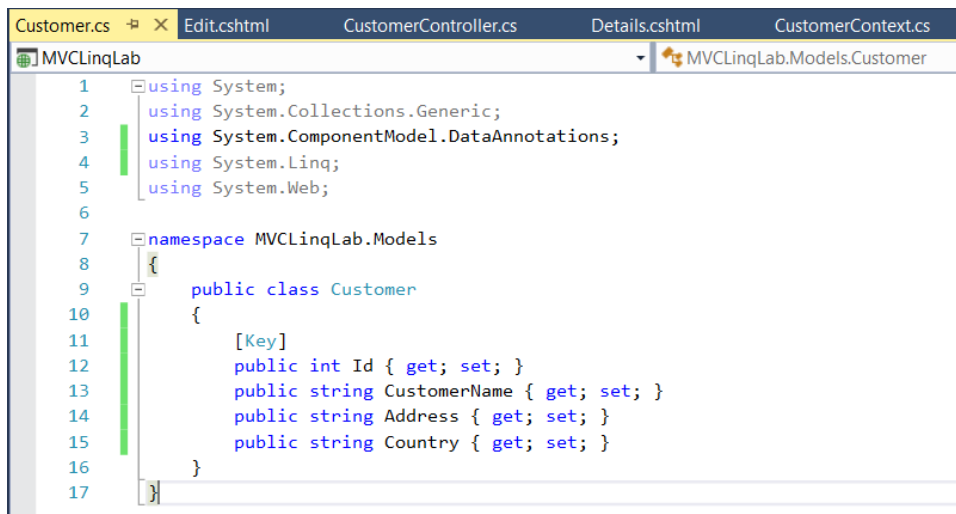


1. สร้างโปรเจกต์และตั้งชื่อตามต้องการเช่น “MVCLinqLab”
2. เลือกรูปแบบ MVC และกด OK
3. Click ขวาที่ Folder “Models” เลือก Add > เลือก Class...
4. ตั้งชื่อ Class ว่า “Customer”
5. เพิ่ม Code เพื่อสร้าง Entity ของ Database ดังตัวอย่าง ที่ Class Models Customer

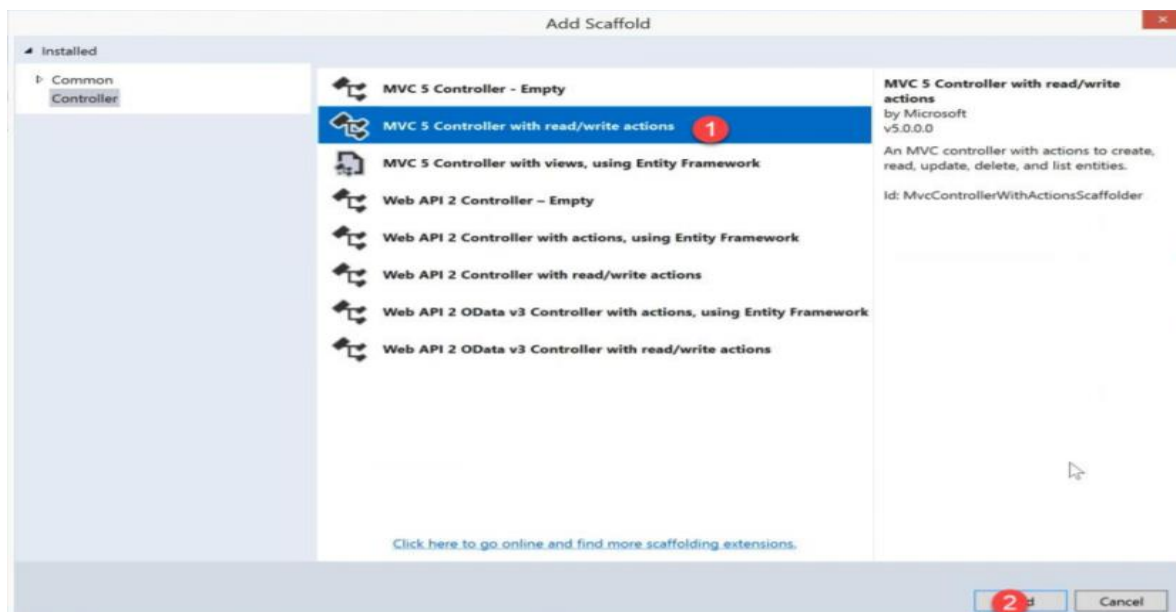


```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations;
4  using System.Linq;
5  using System.Web;
6
7  namespace MVCLinqLab.Models
8  {
9      public class Customer
10     {
11         [Key]
12         public int Id { get; set; }
13         public string CustomerName { get; set; }
14         public string Address { get; set; }
15         public string Country { get; set; }
16     }
17 }

```

6. Click ขวาที่ “Controllers”> Add > Controller
7. เลือกตัวเลือก ดังหมายเลข 1



8. ตั้งชื่อ “CustomerController”
9. สร้าง Folder ใหม่ตั้งชื่อ “Contexttt”

