Trần Thị Ngọc Huyền

1150080138

11CNPM2

LAB 3:

WINDOWS FORMS APPLICATION NÂNG CAO

Thực hành 1: Thiết kế và lập trình ứng dụng Winform theo mô tả sau:

Tại một quán ăn nhanh, người quản lý sẽ cung cấp cho mỗi nhân vật phục vụ bàn

một máy PDA để order món ăn theo phương thức số hóa. Bởi vậy, khi order xong

các món theo bàn, nhân viên phục vụ bàn sẽ không phải xuống trực tiếp với nhà

bếp mà chuyển thông tin các món khách hàng yêu cầu thông qua hệ thống mạng

không dây. Dưới đây là hình ảnh minh hoạ màn hình làm việc máy PDA đó:

Nguyên tắc sử dụng:

− Nhân vật phục vụ bàn sẽ chọn tên bàn trong danh sách sổ xuống của control

ComboBox. Sau đó, tùy theo nhu cầu ăn uống mà khách hàng gọi món. Đồ

ăn và thức uống được thể hiện bằng cách button. Mỗi một lần nhấn button

thì số lượng món ăn sẽ tăng lên 1. Ví dụ, nhấn 2 lần món Cơm ràng Dương

Châu, thì số lượng món ăn này là 2 và danh mục các món cùng số lượng

đặtsẽ được hiển thị bằng control DataGridView hoặc ListView ở bên dưới.

− Kết thúc quá trình gọi món, nhân viên phục vụ sẽ chọn chức năng Order

để chuyển thông tin món ăn và số lượng xuống nhà bếp. Ở đây, ta sẽ thay

thế chức năng chuyển thông tin này bằng chức năng ghi tệp văn bản.

B1: tạo project

A screenshot of a computer

AI-generated content may be incorrect.

B2 chọn Winform App

A screenshot of a computer

AI-generated content may be incorrect.

B3 đặt tên Project

A screenshot of a computer

AI-generated content may be incorrect.

**Code Propram.cs**

using System;

using System.Windows.Forms;

namespace lab3\_1\_1150080138

{

internal static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new FormQuanAn()); // chạy form chính

}

}

}

**Code Form1**

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Globalization;

using System.Linq;

using System.Windows.Forms;

namespace lab3\_1\_1150080138

{

public class FormQuanAn : Form

{

// format VNĐ

private readonly CultureInfo viVN = new CultureInfo("vi-VN");

// UI

private readonly Panel headerPanel;

private readonly PictureBox picLogo;

private readonly Panel bannerPanel;

private readonly Label lblBanner;

private readonly GroupBox grpMenu;

private readonly Panel toolPanel;

private readonly Button btnDelete;

private readonly ComboBox cboBan;

private readonly Label lblBan;

private readonly Button btnOrder;

private readonly ListView lvHoaDon;

// dữ liệu món & giá

private readonly Dictionary<string, int> menu = new Dictionary<string, int>

{

// món ăn

{"Cơm chiên trứng", 35000},

{"Bánh mỳ ốp la", 30000},

{"Ốc rang muối", 80000},

{"Khoai tây chiên", 25000},

{"Mỳ xào hải sản", 65000},

{"Cá viên chiên", 25000},

{"Burger bò nướng", 55000},

{"Đùi gà rán", 45000},

{"Bún bò Huế", 45000},

// đồ uống

{"Coca", 15000},

{"Pepsi", 15000},

{"7 up", 15000},

{"Lipton", 18000},

{"Cam", 18000},

{"Cafe", 20000},

};

public FormQuanAn()

{

Text = "Form1";

StartPosition = FormStartPosition.CenterScreen;

ClientSize = new Size(760, 560);

MinimumSize = new Size(760, 560);

Font = new Font("Segoe UI", 9F);

DoubleBuffered = true;

