﻿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 frmNhanVien : Form

{

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

public frmNhanVien()

{

InitializeComponent();

}

private void FrmNV\_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");

}

SqlCommand command = new SqlCommand("dbo.GiaiMaDuLieuNhanVien", con);

command.CommandType = CommandType.StoredProcedure;

// Tạo SqlDataAdapter để lấy dữ liệu

SqlDataAdapter adapter = new SqlDataAdapter(command);

DataSet ds = new DataSet();

// Đổ dữ liệu vào DataTable

adapter.Fill(ds, "NhanVien");

// Hiển thị kết quả trên DataGridView

dataGridView1.DataSource = ds.Tables["NhanVien"];

con.Close();

}

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

{

// Lấy thông tin từ các TextBox (hoặc các control khác trong form)

string ho = txtHo.Text;

string ten = txtTen.Text;

string matKhau = txtMatKhau.Text;

string ngaySinh = dTpNgaySinh.Value.ToString("yyyy-MM-dd"); // Giả sử bạn dùng DateTimePicker

int gioiTinh = 0 ;

if (rbNam.Checked == true)

{

gioiTinh = 1;

}

string sdt = txtSDT.Text;

string diaChi = txtDiaChi.Text;

string email = txtEmail.Text;

decimal luongCoBan = Convert.ToDecimal(numLuongCB.Text);

DateTime ngayNhanViec = dTpNgayNhanViec.Value;

bool trangThaiLamViec = cbNghiViec.Checked;

string stk = txtSTK.Text;

string maChucVu = txtMaCV.Text;

string maPhongBan = txtMaPB.Text;

using (SqlConnection con = new SqlConnection(sCon))

{

try

{

con.Open();

// Đăng ký sự kiện InfoMessage để nhận thông báo từ SQL Server

con.InfoMessage += (s, ev) =>

{

foreach (SqlError err in ev.Errors)

{

// Hiển thị thông báo thông qua MessageBox

MessageBox.Show(err.Message, "Thông báo từ SQL Server");

}

};

// Tạo SqlCommand để gọi thủ tục AddNewEmployee

SqlCommand command = new SqlCommand("dbo.spInsertNhanVienWithEncryptionnn", con);

command.CommandType = CommandType.StoredProcedure;

// Thêm các tham số vào thủ tục

command.Parameters.AddWithValue("@LName", ho);

command.Parameters.AddWithValue("@FName", ten);

command.Parameters.AddWithValue("@MatKhau", matKhau);

command.Parameters.AddWithValue("@NgaySinh", ngaySinh);

command.Parameters.AddWithValue("@GioiTinh", gioiTinh);

command.Parameters.AddWithValue("@SDT", sdt);

command.Parameters.AddWithValue("@DiaChi", diaChi);

command.Parameters.AddWithValue("@Email", email);

command.Parameters.AddWithValue("@LuongCoBan", luongCoBan);

command.Parameters.AddWithValue("@NgayNhanViec", ngayNhanViec);

command.Parameters.AddWithValue("@TrangThaiLamViec", trangThaiLamViec);

command.Parameters.AddWithValue("@STK", stk);

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

command.Parameters.AddWithValue("@MaPhongBan", maPhongBan);

// Thêm tham số đầu ra (ret)

SqlParameter retParam = new SqlParameter("@ret", SqlDbType.Bit)

{

Direction = ParameterDirection.Output

};

command.Parameters.Add(retParam);

// Thực thi thủ tục

command.ExecuteNonQuery();

// Lấy giá trị trả về từ tham số OUTPUT

bool isSuccess = Convert.ToBoolean(retParam.Value);

// Kiểm tra kết quả thực thi

if (isSuccess)

{

MessageBox.Show("Thêm nhân viên mới thành công", "Thông báo");

ResetForm();

LoadData(); // Hàm cập nhật lại dữ liệu hiển thị

}

}

catch (SqlException ex)

{

// Hiển thị thông báo lỗi từ SQL

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

}

catch (Exception ex)

{

// Xử lý lỗi

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

}

}

}

private void ResetForm()

