

namespace Lab02

{

public partial class frmForm : Form

{

public static DataTable myData;

public frmForm()

{

InitializeComponent();

}

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

{

MessageBox.Show("Click OK to Exit");

Application.Exit();

}

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

{

this.ActiveControl = txtbxUser;

}

// connect to the data and check if the user and password is correct

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

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=Lab02;Integrated Security=True";

conn.Open();

String sSQL = "SELECT UserID, Password FROM tblUsers WHERE " +

"UserID='" + txtbxUser.Text + "' and Password='" + txtbxPassoword.Text + "'";

SqlCommand cmd = new SqlCommand(sSQL, conn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

if (dt.Rows.Count > 0)

{

MessageBox.Show("Login Successful!");

this.Close();

}

else

{

MessageBox.Show("Invalid Login. Please check userID or Password!");

}

myData = dt;

}

private void frmForm\_FormClosing(object sender, FormClosingEventArgs e)

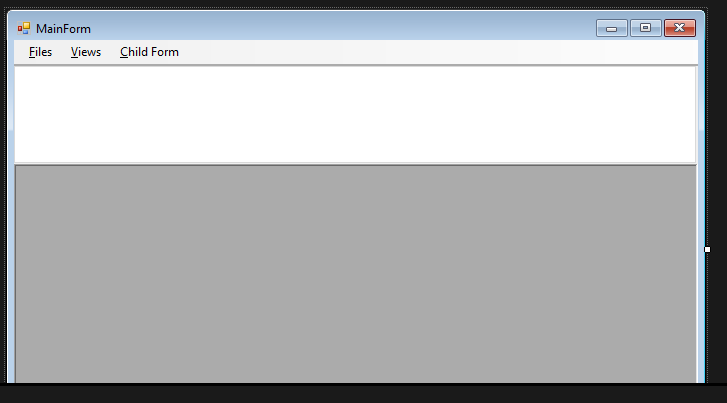
{

//Application.Exit();

}

}

Phần quan trọng nhất là khúc kết nối với SQL



namespace Lab02

{

public partial class MainForm : Form

{

int childCount = 1;

public MainForm()

{

InitializeComponent();

}

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

{

frmForm f = new frmForm();

f.ShowDialog();

}

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

{

this.IsMdiContainer = true;

DataGridForm datag = new DataGridForm();

datag.MdiParent = this;

datag.Show();

}

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

{

this.LayoutMdi(MdiLayout.TileHorizontal);

}

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

{

this.LayoutMdi(MdiLayout.TileVertical);

}

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

{

//MessageBox.Show("Click OK to Exit");

//Application.Exit();

this.Close();

}

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

{

DataGridForm datag = new DataGridForm();

datag.Show();

}

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

{

this.LayoutMdi(MdiLayout.Cascade);

}

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

{

//Creating MDI child form and initialize its fields

ChildForm childForm = new ChildForm();

childForm.Text = "MDIChild " + childCount.ToString();

childForm.MdiParent = this;

//child Form will now hold a reference value to the tab control

childForm.TabCtrl = tabControlDataGridView;

//Add a Tabpage and enables it

TabPage tp = new TabPage();

tp.Parent = tabControlDataGridView;

tp.Text = childForm.Text;

tp.Show();

//child Form will now hold a reference value to a tabpage

childForm.TabPag = tp;

//Activate the MDI child form

childForm.Show();

childCount++;

//Activate the newly created Tabpage

tabControlDataGridView.SelectedTab = tp;

}

private void tabControlDataGridView\_SelectedIndexChanged(object sender, EventArgs e)

{

foreach (ChildForm childForm in this.MdiChildren)

{

if (childForm.TabPag.Equals(tabControlDataGridView.SelectedTab))

{

//Active the child form

childForm.Select();

