

មេរៀនទី ៥

អនុវត្តជា ASP.NET និង Ms. SQL Server

I. អំពី ADO.NET

ADO.NET មកពីពាក្យថា ActiveX Data Object ជាបច្ចេកវិទ្យាមួយដែលត្រូវបានបង្កើតឡើងដោយក្រុមហ៊ុន Microsoft ហើយវាក៏ជាផ្នែកមួយនៃ .Net Framework ដែលគេប្រើដើម្បីចូលទៅប្រើប្រាស់ជាតាមបេស (ទាញ, បញ្ចូល, លុបនិងធ្វើបច្ចុប្បន្នទិន្នន័យ) និងអាចទ្រទ្រង់ប្រភេទជាតិបេសជាច្រើនប្រភេទ។ ហើយវាជាបណ្តុំនៃ object-oriented classes ជាច្រើនដែលវាផ្តល់នូវសំនុំនៃសមាសធាតុទិន្នន័យដ៏សម្បូរបែបដើម្បីបង្កើតកម្មវិធីគ្រប់គ្រងទិន្នន័យប្រកបដោយ high-performance, ទុកចិត្តបាននិងពង្រីកនូវមុខងារ (ADO.NET stands for ActiveX Data Object is a database access technology created by Microsoft as part of its .NET framework that can access any kind of data source. It's a set of object-oriented classes that provides a rich set of data components to create high-performance, reliable and scalable database applications for client- server applications as well as distributed environments over the Internet and intranets.)¹

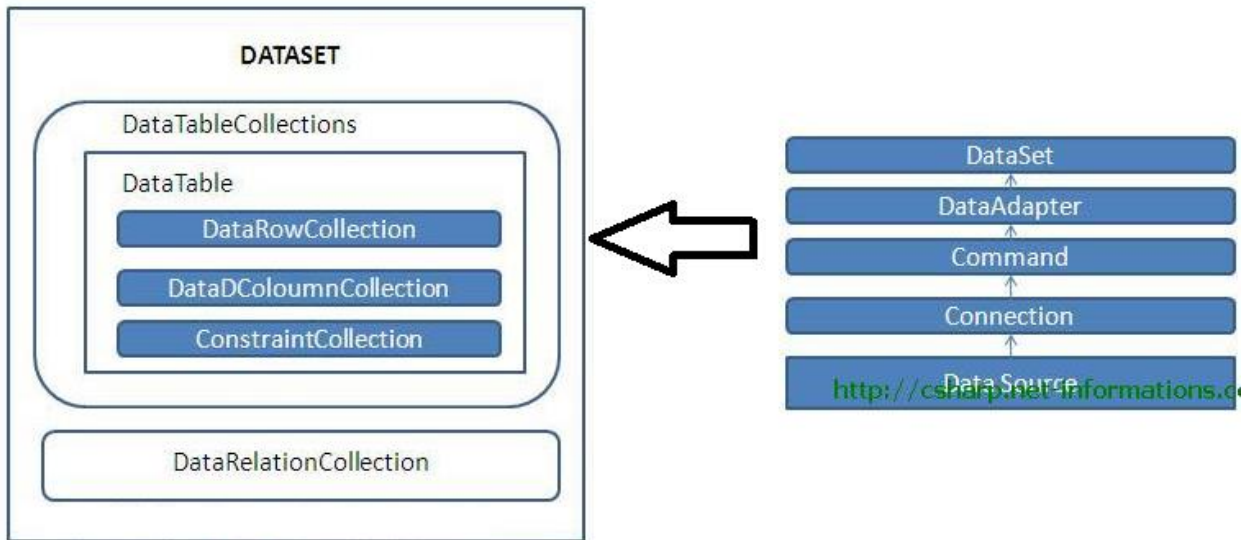
1. សមាសធាតុរបស់ ADO.NET ជាពីរគឺ Data Provider និង DataSet

- Data Provider

Data Provider	ការពណ៌នា
Connection	ជាស្ថានសម្រាប់ភ្ជាប់ទំនាក់ទំនងរវាង command object ជាមួយនិង data source ¹
Command	សម្រាប់ execute នូវ SQL statement និង store procedure
DataAdapter	ជាអ្នកសម្របសម្រួលរវាង datasource ជាមួយ DataSet ដែលវាមានតួនាទីក្នុងការទាញទិន្នន័យពី data source ចូលទៅ dataset (memory)
DataReader	សម្រាប់អានទិន្នន័យពី data source ក្នុងទំរង់ជា readonly និង forward only

- Data Set

វាជាបណ្តុំនៃ DataTable objects ជាច្រើនដែលវាមានតួនាទីរក្សាទុកទិន្នន័យពី data source នៅក្នុង memory ក្នុងទំរង់ read-write¹



2. Store Procedure

គឺជាដូចក្នុងគ្នានិង user-define functionដែលជាបណ្តុំនៃ SQL (structure query language) statementជាច្រើនក្នុងមួយឯកតាដែលត្រូវរក្សាទុកក្នុង database management systemដែលគេអាចហៅមកប្រើម្តងហើយម្តងទៀតបាន។

- ការបង្កើត store procedure

រូបមន្ត

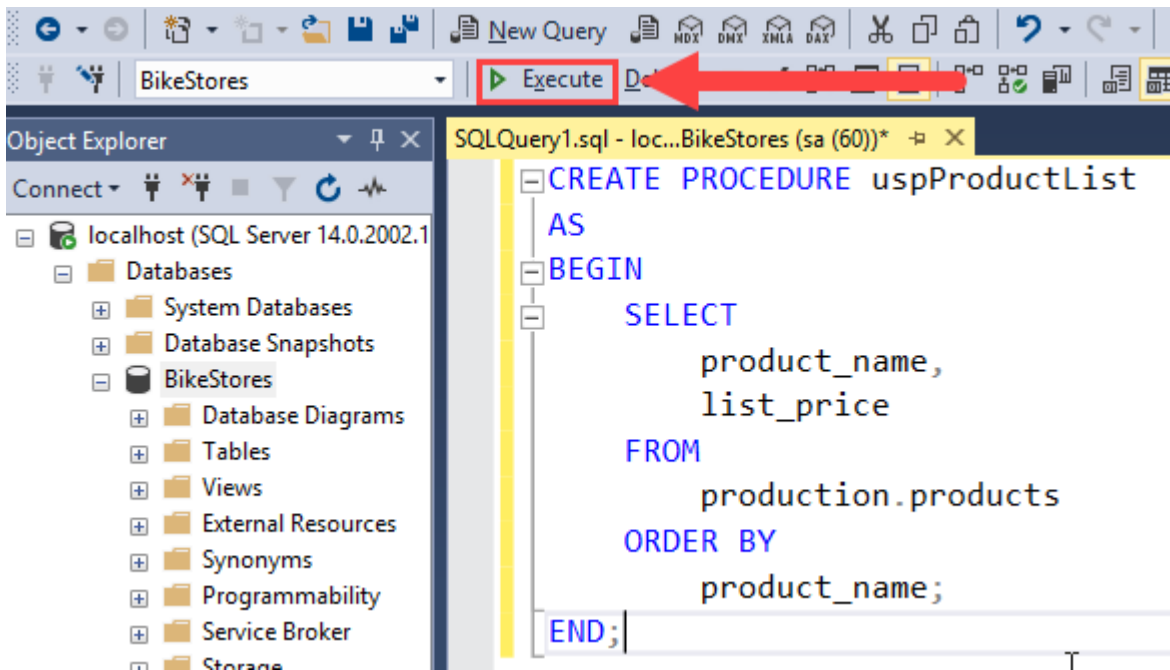
```

CREATE PROC/PROCEDURE ProcureName(<paramenter list>)
AS
Variables here
BEGIN
    SQL statement here;
END
    
```

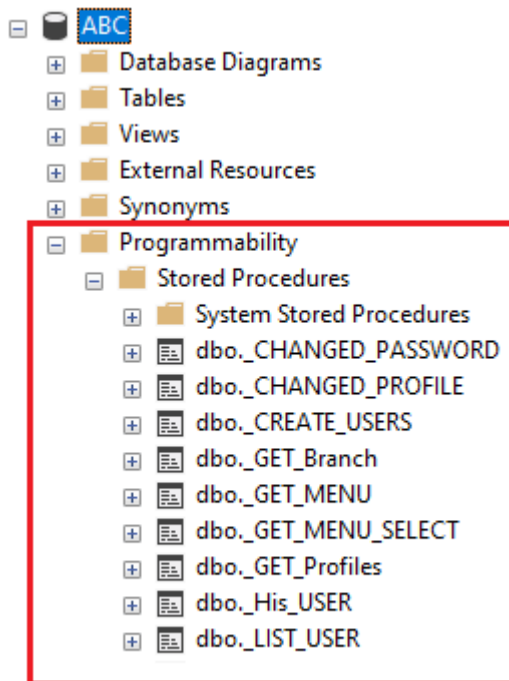
ឧទាហរណ៍៖

- បើក Microsoft SQL Server studio → ជ្រើសរើស database → ចុចមើលស្តាំលើ database → new query នោះវានឹងបង្ហាញផ្ទាំងដូចខាងក្រោម៖
- ក្រោយពីសរសេរ sql statement រួចហើយចុចលើexecute button តាមរូប

សកលវិទ្យាល័យបៀលប្រាស



- គ្រប់ store procedures វា នៅក្នុង database name → programmability → store procedure



3. Trigger

Database trigger ជា store procedure code ពីសេសមួយដែលត្រូវបានគំរើការឡើងដោយស្វ័យប្រវត្តិដើម្បីឆ្លើយតបទៅនឹងព្រឹត្តិការណ៍ពិតណាមួយ (insert, update, delete) នៅលើ view ឬ table ក្នុង database ។

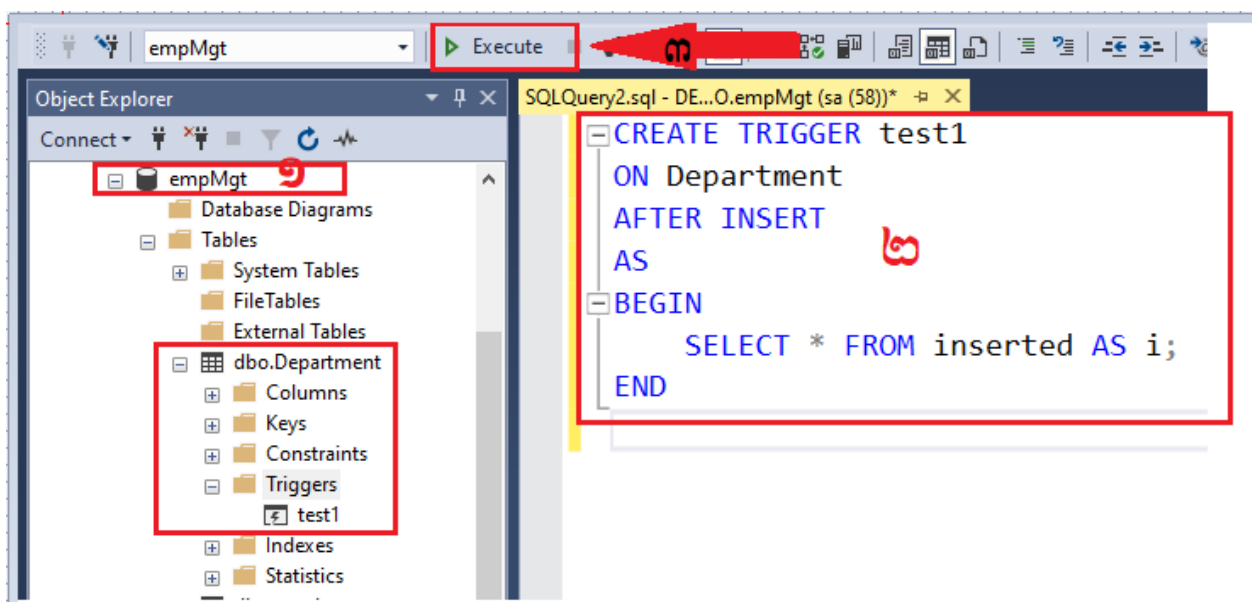
សកលវិទ្យាល័យបៀលប្រាស

Syntax:

```
CREATE TRIGGER triggerName ON tableName
{ AFTER | BEFORE } { INSERT | DELETE | UPDATE }
AS
BEGIN
    --SQL Statement here;
END
```

ឧទាហរណ៍៖

- បើក Microsoft SQL Server studio → ជ្រើសរើស database → ចុចមើល database → new query នោះវានឹងបង្ហាញផ្ទាំងដូចខាងក្រោម៖
- ក្រោយពីសរសេរ sql statement រួចហើយចុចលើ execute button តាមរូប