{

// Đặt lại các TextBox về rỗng

txtMaNV.Text = "";

txtHo.Text = "";

txtTen.Text = "";

txtMatKhau.Text = "";

txtSDT.Text = "";

txtDiaChi.Text = "";

txtEmail.Text = "";

txtSTK.Text = "";

txtMaCV.Text = "";

txtMaPB.Text = "";

// Đặt giá trị mặc định cho DateTimePicker

dTpNgaySinh.Value = DateTime.Now;

dTpNgayNhanViec.Value = DateTime.Now;

// Đặt lại trạng thái của RadioButton

rbNam.Checked = true; // Nam là mặc định

cbNghiViec.Checked = false;

// Đặt lại trạng thái của NumericUpDown (nếu có)

numLuongCB.Value = 0;

// Đặt lại trạng thái của CheckBox

cbNghiViec.Checked = false;

txtMaNV.Enabled = true;

}

private void LoadData()

{

using (SqlConnection con = new SqlConnection(sCon))

{

try

{

con.Open();

}

catch (Exception ex)

{

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

}

SqlCommand command = new SqlCommand("dbo.GiaiMaDuLieuNhanVien", con);

command.CommandType = CommandType.StoredProcedure;

// Tạo SqlDataAdapter để lấy dữ liệu

SqlDataAdapter adapter = new SqlDataAdapter(command);

DataSet ds = new DataSet();

// Đổ dữ liệu vào DataTable

adapter.Fill(ds, "NhanVien");

// Hiển thị kết quả trên DataGridView

dataGridView1.DataSource = ds.Tables["NhanVien"];

con.Close();

}

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (e.RowIndex >= 0)

{

// Lấy dòng được chọn

DataGridViewRow selectedRow = dataGridView1.Rows[e.RowIndex];

// Gán giá trị từ DataGridView vào các control

txtMaNV.Text = selectedRow.Cells["MaNhanVien"].Value?.ToString();

txtHo.Text = selectedRow.Cells["Ho"].Value?.ToString();

txtTen.Text = selectedRow.Cells["Ten"].Value?.ToString();

txtMatKhau.Text = selectedRow.Cells["MatKhau"].Value?.ToString();

dTpNgaySinh.Value = Convert.ToDateTime(selectedRow.Cells["NgaySinh"].Value);

int iGioiTinh = Convert.ToInt16(dataGridView1.Rows[e.RowIndex].Cells["GioiTinh"].Value);

if(iGioiTinh == 1)

{

rbNam.Checked = true;

}

else

{

rbNu.Checked = true;

}

txtSDT.Text = selectedRow.Cells["SDT"].Value?.ToString();

txtDiaChi.Text = selectedRow.Cells["DiaChi"].Value?.ToString();

txtEmail.Text = selectedRow.Cells["Email"].Value?.ToString();

numLuongCB.Value = Convert.ToDecimal(selectedRow.Cells["LuongCoBan"].Value);

dTpNgayNhanViec.Value = Convert.ToDateTime(selectedRow.Cells["NgayNhanViec"].Value);

cbNghiViec.Checked = Convert.ToBoolean(selectedRow.Cells["TrangThaiLamViec"].Value);

int iTrangThaiLV = Convert.ToInt16(dataGridView1.Rows[e.RowIndex].Cells["TrangThaiLamViec"].Value);

if (iTrangThaiLV == 1)

{

cbNghiViec.Checked = false;

}

else

{

cbNghiViec.Checked = true;

}

txtSTK.Text = selectedRow.Cells["STK"].Value?.ToString();

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

txtMaPB.Text = selectedRow.Cells["MaPhongBan"].Value?.ToString();

txtMaNV.Enabled = false;

}

}

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

{

// Lấy dữ liệu từ các control trên form

string maNV = txtMaNV.Text.Trim();

string ho = txtHo.Text.Trim();

string ten = txtTen.Text.Trim();

string matKhau = txtMatKhau.Text.Trim();

DateTime ngaySinh = dTpNgaySinh.Value;

int gioiTinh = rbNam.Checked ? 1 : 0;

string sdt = txtSDT.Text.Trim();

string diaChi = txtDiaChi.Text.Trim();

string email = txtEmail.Text.Trim();

decimal luongCoBan = numLuongCB.Value;