}

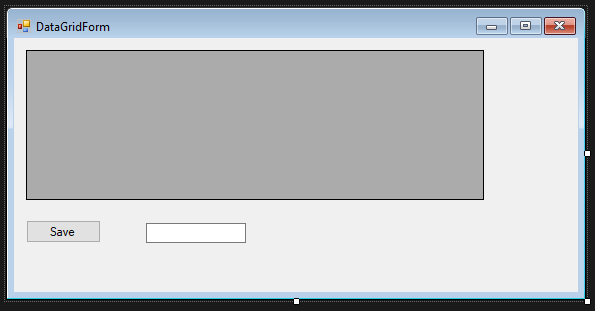
}

}

}

Chúng ta muốn mở form trong form thì phải bật isMDI true và đưa vào DataGridFrom với MDI parent có nó là form này (this)

Để làm các tab, trước tiên chúng ta phải declare có childCount và paste tất cả vào. Chúng ta phải tạo 1 cái để kiểm soát các tab dùng tabControlDataGridView\_SelectedIndexChanged

}

namespace Lab02

{

public partial class DataGridForm : Form

{

// get rows that are initially in SQL

int preDataRows = -1;

public DataGridForm()

{

InitializeComponent();

}

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

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=Lab02;Integrated Security=True";

conn.Open();

String sSQL = "SELECT \* FROM tblUsers";

SqlCommand cmd = new SqlCommand(sSQL, conn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

if (dt.Rows.Count > 0)

{

dataGridViewUser.DataSource = dt;

}

else

{

MessageBox.Show("No data");

}

preDataRows = dt.Rows.Count;

}

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

{

try

{

//String returnData = "";

for (int rows = preDataRows; rows < dataGridViewUser.Rows.Count; rows++)

{

if (dataGridViewUser.Rows[rows].Cells[0].Value == null)

break;

String UserID = dataGridViewUser.Rows[rows].Cells[0].Value.ToString().Trim();

String Name = dataGridViewUser.Rows[rows].Cells[1].Value.ToString().Trim();

String Password = dataGridViewUser.Rows[rows].Cells[2].Value.ToString().Trim();

String temp = dataGridViewUser.Rows[rows].Cells[3].Value.ToString();

int isActive = (dataGridViewUser.Rows[rows].Cells[3].Value.ToString() == "") ? 0 : 1;

//returnData = UserID + ";" + Name + ";" + Password + ";" + isActive;

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=Lab02;Integrated Security=True";

conn.Open();

String sSQL = "INSERT INTO tblUsers (UserID, Name, Password, isActive) VALUES ('"+UserID+"', '"+Name+"', '"+Password+"', '"+isActive+"')";

SqlCommand cmd = new SqlCommand(sSQL, conn);

int i = cmd.ExecuteNonQuery();

if (i != 0)

MessageBox.Show("Saved");

else

MessageBox.Show("Error");

conn.Close();

}

} catch (Exception ex)

{

MessageBox.Show("Some error occur: " + ex.Message + " - " +ex.Source);

}

}

private void dataGridViewUser\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dataGridViewUser\_SelectionChanged(object sender, EventArgs e)

{

string value1 = "";

foreach (DataGridViewRow row in dataGridViewUser.SelectedRows)

{

value1 = row.Cells[0].Value.ToString();

string value2 = row.Cells[1].Value.ToString();

}

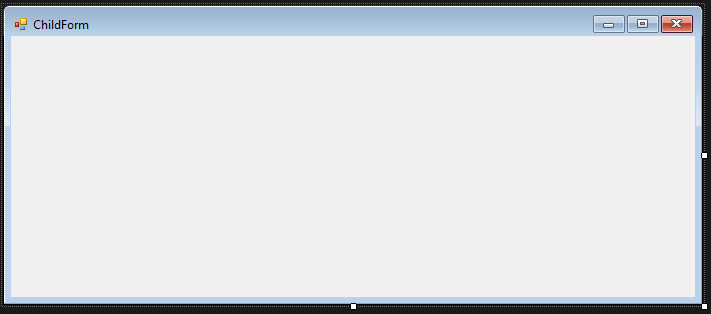
txtbxTest.Text = value1;

}

}

}

Khi load thì form sẽ kết nói với SQL và tải lấy dữ liệu về, chúng ta có thể thay đổi ngay trực tiếp trong datagridview khi đó sẽ tải lên SQL



namespace Lab02

{

public partial class ChildForm : Form

{

//private System.ComponentModel.Container components = null;

private TabControl tabCtrl;

private TabPage tabPag;

public ChildForm()

