

## Lab 08: Database programming with ASP.net C# 3

### จุดประสงค์

ศึกษาและเรียนรู้ Database programming ด้วย ASP.net C#

### ความต้องการ

1. MS SQL Server Express or above [link download](#)
2. SQL Server Management Studio version 18 or above editor [link download](#)
3. Microsoft Visual Studio community 2019 or above [link download](#)

### การทดลอง

1. เปิดโปรแกรม visual studio 2019 แล้วเปิด Solution ล่าสุดที่ได้สร้างไว้ชื่อ "LABDB20121.sln"
2. เมื่อเข้าสู่หน้าการทำงานให้ไปยัง profile.aspx.cs เพื่อดำเนินการแก้ไขในส่วน function saveBtn\_Click เพื่อให้สามารถบันทึกข้อมูล Address ได้ด้วยการแทนที่ด้วย code ในกรอบสีขวาด้านล่าง

```
SqlConnection SqlCon = new
SqlConnection(WebConfigurationManager.ConnectionStrings["strconn"].ConnectionString);
    SqlCon.Open();
    string q = "INSERT INTO student VALUES(@title, @fname, @lname, @blood, '5222' ) " +
        "SELECT SCOPE_IDENTITY() as studentid ";
    var cmd = new SqlCommand(q, SqlCon);
    cmd.Parameters.AddWithValue("@title", titleDpDwn.SelectedValue);
    cmd.Parameters.AddWithValue("@fname", nameTxbx.Text);
    cmd.Parameters.AddWithValue("@lname", lnameTxbx.Text);
    cmd.Parameters.AddWithValue("@blood", bloodRBL.SelectedValue);

    var reader = cmd.ExecuteReader();
    string studentid = "";

    if (reader.Read())
    {
        studentid = reader["studentid"].ToString();
    }
```

```

reader.Close();

if (gotAddrCB.Checked == true)
{
    q = "INSERT INTO address VALUES (@stid, @no, @road, @province, @amphor, @tumbon, @postcode)";
    cmd = new SqlCommand(q, SqlCon);
    cmd.Parameters.AddWithValue("@stid", studentid);
    cmd.Parameters.AddWithValue("@no", noTxBx.Text);
    cmd.Parameters.AddWithValue("@road", RoadTxBx.Text);
    cmd.Parameters.AddWithValue("@province", provinceTxBx.Text);
    cmd.Parameters.AddWithValue("@amphor", amphorTxBx.Text);
    cmd.Parameters.AddWithValue("@tumbon", tumbonTxBx.Text);
    cmd.Parameters.AddWithValue("@postcode", postcodeTxBx.Text);
    cmd.ExecuteNonQuery();
}

SqlCon.Close();

```

3. เมื่อทำการแก้ไข code จากขั้นตอนด้านบนแล้ว ให้ทำการ run WebApplication แล้วทดลองกรอกข้อมูลดังนี้

TITLE	FNAME	LNAME	BLOOD	GOT ADDRESS
MR	taka	shike	A	not
MISS	sora	aoi	B	got

และ sora มีข้อมูลที่อยู่ดังนี้

no	road	province	amphor	tumbon	postcode
123	road monor	phitsanulok	muang	tapho	65000

แล้วทำการ capture ข้อมูลในฐานข้อมูล student และ address ลงในกรอบสี่เหลี่ยมด้านล่างนี้

	id	title	fname	lname	blood	studentid	stid	no	road	province	amphor	tumbon	postcode
1	1	1	Supawitch	Srichaichana	3	5222	3	123	road monor	phitsanulok	muang	tapho	65000
2	2	1	taka	shike	1	5222	3	123	road monor	phitsanulok	muang	tapho	65000
3	3	1	sora	aoi	2	5222	3	123	road monor	phitsanulok	muang	tapho	65000

4. แต่จากการบันทึกข้อมูลที่แล้วนั้นจะสังเกตได้ว่าผู้ใช้งานแทบจะไม่สามารถรับรู้ได้ว่าข้อมูลได้บันทึกลงเรียบร้อยแล้ว หรือมีปัญหาอะไรเกิดขึ้น ในโลกแห่งความเป็นจริงสิ่งหนึ่งที่นิยมใช้ในการแจ้งเหตุการณ์ต่างให้กับผู้ใช้งานนั้น คือ

MessageBox หรือกล่องข้อความที่ปรากฏขึ้นมาแจ้งผู้ใช้ แต่ใน asp.net Web Application นั้นไม่มี element นี้ให้  
เลือกใช้งาน จำเป็นที่จะต้องทำการผลิตแปลงใช้งาน client script โดยในที่นี้จำเป็นต้องสร้าง function ขึ้นมาก่อน  
เพื่อให้สะดวกในการเรียกใช้งาน