// ===== HEADER (logo + banner xanh) =====

headerPanel = new Panel { Left = 10, Top = 10, Width = ClientSize.Width - 20, Height = 90, Anchor = AnchorStyles.Left | AnchorStyles.Right | AnchorStyles.Top };

picLogo = new PictureBox { Left = 0, Top = 0, Width = 110, Height = 90, BackColor = Color.White, SizeMode = PictureBoxSizeMode.CenterImage };

var lblEmoji = new Label { Text = "🍔", Font = new Font("Segoe UI Emoji", 40), Dock = DockStyle.Fill, TextAlign = ContentAlignment.MiddleCenter };

picLogo.Controls.Add(lblEmoji);

bannerPanel = new Panel

{

Left = picLogo.Right + 8,

Top = 0,

Width = headerPanel.Width - picLogo.Width - 8,

Height = 90,

BackColor = Color.FromArgb(0, 153, 51),

Anchor = AnchorStyles.Left | AnchorStyles.Right | AnchorStyles.Top

};

lblBanner = new Label

{

Text = "Quán ăn nhanh Hùng Thịnh",

ForeColor = Color.White,

Font = new Font("Segoe UI Semibold", 20, FontStyle.Bold),

Dock = DockStyle.Fill,

TextAlign = ContentAlignment.MiddleLeft,

Padding = new Padding(16, 0, 0, 0)

};

bannerPanel.Controls.Add(lblBanner);

headerPanel.Controls.Add(picLogo);

headerPanel.Controls.Add(bannerPanel);

// ===== GROUP MENU (lưới nút) =====

grpMenu = new GroupBox

{

Text = "Danh sách món ăn:",

Left = 10,

Top = headerPanel.Bottom + 8,

Width = ClientSize.Width - 20,

Height = 180,

Anchor = AnchorStyles.Left | AnchorStyles.Right | AnchorStyles.Top

};

CreateMenuButtons();

// ===== TOOL BAR (Xóa / Chọn bàn / Order) =====

toolPanel = new Panel

{

Left = 10,

Top = grpMenu.Bottom + 6,

Width = ClientSize.Width - 20,

Height = 40,

Anchor = AnchorStyles.Left | AnchorStyles.Right | AnchorStyles.Top

};

btnDelete = new Button { Text = "Xóa", Left = 0, Top = 6, Width = 80, Height = 28 };

btnDelete.Click += (s, e) => RemoveSelectedLines();

lblBan = new Label { Text = "Chọn bàn:", Left = 100, Top = 11, AutoSize = true };

cboBan = new ComboBox { Left = lblBan.Right + 6, Top = 7, Width = 120, DropDownStyle = ComboBoxStyle.DropDownList };

cboBan.Items.AddRange(Enumerable.Range(1, 20).Select(i => (object)$"Bàn {i}").ToArray());

if (cboBan.Items.Count > 0) cboBan.SelectedIndex = 0;

btnOrder = new Button { Text = "Order", Width = 90, Height = 28, Top = 6, Left = toolPanel.Width - 90, Anchor = AnchorStyles.Top | AnchorStyles.Right };

btnOrder.Click += BtnOrder\_Click;

toolPanel.Controls.Add(btnDelete);

toolPanel.Controls.Add(lblBan);

toolPanel.Controls.Add(cboBan);

toolPanel.Controls.Add(btnOrder);

// ===== LISTVIEW HÓA ĐƠN =====

lvHoaDon = new ListView

{

Left = 10,

Top = toolPanel.Bottom + 6,

Width = ClientSize.Width - 20,

Height = ClientSize.Height - (toolPanel.Bottom + 16),

Anchor = AnchorStyles.Left | AnchorStyles.Right | AnchorStyles.Top | AnchorStyles.Bottom,

View = View.Details,

FullRowSelect = true,

GridLines = true

};

lvHoaDon.Columns.Add("Món", 280);

lvHoaDon.Columns.Add("SL", 60, HorizontalAlignment.Center);

lvHoaDon.Columns.Add("Đơn giá", 120, HorizontalAlignment.Right);

lvHoaDon.Columns.Add("Thành tiền", 140, HorizontalAlignment.Right);

Controls.Add(headerPanel);

Controls.Add(grpMenu);

Controls.Add(toolPanel);

Controls.Add(lvHoaDon);