{

InitializeComponent();

}

public TabPage TabPag

{

get

{

return tabPag;

}

set

{

tabPag = value;

}

}

public TabControl TabCtrl

{

set

{

tabCtrl = value;

}

}

private void ChildForm\_Activated(object sender, EventArgs e)

{

//Activate the correspoding Tabpage

tabCtrl.SelectedTab = tabPag;

if (!tabCtrl.Visible)

{

tabCtrl.Visible = true;

}

}

private void ChildForm\_FormClosing(object sender, FormClosingEventArgs e)

{

//Destroy the corresponding Tabpage when closing MDI child form

this.tabPag.Dispose();

//If no Tabpage left

if (!tabCtrl.HasChildren)

{

tabCtrl.Visible = false;

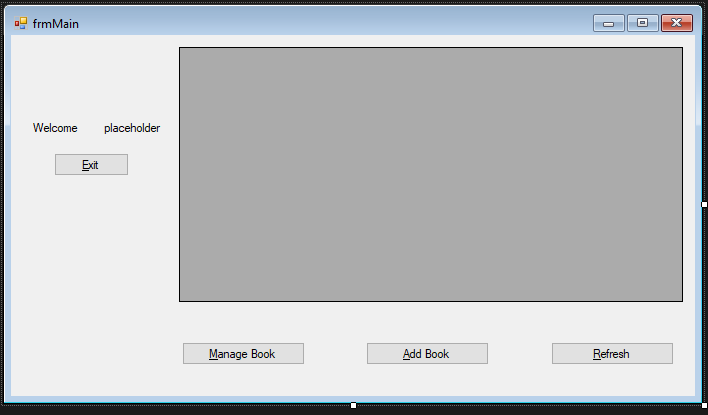
}

}

}

}

Chứa các câu lệnh để thực thi mở các tab



namespace Lab02

{

public partial class frmMain : Form

{

public frmMain()

{

InitializeComponent();

}

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

{

MessageBox.Show("Click OK to Exit", "Warning Box");

Application.Exit();

}

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

{

frmForm frm = new frmForm();

frm.ShowDialog();

try

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=Lab02;Integrated Security=True";

conn.Open();

String sSQL = "SELECT \* FROM tblBooks";

SqlCommand cmd = new SqlCommand(sSQL, conn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

if (dt.Rows.Count > 0)

{

dataGridViewBook.DataSource = dt;

}

else

{

MessageBox.Show("No data");

}

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

frmManageBook frmManageBook = new frmManageBook();

frmManageBook.ShowDialog();

}

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

{

frmAddBook frmAddBook = new frmAddBook();

frmAddBook.ShowDialog();

}

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

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=Lab02;Integrated Security=True";

conn.Open();

String sSQL = "SELECT \* FROM tblBooks";

SqlCommand cmd = new SqlCommand(sSQL, conn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

if (dt.Rows.Count > 0)

{

dataGridViewBook.DataSource = dt;

}

else

{

MessageBox.Show("No data");

}

conn.Close();

}

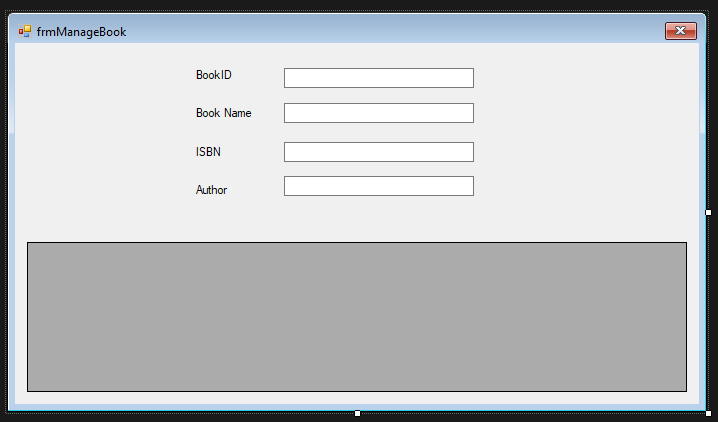
}

}

Đầu tiên, form sẽ thể hiện danh sách các book khi khở động lên, khúc kết nói cần lưu ý đến conn.ConnectionString vì đây sẽ là câu lệnh kết nối database, chú ý đến sSQL vì đây chứa câu lệnh SQL