5. สร้าง function ใหม่ด้วยการแทรก code จากกรอบสีขาวด้านล่าง ลงใน class profile

```
protected void alert(string message)
{
    ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" +
message + "');", true);
}
```

6. แล้วจึงเพิ่ม code จากกรอบสีขาวด้านล่าง เพื่อเรียกใช้งานในส่วนท้ายสุดของ function saveBtn\_Click เพื่อให้  
แจ้งว่าบันทึกเรียบร้อยแล้ว

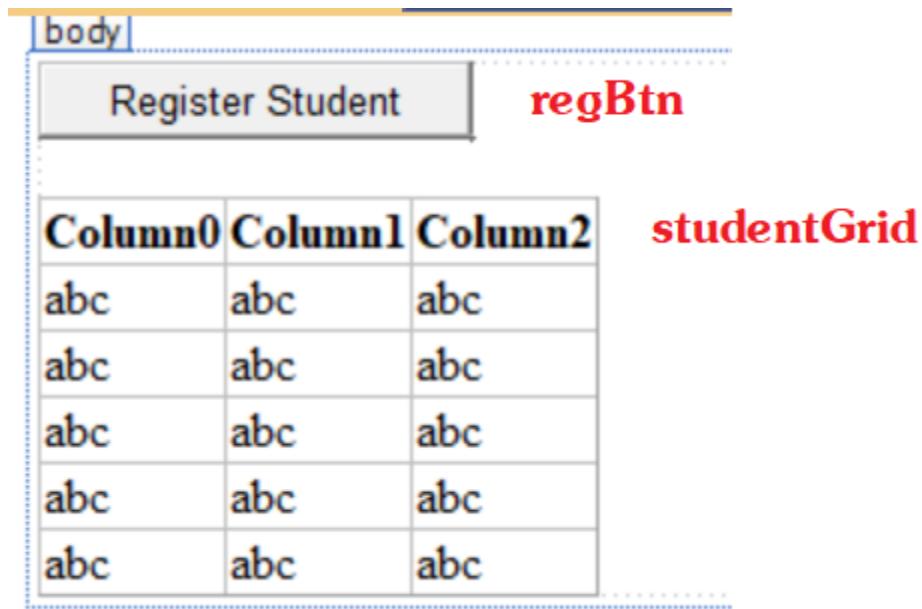
```
alert("บันทึกข้อมูลสำเร็จแล้ว");
```

7. แล้วทดสอบการทำงานด้วยการ run Web Application แล้วบันทึกข้อมูลอีกครั้ง แต่จะเห็นได้ว่าเมื่อทำการบันทึก  
เรียบร้อยแล้ว มีการแจ้งเตือนแล้ว แต่ไม่มีการล้างข้อมูลที่ได้ทำการกรอกแล้วนั้น เป็นภาระต่อการทำงานในกรณีที่ต้องทำการ  
กรอกข้อมูลมากกว่า 1 record ในขั้นตอนนี้เลย ให้นิสิตทำการเพิ่ม code เพื่อให้ระบบมีการล้างข้อมูลเมื่อมีการบันทึกข้อมูล  
เรียบร้อยแล้ว โดยให้นิสิต copy code ที่ตนเองได้แก้ไขเพิ่มเติมลงในกรอบสีส้มด้านล่าง

```
titleDrpDwn.ClearSelection();
nameTxBx.Text = string.Empty;
lnameTxBx.Text = string.Empty;
bloodRBL.ClearSelection();

noTxBx.Text = string.Empty;
roadTxBx.Text = string.Empty;
provinceTxBx.Text = string.Empty;
amphorTxBx.Text = string.Empty;
tumbonTxBx.Text = string.Empty;
postcodeTxBx.Text = string.Empty;
```

8. โดยปกติแล้วเมื่อมีการบันทึกข้อมูล ก็ต้องมีส่วนเรียกดูข้อมูลที่ได้ลงไปแล้ว ในที่นี้ให้สร้าง Web Form ใหม่ โดยให้มีชื่อ  
ว่า studentlist และทำการจัด Element ที่จำเป็นดังรูปที่ 1 โดยให้ตั้ง ID แต่ละ element ตามตัวหนังสือสีแดงด้านข้าง  
element



รูปที่ 1 แสดงการจัดวาง element สำหรับ studentlist Web Form

9. แล้วทำการแก้ไขใน function Page\_Load ของ Form studentlist ด้วย code ในกรอบสีชาวด้านล่าง

```
if (!IsPostBack)
{
    bindStudentList();
}
```