Resize += (s, e) =>

{

bannerPanel.Width = headerPanel.Width - picLogo.Width - 8;

btnOrder.Left = toolPanel.Width - btnOrder.Width;

lvHoaDon.Height = ClientSize.Height - (toolPanel.Bottom + 16);

};

}

private void CreateMenuButtons()

{

// tạo các nút dạng grid 4 cột x 4 hàng (gần giống ảnh mẫu)

int cols = 4;

int btnW = 150, btnH = 32, gapX = 12, gapY = 12;

int startX = 12, startY = 28;

var names = new[]

{

"Cơm chiên trứng","Bánh mỳ ốp la","Coca","Lipton",

"Ốc rang muối","Khoai tây chiên","7 up","Cam",

"Mỳ xào hải sản","Cá viên chiên","Pepsi","Cafe",

"Burger bò nướng","Đùi gà rán","Bún bò Huế"

};

for (int i = 0; i < names.Length; i++)

{

int r = i / cols, c = i % cols;

var b = new Button

{

Text = names[i],

Width = btnW,

Height = btnH,

Left = startX + c \* (btnW + gapX),

Top = startY + r \* (btnH + gapY)

};

b.Click += (s, e) => AddItem(((Button)s).Text);

grpMenu.Controls.Add(b);

}

}

private void AddItem(string name)

{

int price = menu.ContainsKey(name) ? menu[name] : 0;

// nếu món đã có → tăng SL

foreach (ListViewItem it in lvHoaDon.Items)

{

if (string.Equals(it.Text, name, StringComparison.OrdinalIgnoreCase))

{

int sl = int.Parse(it.SubItems[1].Text);

sl++;

it.SubItems[1].Text = sl.ToString();

it.SubItems[3].Text = (sl \* price).ToString("c0", viVN);

return;

}

}

var item = new ListViewItem(name);

item.SubItems.Add("1");

item.SubItems.Add(price.ToString("c0", viVN));

item.SubItems.Add(price.ToString("c0", viVN));

lvHoaDon.Items.Add(item);

}

private void RemoveSelectedLines()

{

if (lvHoaDon.SelectedItems.Count == 0) return;

while (lvHoaDon.SelectedItems.Count > 0)

lvHoaDon.Items.Remove(lvHoaDon.SelectedItems[0]);

}

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

{

if (lvHoaDon.Items.Count == 0)

{

MessageBox.Show("Chưa chọn món nào!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

int tong = 0;

foreach (ListViewItem it in lvHoaDon.Items)

{

string name = it.Text;

int sl = int.Parse(it.SubItems[1].Text);

int price = menu.ContainsKey(name) ? menu[name] : 0;

tong += sl \* price;

}

string ban = cboBan.SelectedItem?.ToString() ?? "Bàn ?";

MessageBox.Show($"{ban}\nTổng tiền: {tong.ToString("c0", viVN)}",

"Order thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);

// muốn clear sau khi order thì bỏ comment dòng dưới:

// lvHoaDon.Items.Clear();

}

}

}

A screenshot of a computer

AI-generated content may be incorrect.

**Thực hành 2: Tạo mới 1 project có tên là “ThucHanh2” với form chính được**

**minh họa như sau:**

B1: tạo project

A screenshot of a computer

AI-generated content may be incorrect.

B2 chọn Winform App

A screenshot of a computer

AI-generated content may be incorrect.

B3 đặt tên Project

A screenshot of a computer

AI-generated content may be incorrect.