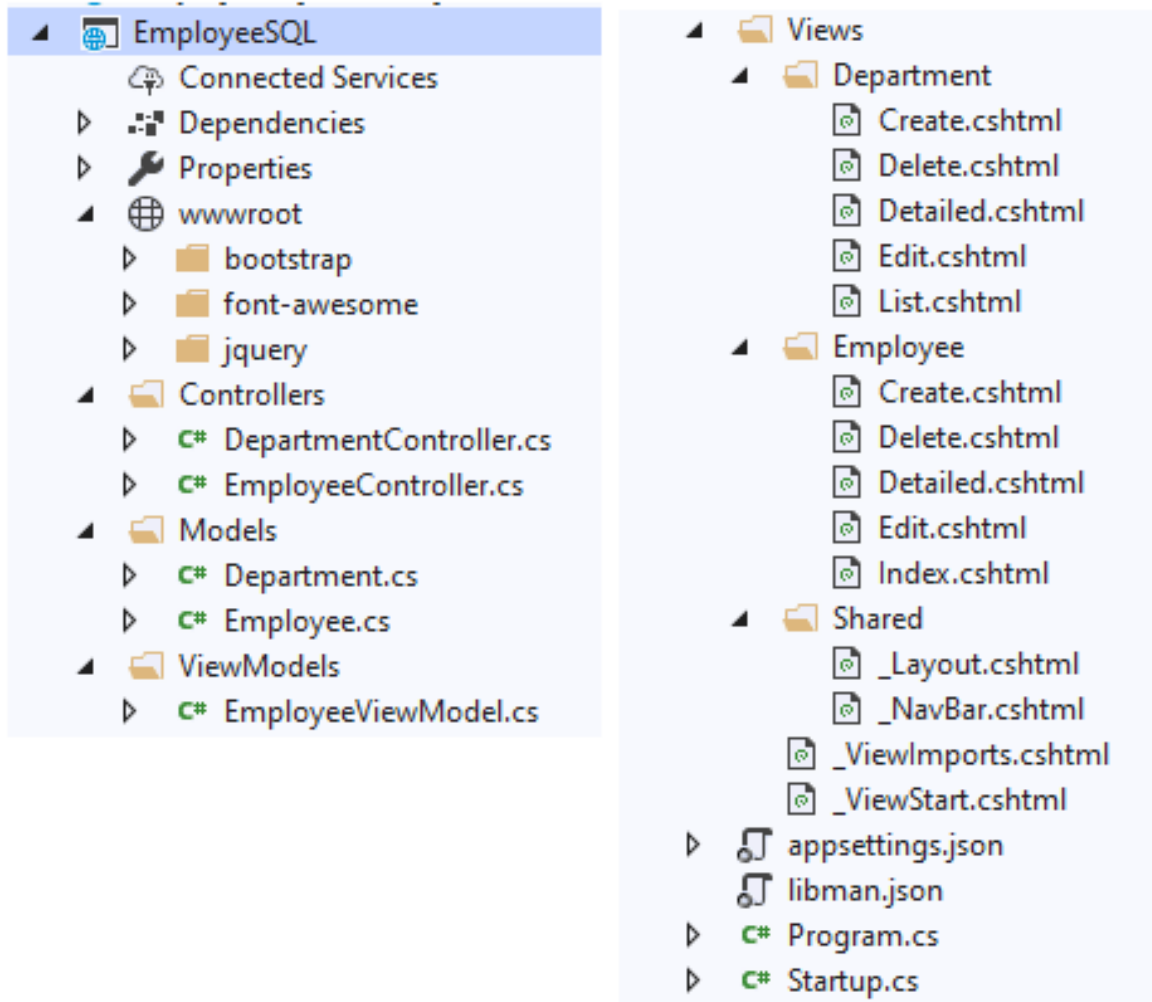


II. ASP.NET និង ADO.NET(Ms. SQL Server)

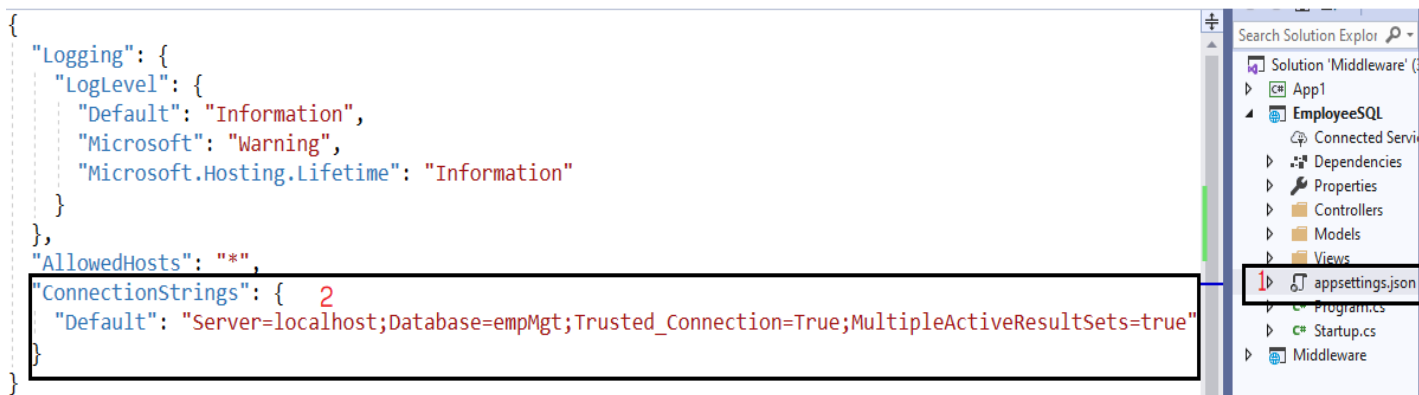
1. ការប្រើប្រាស់ជាមួយ SQL Statement

- សូមបង្កើត Project ដែលមានឈ្មោះ និងទំរង់ខាងក្រោម៖

សកលវិទ្យាល័យបៀលប្រាស

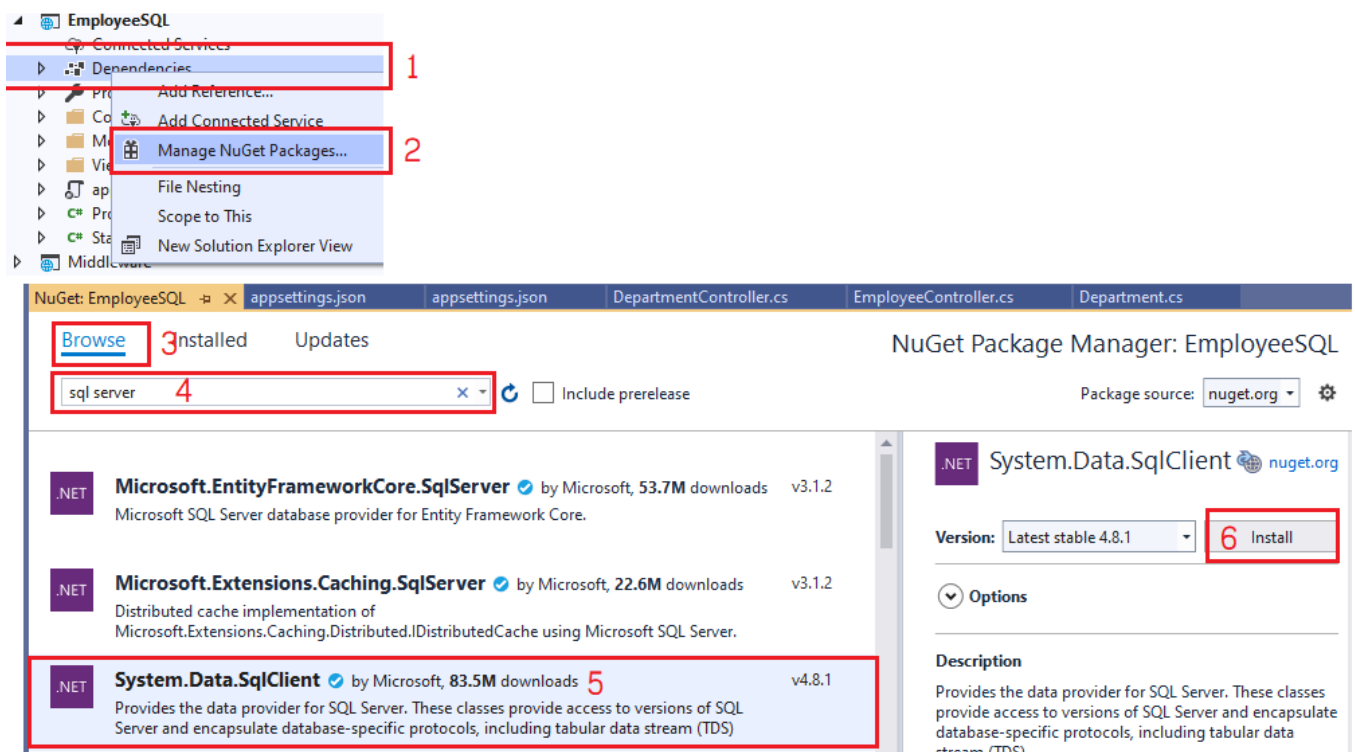


បង្កើត connection string → appsettings.json → សរសេរដូចរូបខាងក្រោម៖

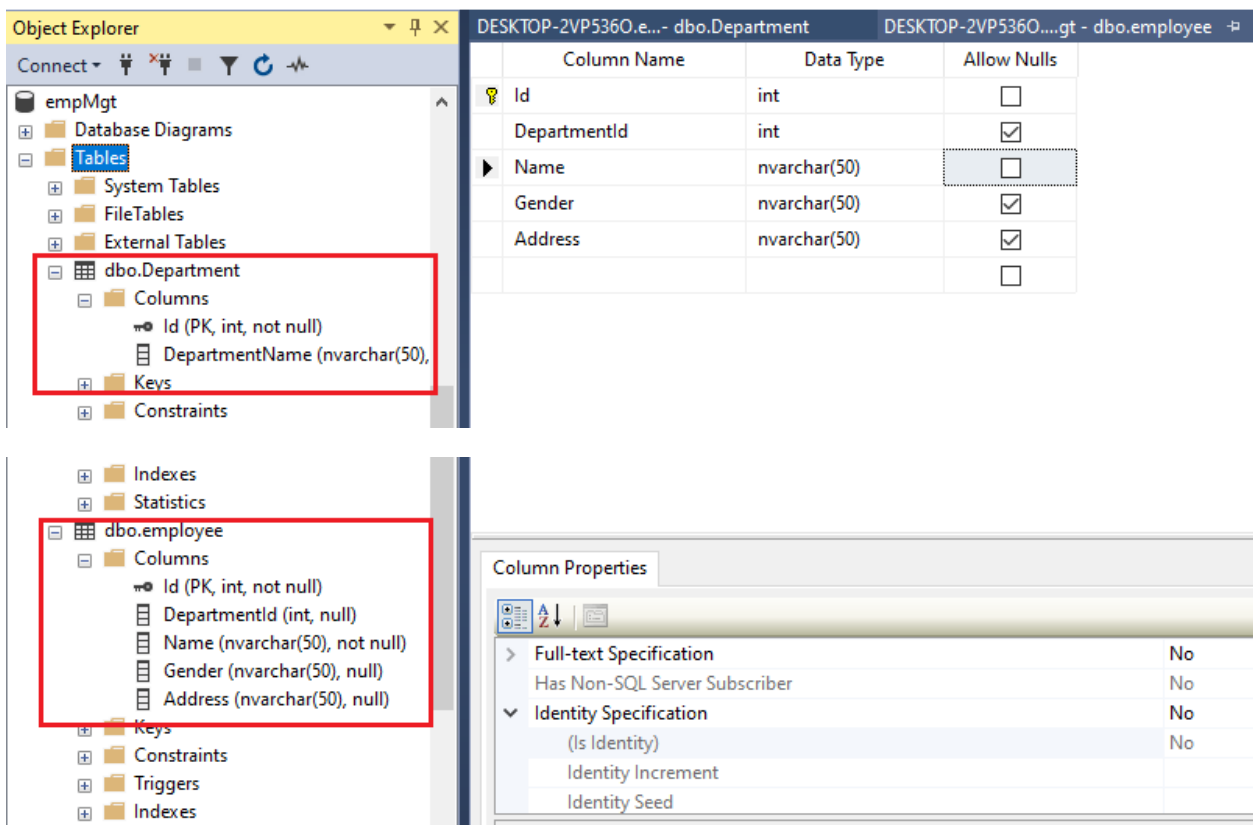


បន្ថែម package ចូលក្នុង project តាមរូបខាងក្រោម៖ ដោយ right click លើ dependency

សកលវិទ្យាល័យបៀលប្រាស



បង្កើត database ដូចរូបខាងក្រោម (empMgt) និងតារាងចំនួនពីរគឺ employee និង department:



→ Models Directory

Department.cs

```
using System.ComponentModel.DataAnnotations;

namespace EmployeeSQL.Models
{
    public class Department
    {
        public int Id { get; set; }

        [Required]
        [Display(Name = "Department")]
        public string DepartmentName { get; set; }
    }
}
```

Employee.cs

```
using System.ComponentModel.DataAnnotations;

namespace EmployeeSQLEntity.Models
{
    public class Employee
    {
        public int Id { get; set; }
        [Required]
        public string Name { get; set; }
        [Required]
        public string Gender { get; set; }
        [Required]
        [Display(Name = "Department")]
        public int DepartmentId { get; set; }
        [Required]
        public string Address { get; set; }
    }
}
```

→ViewModel Directory

EmployeeViewModel.cs

```
namespace EmployeeSQL.ViewModels
{
    public class EmployeeViewModel
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Gender { get; set; }
        public string Department { get; set; }
        public string Address { get; set; }
    }
}
```

```
}
```

→Controllers directory

DepartmentController.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using EmployeeSQL.Models;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Configuration;

namespace EmployeeSQL.Controllers
{
    public class DepartmentController : Controller
    {
        private readonly IConfiguration _config;
        public DepartmentController(IConfiguration configuration)
        {
            this._config = configuration;
        }
        [ActionName("List")]
        public IActionResult Index()
        {
            using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
            string sql = "SELECT * FROM Department ORDER BY
DepartmentName DESC";
            using SqlCommand command = new SqlCommand(sql, connection);
            connection.Open();
            try
            {
                using SqlDataAdapter da = new SqlDataAdapter(command);
                DataTable table = new DataTable();
                da.Fill(table);
                var deps = (from t in table.AsEnumerable()
                    select new Department
                    {
                        Id = t.Field<int>(0),
                        DepartmentName = t.Field<string>(1)
                    }).ToList();

                return View("list", deps);
            }
            catch (Exception ex)
            {
                throw ex;
            }
        }
    }
}
```



```
    }

    }
    public IActionResult Create()
    {
        return View();
    }

    [HttpPost]
    public IActionResult store([Bind("DepartmentName")] Department
dep)
    {
        if (ModelState.IsValid)
        {
            using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
            string sql = "INSERT INTO department(departmentname)
VALUES(@name)";
            using SqlCommand command = new SqlCommand(sql,
connection);
            connection.Open();
            command.Parameters.AddWithValue("@name",
dep.DepartmentName);
            try
            {
                command.ExecuteNonQuery();
                return RedirectToAction("List");
            }
            catch (Exception ex)
            {
                throw ex;
            }
        }
        return View("create", dep);
    }

    public IActionResult Edit(int? id)
    {
        using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
        string sql = "SELECT * FROM department WHERE Id=@id;";
        using SqlCommand command = new SqlCommand(sql, connection);
        connection.Open();
        command.Parameters.AddWithValue("@id", id.Value);
        try
        {
            using SqlDataAdapter da = new SqlDataAdapter(command);
            DataTable table = new DataTable();
            da.Fill(table);
        }
    }
}
```

```
var dep = (from t in table.AsEnumerable()
            select new Department
            {
                Id = t.Field<int>(0),
                DepartmentName = t.Field<string>(1)
            }).FirstOrDefault();
if (dep != null)
    return View("edit", dep);
else
    return NotFound();

}
catch (Exception ex)
{
    throw ex;
}

}

[HttpPost]
public IActionResult Update([Bind("Id", "DepartmentName")]
Department dep)
{
    if (ModelState.IsValid)
    {
        using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
        string sql = "UPDATE department SET departmentname=@name
WHERE Id=@id;";
        using SqlCommand command = new SqlCommand(sql,
connection);
        connection.Open();
        command.Parameters.AddWithValue("@id", dep.Id);
        command.Parameters.AddWithValue("@name",
dep.DepartmentName);
        try
        {
            command.ExecuteNonQuery();
            return RedirectToAction("List");
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
    return View("create", dep);
}

public IActionResult Destroy(int? id)
{

```

```
        using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
        string sql = "SELECT * FROM department WHERE Id=@id;";
        using SqlCommand command = new SqlCommand(sql, connection);
        connection.Open();
        command.Parameters.AddWithValue("@id", id.Value);
        try
        {
            using SqlDataAdapter da = new SqlDataAdapter(command);
            DataTable table = new DataTable();
            da.Fill(table);
            var dep = (from t in table.AsEnumerable()
                       select new Department
                       {
                           Id = t.Field<int>(0),
                           DepartmentName = t.Field<string>(1)
                       }).FirstOrDefault();
            if (dep != null)
                return View("delete", dep);
            else
                return NotFound();
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

    [HttpPost]
    public IActionResult Delete(int? id)
    {
        using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
        string sql = "DELETE FROM department WHERE Id=@id;";
        using SqlCommand command = new SqlCommand(sql, connection);
        connection.Open();
        command.Parameters.AddWithValue("@id", id.Value);
        try
        {
            var rows = command.ExecuteNonQuery();
            if (rows > 0)
                return RedirectToAction("List");
            else
                return NotFound();
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
}
```

```
        throw ex;
    }
}

public IActionResult Detailed(int? id)
{
    using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
    string sql = "SELECT * FROM department WHERE Id=@id;";
    using SqlCommand command = new SqlCommand(sql, connection);
    connection.Open();
    command.Parameters.AddWithValue("@id", id.Value);
    try
    {
        using SqlDataAdapter da = new SqlDataAdapter(command);
        DataTable table = new DataTable();
        da.Fill(table);
        var dep = (from t in table.AsEnumerable()
                    select new Department
                    {
                        Id = t.Field<int>(0),
                        DepartmentName = t.Field<string>(1)
                    }).FirstOrDefault();
        if (dep != null)
            return View("detailed", dep);
        else
            return NotFound();
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
}
```

EmployeeController.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using EmployeeSQL.Models;
using EmployeeSQL.ViewModels;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
```

សកលវិទ្យាល័យបៀលប្រាស

```
using Microsoft.Extensions.Configuration;

namespace EmployeeSQL.Controllers
{
    public class EmployeeController : Controller
    {
        private readonly IConfiguration _config;

        public EmployeeController(IConfiguration config)
        {
            this._config = config;
        }

        public IActionResult Index()
        {
            try
            {
                var emps = Gets();
                return View("Index", emps);
            }
            catch (Exception ex)
            {
                throw ex;
            }
        }

        [HttpPost]
        public IActionResult store(Employee emp)
        {
            try
            {
                using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
                string sql = "INSERT INTO
employee(name,departmentId,gender,address)";
                sql += " VALUES(@name,@dep,@gender,@address)";
                using SqlCommand command = new SqlCommand(sql,
connection);
                connection.Open();
                command.Parameters.AddWithValue("@name", emp.Name);
                command.Parameters.AddWithValue("@dep",
emp.DepartmentId);
                command.Parameters.AddWithValue("@gender", emp.Gender);
                command.Parameters.AddWithValue("@address",
emp.Address);
                if (ModelState.IsValid)
                {
                    if (command.ExecuteNonQuery() > 0){
                        return RedirectToAction("Index");
                    }
                }
            }
        }
    }
}
```

```
        else
        {
            throw new Exception();
        }
    }
    ViewData["Department"] = DepartmentList();
    return View("Create", emp);
}
catch (Exception ex)
{
    throw ex;
}
}
```

```
[HttpGet]
public IActionResult Create()
{
    try
    {
        ViewData["Department"] = DepartmentList();
        return View();
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