10. Add Class ใน Folder Context

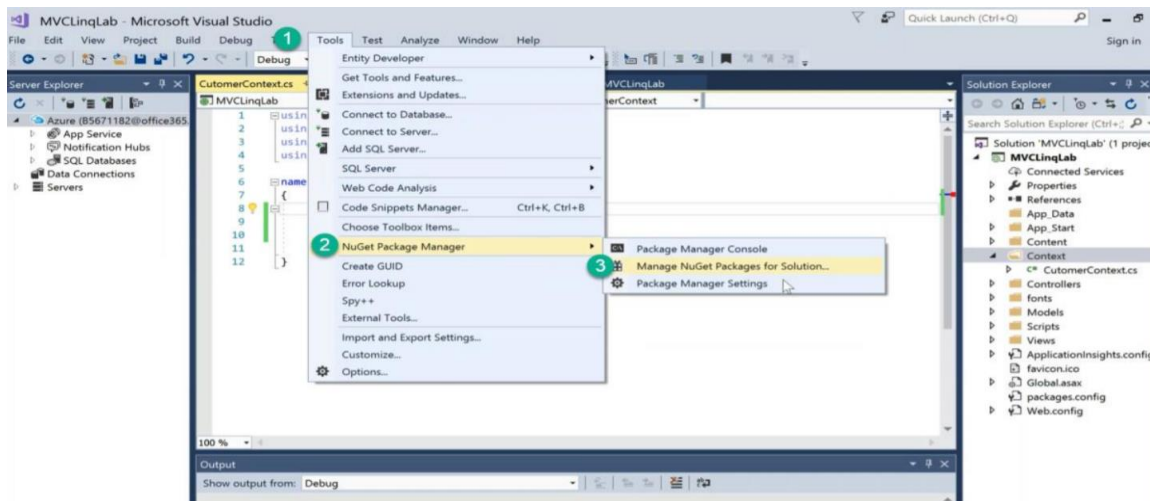
11. ตั้งชื่อว่า "CustomerContext"

12. เพิ่ม Code ต่อส่วนท้ายรูปภาพ (:) หากมี Warning ขึ้น ให้เลือก Add Data Entity ให้ทำการ Download เพิ่มเติม

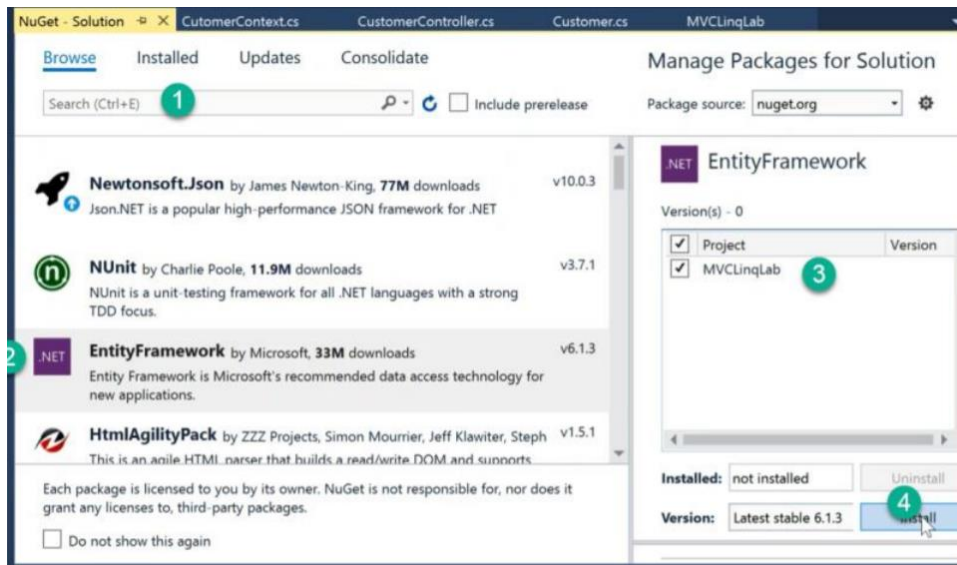
หลังจากนั้นพิมพ์ Code ลงในที่ Class Customer Context `public DbSet Customer { get; set; }` เพื่อทำการกำหนดชุดข้อมูล

```
CustomerContext.cs  Customer.cs  CustomerController.cs  Edit.cshtml
MVCLinqLab
1  using MVCLinqLab.Models;
2  using System;
3  using System.Collections.Generic;
4  using System.Data.Entity;
5  using System.Linq;
6  using System.Web;
7
8  namespace MVCLinqLab.Context
9  {
10     public class CustomerContext : DbContext
11     {
12         public DbSet<Customer> Customer { get; set; }
13     }
14 }
```

