTenCVu.Text.Trim();

decimal heSoLuong = numHesoLuong.Value;

decimal phuCap = numPhuCap.Value;

// Kiểm tra dữ liệu đầu vào

if (string.IsNullOrEmpty(maChucVu))

{

MessageBox.Show("Mã chức vụ không được để trống!", "Thông báo");

return;

}

if (string.IsNullOrEmpty(tenChucVu))

{

MessageBox.Show("Tên chức vụ không được để trống!", "Thông báo");

return;

}

// Kết nối đến cơ sở dữ liệu

using (SqlConnection con = new SqlConnection(sCon))

{

try

{

con.Open();

// Câu lệnh SQL để cập nhật dữ liệu

string query = @"UPDATE ChucVu

SET TenChucVu = @TenChucVu,

HeSoLuong = @HeSoLuong,

PhuCap = @PhuCap

WHERE MaChucVu = @MaChucVu";

// Tạo SqlCommand

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

{

// Thêm tham số vào câu lệnh

cmd.Parameters.AddWithValue("@MaChucVu", maChucVu);

cmd.Parameters.AddWithValue("@TenChucVu", tenChucVu);

cmd.Parameters.AddWithValue("@HeSoLuong", heSoLuong);

cmd.Parameters.AddWithValue("@PhuCap", phuCap);

// Thực thi lệnh SQL

int rowsAffected = cmd.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Cập nhật thành công!", "Thông báo");

}

else

{

MessageBox.Show("Không tìm thấy chức vụ để cập nhật!", "Thông báo");

}

}

// Cập nhật lại DataGridView

LoadDataChucVu();

ResetForm();

}

catch (SqlException ex)

{

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

}

catch (Exception ex)

{

MessageBox.Show($"Lỗi: {ex.Message}", "Lỗi");

}

}

}

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

{

string maChucVu = txtMaCVu.Text.Trim();

// Kiểm tra xem mã chức vụ có trống hay không

if (string.IsNullOrEmpty(maChucVu))

{

MessageBox.Show("Vui lòng chọn chức vụ cần xóa!", "Thông báo");

return;

}

// Hiển thị hộp thoại xác nhận trước khi xóa chức vụ

DialogResult result = MessageBox.Show($"Bạn có chắc chắn muốn xóa chức vụ có mã '{maChucVu}' không?",

"Xác nhận xóa chức vụ",

MessageBoxButtons.YesNo,

MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

// Tạo một hộp thoại yêu cầu người dùng chọn chức vụ mới

string queryChucVuMoi = "SELECT MaChucVu, TenChucVu FROM ChucVu WHERE MaChucVu != @MaChucVu";

// Mở kết nối với cơ sở dữ liệu và lấy dữ liệu chức vụ

using (SqlConnection con = new SqlConnection(sCon))

{

try

{

con.Open();

SqlCommand cmd = new SqlCommand(queryChucVuMoi, con);

cmd.Parameters.AddWithValue("@MaChucVu", maChucVu);

SqlDataReader reader = cmd.ExecuteReader();

// Tạo một ComboBox tạm để người dùng chọn chức vụ mới

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

while (reader.Read())

{

chucVuList.Add(reader["MaChucVu"].ToString() + " - " + reader["TenChucVu"].ToString());

}

reader.Close();

// Kiểm tra nếu có chức vụ khác để chọn

if (chucVuList.Count == 0)

{

MessageBox.Show("Không có chức vụ nào để chọn", "Thông báo");

return;

}

// Hiển thị ComboBox cho người dùng chọn chức vụ mới

string selectedChucVu = ShowComboBoxDialog(chucVuList);

// Kiểm tra người dùng có chọn chức vụ mới không

if (string.IsNullOrEmpty(selectedChucVu))

{

MessageBox.Show("Vui lòng chọn chức vụ mới để chuyển các nhân viên.", "Thông báo");

return;

}

// Cập nhật tất cả nhân viên có chức vụ này sang chức vụ mới

string updateQuery = @"

UPDATE NhanVien

SET MaChucVu = @MaChucVuMoi

WHERE MaChucVu = @MaChucVu";

SqlCommand updateCommand = new SqlCommand(updateQuery, con);

updateCommand.Parameters.AddWithValue("@MaChucVuMoi", selectedChucVu.Split('-')[0].Trim()); // Mã chức vụ mới

updateCommand.Parameters.AddWithValue("@MaChucVu", maChucVu);

updateCommand.ExecuteNonQuery();

// Xóa chức vụ

string deleteQuery = "DELETE FROM ChucVu WHERE MaChucVu = @MaChucVu";

SqlCommand deleteCommand = new SqlCommand(deleteQuery, con);

deleteCommand.Parameters.AddWithValue("@MaChucVu", maChucVu);

deleteCommand.ExecuteNonQuery();

MessageBox.Show("Xóa chức vụ thành công!", "Thông báo");

LoadDataChucVu();