Nút Refresh được dùng để refresh database sau khi thêm quyển sách mới, em có thể dùng event form closing để tự động refresh database nhưng hiện tại thì chấp nhận phương án này

Ngoài ra, form có nút add và manage book để thêm và thể hiện số lượng book



namespace Lab02

{

public partial class frmManageBook : Form

{

public frmManageBook()

{

InitializeComponent();

}

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

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=Lab02;Integrated Security=True";

conn.Open();

String sSQL = "SELECT \* FROM tblBooks";

SqlCommand cmd = new SqlCommand(sSQL, conn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

if (dt.Rows.Count > 0)

{

dataGridViewBook.DataSource = dt;

}

else

{

MessageBox.Show("No data");

}

}

private void dataGridViewBook\_SelectionChanged(object sender, EventArgs e)

{

string BookID = "";

string BookName = "";

string ISBN = "";

string Author = "";

foreach (DataGridViewRow row in dataGridViewBook.SelectedRows)

{

BookID = row.Cells[0].Value.ToString();

BookName = row.Cells[1].Value.ToString();

ISBN = row.Cells[2].Value.ToString();

Author = row.Cells[3].Value.ToString();

}

txtbxBookID.Text = BookID;

txtbxBookName.Text = BookName;

txtbxISBN.Text = ISBN;

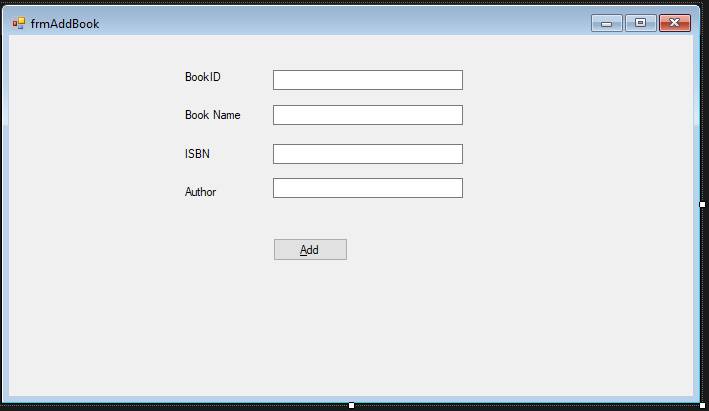
txtbxAuthor.Text = Author;

}

}

}

ManageBook thì thể hiện tổng sớ lượng quyển sách, em dùng event Selection change để kết nối data trên datagridview với các ô textbox (có thể dùng CellContentClick nhưng Selection này dùng ý bị lag hơn) chúng ta sẽ lấy DataGridViewRow từ SelectedRows và chúng ta dùng index để lấy dữ liệu từ từ Row



namespace Lab02

{

public partial class frmAddBook : Form

{

public frmAddBook()

{

InitializeComponent();

}

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

{

try

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=Lab02;Integrated Security=True";

conn.Open();

String sSQL = "INSERT INTO tblBooks (BookID, BookTitle, ISBN, Author) VALUES ('" + txtbxBookID.Text + "', '" + txtbxBookName.Text + "', '" + txtbxISBN.Text + "', '" + txtbxAuthor.Text + "')";

SqlCommand cmd = new SqlCommand(sSQL, conn);

int i = cmd.ExecuteNonQuery();

if (i != 0)

MessageBox.Show("Saved");

else

MessageBox.Show("Error");

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Some error occur: " + ex.Message + " - " + ex.Source);

}

finally

{

this.Close();

}

}

private void txtbxBookID\_Leave(object sender, EventArgs e)

{

if(txtbxBookID.Text == "")