DateTime ngayNhanViec = dTpNgayNhanViec.Value;

bool trangThaiLamViec = cbNghiViec.Checked;

string stk = txtSTK.Text.Trim();

string maChucVu = txtMaCV.Text.Trim();

string maPhongBan = txtMaPB.Text.Trim();

using (SqlConnection con = new SqlConnection(sCon))

{

try

{

con.Open();

// Lệnh SQL cập nhật với mã hóa

string query = @"UPDATE NhanVien

SET Ho = @LName,

Ten = @FName,

encryptedMatKhau = ENCRYPTBYKEY(KEY\_GUID('SymKey\_test'), @MatKhau),

NgaySinh = @NgaySinh,

GioiTinh = @GioiTinh,

SDT = @SDT,

DiaChi = @DiaChi,

Email = @Email,

LuongCoBan = @LuongCoBan,

NgayNhanViec = @NgayNhanViec,

TrangThaiLamViec = @TrangThaiLamViec,

encryptedSTK = ENCRYPTBYKEY(KEY\_GUID('SymKey\_test'), @STK),

MaChucVu = @MaChucVu,

MaPhongBan = @MaPhongBan

WHERE MaNhanVien = @MaNhanVien";

// Tạo SqlCommand

SqlCommand command = new SqlCommand(query, con);

// Thêm tham số với kiểu dữ liệu cụ thể

command.Parameters.Add("@MaNhanVien", SqlDbType.Char, 6).Value = maNV;

command.Parameters.Add("@LName", SqlDbType.NVarChar, 50).Value = ho;

command.Parameters.Add("@FName", SqlDbType.NVarChar, 50).Value = ten;

command.Parameters.Add("@MatKhau", SqlDbType.VarChar, 50).Value = matKhau;

command.Parameters.Add("@NgaySinh", SqlDbType.Date).Value = ngaySinh;

command.Parameters.Add("@GioiTinh", SqlDbType.Bit).Value = gioiTinh;

command.Parameters.Add("@SDT", SqlDbType.Char, 10).Value = sdt;

command.Parameters.Add("@DiaChi", SqlDbType.NVarChar, 50).Value = diaChi;

command.Parameters.Add("@Email", SqlDbType.VarChar, 50).Value = email;

command.Parameters.Add("@LuongCoBan", SqlDbType.Decimal).Value = luongCoBan;

command.Parameters.Add("@NgayNhanViec", SqlDbType.Date).Value = ngayNhanViec;

command.Parameters.Add("@TrangThaiLamViec", SqlDbType.Bit).Value = trangThaiLamViec;

command.Parameters.Add("@STK", SqlDbType.VarChar, 50).Value = stk;

command.Parameters.Add("@MaChucVu", SqlDbType.Char, 6).Value = maChucVu;

command.Parameters.Add("@MaPhongBan", SqlDbType.Char, 6).Value = maPhongBan;

// Thực thi lệnh cập nhật

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

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

// Cập nhật lại dữ liệu trong DataGridView

LoadData();

ResetForm();

}

else

{

MessageBox.Show("Không tìm thấy nhân viên để cập nhật.", "Thông báo");

}

}

catch (SqlException ex)

{

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

}

catch (Exception ex)

{

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

}

}

}

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

{

// Lấy mã nhân viên từ TextBox

string maNV = txtMaNV.Text.Trim();

if (string.IsNullOrEmpty(maNV))

{

MessageBox.Show("Vui lòng nhập mã nhân viên để xóa.", "Thông báo");

return;

}

// Hiển thị hộp thoại xác nhận

DialogResult result = MessageBox.Show(

$"Bạn có chắc chắn muốn xóa nhân viên với Mã nhân viên: {maNV}?",

"Xác nhận xóa",

MessageBoxButtons.YesNo,

MessageBoxIcon.Warning

);

if (result == DialogResult.Yes)

{

using (SqlConnection con = new SqlConnection(sCon))

{

try

{

con.Open();

// Tạo SqlCommand để xóa nhân viên

string query = @"DELETE FROM NhanVien WHERE MaNhanVien = @MaNhanVien";