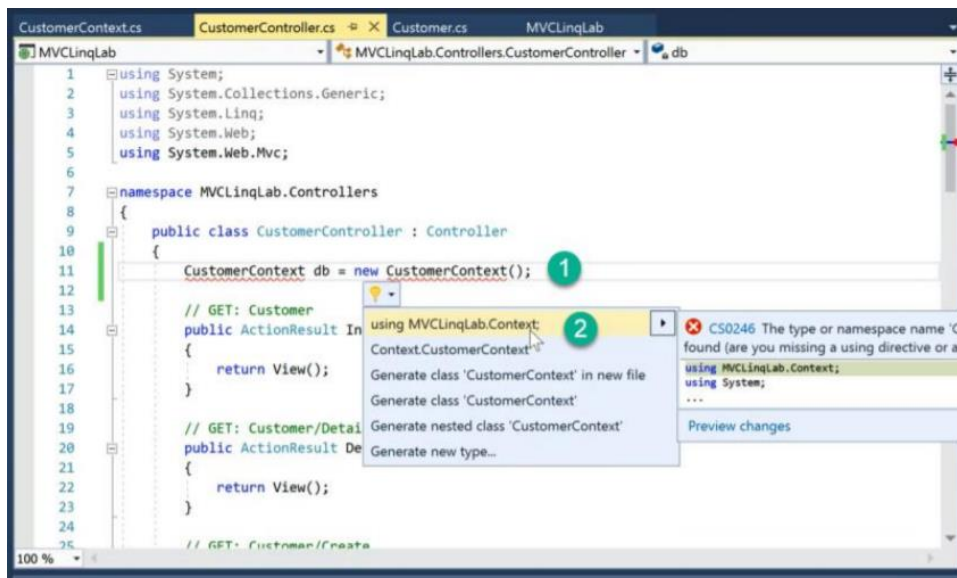
13. Download Package ดังภาพ



14. เลือก EntityFramework ดังภาพ เพื่อทำการ Add ไป install ที่โปรเจค



15. สร้าง Instant เพื่อเรียกใช้ Class หากมีการ Warning ให้เลือก “Using xxxx.Context;” เพื่อเรียกใช้ Class Context ในไฟล์ Customer Controller



เพิ่มโค้ด ส่วน Index () หน้า CustomerController.cs

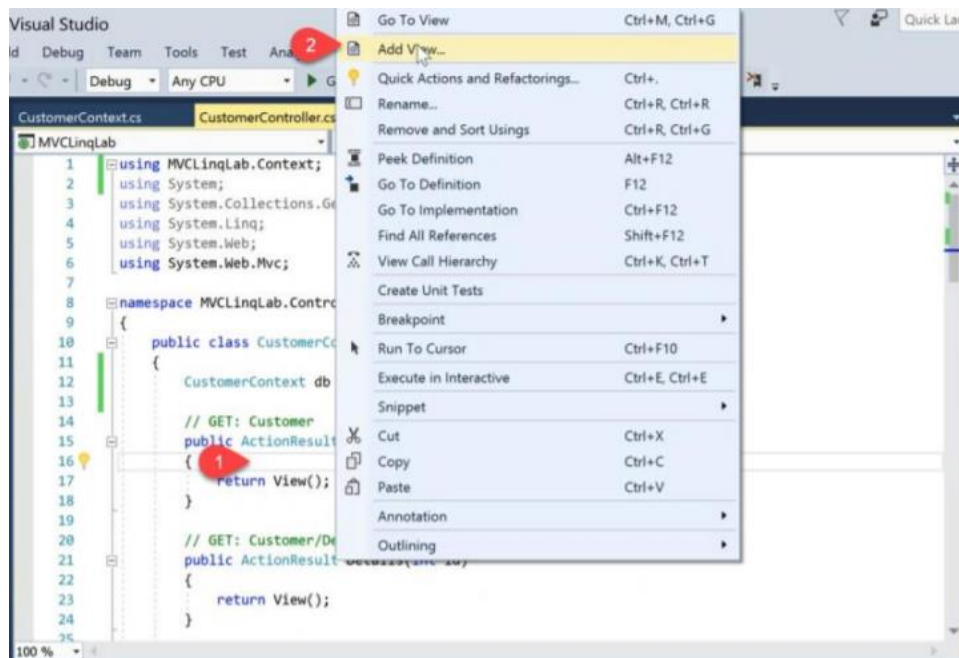
```

9 namespace MVCLinqLab.Controllers
10 {
11     public class CustomerController : Controller
12     {
13         CustomerContext db = new CustomerContext();
14
15         // GET: Customer
16         public ActionResult Index()
17         {
18             return View(db.Customer.ToList());
19         }
20
21         // GET: Customer/Details/5
22         public ActionResult Details(int id)
23         {
24             return View();
25         }
26     }

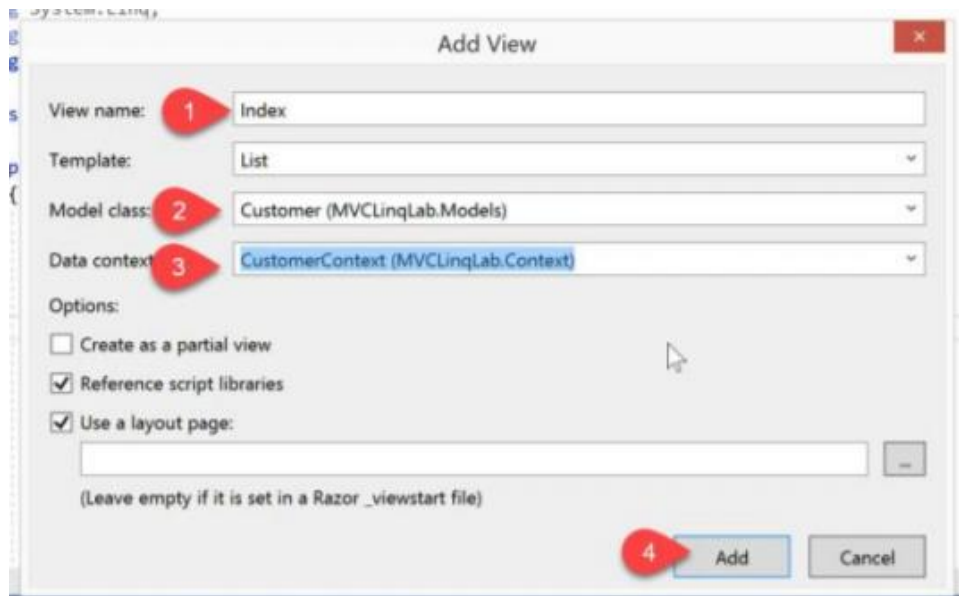
```