```
public IActionResult Edit(int? id)
{
    try
    {
        var emp = GetEmployeeList(id);
        ViewBag.Departments = DepartmentList();
        return View("Edit", emp);
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

```
[HttpPost]
public IActionResult Update(Employee emp)
{
    try
    {
```

```
        using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
        string sql = "UPDATE employee SET
name=@name,departmentId=@dep,gender=@gender,address=@address";
        sql += " WHERE Id=@Id";
        using SqlCommand command = new SqlCommand(sql,
connection);
        connection.Open();
        command.Parameters.AddWithValue("@Id", emp.Id);
        command.Parameters.AddWithValue("@name", emp.Name);
        command.Parameters.AddWithValue("@dep",
emp.DepartmentId);
        command.Parameters.AddWithValue("@gender", emp.Gender);
        command.Parameters.AddWithValue("@address",
emp.Address);
        if (ModelState.IsValid)
        {
            if (command.ExecuteNonQuery() > 0)
            {
                return RedirectToAction("Index");
            }
            else
            {
                throw new Exception();
            }
        }
        return View(emp);
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

public IActionResult Detailed(int? id)
{
    var emp = Gets().SingleOrDefault(emp => emp.Id == id.Value);
    if(emp != null)
    {
        return View("Detailed", emp);
    }
    return NotFound();
}
public IActionResult Delete(int? id)
{
    var emp = Gets().SingleOrDefault(emp => emp.Id == id.Value);
    if (emp != null)
    {
        return View("delete", emp);
    }
}
```

```
    }
    return NotFound();
}
public IActionResult Destroy(int? id)
{
    try
    {
        using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
        string sql = "DELETE FROM Employee WHERE Id=@Id";
        using SqlCommand command = new SqlCommand(sql,
connection);

        connection.Open();
        command.Parameters.AddWithValue("@Id", id.Value);
        if (command.ExecuteNonQuery() > 0)
        {
            return RedirectToAction("Index");
        }
        return NotFound();
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

public IActionResult deletes(int? id)
{
    var emp = Gets().SingleOrDefault(emp => emp.Id == id.Value);
    if (emp != null)
    {
        return View("deletes", emp);
    }
    return NotFound();
}

private List<SelectListItem> DepartmentList()
{
    try
    {
        using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
        string sql = "SELECT * FROM Department ORDER BY
DepartmentName DESC";
        using SqlCommand command = new SqlCommand(sql,
connection);

        connection.Open();
```



```
        using SqlDataAdapter da = new SqlDataAdapter(command);
        DataTable table = new DataTable();
        da.Fill(table);
        var deps = (from t in table.AsEnumerable()
                    select new SelectListItem
                    {
                        Value = t.Field<int>(0).ToString(),
                        Text = t.Field<string>(1)
                    }).ToList();

        return deps;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

private Employee GetEmployeeList(int? id)
{
    try
    {
        using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
        string sql = "SELECT * FROM employee WHERE Id=@id";
        using SqlCommand command = new SqlCommand(sql,
connection);

        connection.Open();
        command.Parameters.AddWithValue("@id", id);
        using SqlDataAdapter da = new SqlDataAdapter(command);
        DataTable table = new DataTable();
        da.Fill(table);
        var emp = (from t in table.AsEnumerable()
                    select new Employee
                    {
                        Id = t.Field<int>("Id"),
                        Name = t.Field<string>("Name"),
                        Gender = t.Field<string>("Gender"),
                        DepartmentId =
t.Field<int>("DepartmentId"),
                        Address = t.Field<string>("Address")
                    }).FirstOrDefault();

        return emp;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

សកលវិទ្យាល័យបៀលប្រាស

```
private List<EmployeeViewModel> Gets()
{
    try
    {
        using SqlConnection connection = new
SqlConnection(this._config.GetConnectionString("Default"));
        string sql = "SELECT * FROM employee INNER JOIN
department ON employee.departmentid=department.id ORDER BY employee.Id
DESC";
        using SqlCommand command = new SqlCommand(sql,
connection);
        connection.Open();
        using SqlDataAdapter da = new SqlDataAdapter(command);
        DataTable table = new DataTable();
        da.Fill(table);
        var emps = (from t in table.AsEnumerable()
                    select new EmployeeViewModel
                    {
                        Id = t.Field<int>("Id"),
                        Name = t.Field<string>("Name"),
                        Gender = t.Field<string>("Gender"),
                        Department =
t.Field<string>("DepartmentName"),
                        Address = t.Field<string>("Address")
                    }).ToList();
        return emps;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
}
```

➔ Views Directory

➔ Department folder

Create.cshtml

```
@model Department
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="store" asp-controller="department" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
        </div>
    </form>
</div>
```

សកលវិទ្យាល័យបៀលប្រាស

```

        </div>
        <div class="card-body">
            <div class="form-group row">
                <label asp-for="DepartmentName" class="col-form-label col-
md-2"></label>
                <div class="col-md-10">
                    <input asp-for="DepartmentName" class="form-control" />
                    <span asp-validation-for="DepartmentName" class="text-
danger"></span>
                </div>
            </div>
        </div>
        <div class="card-footer">
            <button type="submit" class="btn btn-primary">Create</button>
            <a asp-action="List" asp-controller="department">
                <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
            </a>
        </div>
    </div>
</form>
</div>

```

Detailed.cshtml

```

@model Department
@{
    ViewData["Title"] = "Detailed";
}

<div>
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(d => d.Id)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(d => d.Id)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(d => d.DepartmentName)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(d => d.DepartmentName)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="edit" asp-route-id="@Model.Id">Edit</a>
    |
    <a asp-action="List">Back to List</a>
</div>

```

សកលវិទ្យាល័យបៀលប្រាស

Delete.cshtml

```
@model Department
@{
    ViewData["Title"] = "Delete";
}
<div class="container mt-4">
    <div class="row">
        <div class="col col-sm-8 offset-2">
            <div class="alert alert-danger" role="alert">
                Do want to delete <strong>@Model.DepartmentName </strong>?
            </div>
            <div>
                <form asp-action="delete" asp-controller="department"
method="post">
                    <input type="hidden" asp-for="Id" />
                    <input type="submit" value="Delete" class="btn btn-danger"
/>

                    <a asp-action="List" asp-controller="department">
                        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back
To List
                </a>
            </form>
        </div>
    </div>
</div>
```

List.html

```
@model IEnumerable<Department>
@{
    ViewData["Title"] = "List";
}
<div class="card">
    <div class="card-header">
        <strong>All</strong>
        <a asp-action="create" asp-controller="department" class="float-
right">Create</a>
    </div>
    <div class="card-body">
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>@Html.DisplayNameFor(d=>d.DepartmentName)</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody>
                @if (Model.Count() > 0)
                {
                    @foreach (var dep in Model)
```

សកលវិទ្យាល័យបៀលប្រាស

```

        {
            <tr>
                <td>@dep.DepartmentName</td>
                <td>
                    <a asp-action="detailed" asp-controller="department"
asp-route-id="@dep.Id">
                        <i class="far fa-eye mr-1"></i>Detail
                    </a> |
                    <a asp-action="edit" asp-controller="department" asp-
route-id="@dep.Id">
                        <i class="far fa-edit mr-1"></i>Edit
                    </a> |
                    <a asp-action="destroy" asp-controller="department"
asp-route-id="@dep.Id">
                        <i class="far fa-trash-alt mr-1"></i>
                        Delete
                    </a>
                </td>
            </tr>
        }
    </tbody>
</table>

</div>
</div>

```

Edit.cshtml

```

@model Department
@{
    ViewData["Title"] = "Index";
}

@if (Model != null)
{
    <form asp-action="update" asp-controller="department">
        <div class="card">
            <div class="card-header">
                <b>Edit</b>
            </div>
            <div class="card-body">
                <input type="hidden" asp-for="Id" />
                <div class="form-group frow">
                    <label asp-for="DepartmentName" class="col-form-label-sm
col-sm-2"></label>
                    <div class="col-sm-10">
                        <input asp-for="DepartmentName" class="form-control" />
                        <span asp-validation-for="DepartmentName" class="text-
danger"></span>
                    </div>
                </div>
            </div>
        </div>
    </form>
}

```

```

        <div class="card-footer">
            <button type="submit" class="btn btn-warning">Update</button>
            <a asp-action="List" asp-controller="department">
                <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
            </a>
        </div>
    </div>
</div>

</form>
}

```

➔ Employee Directory

Create.cshtml

```

@model Employee
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="store" asp-controller="employee" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <div class="form-group row">
                    <label asp-for="Name" class="col-form-label col-md-
2"></label>
                    <div class="col-md-10">
                        <input asp-for="Name" class="form-control" />
                        <span asp-validation-for="Name" class="text-
danger"></span>
                    </div>
                </div>
                <div class="form-group row">
                    <label asp-for="Gender" class="col-form-label col-md-
2"></label>
                    <div class="col-md-10">
                        <div>
                            <label asp-for="Gender" class="custom-radio">
                                <input type="radio" asp-for="Gender"
value="femlae" />
                                Female
                            </label>
                            <label asp-for="Gender">
                                <input type="radio" asp-for="Gender"
value="male" />
                                Male
                            </label>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>

```

សកលវិទ្យាល័យបៀលប្រាស

```

        <span asp-validation-for="Gender" class="text-
danger"></span>
        </div>
    </div>
    <div class="form-group row">
        <label asp-for="DepartmentId" class="col-form-label col-md-
2"></label>
        <div class="col-md-10">
            <select asp-for="DepartmentId" class="custom-select"
asp-items="@((ViewBag.Department))">
                <option value="" disabled
selected>Department</option>
            </select>
            <span asp-validation-for="DepartmentId" class="text-
danger"></span>
        </div>
    </div>
    <div class="form-group row">
        <label asp-for="Address" class="col-form-label col-md-
2"></label>
        <div class="col-md-10">
            <input asp-for="Address" class="form-control" />
            <span asp-validation-for="Address" class="text-
danger"></span>
        </div>
    </div>
    </div>
    <div class="card-footer">
        <button type="submit" class="btn btn-primary">Create</button>
        <a asp-action="Index" asp-controller="employee">
            <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
        </a>
    </div>
</div>
</form>
</div>
Delete.cshtml

@model EmployeeViewModel
@{
    ViewData["Title"] = "Delete";
}

<div class="card">
    <div class="card-header">
        <strong>Confirm?</strong>
    </div>
    <div class="card-body">
        <div class="alert alert-danger role="alert">
            <p>Do you want to delete
            &nbsp;&nbsp;&nbsp;<strong>@Model.Name</strong>?</p>
            <form asp-action="destroy" method="post" asp-route-id="@Model.Id">

```

```
        <button type="submit" class="btn btn-danger">Delete</button>
        <a asp-action="Index" class="ml-2">
            <i class="fas fa-arrow-circle-left"></i>
            Back
        </a>
    </form>
</div>
</div>
</div>
Detailed.cshtml
```

```
@model EmployeeViewModel
```

```
@{
    ViewData["Title"] = "Detailed";
}
```

```
<h3>Employee information</h3>
<div>
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Id)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Id)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Name)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Name)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Gender)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Gender)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Department)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Department)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Address)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Address)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="Create">
```


សកលវិទ្យាល័យបៀលប្រាស

```
        <i class="fas fa-plus-circle"></i>
        Create
    </a>
    |
    <a asp-action="Index">
        <i class="fas fa-arrow-circle-left"></i>
        Back To List
    </a>
</div>
```

Edit.cshtml

```
@model Employee
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="update" asp-controller="employee" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <input type="hidden" asp-for="Id" />
                <div class="form-group row">
                    <label asp-for="Name" class="col-form-label col-md-2"></label>
                    <div class="col-md-10">
                        <input asp-for="Name" class="form-control" />
                        <span asp-validation-for="Name" class="text-danger"></span>
                    </div>
                </div>
                <div class="form-group row">
                    <label asp-for="Gender" class="col-form-label col-md-2"></label>
                    <div class="col-md-10">
                        <div>
                            <label asp-for="Gender" class="custom-radio">
                                <input type="radio" asp-for="Gender"
                                value="femlae" />
                                Female
                            </label>
                            <label asp-for="Gender">
                                <input type="radio" asp-for="Gender"
                                value="male" />
                                Male
                            </label>
                        </div>
                        <span asp-validation-for="Gender" class="text-danger"></span>
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>
```

សកលវិទ្យាល័យបៀលប្រាស

```

        </div>
    </div>
    <div class="form-group row">
        <label asp-for="DepartmentId" class="col-form-label col-md-
2"></label>
        <div class="col-md-10">
            <select asp-for="DepartmentId" class="custom-select"
asp-items="@ (ViewBag.Departments)">
                <option value="" disabled
selected>Department</option>
            </select>
            <span asp-validation-for="DepartmentId" class="text-
danger"></span>
        </div>
    </div>
    <div class="form-group row">
        <label asp-for="Address" class="col-form-label col-md-
2"></label>
        <div class="col-md-10">
            <input asp-for="Address" class="form-control" />
            <span asp-validation-for="Address" class="text-
danger"></span>
        </div>
    </div>
</div>
<div class="card-footer">
    <button type="submit" class="btn btn-primary">Create</button>
    <a asp-action="Index" asp-controller="employee">
        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
    </a>
</div>
</div>
</form>
</div>
```

Index.cshtml

```
@model IEnumerable<EmployeeViewModel>
```

```
@{
    ViewData["Title"] = "Index";
}
<div class="card">
    <div class="card-header">
        <strong>Employee Information</strong>
        <a asp-action="Create" class="float-right">Create</a>
    </div>
    <div class="card-body">
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>@Html.DisplayNameFor(emp => emp.Name)</th>
                    <th>@Html.DisplayNameFor(emp => emp.Gender)</th>
```

សកលវិទ្យាល័យបៀលប្រាស

```
<th>@Html.DisplayNameFor(emp => emp.Department)</th>
<th>@Html.DisplayNameFor(emp => emp.Address)</th>
<th>@Html.DisplayName("Actions")</th>
</tr>
</thead>
<tbody>
    @if (Model.Count() > 0 || Model != null)
    {
        @foreach (var emp in Model)
        {
            <tr>
                <td>@Html.DisplayFor(em => emp.Name)</td>
                <td>@Html.DisplayFor(em => emp.Gender)</td>
                <td>@Html.DisplayFor(em => emp.Department)</td>
                <td>@Html.DisplayFor(em => emp.Address)</td>
                <td>
                    <a asp-action="Detailed" asp-route-id="@emp.Id">
                        <i class="far fa-eye mr-1"></i>
                        Detail
                    </a> |
                    <a asp-action="Edit" asp-route-id="@emp.Id">
                        <i class="far fa-edit mr-1"></i>Edit
                    </a> |
                    <a asp-action="Delete" asp-route-id="@emp.Id">
                        <i class="far fa-trash-alt mr-1"></i>
                        Delete
                    </a>
                </td>
            </tr>
        }
    }
    else
    {
        <tr>
            <td colspan="5">
                <h4>there is no data</h4>
            </td>
        </tr>
    }
</tbody>
</table>
</div>
</div>
```

- Share Directory

_Layout.cshtml

```
<!DOCTYPE html>

<html>
<head>
```

```
<meta name="viewport" content="width=device-width" />
<title>@ViewBag.Title</title>
<link href="~/bootstrap/css/bootstrap.css" rel="stylesheet" />
<link href="~/font-awesome/css/all.css" rel="stylesheet" />
</head>
<body>
  <partial name="_Navbar" />
  <div class="container-fluid">
    @RenderBody()
  </div>
  <script src="~/jquery/jquery.js"></script>
  <script src="~/bootstrap/js/bootstrap.js"></script>
  <script src="~/font-awesome/js/all.js"></script>
</body>
</html>
```

_Navbar.cshtml

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-
only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link"
          asp-action="Index" asp-controller="employee">Employee</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" asp-action="list" asp-
controller="department">Department</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search"
placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success my-2 my-sm-0"
type="submit">Search</button>
    </form>
  </div>
</nav>
```