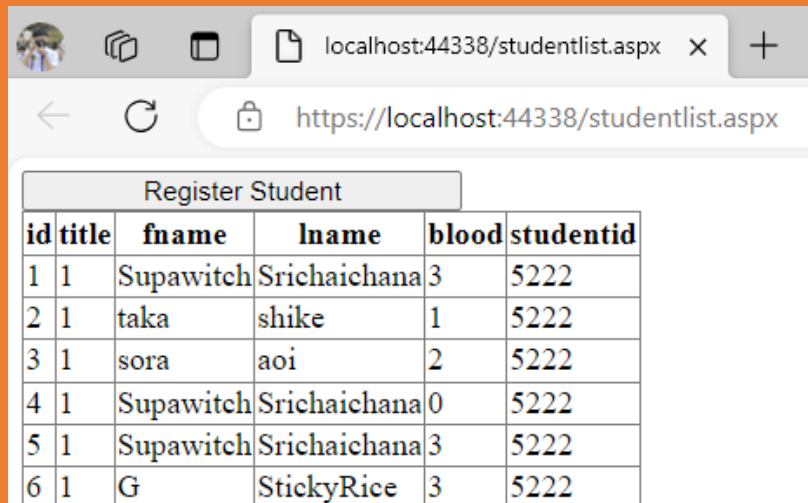
10. ต่อด้วยทำการเพิ่มfunction bindStudentList ลงใน class studentlist ด้วย code ในกรอบสีชาวด้านล่าง ซึ่งอาจจะเกิด error ขึ้นให้ดำเนินการแก้ไขด้วยการเพิ่ม Library ที่จำเป็นเข้ามายังหน้า Web Form แล้วทดลอง run Web Application แล้ว capture ผลลัพธ์ของหน้า Website ลงในกรอบสีส้มด้านล่าง

```
protected void bindStudentList()
{
    SqlConnection sqlCon = new
    SqlConnection(WebConfigurationManager.ConnectionStrings["strconn"].ConnectionString);
    string q = "SELECT * FROM student a ";
    sqlCon.Open();

    var cmd = new SqlCommand(q, sqlCon);
    var reader = cmd.ExecuteReader();

    studentGrid.DataSource = reader;
    studentGrid.DataBind();

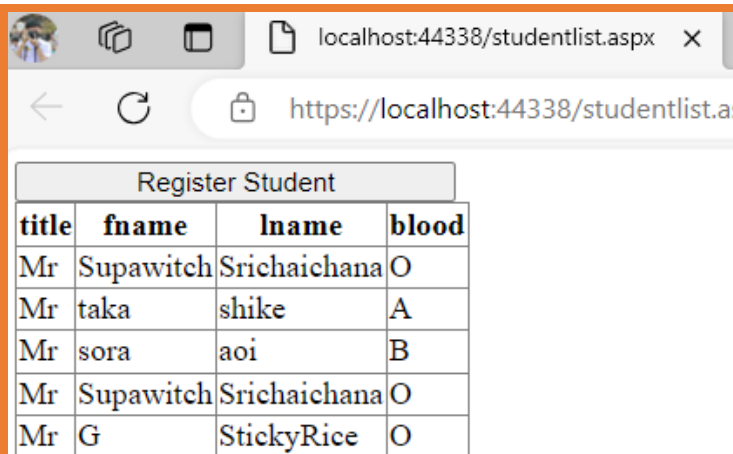
    reader.Close();
    sqlCon.Close();
}
```



Register Student					
id	title	fname	lname	blood	studentid
1	1	Supawitch	Srichaichana	3	5222
2	1	taka	shike	1	5222
3	1	sora	aoi	2	5222
4	1	Supawitch	Srichaichana	0	5222
5	1	Supawitch	Srichaichana	3	5222
6	1	G	StickyRice	3	5222

11. ซึ่งจะได้ข้อมูลมาแสดงแต่ว่าดูไม่สามารถเข้าใจสิ่งที่แสดงได้ ให้ดำเนินการแก้ไข query ภายในฟังก์ชัน bindStudentList ด้วยการแทนที่ query ภายในกรอบสี่เหลี่ยมด้านล่าง แล้วทดลอง run Web Application แล้ว capture ผลลัพธ์ของหน้า Website ลงในกรอบสี่เหลี่ยมด้านล่าง

```
SELECT t.title, s.fname, s.lname, b.blood
FROM student s, title t, blood b
WHERE s.title = t.id AND s.blood = b.id
```