16. อาจต้องทำการ Rebuild ก่อนที่จะทำการทดลอง Run

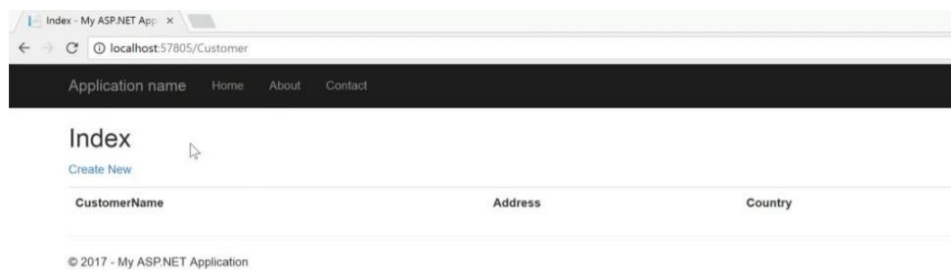
17. ที่ไฟล์ CustomerController Click ขวาที่ Method Index เพื่อทำการสร้างหน้า View สำหรับการแสดงผลให้กับหน้าแรกของข้อมูล Customer ไฟล์ต่าง ๆ ใน View จะถูกสร้างขึ้นโดยอัตโนมัติ



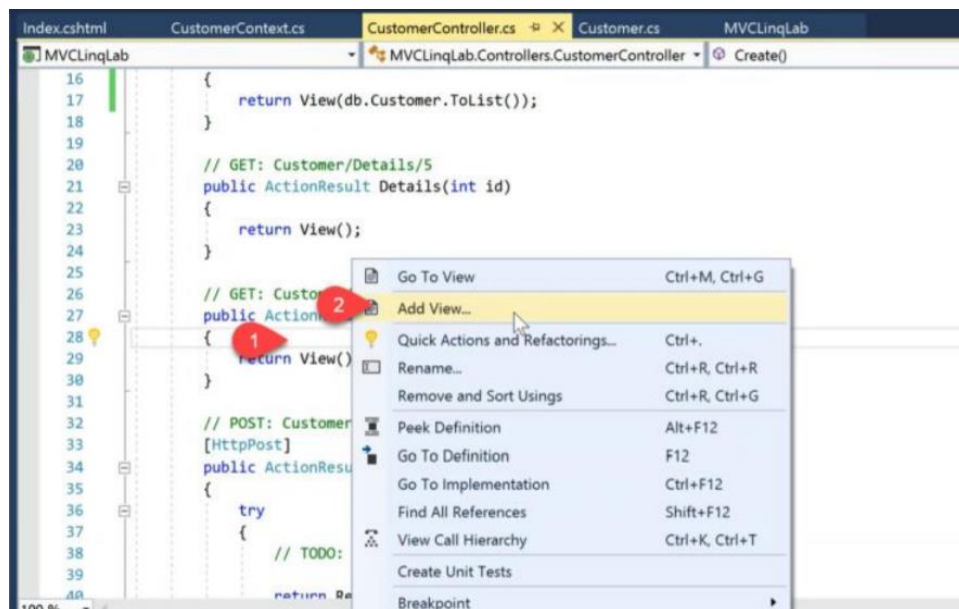
18. เลือก List และ Data Entity Context ดังภาพตัวอย่าง



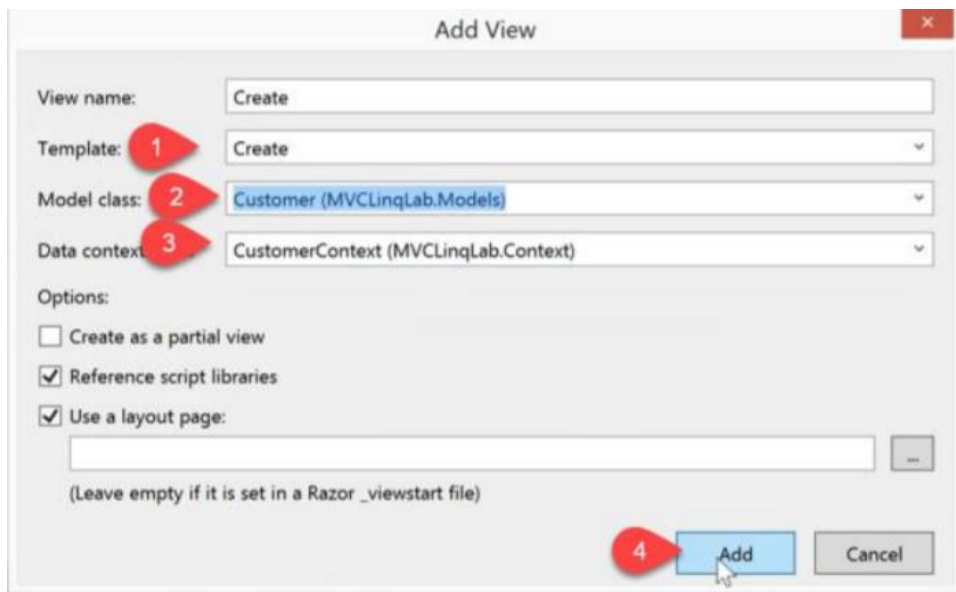
19. ลองทำการ Run โปรแกรมเพื่อดูหน้า Index ของ Customer จะปรากฏเป็นหน้าว่างเปล่าที่ซึ่งไม่มีข้อมูล