{

MessageBox.Show("Can not leave Empty", "Warning Box");

this.ActiveControl = txtbxBookID;

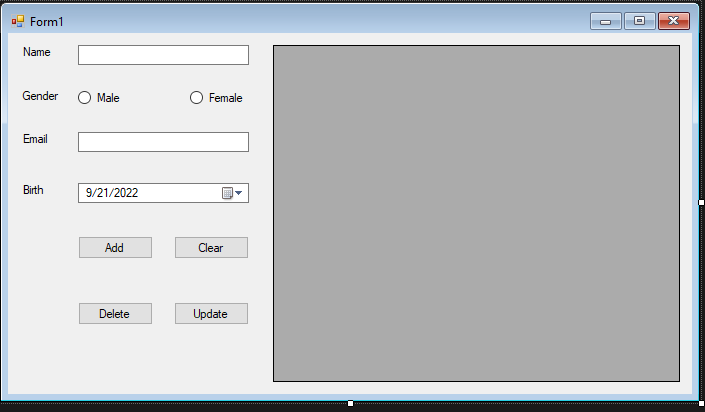
}

}

}

}

Chúng ta sẽ add book, chúng ta lấy dữ liệu từ các textbox và add chúng vào câu lệnh SQL, hoàn thành thì MessageBox Saved, sai thì báo error



namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

try

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=school;Integrated Security=True";

Khi chạy chương trình thì form sẽ kết nối với database và thể hiện data trong datagridview

conn.Open();

String sSQL = "SELECT \* FROM student";

SqlCommand cmd = new SqlCommand(sSQL, conn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

if (dt.Rows.Count > 0)

{

dataGridViewStudent.DataSource = dt;

// this.Close();

}

else

{

MessageBox.Show("No Data");

}

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

this.ActiveControl = txtbxName;

}

private void dataGridViewStudent\_CellContentClick(object sender, DataGridViewCellEventArgs e)

Dùng CellContentClick để lấy giá trị từng bản và hiện thị từng bảng ở các textbox, vì các giá trị bit thể hiện trên datagridvie là true và false nên chúng ta phải so sánh bằng string.

Dùng DataGridVieRow để lưu giá trị từng Row và SelectedRow để lấy từng Row được chọn

{

btnAdd.Enabled = false;

//string maso = "";

string hovaten = "";

string ngaySinh = "";

string gioitinh = "";

string email = "";

foreach (DataGridViewRow row in dataGridViewStudent.SelectedRows)

{

// maso = row.Cells[0].Value.ToString();

hovaten = row.Cells[1].Value.ToString();

ngaySinh = row.Cells[2].Value.ToString();

gioitinh = row.Cells[3].Value.ToString();

email = row.Cells[4].Value.ToString();

}

txtbxName.Text = hovaten;

dateTimePickerStudent.Text = ngaySinh;

if (gioitinh.Equals("True") == true)

{

rdbtnMale.Checked = true;

rdbtnFemale.Checked = false;

}

else

{

rdbtnMale.Checked = false;

rdbtnFemale.Checked = true;

}

txtbxEmail.Text = email;

}

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

{

Clear tất cả các giá trị và gán radio button về false, set datatime picker về ngày hiện tại và focus vào ô đầu tiên là text box Name

btnAdd.Enabled = true;

//btnUpdate.Enabled = false;

//Update.Enabled = false;

txtbxName.Clear();

txtbxEmail.Clear();

rdbtnFemale.Checked = false;

rdbtnMale.Checked = false;

dateTimePickerStudent.Value = DateTime.Today;

this.ActiveControl = txtbxName;

}

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

{

int gioiTinhBitValue = (rdbtnMale.Checked == true) ? 1 : 0;

try

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=school;Integrated Security=True";

conn.Open();

String sSQL = "INSERT INTO student (hoTen, ngaySinh, gioitinh, email) VALUES ('" + txtbxName.Text + "', '" + dateTimePickerStudent.Text + "', '" + gioiTinhBitValue + "', '" + txtbxEmail.Text + "')";

SqlCommand cmd = new SqlCommand(sSQL, conn);

int i = cmd.ExecuteNonQuery();