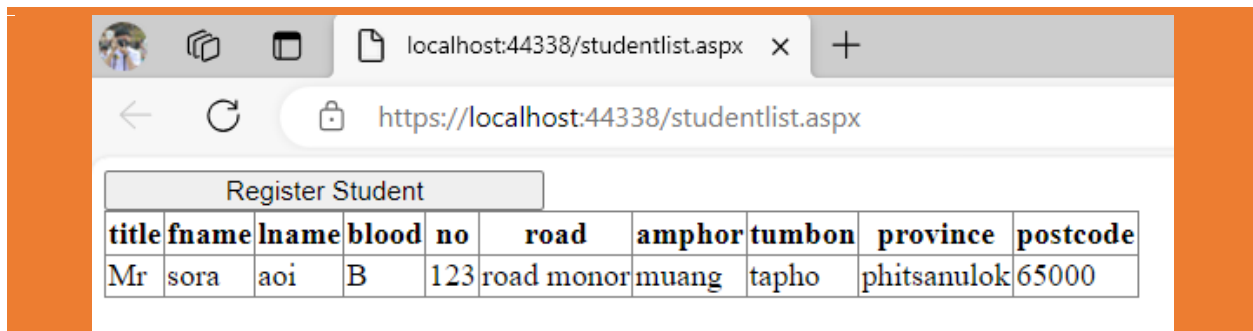


Register Student			
title	fname	lname	blood
Mr	Supawitch	Srichaichana	O
Mr	taka	shike	A
Mr	sora	aoi	B
Mr	Supawitch	Srichaichana	O
Mr	G	StickyRice	O

12. ที่นี้ข้อมูลนักเรียนนั้นไม่มีเพียงแค่ ชื่อ นามสกุล กรุ๊ปเลือด อย่างเดียวเท่านั้นแต่ว่ามีข้อมูลที่อยู่ด้วย จึงมีความจำเป็นที่จะต้องดำเนินการแก้ไข query ภายในฟังก์ชัน bindStudentList ด้วยการแทนที่ query ภายในกรอบสี่เหลี่ยมด้านล่าง แล้วทดลอง run Web Application แล้ว capture ผลลัพธ์ของหน้า Website ลงในกรอบสี่เหลี่ยมด้านล่าง

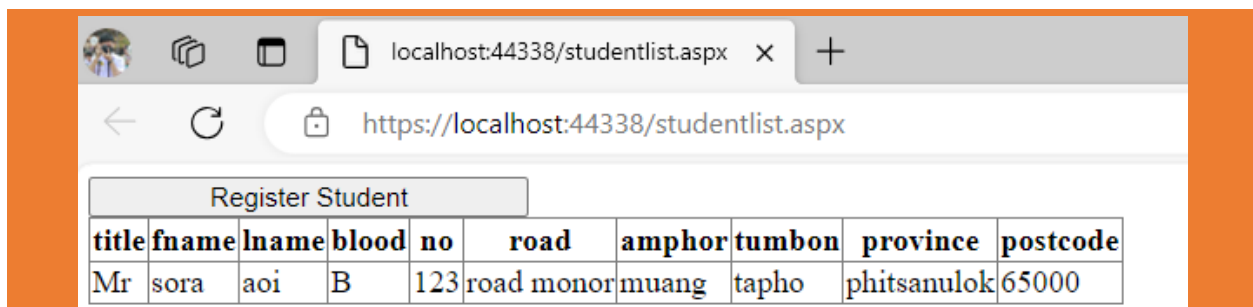
```
SELECT t.title, s.fname, s.lname, b.blood, a.[no], a.road, a.amphor, a.tumbon, a.province, a.postcode
```

```
FROM student s, title t, blood b, address a
WHERE s.title = t.id AND s.blood = b.id AND a.stid = s.id
```



13. แต่สิ่งที่ได้จากการแก้ไข เพิ่ม relation list ที่เป็น address กลับกลายเป็นว่าทำให้ข้อมูลนักเรียนหายไปบางส่วน อันเนื่องมาจากระบบจะมีนักเรียนบางคนที่ไม่ได้อยู่ทำให้ไม่สามารถเรียกดูข้อมูลตามแบบข้างบนได้ ต้องใช้คุณสมบัติที่เรียกว่า JOIN มาดำเนินการแก้ไข query ภายในฟังก์ชัน bindStudentList ด้วยการแทนที่ query ภายในกรอบสี่เหลี่ยมด้านล่าง แล้วทดลอง run Web Application แล้ว capture ผลลัพธ์ของหน้า Website ลงในกรอบสี่เหลี่ยมด้านล่าง

```
SELECT t.title, s.fname, s.lname, b.blood, a.[no], a.road, a.amphor, a.tumbon, a.province, a.postcode
FROM student s
JOIN title t ON s.title = t.id
JOIN blood b ON s.blood = b.id
JOIN address a ON a.stid = s.id
```