**Code Program**

using System;

using System.Windows.Forms;

namespace lab3\_1\_1150080138

{

internal static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new FormDanhMucSinhVien());

}

}

}

**Code Form1**

using System;

using System.Drawing;

using System.Windows.Forms;

namespace lab3\_1\_1150080138

{

public class FormDanhMucSinhVien : Form

{

// Tiêu đề

private readonly Label lblHeader;

// “Thông tin sinh viên”

private readonly GroupBox grpInfo;

private readonly Label lblHoTen, lblLop, lblNgaySinh, lblDiaChi;

private readonly TextBox txtHoTen, txtLop, txtDiaChi;

private readonly DateTimePicker dtNgaySinh;

// Nút lệnh

private readonly GroupBox grpAction;

private readonly Button btnThem, btnSua, btnXoa, btnThoat;

// ListView

private readonly GroupBox grpList;

private readonly ListView lvSV;

public FormDanhMucSinhVien()

{

Text = "Danh sách sinh viên";

StartPosition = FormStartPosition.CenterScreen;

ClientSize = new Size(800, 520);

FormBorderStyle = FormBorderStyle.FixedSingle;

MaximizeBox = false;

// ===== Tiêu đề =====

lblHeader = new Label

{

Text = "DANH MỤC SINH VIÊN",

Font = new Font("Segoe UI", 20, FontStyle.Bold),

ForeColor = Color.RoyalBlue,

AutoSize = false,

TextAlign = ContentAlignment.MiddleCenter,

Left = 10, Top = 10, Width = ClientSize.Width - 20, Height = 48,

Anchor = AnchorStyles.Left | AnchorStyles.Right | AnchorStyles.Top

};

// ===== Group thông tin sinh viên =====

grpInfo = new GroupBox

{

Text = "Thông tin sinh viên:",

Left = 10, Top = 70, Width = ClientSize.Width - 20, Height = 120

};

lblHoTen = new Label { Text = "Họ tên:", Left = 20, Top = 30, AutoSize = true };

txtHoTen = new TextBox { Left = 80, Top = 26, Width = 260 };

lblLop = new Label { Text = "Lớp:", Left = 380, Top = 30, AutoSize = true };

txtLop = new TextBox { Left = 420, Top = 26, Width = 150 };

lblNgaySinh = new Label { Text = "Ngày sinh:", Left = 20, Top = 70, AutoSize = true };

dtNgaySinh = new DateTimePicker { Left = 90, Top = 66, Width = 150, Format = DateTimePickerFormat.Short };

lblDiaChi = new Label { Text = "Địa chỉ:", Left = 280, Top = 70, AutoSize = true };

txtDiaChi = new TextBox { Left = 330, Top = 66, Width = 380 };

grpInfo.Controls.AddRange(new Control[] { lblHoTen, txtHoTen, lblLop, txtLop, lblNgaySinh, dtNgaySinh, lblDiaChi, txtDiaChi });

// ===== Group chức năng =====

grpAction = new GroupBox

{

Text = "Chức năng:",

Left = 10, Top = grpInfo.Bottom + 8, Width = ClientSize.Width - 20, Height = 70

};

btnThem = new Button { Text = "Thêm", Left = 120, Top = 25, Width = 100, Height = 30 };

btnSua = new Button { Text = "Sửa", Left = 240, Top = 25, Width = 100, Height = 30 };

btnXoa = new Button { Text = "Xóa", Left = 360, Top = 25, Width = 100, Height = 30 };

btnThoat= new Button { Text = "Thoát",Left = 480, Top = 25, Width = 100, Height = 30 };

btnThem.Click += BtnThem\_Click;

btnSua.Click += BtnSua\_Click;

btnXoa.Click += BtnXoa\_Click;

btnThoat.Click+= (s, e) => Close();

grpAction.Controls.AddRange(new Control[] { btnThem, btnSua, btnXoa, btnThoat });

// ===== Group danh sách =====

grpList = new GroupBox

{

Text = "Thông tin chung sinh viên:",

Left = 10, Top = grpAction.Bottom + 8, Width = ClientSize.Width - 20, Height = ClientSize.Height - (grpAction.Bottom + 18),

Anchor = AnchorStyles.Left | AnchorStyles.Right | AnchorStyles.Top | AnchorStyles.Bottom

};

lvSV = new ListView

{

Left = 12, Top = 22, Width = grpList.Width - 24, Height = grpList.Height - 34,

Anchor = AnchorStyles.Left | AnchorStyles.Right | AnchorStyles.Top | AnchorStyles.Bottom,