_ViewImports.cshtml

សកលវិទ្យាល័យបៀលប្រាស

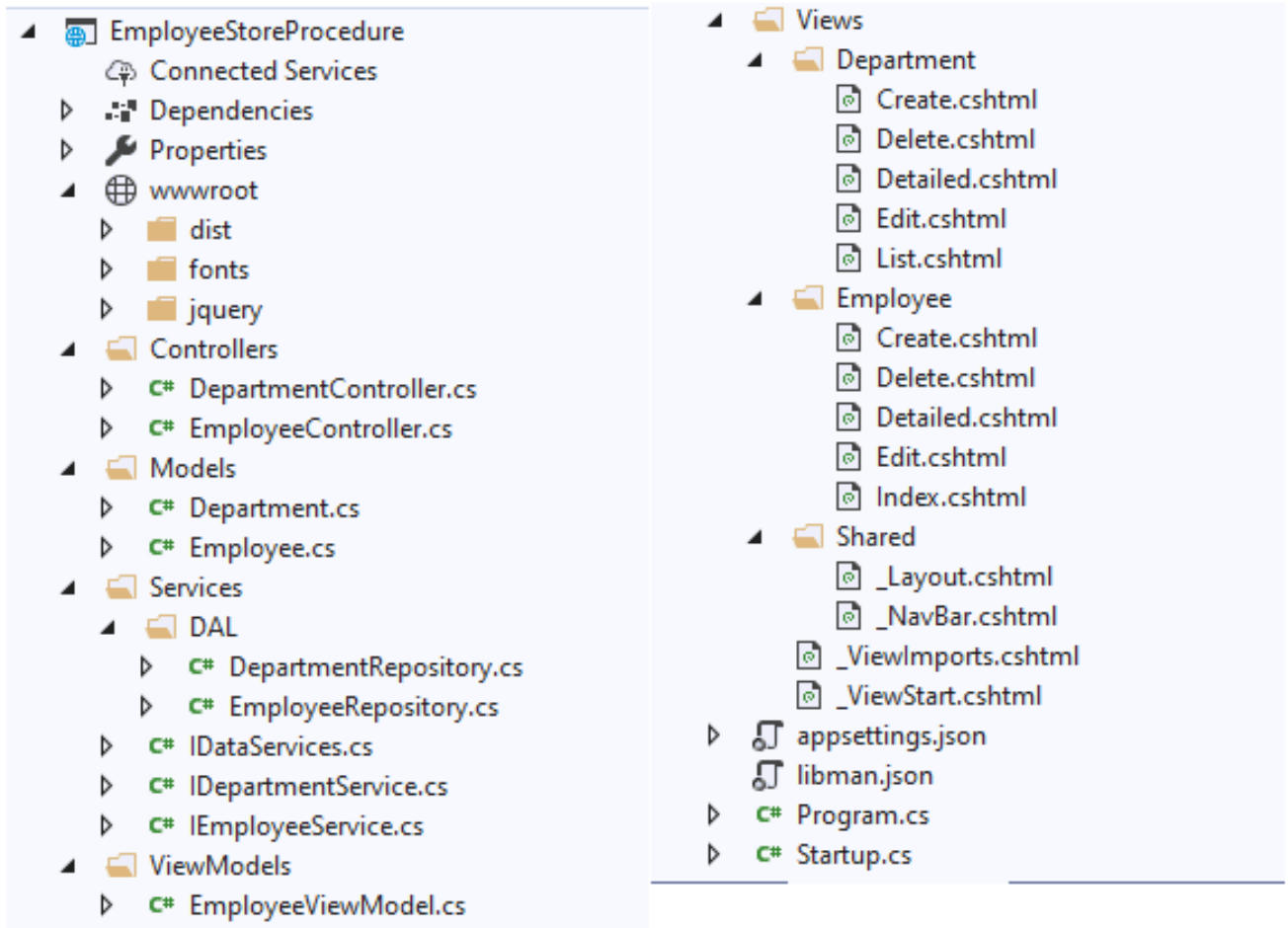
```
@using EmployeeSQL.Models;  
@using EmployeeSQL.ViewModels;  
@addTagHelper *,Microsoft.AspNetCore.Mvc.TagHelpers
```

_ViewStart.cshtml

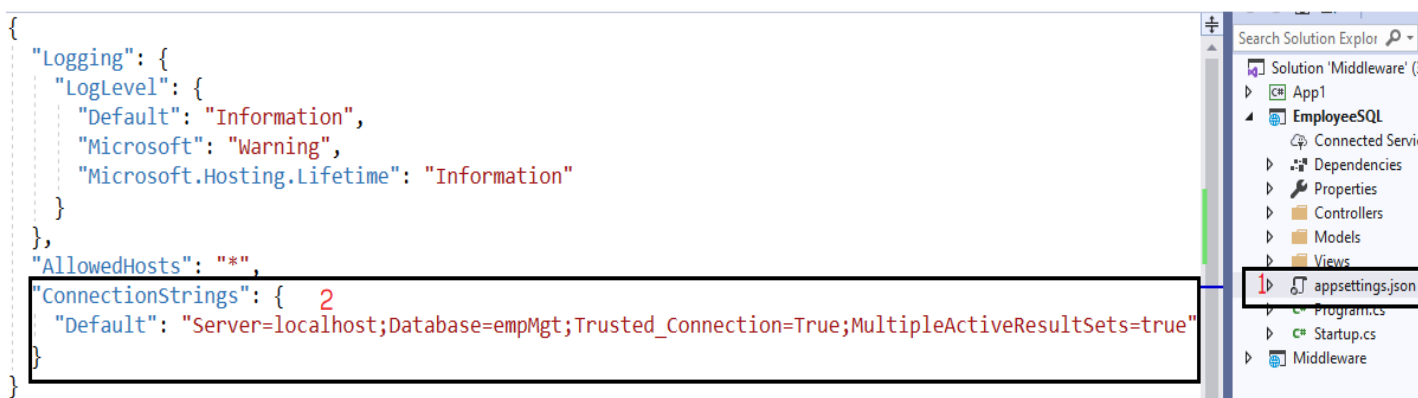
```
@{  
    Layout = "_Layout";  
}
```

2. ការប្រើប្រាស់ជាមួយ Store Procedure

a. បង្កើតproject ដែលមានឈ្មោះនិងទំរង់ដូចខាងក្រោម៖

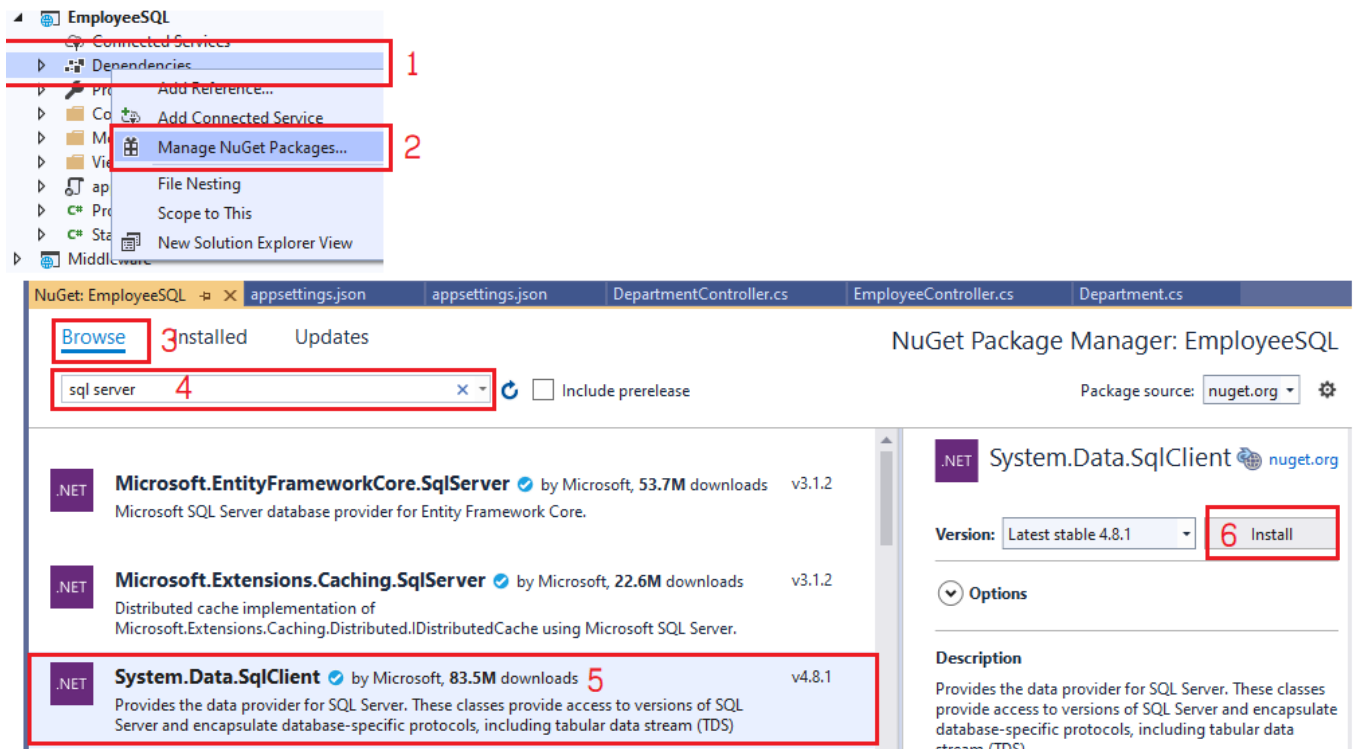


បង្កើត connection string → appsettings.json → សរសេរដូចរូបខាងក្រោម៖

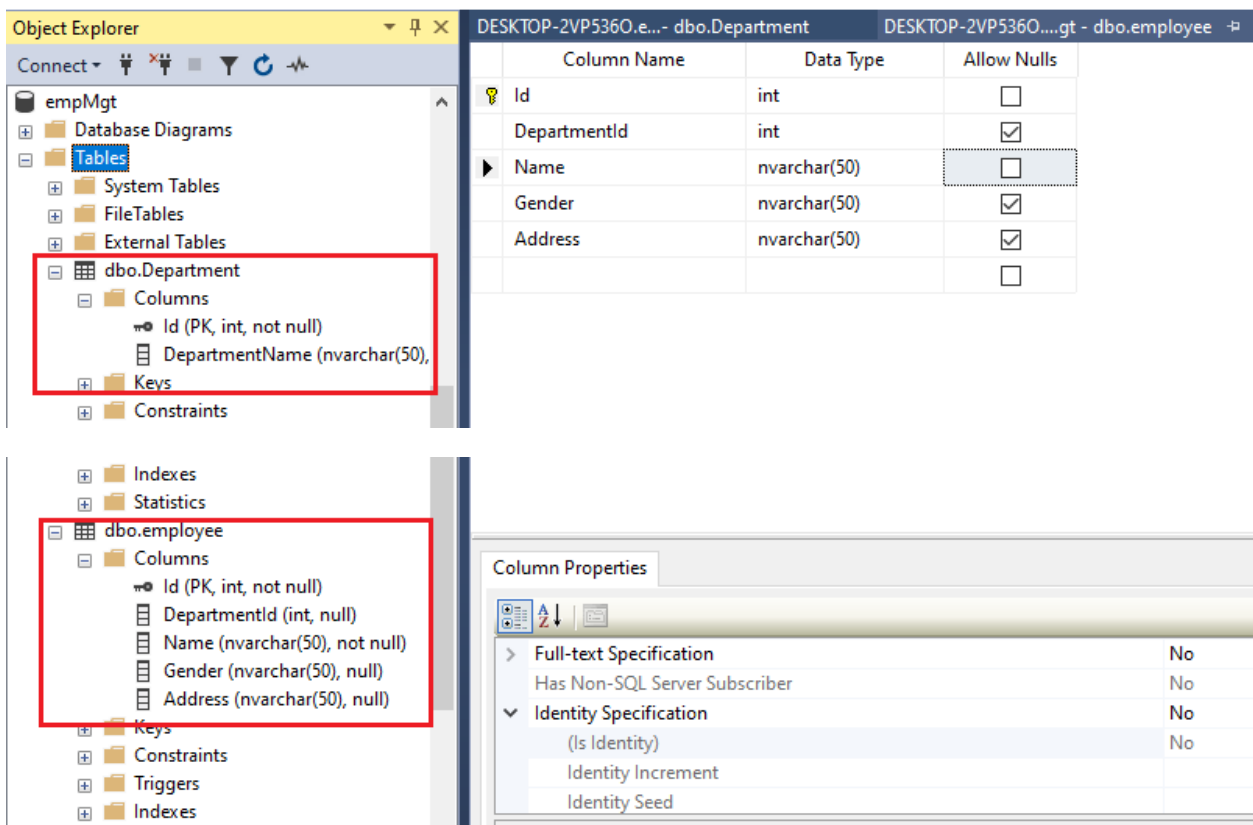


បន្ថែម package ចូលក្នុង project តាមរូបខាងក្រោម៖ ដោយ right click លើ dependency

សកលវិទ្យាល័យបៀលប្រាស



បង្កើត database ដូចរូបខាងក្រោម (empMgt) និងតារាងចំនួនពីរគឺ employee និង department:



→ បង្កើត store procedure មួយចំនួនដូចខាងក្រោម:

សកលវិទ្យាល័យបៀលប្រាស

- CreateDepartment

```
CREATE PROC [dbo].[CreateDepartment](@name nvarchar(50))
AS
BEGIN
INSERT INTO Department(DepartmentName) VALUES(@name);
END
```

- CreateEmp

```
CREATE PROC [dbo].[CreateEmp](@name nvarchar(50),@gender nvarchar(10)
,@dep int,@address nvarchar(100))
AS
BEGIN
INSERT INTO employee([Name],[Gender],[DepartmentId],[Address])
VALUES(@name,@gender,@dep,@address);
END
```

- DeleteDepartment

```
CREATE PROC [dbo].[DeleteDepartment](@id int)
AS
BEGIN
DELETE FROM Department
WHERE Id=@id;
END
```

- DeleteEmployee

```
CREATE PROC [dbo].[DeleteEmployee](@id int)
AS
BEGIN
DELETE FROM employee
WHERE Id=@id;
END
```

- GetAll

```
CREATE PROC [dbo].[GetAll]
AS
BEGIN
SELECT * FROM employee emp INNER JOIN Department dep
ON emp.DepartmentId=dep.Id
ORDER BY [Name] DESC;
END
```

- GetAllEmployee

```
CREATE PROC [dbo].[GetAllEmployees]
```


សកលវិទ្យាល័យបៀលប្រាស

```
AS
BEGIN
    SELECT * FROM employee
    ORDER BY [Name] DESC;
END
```

- GetDepartment

```
CREATE PROC [dbo].[GetDepartment](@id int)
AS
BEGIN

SELECT * FROM Department WHERE [Id]=@id;
END
```

- GetEmployee

```
CREATE PROC [dbo].[GetEmployee](@id int)
AS
BEGIN
    SELECT * FROM employee emp INNER JOIN Department dep
    ON emp.DepartmentId=dep.Id
    WHERE emp.Id=@id
    ORDER BY [Name] DESC;
END
```

- GetOnlyEmployee

```
CREATE PROC [dbo].[GetOnlyEmployee](@id int)
AS
BEGIN
    SELECT * FROM employee
    WHERE Id=@id;
END
```

- GetsDeatment

```
CREATE PROC [dbo].[GetsDepartment]
AS
BEGIN

SELECT * FROM Department;
END
```

- UpdateDepartment

```
CREATE PROC [dbo].[UpdateDepartment](@id int,@name nvarchar(50))
AS
BEGIN
UPDATE Department SET DepartmentName=@name
```

សកលវិទ្យាល័យបៀល(ប្រាស)

```
WHERE Id=@id;  
END
```

- UpdateEmp

```
CREATE PROC [dbo].[UpdateEmp](@id int,@name nvarchar(50),@gender  
nvarchar(10)  
,@dep int,@address nvarchar(100))  
AS  
BEGIN  
    UPDATE employee SET  
        [Name]=@name,[Gender]=@gender,  
        [DepartmentId]=@dep,[Address]=@address  
    WHERE Id=@id;  
END
```

→Models Directory

Department.cs

```
using System.ComponentModel.DataAnnotations;  
  
namespace EmployeeSQL.Models  
{  
    public class Department  
    {  
        public int Id { get; set; }  
  
        [Required]  
        [Display(Name = "Department")]  
        public string DepartmentName { get; set; }  
    }  
}
```

Employee.cs

```
using System.ComponentModel.DataAnnotations;  
  
namespace EmployeeSQLEntity.Models  
{  
    public class Employee  
    {  
        public int Id { get; set; }  
        [Required]  
        public string Name { get; set; }  
        [Required]  
        public string Gender { get; set; }  
        [Required]  
        [Display(Name = "Department")]
```

```
        public int DepartmentId { get; set; }
        [Required]
        public string Address { get; set; }
    }
}
```

→ ViewModel Directory

EmployeeViewModel.cs

```
namespace EmployeeSQL.ViewModels
{
    public class EmployeeViewModel
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Gender { get; set; }
        public string Department { get; set; }
        public string Address { get; set; }
    }
}
```

→ Startup.cs

```
using EmployeeMySQL.Services;
using EmployeeMySQL.Services.DAL;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;

namespace EmployeeMySQL
{
    public class Startup
    {
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddControllersWithViews();
            services.AddScoped<IDepartmentService, DepartmentRepository>();
            services.AddScoped<IEmployeeService, EmployeeRepository>();
        }

        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
            }
            app.UseStaticFiles();
            app.UseRouting();
            app.UseEndpoints(endpoints =>
            {
            })
        }
    }
}
```

សកលវិទ្យាល័យបៀលប្រាស

```
        endpoints.MapControllerRoute("default",
            "{controller}/{action}/{id?}",
            new { controller = "employee", action = "index" });
    });
}
}
```

→Controllers directory

DepartmentController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using EmployeeStoreProcedure.Models;
using EmployeeStoreProcedure.Services;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Configuration;

namespace EmployeeStoreProcedure.Controllers
{
    public class DepartmentController : Controller
    {
        private readonly IDepartmentService _departmentService;
        public DepartmentController(IDepartmentService departmentService)
        {
            this._departmentService = departmentService;
        }
        [ActionName("List")]
        public IActionResult Index()
        {
            var deps = _departmentService.Get();
            return View("List", deps);
        }
        public IActionResult Create()
        {
            return View();
        }