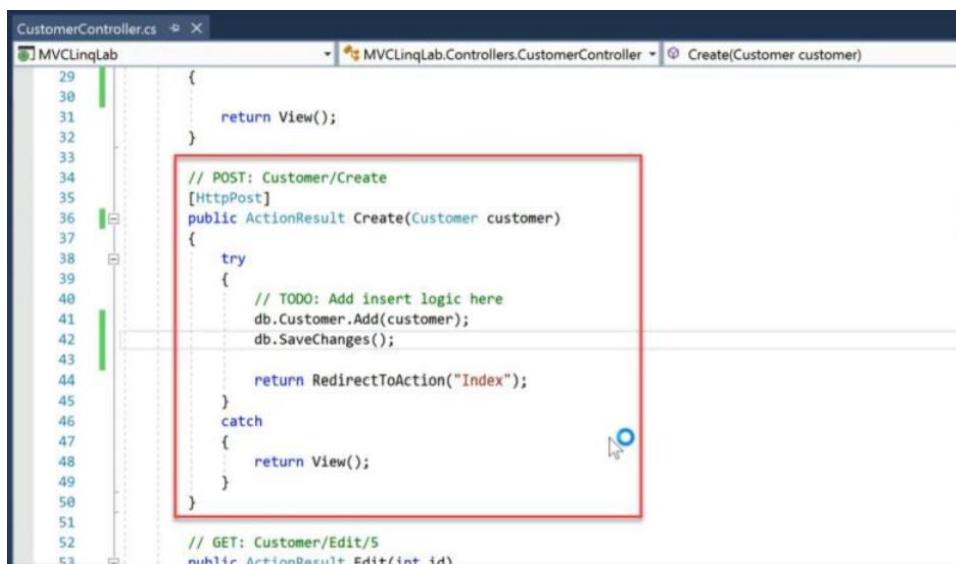
20. ทำการสร้าง View ให้กับ Method Create โดยการ Click ขาวและ Add View