View = View.Details, FullRowSelect = true, GridLines = true

};

lvSV.Columns.Add("Họ tên", 220);

lvSV.Columns.Add("Ngày sinh", 110);

lvSV.Columns.Add("Lớp", 120);

lvSV.Columns.Add("Địa chỉ", 280);

// chọn 1 dòng → đổ dữ liệu lên input

lvSV.SelectedIndexChanged += (s, e) =>

{

if (lvSV.SelectedItems.Count == 0) return;

var it = lvSV.SelectedItems[0];

txtHoTen.Text = it.SubItems[0].Text;

DateTime d;

if (DateTime.TryParse(it.SubItems[1].Text, out d)) dtNgaySinh.Value = d;

txtLop.Text = it.SubItems[2].Text;

txtDiaChi.Text = it.SubItems[3].Text;

};

grpList.Controls.Add(lvSV);

// ===== Add lên form =====

Controls.AddRange(new Control[] { lblHeader, grpInfo, grpAction, grpList });

// phím Enter = Thêm

KeyPreview = true;

KeyDown += (s, e) =>

{

if (e.KeyCode == Keys.Enter) { BtnThem\_Click(s, EventArgs.Empty); e.Handled = true; }

};

}

// ===== Xử lý nút Thêm =====

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

{

var hoten = (txtHoTen.Text ?? "").Trim();

if (string.IsNullOrEmpty(hoten))

{

MessageBox.Show("Họ tên không được rỗng!", "Thiếu thông tin", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtHoTen.Focus();

return;

}

var it = new ListViewItem(hoten);

it.SubItems.Add(dtNgaySinh.Value.ToShortDateString());

it.SubItems.Add((txtLop.Text ?? "").Trim());

it.SubItems.Add((txtDiaChi.Text ?? "").Trim());

lvSV.Items.Add(it);

ClearInputs();

}

// ===== Xử lý nút Sửa =====

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

{

if (lvSV.SelectedItems.Count == 0)

{

MessageBox.Show("Hãy chọn 1 dòng để sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

var hoten = (txtHoTen.Text ?? "").Trim();

if (string.IsNullOrEmpty(hoten))

{

MessageBox.Show("Họ tên không được rỗng!", "Thiếu thông tin", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtHoTen.Focus();

return;

}

var it = lvSV.SelectedItems[0];

it.SubItems[0].Text = hoten;

it.SubItems[1].Text = dtNgaySinh.Value.ToShortDateString();

it.SubItems[2].Text = (txtLop.Text ?? "").Trim();

it.SubItems[3].Text = (txtDiaChi.Text ?? "").Trim();

}

// ===== Xử lý nút Xóa =====

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

{

if (lvSV.SelectedItems.Count == 0)

{

MessageBox.Show("Bạn chưa chọn dòng cần xóa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

var r = MessageBox.Show("Xóa dòng đã chọn?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (r == DialogResult.Yes)

{

lvSV.Items.Remove(lvSV.SelectedItems[0]);

ClearInputs();

}

}

private void ClearInputs()

{

txtHoTen.Clear();

txtLop.Clear();

txtDiaChi.Clear();