        [HttpPost]
        public IActionResult store([Bind("DepartmentName")] Department dep)
        {
            if (ModelState.IsValid)
            {
                if (_departmentService.Save(dep))
                {
                    return RedirectToAction("List");
                }
            }
            return View("create", dep);
        }
    }
}
```

```
}

public IActionResult Edit(int? id)
{
    var dep = _departmentService.Get(id.Value);
    if (dep != null)
        return View(dep);
    else
        return NotFound();
}

[HttpPost]
public IActionResult Update([Bind("Id", "DepartmentName")] Department
dep)
{
    if (ModelState.IsValid)
    {
        if (_departmentService.Update(dep))
        {
            return RedirectToAction("List");
        }
    }
    return View("create", dep);
}

[HttpPost]
public IActionResult Destroy(int? id)
{
    if (_departmentService.Delete(id.Value))
    {
        return RedirectToAction("List");
    }
    return NotFound();
}

[HttpGet]
public IActionResult Delete(int? id)
{
    var dep = _departmentService.Get(id.Value);
    if (dep != null){
        return View("Delete", dep);
    }
    return NotFound();
}

public IActionResult Detailed(int? id)
{
    var dep = _departmentService.Get(id.Value);
    if (dep != null)
    {
        return View("Detailed", dep);
    }
    return NotFound();
}
```

សកលវិទ្យាល័យបៀលប្រាស

```
}  
}
```

EmployeeController.cs

```
using System.Linq;  
using EmployeeStoreProcedure.Models;  
using EmployeeStoreProcedure.Services;  
using Microsoft.AspNetCore.Mvc;  
using Microsoft.AspNetCore.Mvc.Rendering;  
  
namespace EmployeeStoreProcedure.Controllers  
{  
  
    public class EmployeeController : Controller  
    {  
        private readonly IEmployeeService _empService;  
        private readonly IDepartmentService _departmentService;  
        public EmployeeController(IEmployeeService  
iempService, IDepartmentService departmentService)  
        {  
            this._empService = iempService;  
            this._departmentService = departmentService;  
        }  
        public IActionResult Index()  
        {  
            var emps = _empService.AllEmployees();  
            return View("Index", emps);  
        }  
  
        [HttpGet]  
        public IActionResult Create()  
        {  
            ViewBag.Departments = _departmentService.Get().Select(dep => new  
SelectListItem  
            {  
                Text = dep.DepartmentName, Value = dep.Id.ToString()  
            });  
            return View();  
        }  
        [HttpPost]  
        // [ActionName("store")]  
        public IActionResult Store(Employee emp)  
        {  
            if (ModelState.IsValid)  
            {  
                if (_empService.Save(emp))  
                {  
                    return RedirectToAction("Index");  
                }  
            }  
            ViewBag.Departments = _departmentService.Get().Select(dep => new  
SelectListItem
```

```
{
    Text = dep.DepartmentName,
    Value = dep.Id.ToString()
});
return View("Create", emp);
}

public IActionResult Edit(int Id)
{
    var emp = _empService.Get(Id);
    if (emp != null)
    {
        ViewBag.Departments = _departmentService.Get().Select(dep=>new
        SelectListItem
        {
            Text=dep.DepartmentName,
            Value=dep.Id.ToString(),
        });
        return View(emp);
    }
    else
        return NotFound();
}

[HttpPost]
public IActionResult Update(Employee emp)
{
    if (ModelState.IsValid)
    {
        if (_empService.Update(emp))
        {
            return RedirectToAction("Index");
        }
    }
    return View(emp);
}

public IActionResult Detailed(int Id)
{
    var emp = _empService.GetEmployee(Id);
    if (emp != null)
        return View(emp);
    else
        return NotFound();
}

public IActionResult Delete(int Id)
{
    var emp = _empService.GetEmployee(Id);
    if (emp != null)
        return View(emp);
    else
        return NotFound();
}

public IActionResult Destroy(int Id)
```

```
{
    var emp = _empService.Delete(Id);
    if (emp)
        return RedirectToAction("Index");
    else
        return NotFound();
}
}
```

➔ Services Directory

IDataServices.cs

```
using EmployeeStoreProcedure.ViewModels;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace EmployeeStoreProcedure.Services
{
    public interface IDataServices<T> where T:class
    {
        bool Save(T TEntity);
        bool Update(T TEntity);
        bool Delete(int Id);
        T Get(int Id);
        List<T> Gets();
    }
}
```

IDepartmentService.cs

```
using EmployeeStoreProcedure.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace EmployeeStoreProcedure.Services
{
    public interface IDepartmentService:IDataServices<Department>
    {
    }
}
```

IEmployeeService.cs

```
using EmployeeStoreProcedure.Models;
```


សកលវិទ្យាល័យបៀលប្រាស

```
using EmployeeStoreProcedure.ViewModels;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace EmployeeStoreProcedure.Services
{
    public interface IEmployeeService:IDataServices<Employee>
    {
        IList<EmployeeViewModel> AllEmployees();
        EmployeeViewModel GetEmployee(int Id);
    }
}
```

- DAL Directory

DepartmentRepository.cs

```
using EmployeeStoreProcedure.Models;
using Microsoft.Extensions.Configuration;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;

namespace EmployeeStoreProcedure.Services.DAL
{
    public class DepartmentRepository : IDepartmentService
    {
        private readonly IConfiguration _config;
        public DepartmentRepository(IConfiguration configuration)
        {
            this._config = configuration;
        }

        public bool Delete(int Id)
        {
            try
            {
                using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
                {
                    using (SqlCommand command = new
SqlCommand("DeleteDepartment", connection))
                    {
                        connection.Open();
                        command.CommandType = CommandType.StoredProcedure;
                        command.Parameters.AddWithValue("@id", Id);
                        return command.ExecuteNonQuery() > 0;
                    }
                }
            }
        }
    }
}
```

```
    }
  }
  catch (Exception ex)
  {
    throw ex;
  }
}

public Department Get(int Id)
{
  try
  {
    using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
    {
      using (SqlCommand command = new SqlCommand("GetDepartment",
connection))
      {
        connection.Open();
        command.CommandType = CommandType.StoredProcedure;
        command.Parameters.AddWithValue("@id", Id);
        using SqlDataAdapter da = new SqlDataAdapter(command);
        DataTable table = new DataTable();
        da.Fill(table);
        var dep = (from t in table.AsEnumerable()
                    select new Department
                    {
                      Id = t.Field<int>("Id"),
                      DepartmentName =
t.Field<string>("DepartmentName"),
                    }).FirstOrDefault();
        return dep;
      }
    }
  }
  catch (Exception ex)
  {
    throw ex;
  }
}

public List<Department> Gets()
{
  try
  {
    using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
    {
      using (SqlCommand command = new
SqlCommand("GetsDepartment", connection))
      {
        connection.Open();
        using SqlDataAdapter da = new SqlDataAdapter(command);
        DataTable table = new DataTable();
```

```
        da.Fill(table);
        var deps = (from t in table.AsEnumerable()
                     select new Department
                     {
                         Id = t.Field<int>("Id"),
                         DepartmentName =
t.Field<string>("DepartmentName"),
                     }).ToList();
        return deps;
    }
}
catch (Exception ex)
{
    throw ex;
}
}

public bool Save(Department dep)
{
    try
    {
        using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
        {
            using (SqlCommand command = new
SqlCommand("CreateDepartment", connection))
            {
                connection.Open();
                command.CommandType = CommandType.StoredProcedure;
                command.Parameters.AddWithValue("@name",
dep.DepartmentName);
                return command.ExecuteNonQuery() > 0;
            }
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

public bool Update(Department dep)
{
    try
    {
        using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
        {
            using (SqlCommand command = new
SqlCommand("UpdateDepartment", connection))
            {
                connection.Open();
                command.CommandType = CommandType.StoredProcedure;
```

```
        command.Parameters.AddWithValue("@id", dep.Id);
        command.Parameters.AddWithValue("@name",
dep.DepartmentName);
        return command.ExecuteNonQuery() > 0;
    }
}
}
catch (Exception ex)
{
    throw ex;
}
}
}
```

EmployeeRepository.cs

```
using Microsoft.Extensions.Configuration;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using EmployeeStoreProcedure.Models;
using System.Data;
using EmployeeStoreProcedure.ViewModels;

namespace EmployeeStoreProcedure.Services.DAL
{
    public class EmployeeRepository : IEmployeeService
    {
        private readonly IConfiguration _config;
        public EmployeeRepository(IConfiguration configuration)
        {
            this._config = configuration;
        }
        public IList<EmployeeViewModel> AllEmployees()
        {
            try
            {
                using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
                {
                    using (SqlCommand command = new SqlCommand("GetAll",
connection))
                    {
                        connection.Open();
                        command.CommandType = CommandType.StoredProcedure;
                        using SqlDataAdapter da = new SqlDataAdapter(command);
                        DataTable table = new DataTable();
                        da.Fill(table);
                        var emps = (from t in table.AsEnumerable()
select new EmployeeViewModel
```

```
        {
            Id = t.Field<int>("Id"),
            Name = t.Field<string>("Name"),
            Gender = t.Field<string>("Gender"),
            Department =
t.Field<string>("DepartmentName"),
            Address = t.Field<string>("Address")
        }).ToList();
        return emps;
    }
}
}
catch (Exception ex)
{
    throw ex;
}
}

public bool Delete(int Id)
{
    try
    {
        using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
        {
            using (SqlCommand command = new
SqlCommand("DeleteEmployee", connection))
            {
                connection.Open();
                command.CommandType = CommandType.StoredProcedure;
                command.Parameters.AddWithValue("@id", Id);
                return command.ExecuteNonQuery()>0;
            }
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

public Employee Get(int Id)
{
    try
    {
        using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
        {
            using (SqlCommand command = new SqlCommand("GetEmployee",
connection))
            {
                connection.Open();
                command.CommandType = CommandType.StoredProcedure;
                command.Parameters.AddWithValue("@id", Id);
            }
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

```

        using SqlDataAdapter da = new SqlDataAdapter(command);
        DataTable table = new DataTable();
        da.Fill(table);
        var emp = (from t in table.AsEnumerable()
                    select new Employee
                    {
                        Id = t.Field<int>("Id"),
                        Name = t.Field<string>("Name"),
                        Gender = t.Field<string>("Gender"),
                        DepartmentId =
t.Field<int>("DepartmentId"),
                        Address = t.Field<string>("Address")
                    }).FirstOrDefault();
        return emp;
    }
}
catch (Exception ex)
{
    throw ex;
}
}

public EmployeeViewModel GetEmployee(int Id)
{
    try
    {
        using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
        {
            using (SqlCommand command = new SqlCommand("GetEmployee",
connection))
            {
                connection.Open();
                command.CommandType = CommandType.StoredProcedure;
                command.Parameters.AddWithValue("@id", Id);
                using (SqlDataAdapter da = new SqlDataAdapter(command))
                {
                    DataTable table = new DataTable();
                    da.Fill(table);
                    var emp = (from t in table.AsEnumerable()
                                select new EmployeeViewModel
                                {
                                    Id = t.Field<int>("Id"),
                                    Name = t.Field<string>("Name"),
                                    Gender = t.Field<string>("Gender"),
                                    Department =
t.Field<string>("DepartmentName"),
                                    Address = t.Field<string>("Address")
                                }).FirstOrDefault();
                    return emp;
                }
            }
        }
    }
}

```

```
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

public List<Employee> Gets()
{
    try
    {
        using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
        {
            using (SqlCommand command = new
SqlCommand("GetAllEmployees", connection))
            {
                connection.Open();
                command.CommandType = CommandType.StoredProcedure;
                using SqlDataAdapter da = new SqlDataAdapter(command);
                DataTable table = new DataTable();
                da.Fill(table);
                var emps = (from t in table.AsEnumerable()
                    select new Employee
                    {
                        Id = t.Field<int>("Id"),
                        Name = t.Field<string>("Name"),
                        Gender = t.Field<string>("Gender"),
                        DepartmentId =
t.Field<int>("DepartmentId"),
                        Address = t.Field<string>("Address")
                    }).ToList();
                return emps;
            }
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

public bool Save(Employee emp)
{
    try
    {
        using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
        {
            using (SqlCommand command = new SqlCommand("CreateEmp",
connection))
            {
```

```

        connection.Open();
        command.CommandType = CommandType.StoredProcedure;
        command.Parameters.AddWithValue("@name", emp.Name);
        command.Parameters.AddWithValue("@gender", emp.Gender);
        command.Parameters.AddWithValue("@dep",
emp.DepartmentId);
        command.Parameters.AddWithValue("@address",
emp.Address);
        return command.ExecuteNonQuery() > 0;
    }
}
}
catch (Exception ex)
{
    throw ex;
}
}
public bool Update(Employee emp)
{
    try
    {
        using (SqlConnection connection = new
SqlConnection(_config.GetConnectionString("Connection")))
        {
            using (SqlCommand command = new SqlCommand("UpdateEmp",
connection))
            {
                connection.Open();
                command.CommandType = CommandType.StoredProcedure;
                command.Parameters.AddWithValue("@id", emp.Id);
                command.Parameters.AddWithValue("@name", emp.Name);
                command.Parameters.AddWithValue("@gender", emp.Gender);
                command.Parameters.AddWithValue("@dep",
emp.DepartmentId);
                command.Parameters.AddWithValue("@address",
emp.Address);
                return command.ExecuteNonQuery() > 0;
            }
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
}
}

```


➔ Views Directory

➔ Department folder

Create.cshtml

```
@model Department
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="store" asp-controller="department" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <div class="form-group row">
                    <label asp-for="DepartmentName" class="col-form-label col-
md-2"></label>
                    <div class="col-md-10">
                        <input asp-for="DepartmentName" class="form-control" />
                        <span asp-validation-for="DepartmentName" class="text-
danger"></span>
                    </div>
                </div>
            </div>
            <div class="card-footer">
                <button type="submit" class="btn btn-primary">Create</button>
                <a asp-action="List" asp-controller="department">
                    <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
                </a>
            </div>
        </div>
    </form>
</div>
```

Detailed.cshtml

```
@model Department
@{
    ViewData["Title"] = "Detailed";
}
<div>
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(d => d.Id)
        </dt>
        <dd class="col-sm-10">
```

សកលវិទ្យាល័យបៀលប្រាស

```
        @Html.DisplayFor(d => d.Id)
    </dd>
    <dt class="col-sm-2">
        @Html.DisplayNameFor(d => d.DepartmentName)
    </dt>
    <dd class="col-sm-10">
        @Html.DisplayFor(d => d.DepartmentName)
    </dd>
</dl>
</div>
<div>
    <a asp-action="edit" asp-route-id="@Model.Id">Edit</a>
    |
    <a asp-action="List">Back to List</a>
</div>
```

Delete.cshtml

```
@model Department
@{
    ViewData["Title"] = "Delete";
}
<div class="container mt-4">
    <div class="row">
        <div class="col col-sm-8 offset-2">
            <div class="alert alert-danger" role="alert">
                Do want to delete <strong>@Model.DepartmentName </strong>?
            </div>
            <div>
                <form asp-action="delete" asp-controller="department"
method="post">
                    <input type="hidden" asp-for="Id" />
                    <input type="submit" value="Delete" class="btn btn-danger"
/>

                    <a asp-action="List" asp-controller="department">
                        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back
To List
                        </a>
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>
List.html
```

```
@model IEnumerable<Department>
@{
    ViewData["Title"] = "List";
}
<div class="card">
    <div class="card-header">
        <strong>All</strong>
```

សកលវិទ្យាល័យបៀលប្រាស

```
<a asp-action="create" asp-controller="department" class="float-
right">Create</a>
</div>
<div class="card-body">
  <table class="table table-bordered">
    <thead>
      <tr>
        <th>@Html.DisplayNameFor(d=>d.DepartmentName)</th>
        <th>Actions</th>
      </tr>
    </thead>
    <tbody>
      @if (Model.Count() > 0)
      {
        @foreach (var dep in Model)
        {
          <tr>
            <td>@dep.DepartmentName</td>
            <td>
              <a asp-action="detailed" asp-controller="department"
asp-route-id="@dep.Id">
                <i class="far fa-eye mr-1"></i>Detail
              </a> |
              <a asp-action="edit" asp-controller="department" asp-
route-id="@dep.Id">
                <i class="far fa-edit mr-1"></i>Edit
              </a> |
              <a asp-action="destroy" asp-controller="department"
asp-route-id="@dep.Id">
                <i class="far fa-trash-alt mr-1"></i>
                Delete
              </a>
            </td>
          </tr>
        }
      }
    </tbody>
  </table>
</div>
</div>
```

Edit.cshtml

```
@model Department
@{
  ViewData["Title"] = "Index";
}

@if (Model != null)
{
  <form asp-action="update" asp-controller="department">
    <div class="card">
```

