﻿using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Create\_NghiPhep

{

public partial class frm\_NghiPhep : Form

{

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

public frm\_NghiPhep()

{

InitializeComponent();

}

private void label5\_Click(object sender, EventArgs e)

{

}

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

{

}

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

{

}

private void GioVao\_ValueChanged(object sender, EventArgs e)

{

}

private void GioRa\_ValueChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

// Lấy mã nhân viên từ TextBox

string sMaNV = txtMaNV.Text.Trim();

// Kiểm tra mã nhân viên không được để trống

if (string.IsNullOrEmpty(sMaNV))

{

MessageBox.Show("Vui lòng nhập mã nhân viên!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Chuẩn bị câu lệnh SQL

string sQuery = "SELECT MaNghi, MaNhanVien, NgayNghi, LoaiNghi FROM NghiPhep WHERE MaNhanVien = @MaNhanVien";

// Tạo SqlCommand và gán tham số

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

{

cmd.Parameters.AddWithValue("@MaNhanVien", sMaNV);

// Sử dụng SqlDataAdapter để lấy dữ liệu

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

// Hiển thị dữ liệu trong DataGridView

if (dt.Rows.Count > 0)

{

ThongtinNghiPhep.DataSource = dt;

ResetForm();

}

else

{

MessageBox.Show("Không tìm thấy dữ liệu cho mã nhân viên này.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

ThongtinNghiPhep.DataSource = null; // Xóa dữ liệu cũ trong bảng

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi xem ngày nghỉ phép ");

}

con.Close();

}

private void Create\_NghiPhep\_FormClosed(object sender, FormClosedEventArgs e)

{

MessageBox.Show("Hẹn gặp lại lần sau!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void Create\_NghiPhep\_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");

}

string sQuery = "select \* from NghiPhep";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "NghiPhep");

ThongtinNghiPhep.DataSource = ds.Tables["NghiPhep"];

con.Close();

}

private void LoadData()

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

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

}

string sQuery = "select \* from NghiPhep";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "NghiPhep");

ThongtinNghiPhep.DataSource = ds.Tables["NghiPhep"];

con.Close();

}

private string GenerateMaNghi()

{

SqlConnection con = new SqlConnection(sCon);

string newMaNghi = "";

try

{

con.Open();

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

string query = "SELECT TOP 1 MaNghi FROM NghiPhep ORDER BY MaNghi DESC";

SqlCommand cmd = new SqlCommand(query, con);

// Dùng ExecuteScalar để chỉ lấy 1 giá trị

var result = cmd.ExecuteScalar();

if (result != null)

{

string lastMaNghi = result.ToString();

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

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

int newNumber = lastNumber + 1;

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

newMaNghi = "MN" + newNumber.ToString().PadLeft(4, '0');

}

else

{

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

newMaNghi = "MN0001";

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tạo mã nghỉ ");

}

finally

{

if (con.State == ConnectionState.Open)

con.Close();

}

return newMaNghi;

}

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

{

//b1

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

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

}

//b2

// Kiểm tra Mã nhân viên

if (string.IsNullOrEmpty(txtMaNV.Text))

{

MessageBox.Show("Mã nhân viên không được để trống.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Dừng lại nếu Mã nhân viên không hợp lệ

}

// Kiểm tra Ngày nghỉ

if (NgayNghi.Value == null || NgayNghi.Value == DateTime.MinValue)

{

MessageBox.Show("Ngày nghỉ không được để trống.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Dừng lại nếu Ngày nghỉ không hợp lệ

}

// Kiểm tra Loại nghỉ (Có phép hoặc Không phép)

if (CoPhep.Checked && KhongPhep.Checked)

{

MessageBox.Show("Vui lòng chọn loại nghỉ (Có phép hoặc Không phép).", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Dừng lại nếu Loại nghỉ không được chọn

}

//Gan du lieu vao bien

string sMaNV = txtMaNV.Text;

string sNgayNghi = NgayNghi.Value.ToString("yyyy-MM-dd");

int iLoaiNghi = 0;

if (CoPhep.Checked == true)

{

iLoaiNghi = 1;

}

else

{

iLoaiNghi = 0;

}

string sMaNghi = GenerateMaNghi();

string sQuery = "INSERT INTO NghiPhep (MaNghi, NgayNghi, LoaiNghi, MaNhanVien) " +

"VALUES (@MaNghi, @NgayNghi, @LoaiNghi, @MaNhanVien)";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaNghi", sMaNghi);

cmd.Parameters.AddWithValue("@MaNhanVien", sMaNV);

cmd.Parameters.AddWithValue("@NgayNghi", sNgayNghi);

cmd.Parameters.AddWithValue("@LoaiNghi", iLoaiNghi);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm mới thành công!");

ResetForm();

LoadData();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình thêm mới!");

}

con.Close();

}

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

{

txtMaNghi.Text = ThongtinNghiPhep.Rows[e.RowIndex].Cells["MaNghi"].Value.ToString();

txtMaNV.Text = ThongtinNghiPhep.Rows[e.RowIndex].Cells["MaNhanVien"].Value.ToString();