21. เลือก Data Entity, Model, Data Context ดังภาพตัวอย่างที่กำหนด

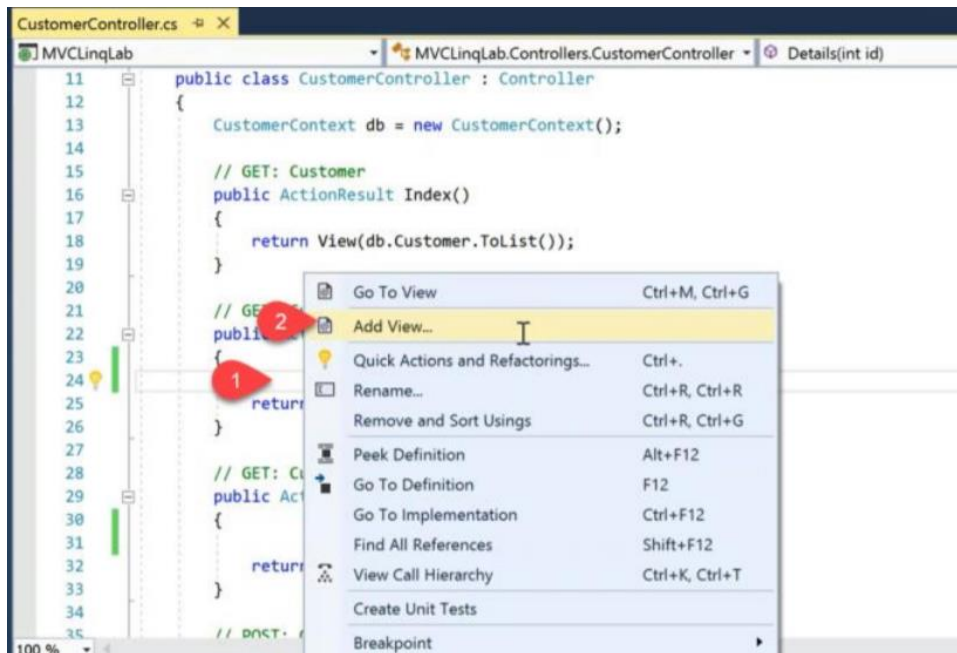


22. เพิ่ม Code ในส่วนของ Method Create ดังภาพ

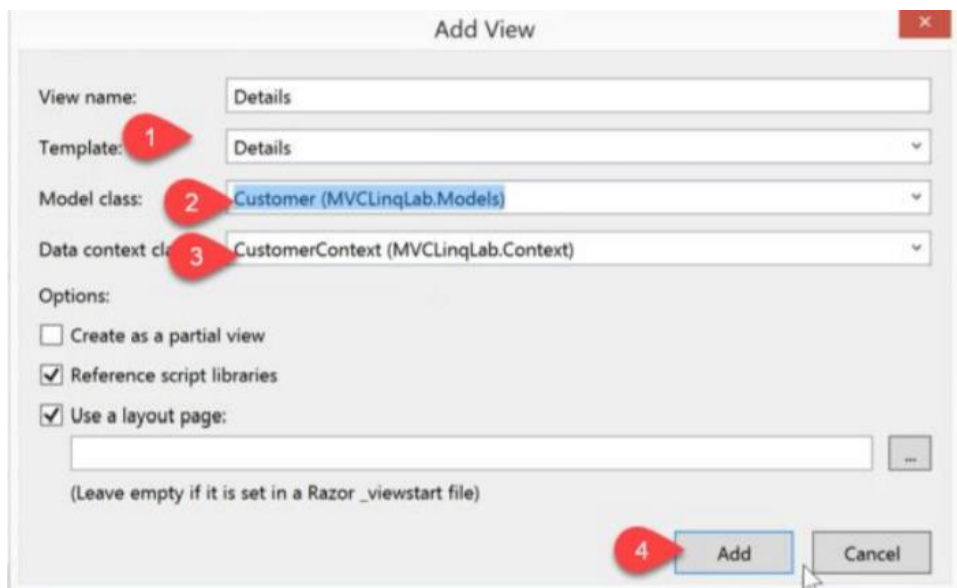


23. ลอง Run โปรแกรมจะได้หน้าที่กำหนด ลองกด “Create New” เพื่อทำการสร้างข้อมูลตัวอย่าง

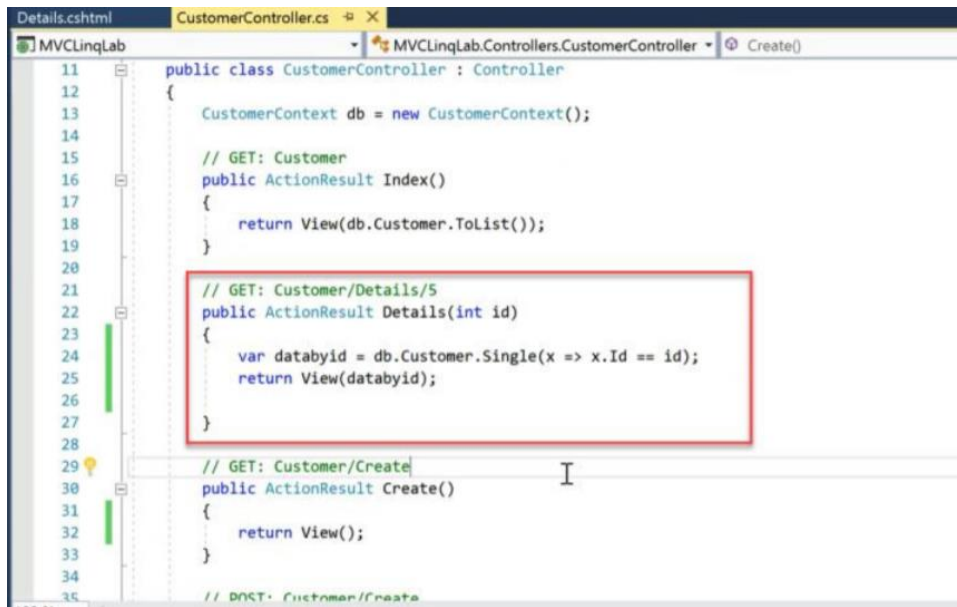
24. เพิ่ม View ในหน้า Details ในการเรียกดูรายละเอียดของข้อมูล



25. เลือกข้อมูลตามรายละเอียดที่กำหนด



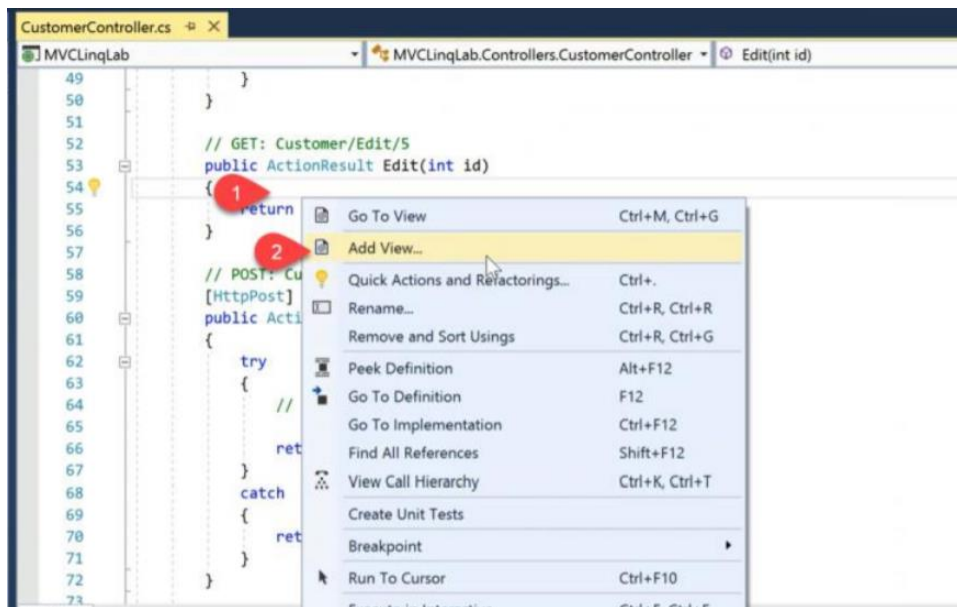
เพิ่ม Code ที่จำเป็นดังภาพ ใน Method Action Result เพื่อทำการ Select ข้อมูลจาก Id ที่เลือก



```
11 public class CustomerController : Controller
12 {
13     CustomerContext db = new CustomerContext();
14
15     // GET: Customer
16     public ActionResult Index()
17     {
18         return View(db.Customer.ToList());
19     }
20
21     // GET: Customer/Details/5
22     public ActionResult Details(int id)
23     {
24         var databyid = db.Customer.Single(x => x.Id == id);
25         return View(databyid);
26     }
27
28     // GET: Customer/Create
29     public ActionResult Create()
30     {
31         return View();
32     }
33
34     // POST: Customer/Create
35 }
```