សកលវិទ្យាល័យបៀលប្រាស

```
<div class="card-header">
    <b>Edit</b>
</div>
<div class="card-body">
    <input type="hidden" asp-for="Id" />
    <div class="form-group row">
        <label asp-for="DepartmentName" class="col-form-label-sm
col-sm-2"></label>
        <div class="col-sm-10">
            <input asp-for="DepartmentName" class="form-control" />
            <span asp-validation-for="DepartmentName" class="text-
danger"></span>
        </div>
    </div>
</div>
<div class="card-footer">
    <button type="submit" class="btn btn-warning">Update</button>
    <a asp-action="List" asp-controller="department">
        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
    </a>
</div>
</div>
</form>
}
```

- Employee Directory

Create.cshtml

```
@model Employee
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="store" asp-controller="employee" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <div class="form-group row">
                    <label asp-for="Name" class="col-form-label col-md-
2"></label>
                    <div class="col-md-10">
                        <input asp-for="Name" class="form-control" />
                        <span asp-validation-for="Name" class="text-
danger"></span>
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>
```

សកលវិទ្យាល័យបៀលប្រាស

```
<label asp-for="Gender" class="col-form-label col-md-2"></label>
<div class="col-md-10">
  <div>
    <label asp-for="Gender" class="custom-radio">
      <input type="radio" asp-for="Gender"
value="female" />
      Female
    </label>
    <label asp-for="Gender">
      <input type="radio" asp-for="Gender"
value="male" />
      Male
    </label>
  </div>
  <span asp-validation-for="Gender" class="text-
danger"></span>
</div>
</div>
<div class="form-group row">
  <label asp-for="DepartmentId" class="col-form-label col-md-2"></label>
  <div class="col-md-10">
    <select asp-for="DepartmentId" class="custom-select"
asp-items="@((ViewBag.Department))">
      <option value="" disabled
selected>Department</option>
    </select>
    <span asp-validation-for="DepartmentId" class="text-
danger"></span>
  </div>
</div>
<div class="form-group row">
  <label asp-for="Address" class="col-form-label col-md-2"></label>
  <div class="col-md-10">
    <input asp-for="Address" class="form-control" />
    <span asp-validation-for="Address" class="text-
danger"></span>
  </div>
</div>
</div>
<div class="card-footer">
  <button type="submit" class="btn btn-primary">Create</button>
  <a asp-action="Index" asp-controller="employee">
    <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
  </a>
</div>
</div>
</form>
</div>
Delete.cshtml
```

សកលវិទ្យាល័យហ្សឺណែវ

[illegible]

```
@model EmployeeViewModel
@{
    ViewData["Title"] = "Detailed";
}

<h3>Employee information</h3>
<div>
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Id)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Id)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Name)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Name)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Gender)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Gender)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Department)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Department)
        </dd>
    </dl>
</div>
```

```
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Department)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Address)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Address)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="Create">
        <i class="fas fa-plus-circle"></i>
        Create
    </a>
    |
    <a asp-action="Index">
        <i class="fas fa-arrow-circle-left"></i>
        Back To List
    </a>
</div>
```

Edit.cshtml

```
@model Employee
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="update" asp-controller="employee" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <input type="hidden" asp-for="Id" />
                <div class="form-group row">
                    <label asp-for="Name" class="col-form-label col-md-2"></label>
                    <div class="col-md-10">
                        <input asp-for="Name" class="form-control" />
                        <span asp-validation-for="Name" class="text-danger"></span>
                    </div>
                </div>
                <div class="form-group row">
                    <label asp-for="Gender" class="col-form-label col-md-2"></label>
                    <div class="col-md-10">
```

សកលវិទ្យាល័យបៀលប្រាស

```
<div>
    <label asp-for="Gender" class="custom-radio">
        <input type="radio" asp-for="Gender"
value="female" />
        Female
    </label>
    <label asp-for="Gender">
        <input type="radio" asp-for="Gender"
value="male" />
        Male
    </label>
</div>
<span asp-validation-for="Gender" class="text-
danger"></span>
</div>
</div>
<div class="form-group row">
    <label asp-for="DepartmentId" class="col-form-label col-md-
2"></label>
    <div class="col-md-10">
        <select asp-for="DepartmentId" class="custom-select"
asp-items="@((ViewBag.Departments))">
            <option value="" disabled
selected>Department</option>
        </select>
        <span asp-validation-for="DepartmentId" class="text-
danger"></span>
    </div>
</div>
<div class="form-group row">
    <label asp-for="Address" class="col-form-label col-md-
2"></label>
    <div class="col-md-10">
        <input asp-for="Address" class="form-control" />
        <span asp-validation-for="Address" class="text-
danger"></span>
    </div>
</div>
</div>
<div class="card-footer">
    <button type="submit" class="btn btn-primary">Create</button>
    <a asp-action="Index" asp-controller="employee">
        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
    </a>
</div>
</div>
</form>
</div>
Index.cshtml
```

@model IEnumerable<EmployeeViewModel>

សកលវិទ្យាល័យបៀលប្រាស

```
@{
    ViewData["Title"] = "Index";
}
<div class="card">
    <div class="card-header">
        <strong>Employee Information</strong>
        <a asp-action="Create" class="float-right">Create</a>
    </div>
    <div class="card-body">
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>@Html.DisplayNameFor(emp => emp.Name)</th>
                    <th>@Html.DisplayNameFor(emp => emp.Gender)</th>
                    <th>@Html.DisplayNameFor(emp => emp.Department)</th>
                    <th>@Html.DisplayNameFor(emp => emp.Address)</th>
                    <th>@Html.DisplayName("Actions")</th>
                </tr>
            </thead>
            <tbody>
                @if (Model.Count() > 0 || Model != null)
                {
                    @foreach (var emp in Model)
                    {
                        <tr>
                            <td>@Html.DisplayFor(em => emp.Name)</td>
                            <td>@Html.DisplayFor(em => emp.Gender)</td>
                            <td>@Html.DisplayFor(em => emp.Department)</td>
                            <td>@Html.DisplayFor(em => emp.Address)</td>
                            <td>
                                <a asp-action="Detailed" asp-route-id="@emp.Id">
                                    <i class="far fa-eye mr-1"></i>
                                    Detail
                                </a> |
                                <a asp-action="Edit" asp-route-id="@emp.Id">
                                    <i class="far fa-edit mr-1"></i>Edit
                                </a> |
                                <a asp-action="Delete" asp-route-id="@emp.Id">
                                    <i class="far fa-trash-alt mr-1"></i>
                                    Delete
                                </a>
                            </td>
                        </tr>
                    }
                }
                else
                {
                    <tr>
                        <td colspan="5">
                            <h4>there is no data</h4>
                        </td>
                    </tr>
                }
            </tbody>
        </table>
    </div>
</div>
```

```
        </tbody>
    </table>
</div>
</div>
```

- Share Directory

_Layout.cshtml

```
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>
    <link href="~/bootstrap/css/bootstrap.css" rel="stylesheet" />
    <link href="~/font-awesome/css/all.css" rel="stylesheet" />
</head>
<body>
    <partial name="_Navbar" />
    <div class="container-fluid">
        @RenderBody()
    </div>
    <script src="~/jquery/jquery.js"></script>
    <script src="~/bootstrap/js/bootstrap.js"></script>
    <script src="~/font-awesome/js/all.js"></script>
</body>
</html>
```

_NavBar.cshtml

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
                <a class="nav-link" href="#">Home <span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link"
asp-action="Index" asp-controller="employee">Employee</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" asp-action="list" asp-
controller="department">Department</a>
            </li>
        </ul>
    </div>
</div>
```

```
        </ul>
        <form class="form-inline my-2 my-lg-0">
            <input class="form-control mr-sm-2" type="search"
placeholder="Search" aria-label="Search">
            <button class="btn btn-outline-success my-2 my-sm-0"
type="submit">Search</button>
        </form>
    </div>
</nav>
```

_ViewImports.cshtml

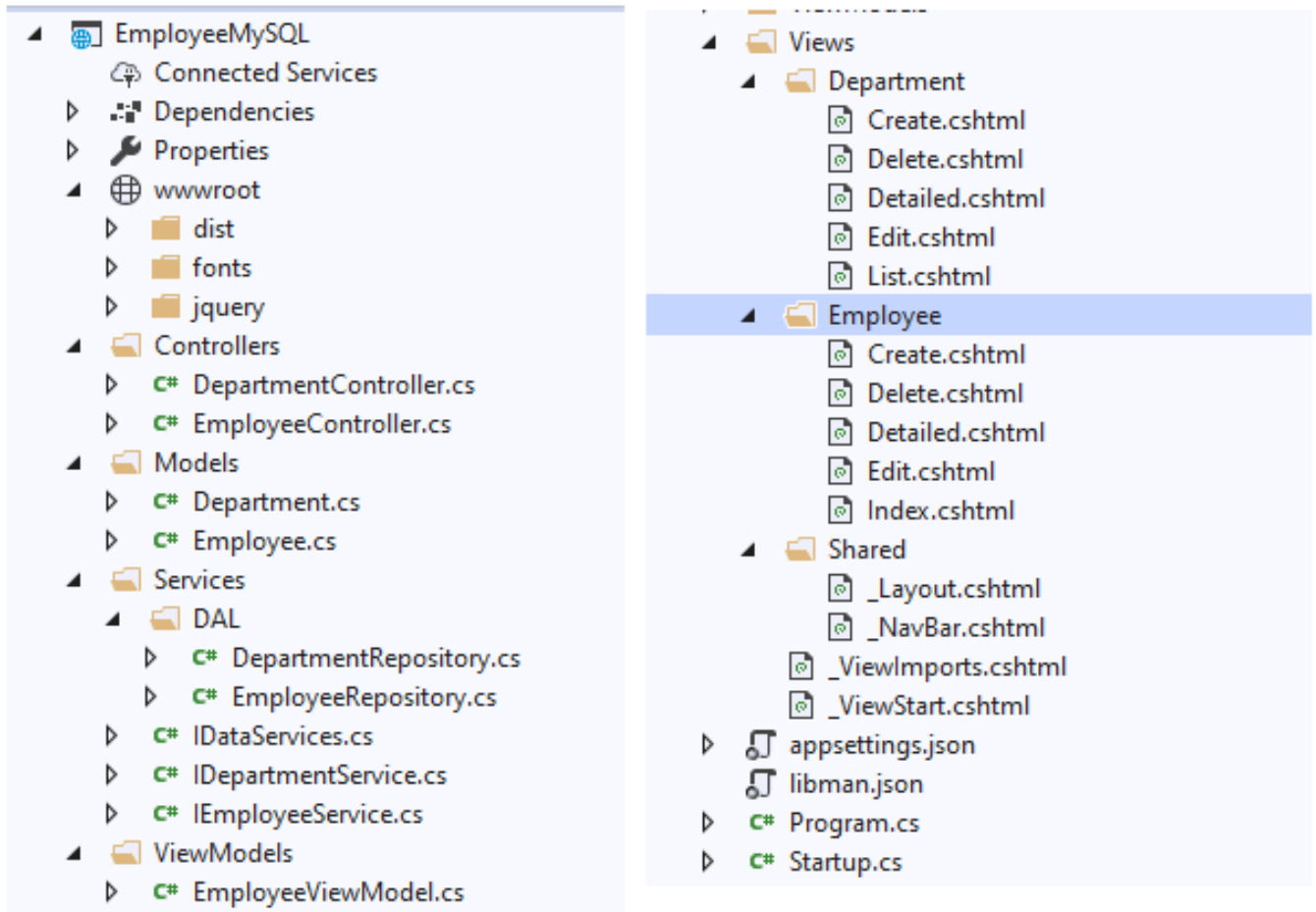
```
@using EmployeeSQL.Models;
@using EmployeeSQL.ViewModels;
@addTagHelper *,Microsoft.AspNetCore.Mvc.TagHelpers
```

_ViewStart.cshtml

```
@{
    Layout = "_Layout";
}
```

III. ASP.NET និង ADO.NET(MySQL Server)

បង្កើតProjectដែលមានឈ្មោះនិងទំរង់ដូចខាងក្រោម៖

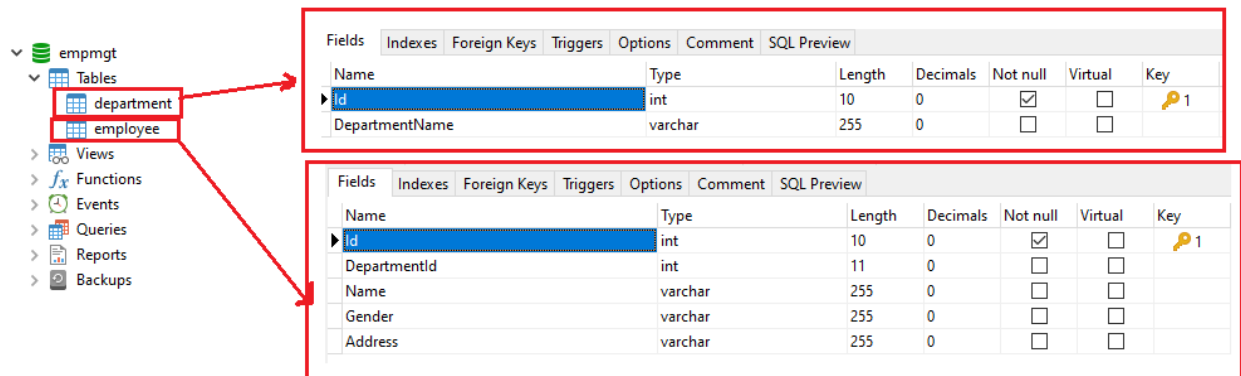


→ConnectionString

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "Connection": "Server=localhost;database=empMgts;user=root;password="
  }
}
```

សកលវិទ្យាល័យបៀលប្រាស

→ បង្កើត Database



→ Startup.cs

```
using EmployeeMySQL.Services;
using EmployeeMySQL.Services.DAL;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;

namespace EmployeeMySQL
{
    public class Startup
    {
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddControllersWithViews();
            services.AddScoped<IDepartmentService, DepartmentRepository>();
            services.AddScoped<IEmployeeService, EmployeeRepository>();
        }

        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
            }
            app.UseStaticFiles();
            app.UseRouting();
            app.UseEndpoints(endpoints =>
            {
                endpoints.MapControllerRoute("default",
                    "{controller}/{action}/{id?}",
                    new { controller = "employee", action = "index" });
            });
        }
    }
}
```

}

→Controllers Directory

DepartmentController.cs

```
using EmployeeMySQL.Models;
using EmployeeMySQL.Services;
using Microsoft.AspNetCore.Mvc;

namespace EmployeeMySQL.Controllers
{
    public class DepartmentController : Controller
    {
        private readonly IDepartmentService _departmentService;
        public DepartmentController(IDepartmentService departmentService)
        {
            this._departmentService = departmentService;
        }
        [ActionName("List")]
        public IActionResult Index()
        {
            var deps = _departmentService.Get();
            return View("List", deps);
        }
        public IActionResult Create()
        {
            return View();
        }
        [HttpPost]
        public IActionResult store([Bind("DepartmentName")] Department dep)
        {
            if (ModelState.IsValid)
            {
                if (_departmentService.Save(dep))
                {
                    return RedirectToAction("List");
                }
            }
            return View("create", dep);
        }
        public IActionResult Edit(int? id)
        {
            var dep = _departmentService.Get(id.Value);
            if (dep != null)
            {
                return View(dep);
            }
            else
            {
                return NotFound();
            }
        }
    }
}
```

```
    }

    [HttpPost]
    public IActionResult Update([Bind("Id", "DepartmentName")]
Department dep)
    {
        if (ModelState.IsValid)
        {
            if (_departmentService.Update(dep))
            {
                return RedirectToAction("List");
            }
        }
        return View("create", dep);
    }

    [HttpPost]
    public IActionResult Destroy(int? id)
    {
        if (_departmentService.Delete(id.Value))
        {
            return RedirectToAction("List");
        }
        return NotFound();
    }

    [HttpGet]
    public IActionResult Delete(int? id)
    {
        var dep = _departmentService.Get(id.Value);
        if (dep != null)
        {
            return View("Delete", dep);
        }
        return NotFound();
    }

    public IActionResult Detailed(int? id)
    {
        var dep = _departmentService.Get(id.Value);
        if (dep != null)
        {
            return View("Detailed", dep);
        }
        return NotFound();
    }
}
}
```