}

catch (SqlException ex)

{

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

}

catch (Exception ex)

{

MessageBox.Show($"Lỗi: {ex.Message}", "Lỗi");

}

}

}

else

{

MessageBox.Show("Đã hủy thao tác xóa chức vụ.", "Thông báo");

}

ResetForm();

}

private string ShowComboBoxDialog(List<string> chucVuList)

{

// Tạo form tạm để hiển thị ComboBox

Form form = new Form();

form.Text = "Chọn chức vụ thay thế cho nhân viên"; // Tiêu đề form

// Tạo Label cho tiêu đề

Label label = new Label();

label.Text = "Chọn chức vụ thay thế cho nhân viên";

label.Dock = DockStyle.Top;

label.TextAlign = ContentAlignment.MiddleCenter;

label.Font = new Font("Arial", 12, FontStyle.Bold);

label.Height = 40;

ComboBox comboBox = new ComboBox();

comboBox.DataSource = chucVuList;

comboBox.DropDownStyle = ComboBoxStyle.DropDownList;

comboBox.Width = 500;

comboBox.Dock = DockStyle.Top;

Button buttonOk = new Button() { Text = "Ok", DialogResult = DialogResult.OK, Dock = DockStyle.Bottom };

Button buttonCancel = new Button() { Text = "Cancel", DialogResult = DialogResult.Cancel, Dock = DockStyle.Bottom };

form.Controls.Add(label);

form.Controls.Add(comboBox);

form.Controls.Add(buttonOk);

form.Controls.Add(buttonCancel);

form.StartPosition = FormStartPosition.CenterScreen;

// Hiển thị form tạm để chọn chức vụ

if (form.ShowDialog() == DialogResult.OK)

{

return comboBox.SelectedItem.ToString();

}

return string.Empty;

}

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

{

// Lấy dữ liệu từ TextBox

string maChucVu = txtMaCVu.Text.Trim();

string tenChucVu = txtTenCVu.Text.Trim();

// Kiểm tra đầu vào: ít nhất một trong hai trường phải được nhập

if (string.IsNullOrEmpty(maChucVu) && string.IsNullOrEmpty(tenChucVu))

{

MessageBox.Show("Vui lòng nhập Mã chức vụ hoặc Tên chức vụ để tìm kiếm!", "Thông báo");

return;

}

// Kết nối đến cơ sở dữ liệu

using (SqlConnection con = new SqlConnection(sCon))

{

try

{

con.Open();

// Câu lệnh SQL để lấy thông tin chức vụ

string query = @"SELECT MaChucVu, TenChucVu, HeSoLuong, PhuCap

FROM ChucVu

WHERE (@MaChucVu = '' OR MaChucVu = @MaChucVu)

AND (@TenChucVu = '' OR TenChucVu LIKE '%' + @TenChucVu + '%')";

// Tạo SqlCommand

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

{

// Thêm tham số vào câu lệnh SQL

cmd.Parameters.AddWithValue("@MaChucVu", maChucVu);

cmd.Parameters.AddWithValue("@TenChucVu", tenChucVu);

// Thực thi lệnh SQL và đọc kết quả

using (SqlDataReader reader = cmd.ExecuteReader())

{

if (reader.HasRows)

{

DataTable dt = new DataTable();

dt.Load(reader);

// Hiển thị kết quả lên DataGridView

dgvChucVu.DataSource = dt;

}

else

{

MessageBox.Show("Không tìm thấy chức vụ phù hợp!", "Thông báo");

}

}

}

}

catch (SqlException ex)

{

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

}

catch (Exception ex)

{

MessageBox.Show($"Lỗi: {ex.Message}", "Lỗi");

}

}

}

private string GenerateMaChucVu()

{

SqlConnection con = new SqlConnection(sCon);

string newMaCVu = "";

try

{

con.Open();

// Lấy mã chấm công cuối cùng từ bảng, thay "MaChamCong" bằng "MaCong"

string query = "SELECT TOP 1 MaChucVu FROM ChucVu ORDER BY MaChucVu DESC";

SqlCommand cmd = new SqlCommand(query, con);

// Dùng ExecuteScalar để chỉ lấy 1 giá trị

var result = cmd.ExecuteScalar();

if (result != null)

{

string lastMaCVu = result.ToString();

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

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

int newNumber = lastNumber + 1;

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

newMaCVu = "CV" + newNumber.ToString().PadLeft(4, '0');

}

else

{

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

newMaCVu = "CV0001";

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tạo mã chấm công!");

}

finally

{

if (con.State == ConnectionState.Open)

con.Close();

}

return newMaCVu;

}

private void btResetClick(object sender, EventArgs e)

{

txtMaCVu.Text = "";

txtTenCVu.Text = "";

numHesoLuong.Value = 0;

numPhuCap.Value = 0;

txtMaCVu.Enabled = true;

LoadDataChucVu();

}

}

}