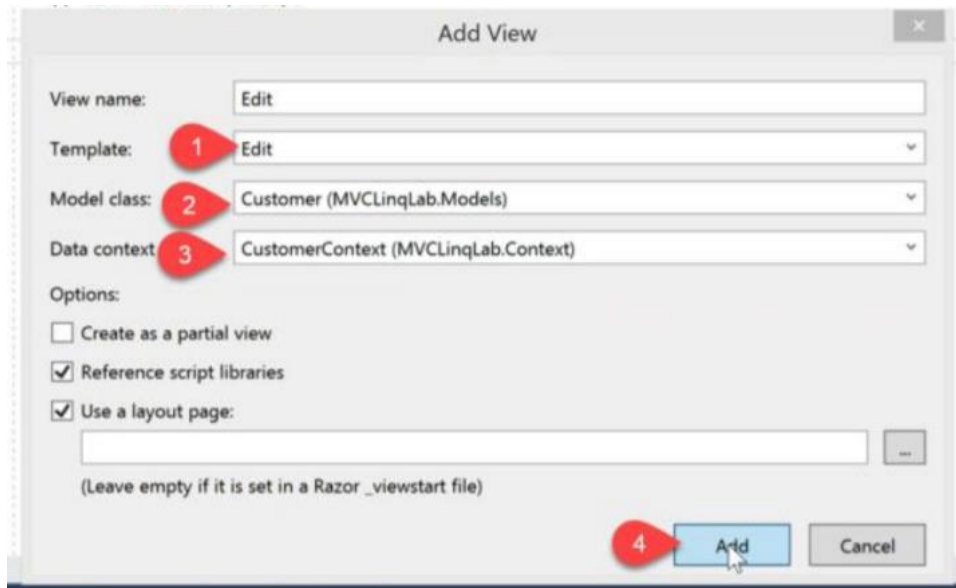
ลอง Run โปรแกรมและกดดูตัวอย่าง Details

26. ทำการเพิ่ม View ในส่วนของ Method Edit โดยการ คลิกขวาที่ Method Edit เลือก Add View...



```
49 }
50 }
51
52 // GET: Customer/Edit/5
53 public ActionResult Edit(int id)
54 {
55     return
56 }
57
58 // POST: Customer/Edit/5
59 [HttpPost]
60 public ActionResult Edit(int id)
61 {
62     try
63     {
64         //
65     }
66     return
67 }
68 catch
69 {
70     return
71 }
72 }
73 }
```

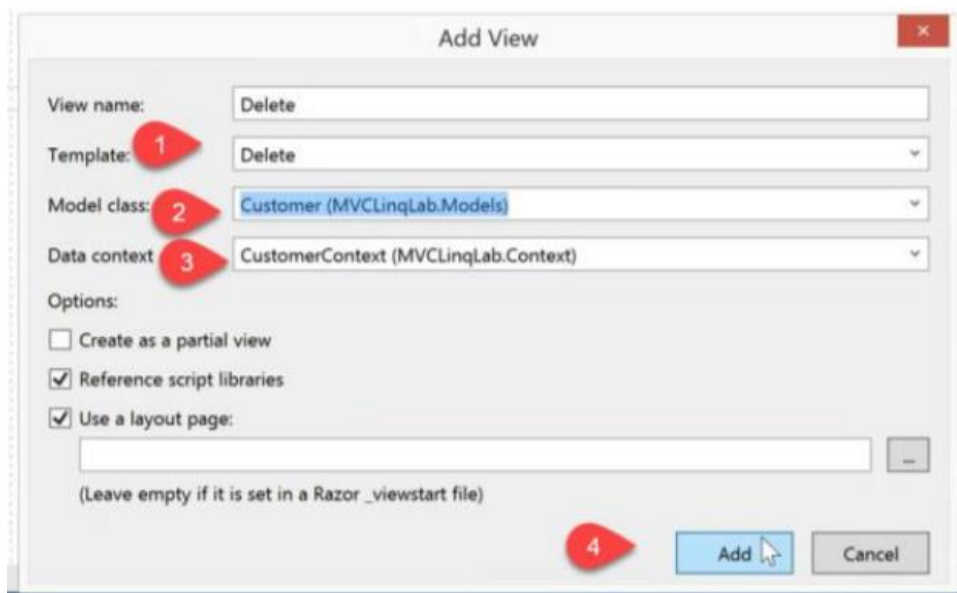
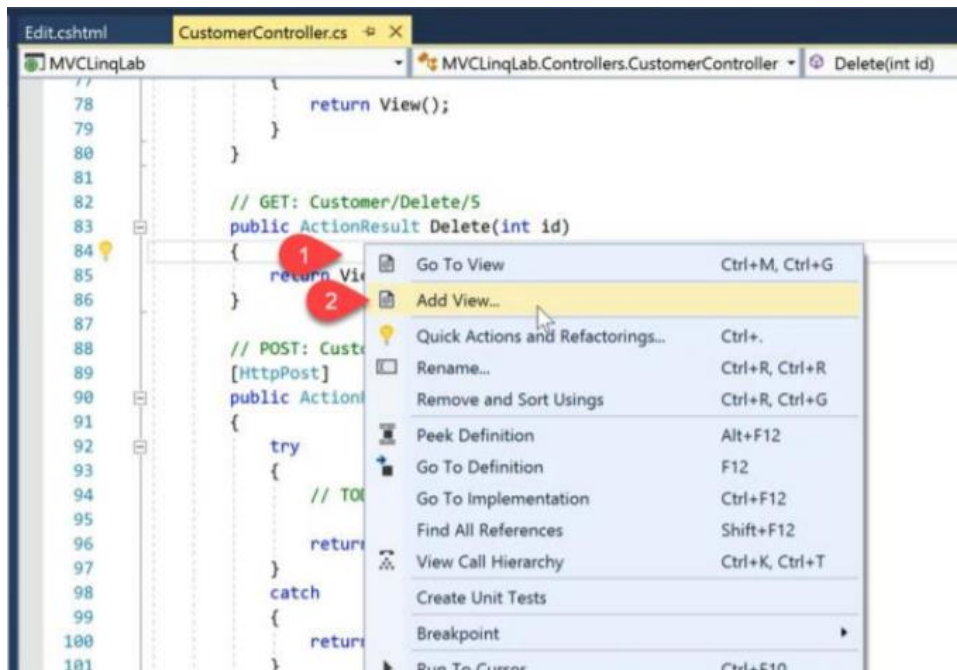
27. เลือกข้อมูลต่าง ๆ ดังภาพที่กำหนด