EmployeeController.cs

```
using System.Linq;
using EmployeeMySQL.Models;
```

សកលវិទ្យាល័យបៀលប្រាស

```
using EmployeeMySQL.Services;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;

namespace EmployeeMySQL.Controllers
{
    public class EmployeeController : Controller
    {
        private readonly IEmployeeService _empService;
        private readonly IDepartmentService _departmentService;
        public EmployeeController(IEmployeeService iempService,
IDepartmentService departmentService)
        {
            this._empService = iempService;
            this._departmentService = departmentService;
        }
        public IActionResult Index()
        {
            var emps = _empService.AllEmployees();
            return View("Index", emps);
        }

        [HttpGet]
        public IActionResult Create()
        {
            ViewBag.Departments = _departmentService.Get().Select(dep => new
SelectListItem
            {
                Text = dep.DepartmentName,
                Value = dep.Id.ToString()
            });
            return View();
        }
        [HttpPost]
        public IActionResult Store(Employee emp)
        {
            if (ModelState.IsValid)
            {
                if (_empService.Save(emp))
                {
                    return RedirectToAction("Index");
                }
            }
            ViewBag.Departments = _departmentService.Get().Select(dep => new
SelectListItem
            {
                Text = dep.DepartmentName,
                Value = dep.Id.ToString()
            });
            return View("Create", emp);
        }

        public IActionResult Edit(int Id)
        {

```



```
new
    var emp = _empService.Get(Id);
    if (emp != null)
    {
        ViewBag.Departments = _departmentService.Get().Select(dep =>
            SelectListItem
            {
                Text = dep.DepartmentName,
                Value = dep.Id.ToString(),
            });
        return View(emp);
    }
    else
        return NotFound();
}
[HttpPost]
public IActionResult Update(Employee emp)
{
    if (ModelState.IsValid)
    {
        if (_empService.Update(emp))
        {
            return RedirectToAction("Index");
        }
    }
    return View(emp);
}

public IActionResult Detailed(int Id)
{
    var emp = _empService.GetEmployee(Id);
    if (emp != null)
        return View(emp);
    else
        return NotFound();
}

public IActionResult Delete(int Id)
{
    var emp = _empService.GetEmployee(Id);
    if (emp != null)
        return View(emp);
    else
        return NotFound();
}

public IActionResult Destroy(int Id)
{
    var emp = _empService.Delete(Id);
    if (emp)
        return RedirectToAction("Index");
    else
        return NotFound();
}
}
```

}

→ Model Directory

Department.cs

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace EmployeeMySQLEntity.Models
{
    public class Department
    {
        public int Id { get; set; }
        [Required]
        [Display(Name = "Department")]
        public string DepartmentName { get; set; }

        public ICollection<Employee> Employees { get; set; }
    }
}
```

Employee.cs

```
using System.ComponentModel.DataAnnotations;

namespace EmployeeMySQLEntity.Models
{
    public class Employee
    {
        public int Id { get; set; }
        [Required]
        public string Name { get; set; }
        [Required]
        public string Gender { get; set; }
        [Required]
        [Display(Name = "Department")]
        public int DepartmentId { get; set; }
        [Required]
        public string Address { get; set; }
    }
}
```

→ Views Directory

→ Department folder

Create.cshtml

សកលវិទ្យាល័យបៀលប្រាស

```
@model Department
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="store" asp-controller="department" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <div class="form-group row">
                    <label asp-for="DepartmentName" class="col-form-label col-
md-2"></label>
                    <div class="col-md-10">
                        <input asp-for="DepartmentName" class="form-control" />
                        <span asp-validation-for="DepartmentName" class="text-
danger"></span>
                    </div>
                </div>
            </div>
            <div class="card-footer">
                <button type="submit" class="btn btn-primary">Create</button>
                <a asp-action="List" asp-controller="department">
                    <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
                </a>
            </div>
        </div>
    </form>
</div>
```

Detailed.cshtml

```
@model Department
@{
    ViewData["Title"] = "Detailed";
}
<div>
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(d => d.Id)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(d => d.Id)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(d => d.DepartmentName)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(d => d.DepartmentName)
        </dd>
    </dl>
</div>
```

សកលវិទ្យាល័យបៀលប្រាស

```
        </dd>
    </dl>
</div>
<div>
    <a asp-action="edit" asp-route-id="@Model.Id">Edit</a>
    |
    <a asp-action="List">Back to List</a>
</div>
```

Delete.cshtml

```
@model Department
@{
    ViewData["Title"] = "Delete";
}
<div class="container mt-4">
    <div class="row">
        <div class="col col-sm-8 offset-2">
            <div class="alert alert-danger" role="alert">
                Do want to delete <strong>@Model.DepartmentName </strong>?
            </div>
            <div>
                <form asp-action="delete" asp-controller="department"
method="post">
                    <input type="hidden" asp-for="Id" />
                    <input type="submit" value="Delete" class="btn btn-danger"
/>
                    <a asp-action="List" asp-controller="department">
                        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back
To List
                    </a>
                </form>
            </div>
        </div>
    </div>
</div>
```

List.html

```
@model IEnumerable<Department>
@{
    ViewData["Title"] = "List";
}
<div class="card">
    <div class="card-header">
        <strong>All</strong>
        <a asp-action="create" asp-controller="department" class="float-
right">Create</a>
    </div>
    <div class="card-body">
        <table class="table table-bordered">
            <thead>
                <tr>
```

សកលវិទ្យាល័យបៀលប្រាស

```

        <th>@Html.DisplayNameFor(d=>d.DepartmentName)</th>
        <th>Actions</th>
    </tr>
</thead>
<tbody>
    @if (Model.Count() > 0)
    {
        @foreach (var dep in Model)
        {
            <tr>
                <td>@dep.DepartmentName</td>
                <td>
                    <a asp-action="detailed" asp-controller="department"
asp-route-id="@dep.Id">
                        <i class="far fa-eye mr-1"></i>Detail
                    </a> |
                    <a asp-action="edit" asp-controller="department" asp-
route-id="@dep.Id">
                        <i class="far fa-edit mr-1"></i>Edit
                    </a> |
                    <a asp-action="destroy" asp-controller="department"
asp-route-id="@dep.Id">
                        <i class="far fa-trash-alt mr-1"></i>
                        Delete
                    </a>
                </td>
            </tr>
        }
    }
</tbody>
</table>

</div>
</div>

```

Edit.cshtml

```

@model Department
@{
    ViewData["Title"] = "Index";
}

@if (Model != null)
{
    <form asp-action="update" asp-controller="department">
        <div class="card">
            <div class="card-header">
                <b>Edit</b>
            </div>
            <div class="card-body">
                <input type="hidden" asp-for="Id" />
                <div class="form-group frow">

```

សកលវិទ្យាល័យបៀលប្រាស

```

        <label asp-for="DepartmentName" class="col-form-label-sm
col-sm-2"></label>
        <div class="col-sm-10">
            <input asp-for="DepartmentName" class="form-control" />
            <span asp-validation-for="DepartmentName" class="text-
danger"></span>
        </div>
    </div>
</div>
<div class="card-footer">
    <button type="submit" class="btn btn-warning">Update</button>
    <a asp-action="List" asp-controller="department">
        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
    </a>
</div>
</div>
</div>
</form>
}

```

- Employee Directory

Create.cshtml

```

@model Employee
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="store" asp-controller="employee" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <div class="form-group row">
                    <label asp-for="Name" class="col-form-label col-md-
2"></label>
                    <div class="col-md-10">
                        <input asp-for="Name" class="form-control" />
                        <span asp-validation-for="Name" class="text-
danger"></span>
                    </div>
                </div>
                <div class="form-group row">
                    <label asp-for="Gender" class="col-form-label col-md-
2"></label>
                    <div class="col-md-10">
                        <div>
                            <label asp-for="Gender" class="custom-radio">

```

សកលវិទ្យាល័យបៀលប្រាស

```
<input type="radio" asp-for="Gender"
value="female" />
    Female
</label>
<label asp-for="Gender">
    <input type="radio" asp-for="Gender"
value="male" />
    Male
</label>
</div>
<span asp-validation-for="Gender" class="text-
danger"></span>
</div>
</div>
<div class="form-group row">
    <label asp-for="DepartmentId" class="col-form-label col-md-
2"></label>
    <div class="col-md-10">
        <select asp-for="DepartmentId" class="custom-select"
asp-items="@((ViewBag.Department))">
            <option value="" disabled
selected>Department</option>
        </select>
        <span asp-validation-for="DepartmentId" class="text-
danger"></span>
    </div>
</div>
<div class="form-group row">
    <label asp-for="Address" class="col-form-label col-md-
2"></label>
    <div class="col-md-10">
        <input asp-for="Address" class="form-control" />
        <span asp-validation-for="Address" class="text-
danger"></span>
    </div>
</div>
</div>
<div class="card-footer">
    <button type="submit" class="btn btn-primary">Create</button>
    <a asp-action="Index" asp-controller="employee">
        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
    </a>
</div>
</div>
</div>
</form>
</div>
Delete.cshtml

@model EmployeeViewModel
@{
    ViewData["Title"] = "Delete";
}
```

សកលវិទ្យាល័យហ្សឺនឡា

```
<div class="card">
  <div class="card-header">
    <strong>Confirm?</strong>
  </div>
  <div class="card-body">
    <div class="alert alert-danger" role="alert">
      <p>Do you want to delete
        &nbsp;&nbsp;&nbsp;&nbsp;<strong>@Model.Name</strong>?</p>
      <form asp-action="destroy" method="post" asp-route-id="@Model.Id">
        <button type="submit" class="btn btn-danger">Delete</button>
        <a asp-action="Index" class="ml-2">
          <i class="fas fa-arrow-circle-left"></i>
          Back
        </a>
      </form>
    </div>
  </div>
</div>
```

Detailed.cshtml

```
@model EmployeeViewModel
```

```
@{
    ViewData["Title"] = "Detailed";
}
```

Employee information

```
<div>
<dl class="row">
  <dt class="col-sm-2">
    @Html.DisplayNameFor(emp => emp.Id)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(emp => emp.Id)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(emp => emp.Name)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(emp => emp.Name)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(emp => emp.Gender)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(emp => emp.Gender)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(emp => emp.Department)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(emp => emp.Department)
  </dd>
</div>
```


សកលវិទ្យាល័យបៀលប្រាស

```
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Address)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Address)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="Create">
        <i class="fas fa-plus-circle"></i>
        Create
    </a>
    |
    <a asp-action="Index">
        <i class="fas fa-arrow-circle-left"></i>
        Back To List
    </a>
</div>
```

Edit.cshtml

```
@model Employee
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="update" asp-controller="employee" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <input type="hidden" asp-for="Id" />
                <div class="form-group row">
                    <label asp-for="Name" class="col-form-label col-md-2"></label>
                    <div class="col-md-10">
                        <input asp-for="Name" class="form-control" />
                        <span asp-validation-for="Name" class="text-danger"></span>
                    </div>
                </div>
                <div class="form-group row">
                    <label asp-for="Gender" class="col-form-label col-md-2"></label>
                    <div class="col-md-10">
                        <div>
                            <label asp-for="Gender" class="custom-radio">
                                <input type="radio" asp-for="Gender"
value="femlae" />
```

សកលវិទ្យាល័យបៀលប្រាស

```

        Female
        </label>
        <label asp-for="Gender">
            <input type="radio" asp-for="Gender"
value="male" />
        Male
        </label>
    </div>
    <span asp-validation-for="Gender" class="text-
danger"></span>
</div>
</div>
<div class="form-group row">
    <label asp-for="DepartmentId" class="col-form-label col-md-
2"></label>
    <div class="col-md-10">
        <select asp-for="DepartmentId" class="custom-select"
asp-items="@ (ViewBag.Departments)">
            <option value="" disabled
selected>Department</option>
        </select>
        <span asp-validation-for="DepartmentId" class="text-
danger"></span>
    </div>
</div>
<div class="form-group row">
    <label asp-for="Address" class="col-form-label col-md-
2"></label>
    <div class="col-md-10">
        <input asp-for="Address" class="form-control" />
        <span asp-validation-for="Address" class="text-
danger"></span>
    </div>
</div>
</div>
<div class="card-footer">
    <button type="submit" class="btn btn-primary">Create</button>
    <a asp-action="Index" asp-controller="employee">
        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
    </a>
</div>
</div>
</div>
</form>
</div>
```

Index.cshtml

```
@model IEnumerable<EmployeeViewModel>

@{
    ViewData["Title"] = "Index";
}
<div class="card">
```

```

<div class="card-header">
    <strong>Employee Information</strong>
    <a asp-action="Create" class="float-right">Create</a>
</div>
<div class="card-body">
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>@Html.DisplayNameFor(emp => emp.Name)</th>
                <th>@Html.DisplayNameFor(emp => emp.Gender)</th>
                <th>@Html.DisplayNameFor(emp => emp.Department)</th>
                <th>@Html.DisplayNameFor(emp => emp.Address)</th>
                <th>@Html.DisplayName("Actions")</th>
            </tr>
        </thead>
        <tbody>
            @if (Model.Count() > 0 || Model != null)
            {
                @foreach (var emp in Model)
                {
                    <tr>
                        <td>@Html.DisplayFor(em => emp.Name)</td>
                        <td>@Html.DisplayFor(em => emp.Gender)</td>
                        <td>@Html.DisplayFor(em => emp.Department)</td>
                        <td>@Html.DisplayFor(em => emp.Address)</td>
                        <td>
                            <a asp-action="Detailed" asp-route-id="@emp.Id">
                                <i class="far fa-eye mr-1"></i>
                                Detail
                            </a> |
                            <a asp-action="Edit" asp-route-id="@emp.Id">
                                <i class="far fa-edit mr-1"></i>Edit
                            </a> |
                            <a asp-action="Delete" asp-route-id="@emp.Id">
                                <i class="far fa-trash-alt mr-1"></i>
                                Delete
                            </a>
                        </td>
                    </tr>
                }
            }
            else
            {
                <tr>
                    <td colspan="5">
                        <h4>there is no data</h4>
                    </td>
                </tr>
            }
        </tbody>
    </table>
</div>
</div>

```

- Share Directory

_Layout.cshtml

```
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>
    <link href="~/bootstrap/css/bootstrap.css" rel="stylesheet" />
    <link href="~/font-awesome/css/all.css" rel="stylesheet" />
</head>
<body>
    <partial name="_Navbar" />
    <div class="container-fluid">
        @RenderBody()
    </div>
    <script src="~/jquery/jquery.js"></script>
    <script src="~/bootstrap/js/bootstrap.js"></script>
    <script src="~/font-awesome/js/all.js"></script>
</body>
</html>
```

_Navbar.cshtml

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
                <a class="nav-link" href="#">Home <span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link"
                    asp-action="Index" asp-controller="employee">Employee</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" asp-action="list" asp-
controller="department">Department</a>
            </li>
        </ul>
        <form class="form-inline my-2 my-lg-0">
            <input class="form-control mr-sm-2" type="search"
placeholder="Search" aria-label="Search">
        </form>
    </div>
</div>
```

សកលវិទ្យាល័យបៀលប្រាស

```
        <button class="btn btn-outline-success my-2 my-sm-0"
type="submit">Search</button>
    </form>
</div>
</nav>
```