Khi add thì function sẽ add trực tiếp vào trong database đồng thời em tạo function refreshdatabase để khi add vào rồi nó sẽ tự động lấy lại database, tạo cảm giác refresh

if (i != 0)

MessageBox.Show("Saved");

else

MessageBox.Show("Error");

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Some error occur: " + ex.Message + " - " + ex.Source);

}

finally

{

refreshDataBase();

//this.Close();

}

}

private void refreshDataBase()

{

try

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=school;Integrated Security=True";

conn.Open();

String sSQL = "SELECT \* FROM student";

SqlCommand cmd = new SqlCommand(sSQL, conn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

if (dt.Rows.Count > 0)

{

dataGridViewStudent.DataSource = dt;

// this.Close();

}

else

{

MessageBox.Show("No Data");

}

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Some error occur: " + ex.Message + " - " + ex.Source);

}

}

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

{

DataGridViewRow row = dataGridViewStudent.SelectedRows[0];

//MessageBox.Show(row.Cells[0].Value.ToString());

int gioiTinhBitValue = (rdbtnMale.Checked == true) ? 1 : 0;

try

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=school;Integrated Security=True";

conn.Open();

String sSQL = "UPDATE student SET hoTen = @hoTen, ngaySinh = @ngaySinh, gioiTinh=@gioiTinh, email=@email WHERE maso=@maso";

SqlCommand cmd = new SqlCommand(sSQL, conn);

cmd.Parameters.AddWithValue("@hoten", txtbxName.Text);

cmd.Parameters.AddWithValue("@ngaySinh", dateTimePickerStudent.Text);

cmd.Parameters.AddWithValue("@gioiTinh", gioiTinhBitValue);

cmd.Parameters.AddWithValue("@email", txtbxEmail.Text);

cmd.Parameters.AddWithValue("@maso", int.Parse(row.Cells[0].Value.ToString()));

Update thì dùng câu lệnh Update trong SQL và refresh lại cái database

Ở đây có 1 cách để thêm các value vào thay vì thêm tại sSQL thì chúng ta thêm các biến và sau đó gán giá trị vào các biến đó ở dưới

int i = cmd.ExecuteNonQuery();

if (i != 0)

MessageBox.Show("Saved");

else

MessageBox.Show("Error");

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Some error occur: " + ex.Message + " - " + ex.Source);

}

finally

{

refreshDataBase();

//this.Close();

}

}

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

{

DataGridViewRow row = dataGridViewStudent.SelectedRows[0];

try

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = @"Data Source=DESKTOP-N5KPC9G\SQLEXPRESS; Initial Catalog=school;Integrated Security=True";

conn.Open();

String sSQL = "DELETE FROM student WHERE maso = @maso";

SqlCommand cmd = new SqlCommand(sSQL, conn);

Function update này thật chất là function delete, chúng ta chỉ cần dùng câu lệnh delete trong SQL

Bởi vì tât cả student đều có ID nên chúng ta chỉ cần dùng DataGridViewRow row = dataGridViewStudent.SelectedRows[0]; để lấy ID của từng SV sau đó dùng câu lênh SQL để xóa

cmd.Parameters.AddWithValue("@maso", int.Parse(row.Cells[0].Value.ToString()));

int i = cmd.ExecuteNonQuery();

if (i != 0)

MessageBox.Show("Deleted");

else

MessageBox.Show("Error");

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Some error occur: " + ex.Message + " - " + ex.Source);

}

finally

{

refreshDataBase();

//this.Close();

}

}

private void txtbxName\_Leave(object sender, EventArgs e)

{

//if (txtbxName.Text == "")

//{

// MessageBox.Show("Can not leave Name empty", "Warning Box");

// this.ActiveControl = txtbxName;

//}

}

private void txtbxEmail\_Leave(object sender, EventArgs e)

{

//if(txtbxEmail.Text == "" || txtbxEmail.Text.Contains("@") == false)

//{

// MessageBox.Show("Can not leave Email empty or missing and @", "Warning Box");

// this.ActiveControl=txtbxEmail;

//}

}

}