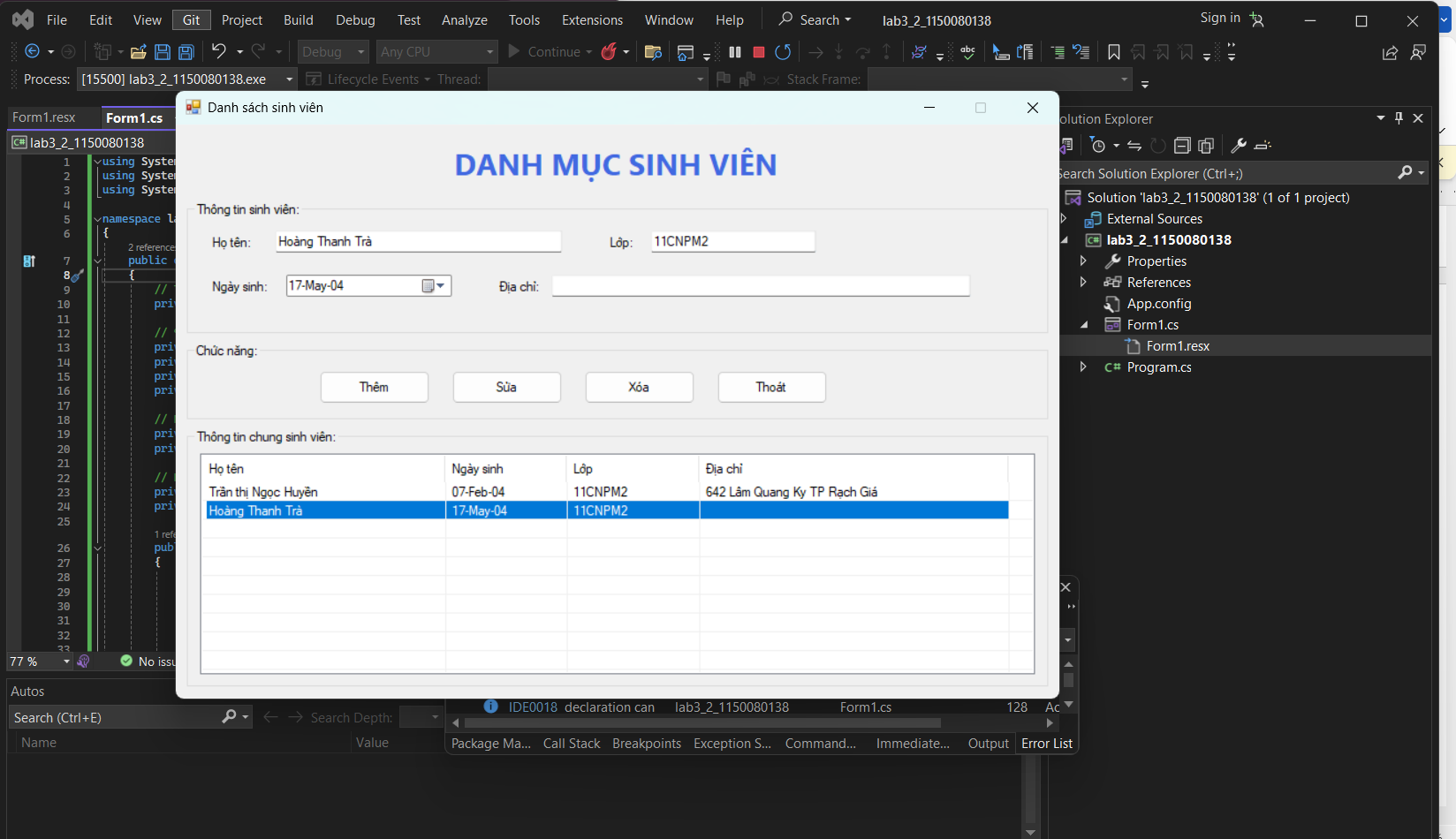
dtNgaySinh.Value = DateTime.Today;

txtHoTen.Focus();

}

}

}

****

**LAB 4:**

**WINFORM KẾT NỐI CƠ SỞ DỮ LIỆU CƠ BẢN**

**B. Nội dung**

**Thực hành 1: Tạo kết nối giữa Winform và CSDL.**

**Đầu tiên, ta phải có được Server name của máy local mình đang sử dụng:** **A screenshot of a computer

AI-generated content may be incorrect.**

**Tiếp theo, trong VS, check trong TOOLS/Connect to database...**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Paste Server name vào (1) → chọn databse đã có trên máy local trong (2)**

**Nhấn (3) để lấy chuỗi kết nối:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Nhấn (4) để kiểm tra kết nối:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Nhấn (5) để hoàn tất kết nối!**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Thực hành 2: Truy vấn dữ liệu**

**1. Truy vấn lấy 1 giá trị**

− Để truy vấn lấy 1 giá trị, ta dùng hàm ExecuteScalar.

− Ví dụ: Đếm số lượng sinh viên trong bảng SinhVien

A screenshot of a computer

AI-generated content may be incorrect.