SqlCommand command = new SqlCommand(query, con);

// Thêm tham số MaNhanVien vào câu lệnh SQL

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

// Thực thi câu lệnh xóa

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Xóa nhân viên thành công!", "Thông báo");

// Cập nhật lại dữ liệu trong DataGridView (nếu cần)

LoadData();

ResetForm();

}

else

{

MessageBox.Show("Không tìm thấy nhân viên để xóa.", "Thông báo");

}

}

catch (SqlException ex)

{

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

}

catch (Exception ex)

{

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

}

}

}

}

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

{

// Lấy giá trị từ các TextBox tìm kiếm

string maNV = txtMaNV.Text.Trim();

string ho = txtHo.Text.Trim();

string ten = txtTen.Text.Trim();

string email = txtEmail.Text.Trim();

string sdt = txtSDT.Text.Trim();

using (SqlConnection con = new SqlConnection(sCon))

{

try

{

con.Open();

// Tạo câu lệnh SQL với điều kiện LIKE cho tất cả các trường

string query = @"

SELECT MaNhanVien, Ho, Ten, SDT, Email, LuongCoBan, NgaySinh, NgayNhanViec

FROM NhanVien

WHERE (@MaNhanVien IS NULL OR MaNhanVien LIKE @MaNhanVien)

AND (@Ho IS NULL OR Ho LIKE @Ho)

AND (@Ten IS NULL OR Ten LIKE @Ten)

AND (@Email IS NULL OR Email LIKE @Email)

AND (@SDT IS NULL OR SDT LIKE @SDT)";

// Tạo SqlCommand với câu lệnh SQL

SqlCommand command = new SqlCommand(query, con);

// Thêm các tham số vào câu lệnh SQL

command.Parameters.AddWithValue("@MaNhanVien", string.IsNullOrEmpty(maNV) ? (object)DBNull.Value : "%" + maNV + "%");

command.Parameters.AddWithValue("@Ho", string.IsNullOrEmpty(ho) ? (object)DBNull.Value : "%" + ho + "%");

command.Parameters.AddWithValue("@Ten", string.IsNullOrEmpty(ten) ? (object)DBNull.Value : "%" + ten + "%");

command.Parameters.AddWithValue("@Email", string.IsNullOrEmpty(email) ? (object)DBNull.Value : "%" + email + "%");

command.Parameters.AddWithValue("@SDT", string.IsNullOrEmpty(sdt) ? (object)DBNull.Value : "%" + sdt + "%");

// Tạo SqlDataAdapter để lấy dữ liệu

SqlDataAdapter adapter = new SqlDataAdapter(command);

// Tạo DataTable để chứa dữ liệu

DataTable dt = new DataTable();

// Đổ dữ liệu vào DataTable

adapter.Fill(dt);

// Kiểm tra nếu có kết quả

if (dt.Rows.Count > 0)

{

// Gán dữ liệu vào DataGridView

dataGridView1.DataSource = dt;

}

else

{

MessageBox.Show("Không tìm thấy nhân viên nào khớp với từ khóa tìm kiếm!", "Thông báo");

}

}

catch (SqlException ex)

{

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

}

catch (Exception ex)

{

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

}

}

}

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

{

txtMaNV.Text = "";

txtHo.Text = "";

txtTen.Text = "";

txtMatKhau.Text = "";

txtSDT.Text = "";

txtDiaChi.Text = "";

txtEmail.Text = "";

txtSTK.Text = "";

txtMaCV.Text = "";

txtMaPB.Text = "";

// Đặt giá trị mặc định cho DateTimePicker

dTpNgaySinh.Value = DateTime.Now;

dTpNgayNhanViec.Value = DateTime.Now;

// Đặt lại trạng thái của RadioButton

rbNam.Checked = true; // Nam là mặc định

cbNghiViec.Checked = false;

// Đặt lại trạng thái của NumericUpDown (nếu có)

numLuongCB.Value = 0;

// Đặt lại trạng thái của CheckBox

cbNghiViec.Checked = false;

txtMaNV.Enabled = true;

LoadData();

}

}

}