﻿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 BangKyNang

{

public partial class KyNang : 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()

{

command = connection.CreateCommand();

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

adapter.SelectCommand = command;

table.Clear();

adapter.Fill(table);

dgv.DataSource = table;

}

public KyNang()

{

InitializeComponent();

}

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

{

connection = new SqlConnection(str);

connection.Open();

LoadData();

}

private void DgvCellContentClick(object sender, DataGridViewCellEventArgs e)

{

int i;

i = dgv.CurrentRow.Index;

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

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

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

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

tb\_makynang.Enabled = false;

}

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

{

connection = new SqlConnection(str);

connection.Open();

LoadData();

}

private void BtThemClick(object sender, EventArgs e)

{

command = connection.CreateCommand();

command.CommandText = "SELECT TOP 1 MaKyNang FROM KYNANG ORDER BY MaKyNang DESC";

object result = command.ExecuteScalar();

string newMaKyNang;

if (result != null)

{

string lastMaKyNang = result.ToString();

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

newMaKyNang = "K" + number.ToString("D5");

}

else

{

newMaKyNang = "K00001";

}

// 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 kỹ năng mới

command = connection.CreateCommand();

command.CommandText = "INSERT INTO KYNANG VALUES (@MaKyNang, @TenKyNang, @TrinhDo, @MaNhanVien)";

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

command.Parameters.AddWithValue("@TenKyNang", tb\_tenkynang.Text.Trim());

command.Parameters.AddWithValue("@TrinhDo", tb\_trinhdo.Text.Trim());

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 BtCapNhatClick(object sender, EventArgs e)

{

string maKyNang = tb\_makynang.Text.Trim(); // TextBox nhập Mã kỹ năng

string tenKyNang = tb\_tenkynang.Text.Trim(); // TextBox nhập Tên kỹ năng

string trinhDo = tb\_trinhdo.Text.Trim(); // ComboBox hoặc TextBox nhập Trình độ

if (string.IsNullOrEmpty(maKyNang))

{

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

return;

}

// Tạo lệnh SQL động

StringBuilder query = new StringBuilder("UPDATE KYNANG SET ");

List<SqlParameter> parameters = new List<SqlParameter>();

if (!string.IsNullOrEmpty(tenKyNang))

{

query.Append("TenKyNang = @TenKyNang, ");

parameters.Add(new SqlParameter("@TenKyNang", tenKyNang));

}

if (!string.IsNullOrEmpty(trinhDo))

{

query.Append("TrinhDo = @TrinhDo, ");

parameters.Add(new SqlParameter("@TrinhDo", trinhDo));

}

// Loại bỏ dấu phẩy cuối cùng trong câu lệnh SQL

query.Length -= 2;

query.Append(" WHERE MaKyNang = @MaKyNang");

parameters.Add(new SqlParameter("@MaKyNang", maKyNang));

// Tạo và thực thi lệnh SQL

command = connection.CreateCommand();

command.CommandText = query.ToString();

command.Parameters.AddRange(parameters.ToArray());

try

{

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

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

LoadData(); // Tải lại dữ liệu lên DataGridView

ResetForm();

}

else

{

MessageBox.Show("Không tìm thấy Mã kỹ năng 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 BtXoaClick(object sender, EventArgs e)

{

string maKyNang = tb\_makynang.Text.Trim(); // TextBox nhập Mã kỹ năng

string maNhanVien = tb\_manhanvien.Text.Trim(); // TextBox nhập Mã nhân viên

// Kiểm tra dữ liệu đầu vào

if (string.IsNullOrEmpty(maKyNang))

{

MessageBox.Show("Vui lòng nhập Mã kỹ năng 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 kỹ năng và Mã nhân viên

command.CommandText = "DELETE FROM KYNANG WHERE MaKyNang = @MaKyNang AND MaNhanVien = @MaNhanVien";

command.Parameters.AddWithValue("@MaKyNang", maKyNang);

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

}

else

{

// Xóa chỉ theo Mã kỹ năng

command.CommandText = "DELETE FROM KYNANG WHERE MaKyNang = @MaKyNang";

command.Parameters.AddWithValue("@MaKyNang", maKyNang);

}

// 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)

{

try

{

// 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(); // Tải lại dữ liệu lên DataGridView

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 khi xóa: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void BtTimKiemClick(object sender, EventArgs e)

{

// Lấy giá trị từ các TextBox tìm kiếm

string maKyNang = tb\_makynang.Text.Trim();

string tenKyNang = tb\_tenkynang.Text.Trim();

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

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

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

{

try

{

con.Open();

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

string query = @"

SELECT MaKyNang, TenKyNang, TrinhDo, MaNhanVien

FROM KyNang

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

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

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

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

// Tạo SqlCommand với câu lệnh SQL

SqlCommand command = new SqlCommand(query, con);

// Thêm các tham số vào câu lệnh SQL

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

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

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

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

// 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

dgv.DataSource = dt;

}

else

{

MessageBox.Show("Không tìm thấy kỹ năng 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 label5\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

private void ResetForm()

{

// Xóa nội dung các TextBox

tb\_makynang.Text = string.Empty;

tb\_tenkynang.Text = string.Empty;

tb\_trinhdo.Text = string.Empty;

tb\_manhanvien.Text = string.Empty;

// Đặt lại tiêu điểm về TextBox đầu tiên

tb\_makynang.Focus();

tb\_makynang.Enabled = true;

}

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

{

LoadData();

ResetForm();

}

}

}