NgayNghi.Value = Convert.ToDateTime(ThongtinNghiPhep.Rows[e.RowIndex].Cells["NgayNghi"].Value);

int iLoaiNghi = Convert.ToInt16(ThongtinNghiPhep.Rows[e.RowIndex].Cells["LoaiNghi"].Value);

if (iLoaiNghi == 1)

{

CoPhep.Checked = true;

}

else

{

KhongPhep.Checked = true;

}

txtMaNghi.Enabled = false;

}

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

{

// Kiểm tra kết nối và xử lý ngoại lệ

using (SqlConnection con = new SqlConnection(sCon))

{

try

{

con.Open();

// Lấy thông tin từ các trường nhập liệu

string sMaNV = txtMaNV.Text.Trim();

string sNgayNghi = NgayNghi.Value.ToString("yyyy-MM-dd");

int iLoaiNghi = CoPhep.Checked ? 1 : 0;

// Kiểm tra hàng được chọn trong DataGridView

if (ThongtinNghiPhep.SelectedCells.Count > 0)

{

int rowIndex = ThongtinNghiPhep.SelectedCells[0].RowIndex;

string MaNghi = ThongtinNghiPhep.Rows[rowIndex].Cells["MaNghi"].Value?.ToString();

if (string.IsNullOrEmpty(MaNghi))

{

MessageBox.Show("Không tìm thấy mã nghỉ để cập nhật.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Câu lệnh SQL

string sQuery = "UPDATE NghiPhep SET MaNhanVien = @MaNhanVien, NgayNghi = @NgayNghi, LoaiNghi = @LoaiNghi " +

"WHERE MaNghi = @MaNghi";

// Tạo SqlCommand và gán kết nối

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

{

// Thêm tham số

cmd.Parameters.AddWithValue("@MaNhanVien", sMaNV);

cmd.Parameters.AddWithValue("@NgayNghi", sNgayNghi);

cmd.Parameters.AddWithValue("@LoaiNghi", iLoaiNghi);

cmd.Parameters.AddWithValue("@MaNghi", MaNghi);

// Thực thi câu lệnh

int result = cmd.ExecuteNonQuery();

if (result > 0)

{

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

ResetForm();

LoadData();

}

else

{

MessageBox.Show("Không có dữ liệu nào được cập nhật.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

else

{

MessageBox.Show("Vui lòng chọn một hàng để cập nhật.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show($"Xảy ra lỗi trong quá trình cập nhật", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

con.Close();

}

}

}

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

{

// Kiểm tra mã nghỉ

string sMaNghi = txtMaNghi.Text.Trim(); // Lấy mã nghỉ từ TextBox

// Kiểm tra nếu mã nghỉ trống

if (string.IsNullOrEmpty(sMaNghi))

{

MessageBox.Show("Vui lòng nhập mã nghỉ phép để xóa.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Xác nhận trước khi xóa

DialogResult confirmResult = MessageBox.Show($"Bạn có chắc muốn xóa ngày nghỉ phép với mã nghỉ {sMaNghi} không?",

"Xác nhận xóa",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (confirmResult == DialogResult.No)

{

return;

}

// Câu lệnh SQL để kiểm tra xem mã nghỉ có tồn tại không

string checkQuery = "SELECT COUNT(\*) FROM NghiPhep WHERE MaNghi = @MaNghi";

string sQuery = "DELETE FROM NghiPhep WHERE MaNghi = @MaNghi";

// Sử dụng khối 'using' để tự động quản lý tài nguyên kết nối

using (SqlConnection con = new SqlConnection(sCon))

{

try

{

// Mở kết nối

con.Open();

// Kiểm tra xem mã nghỉ có tồn tại trong bảng không

using (SqlCommand checkCmd = new SqlCommand(checkQuery, con))

{

checkCmd.Parameters.AddWithValue("@MaNghi", sMaNghi);

int count = (int)checkCmd.ExecuteScalar();

if (count == 0)

{

MessageBox.Show("Mã nghỉ phép không tồn tại trong hệ thống.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

}

// Thực hiện xóa dữ liệu

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

{

cmd.Parameters.AddWithValue("@MaNghi", sMaNghi);

// Thực thi câu lệnh xóa

int rowsAffected = cmd.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Xóa thành công!");

LoadData(); // Tải lại dữ liệu sau khi xóa

ResetForm(); // Reset lại form sau khi xóa

}

else

{

MessageBox.Show("Không thể xóa ngày nghỉ phép. Vui lòng thử lại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

catch (Exception ex)

{

// Hiển thị thông báo lỗi chi tiết

MessageBox.Show($"Xảy ra lỗi trong quá trình xóa!\n{ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void ResetForm()

{

txtMaNghi.Text = "";

txtMaNV.Text = "";

NgayNghi.Value = DateTime.Now;

CoPhep.Checked = true;

}

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

{

txtMaNghi.Text = "";

txtMaNV.Text = "";

NgayNghi.Value = DateTime.Now;

CoPhep.Checked = true;

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

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

}

string sQuery = "select \* from NghiPhep";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "NghiPhep");

ThongtinNghiPhep.DataSource = ds.Tables["NghiPhep"];

con.Close();

}

}

}