14. แต่สิ่งที่ได้จาก JOIN แบบปกตินี้ยังไม่ได้ช่วยแก้ปัญหา เพราะว่าเป็นเลือกใช้ JOIN ที่ไม่สอดคล้องกับข้อมูลที่มี ดำเนินการแก้ไข query ภายในฟังก์ชัน bindStudentList ด้วยการแทนที่ query ภายในกรอบสี่เหลี่ยมด้านล่าง แล้วทดลอง run Web Application แล้ว capture ผลลัพธ์ของหน้า Website ลงในกรอบสี่เหลี่ยมด้านล่าง

```
SELECT t.title, s.fname, s.lname, b.blood, a.[no], a.road, a.amphor, a.tumbon, a.province, a.postcode
FROM student s
JOIN title t ON s.title = t.id
```

```
JOIN blood b ON s.blood = b.id
LEFT JOIN address a ON a.stid = s.id
```

title	fname	lname	blood	no	road	amphor	tumbon	province	postcode
Mr	Supawitch	Srichaichana	O						
Mr	taka	shike	A						
Mr	sora	aoi	B	123	road monor	muang	tapho	phitsanulok	65000
Mr	Supawitch	Srichaichana	O						
Mr	G	StickyRice	O						

15. หลังจากที่สามารถแสดงข้อมูลนักเรียนได้ครบถ้วนตามที่มึพื้นฐานข้อมูลแล้วนั้น จะเห็นว่ามีช่องที่เป็น empty string ในกรณีที่ไม่มีข้อมูล ซึ่งไม่สอดคล้องกับความจริงเพราะว่าผู้ใช้ไม่ได้กรอกมาแบบนั้น ในขั้นตอนนี้จะทดลองเปลี่ยนข้อมูลที่ไม่มีเป็น “-” ด้วยการดำเนินการแก้ไข query ภายในฟังก์ชัน bindStudentList ด้วยการแทนที่ query ภายในกรอบสี่เหลี่ยมด้านล่าง แล้ว ทดลอง run Web Application แล้ว capture ผลลัพธ์ของหน้า Website ลงในกรอบสี่เหลี่ยมด้านล่าง

```
SELECT t.title, s.fname, s.lname, b.blood,
ISNULL(a.[no], '') AS [no], ISNULL(a.[road], '-') AS road,
ISNULL(a.[amphor], '-') AS amphor, ISNULL(a.[tumbon], '-') AS tumbon,
ISNULL(a.[province], '-') AS province, ISNULL(a.[postcode], '-') AS postcode
FROM student s
JOIN title t ON s.title = t.id
JOIN blood b ON s.blood = b.id
LEFT JOIN address a ON a.stid = s.id
```

title	fname	lname	blood	no	road	amphor	tumbon	province	postcode
Mr	Supawitch	Srichaichana	O	0	-	-	-	-	-
Mr	taka	shike	A	0	-	-	-	-	-
Mr	sora	aoi	B	123	road monor	muang	tapho	phitsanulok	65000
Mr	Supawitch	Srichaichana	O	0	-	-	-	-	-
Mr	G	StickyRice	O	0	-	-	-	-	-

16. หลังจากที่เราทำหน้าเว็บ เราสามารถแสดงผลข้อมูลที่มีในฐานแล้ว เราจะสร้างไปยังหน้า Design สำหรับ studentlist.aspx แล้วทำการ double click ที่ปุ่ม “Register Student” เพื่อสร้าง event click แล้วทำการแก้ไข event ด้วย code ที่อยู่ในกรอบสี่เหลี่ยมด้านล่าง แล้ว run Web Application เพื่อทดลองกดปุ่ม “Register Student”

```
Response.Redirect("~/profile.aspx");
```

17. หลังจากที่เราทำปุ่มไปยังหน้าลงทะเบียนได้แล้วนั้น ให้นิสิตทำการทดลอง สร้างปุ่มในหน้า profile.aspx ให้มีชื่อว่า “Back to Student List” แล้วแก้ไข code ให้สามารถกลับมายังหน้าของ รายชื่อ นักเรียนให้ได้ แล้วทำการ copy code ที่ได้แก้ไขเองลงในกรอบสี่เหลี่ยมด้านล่าง

```
0 references
protected void BacktoStuListBtn_Click(object sender, EventArgs e)
{
    Response.Redirect("~/studentlist.aspx");
}
```