28. เพิ่ม Code ที่จำเป็นสำหรับการ Edit

```
CustomerContext.cs  Customer.cs  CustomerController.cs  Edit.cshtml  Details.cshtml  Delete.cshtml  Index.cshtml
MVCLinqLab
51
52 // GET: Customer/Edit/5
53 public ActionResult Edit(int id)
54 {
55     return View();
56 }
57
58 // POST: Customer/Edit/5
59 [HttpPost]
60 public ActionResult Edit(int id, Customer collection)
61 {
62     try
63     {
64         // TODO: Add update logic here
65         Customer cusObj = db.Customer.Single(x => x.Id == id);
66
67         cusObj.Id = collection.Id;
68         cusObj.CustomerName = collection.CustomerName;
69         cusObj.Address = collection.Address;
70         cusObj.Country = collection.Country;
71
72         db.SaveChanges();
73
74         return RedirectToAction("Index");
75     }
76     catch
77     {
78         return View();
79     }
80 }
81
82 // GET: Customer/Delete/5
83 public ActionResult Delete(int id)
```

28. เพิ่ม View ใน Method Delete เพื่อทำการดูข้อมูลก่อนการตัดสินใจลบ



เพิ่ม Code ที่สำคัญในส่วนของ Delete ดังภาพ

```
CustomerContext.cs Customer.cs CustomerController.cs Edit.cshtml Details.cshtml Delete.cshtml
MVCLinqLab MVCLinqLab.Controllers.CustomerController
78         return View();
79     }
80 }
81
82 // GET: Customer/Delete/5
83 public ActionResult Delete(int id)
84 {
85     var dataid = db.Customer.Single(x => x.Id == id);
86     return View();
87 }
88
89 // POST: Customer/Delete/5
90 [HttpPost]
91 public ActionResult Delete(int id, FormCollection collection)
92 {
93     try
94     {
95         // TODO: Add delete logic here
96         var customerdataid = db.Customer.Single(x => x.Id == id);
97         db.Customer.Remove(customerdataid);
98         db.SaveChanges();
99
100        return RedirectToAction("Index");
101    }
102    catch
103    {
104        return View();
105    }
106 }
107
108 }
```

เพิ่มลิงก์ `@Html.ActionLink("Customer", "Index", "Customer")`

(ไฟล์ `_layout.cshtml` ที่ `...Views/Shared/`)

```
_Layout.cshtml CustomerContext.cs Customer.cs CustomerController.cs Edit.cshtml Details.cshtml Delete.cshtml Index.cshtml
4 <meta charset="utf-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>@ViewBag.Title - My ASP.NET Application</title>
7 @Styles.Render("~/Content/css")
8 @Scripts.Render("~/bundles/modernizr")
9
10 </head>
11 <body>
12 <div class="navbar navbar-inverse navbar-fixed-top">
13 <div class="container">
14 <div class="navbar-header">
15 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
16 <span class="icon-bar"></span>
17 <span class="icon-bar"></span>
18 <span class="icon-bar"></span>
19 </button>
20 @Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
21 </div>
22 <div class="navbar-collapse collapse">
23 <ul class="nav navbar-nav">
24 <li>@Html.ActionLink("Home", "Index", "Home")</li>
25 <li>@Html.ActionLink("About", "About", "Home")</li>
26 <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
27 <li>@Html.ActionLink("Customer", "Index", "Customer")</li>
28 </ul>
29 @Html.Partial("_LoginPartial")
30 </div>
31 </div>
32 </div>
33 <div class="container body-content">
34 @RenderBody()
35 </div>
```