_ViewImports.cshtml

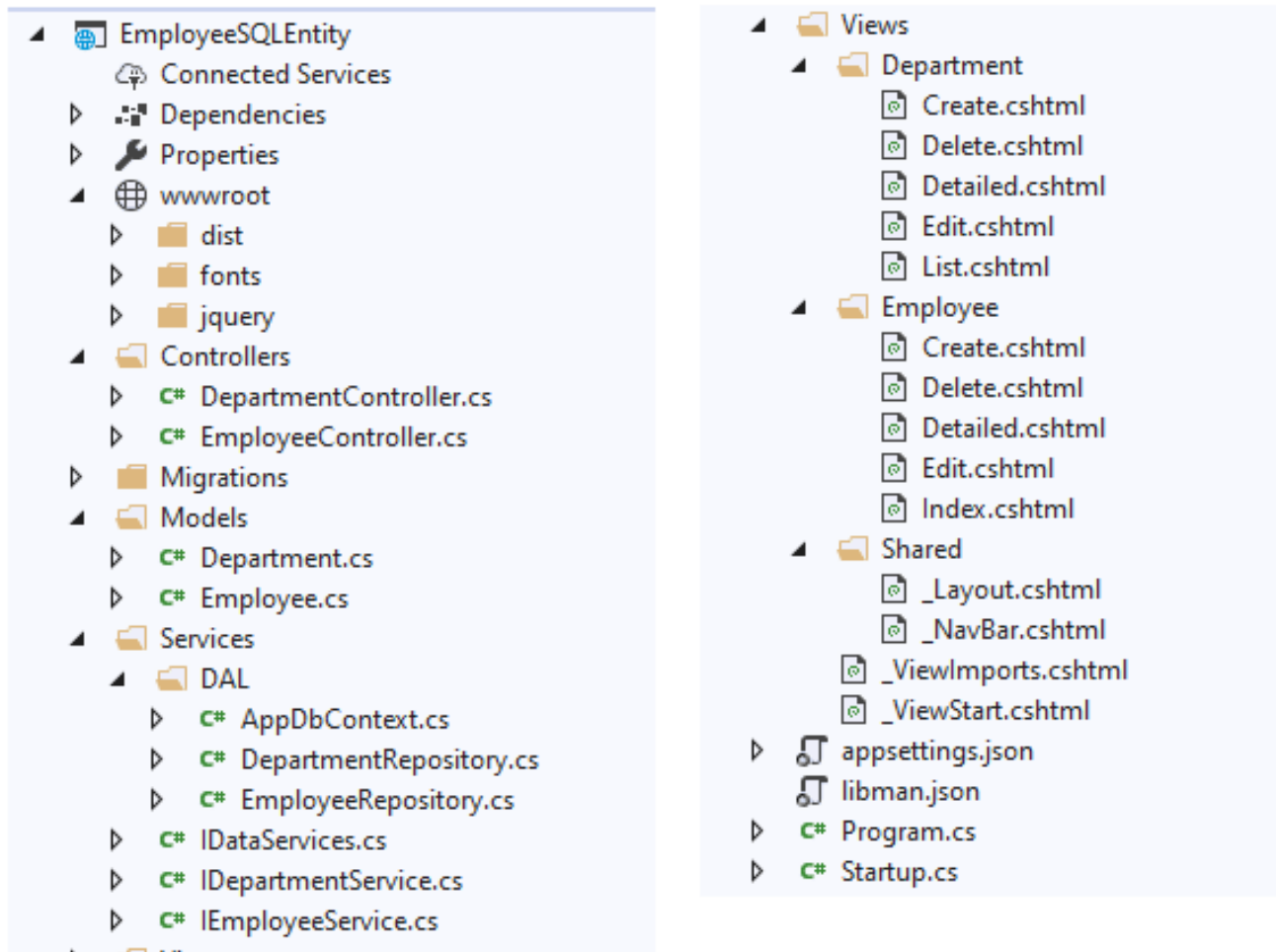
```
@using EmployeeSQL.Models;
@using EmployeeSQL.ViewModels;
@addTagHelper *,Microsoft.AspNetCore.Mvc.TagHelpers
```

_ViewStart.cshtml

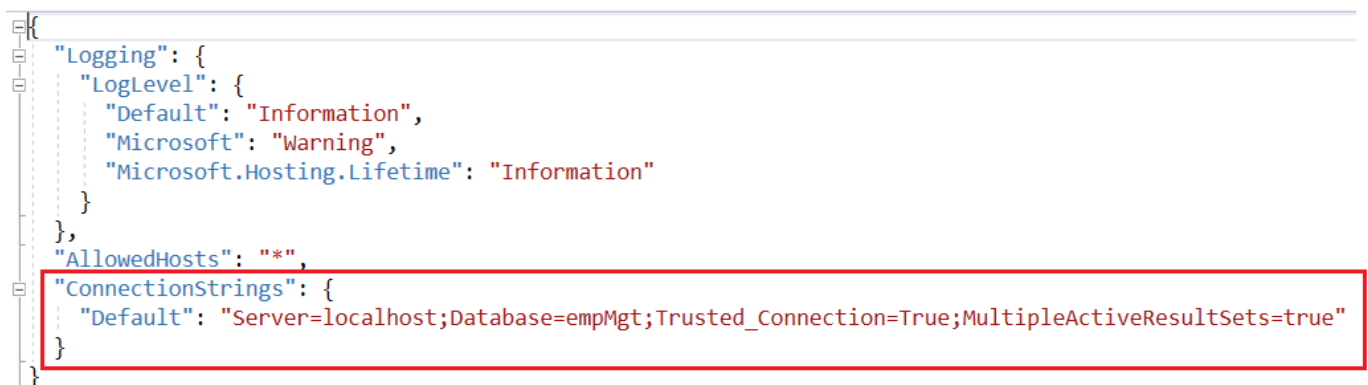
```
@{
    Layout = "_Layout";
}
```

IV. ASP.NET និង Entity Framework(Ms. SQL Server)

- បង្កើតProjectនិងឈ្មោះដូចខាងក្រោម៖

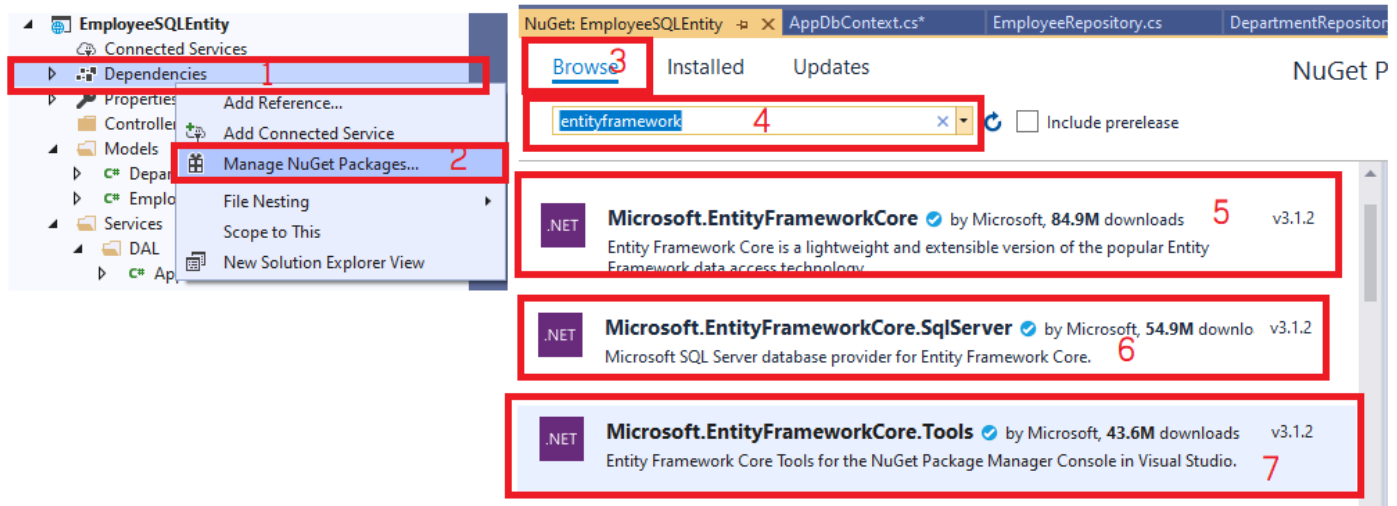


→ Configure connectionString ក្នុង appsetting.json



→install Package(right click on dependency)

សកលវិទ្យាល័យបៀលប្រាស



→Startup.cs

```
using EmployeesSQL.Services.DAL;
using EmployeesSQLEntity.Services;
using EmployeesSQLEntity.Services.DAL;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;

namespace EmployeesSQLEntity
{
    public class Startup
    {
        private readonly IConfiguration configuration;

        public Startup(IConfiguration configuration)
        {
            this.configuration = configuration;
        }

        public void ConfigureServices(IServiceCollection services)
        {
            services.AddControllersWithViews();
            services.AddDbContext<AppDbContext>(
                options =>
                options.UseMySQL(configuration.GetConnectionString("Connection"))
            );
            services.AddScoped<IEmployeeService, EmployeeRepository>();
            services.AddScoped<IDepartmentService, DepartmentRepository>();
        }

        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
        {
            if (env.IsDevelopment())
```

```
        {
            app.UseDeveloperExceptionPage();
        }
        app.UseStaticFiles();

        app.UseRouting();

        app.UseEndpoints(endpoints =>
        {
            endpoints.MapControllerRoute("default",
                "{controller}/{action}/{id?}",
                new { controller = "Employee", action = "Index" });
        });
    }
}
```

→Model Directory

Department.cs

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace EmployeeMySQLEntity.Models
{
    public class Department
    {
        public Department()
        {
            Employees = new HashSet<Employee>();
        }
        public int Id { get; set; }

        [Required]
        [Display(Name = "Department")]
        public string DepartmentName { get; set; }

        public ICollection<Employee> Employees { get; set; }
    }
}
```

Employee.cs

```
using System.ComponentModel.DataAnnotations;

namespace EmployeeMySQLEntity.Models
{
    public class Employee
    {
        public int Id { get; set; }
        [Required]
        public string Name { get; set; }
    }
}
```

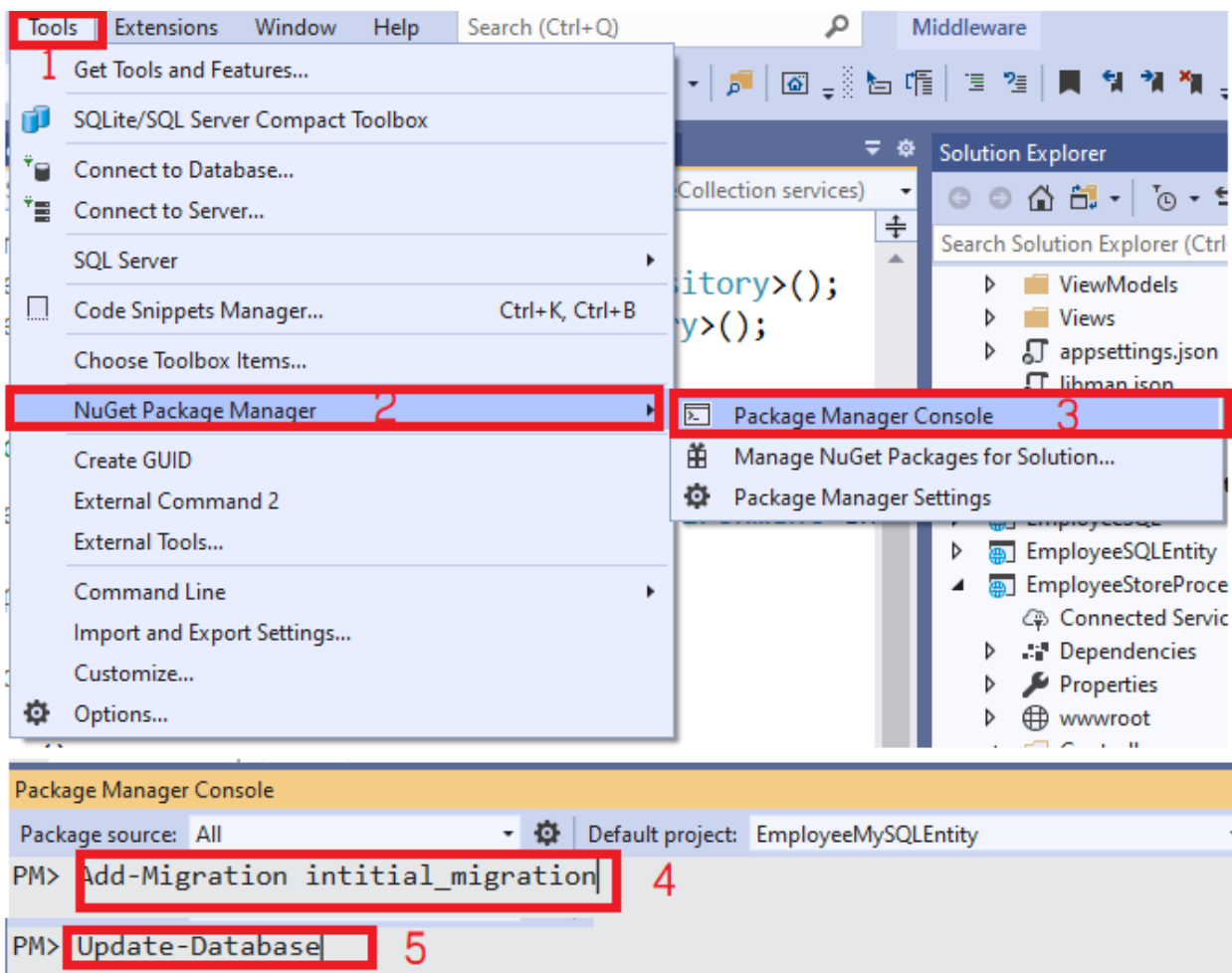

សកលវិទ្យាល័យបៀលប្រាស

```
[Required]
public string Gender { get; set; }
[Required]
[Display(Name = "Department")]
public int DepartmentId { get; set; }
[Required]
public string Address { get; set; }

public Department Department { get; set; }
}
}
```

→បង្កើត migrationនិងជាតាមស

- ចូល tools→NuGet Package Manager→Package manager console
- ក្នុងconsole វាយ command ដូចខាងក្រោម៖
 - Add-Migration <your_migration_name> រួច enter
 - Update-database រួចenter



→Controllers Directory

DepartmentController.cs

```
using EmployeeMySQL.Models;
using EmployeeMySQL.Services;
using Microsoft.AspNetCore.Mvc;

namespace EmployeeMySQL.Controllers
{
    public class DepartmentController : Controller
    {
        private readonly IDepartmentService _departmentService;
        public DepartmentController(IDepartmentService departmentService)
        {
            this._departmentService = departmentService;
        }
        [ActionName("List")]
        public IActionResult Index()
        {
            var deps = _departmentService.Get();
            return View("List", deps);
        }
        public IActionResult Create()
        {
            return View();
        }

        [HttpPost]
        public IActionResult store([Bind("DepartmentName")] Department dep)
        {
            if (ModelState.IsValid)
            {
                if (_departmentService.Save(dep))
                {
                    return RedirectToAction("List");
                }
            }
            return View("create", dep);
        }

        public IActionResult Edit(int? id)
        {
            var dep = _departmentService.Get(id.Value);
            if (dep != null)
            {
                return View(dep);
            }
            else
            {
                return NotFound();
            }
        }

        [HttpPost]
```

```
public IActionResult Update([Bind("Id", "DepartmentName")]
Department dep)
{
    if (ModelState.IsValid)
    {
        if (_departmentService.Update(dep))
        {
            return RedirectToAction("List");
        }
    }
    return View("create", dep);
}

[HttpPost]
public IActionResult Destroy(int? id)
{
    if (_departmentService.Delete(id.Value))
    {
        return RedirectToAction("List");
    }
    return NotFound();
}

[HttpGet]
public IActionResult Delete(int? id)
{
    var dep = _departmentService.Get(id.Value);
    if (dep != null)
    {
        return View("Delete", dep);
    }
    return NotFound();
}

public IActionResult Detailed(int? id)
{
    var dep = _departmentService.Get(id.Value);
    if (dep != null)
    {
        return View("Detailed", dep);
    }
    return NotFound();
}
}
```

EmployeeController.cs

```
using System.Linq;
using EmployeeMySQL.Models;
using EmployeeMySQL.Services;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
```

សកលវិទ្យាល័យបៀលប្រាស

```
namespace EmployeeMySQL.Controllers
{
    public class EmployeeController : Controller
    {
        private readonly IEmployeeService _empService;
        private readonly IDepartmentService _departmentService;
        public EmployeeController(IEmployeeService iempService,
IDepartmentService departmentService)
        {
            this._empService = iempService;
            this._departmentService = departmentService;
        }
        public IActionResult Index()
        {
            var emps = _empService.AllEmployees();
            return View("Index", emps);
        }

        [HttpGet]
        public IActionResult Create()
        {
            ViewBag.Departments = _departmentService.Get().Select(dep => new
SelectListItem
            {
                Text = dep.DepartmentName,
                Value = dep.Id.ToString()
            });
            return View();
        }
        [HttpPost]
        public IActionResult Store(Employee emp)
        {
            if (ModelState.IsValid)
            {
                if (_empService.Save(emp))
                {
                    return RedirectToAction("Index");
                }
            }
            ViewBag.Departments = _departmentService.Get().Select(dep => new
SelectListItem
            {
                Text = dep.DepartmentName,
                Value = dep.Id.ToString()
            });
            return View("Create", emp);
        }

        public IActionResult Edit(int Id)
        {
            var emp = _empService.Get(Id);
            if (emp != null)
            {
```

```
new ViewBag.Departments = _departmentService.Get().Select(dep =>
    SelectListItem
    {
        Text = dep.DepartmentName,
        Value = dep.Id.ToString(),
    });
    return View(emp);
}
else
    return NotFound();
}
[HttpPost]
public IActionResult Update(Employee emp)
{
    if (ModelState.IsValid)
    {
        if (_empService.Update(emp))
        {
            return RedirectToAction("Index");
        }
    }
    return View(emp);
}

public IActionResult Detailed(int Id)
{
    var emp = _empService.GetEmployee(Id);
    if (emp != null)
    {
        return View(emp);
    }
    else
        return NotFound();
}

public IActionResult Delete(int Id)
{
    var emp = _empService.GetEmployee(Id);
    if (emp != null)
    {
        return View(emp);
    }
    else
        return NotFound();
}

public IActionResult Destroy(int Id)
{
    var emp = _empService.Delete(Id);
    if (emp)
    {
        return RedirectToAction("Index");
    }
    else
        return NotFound();
}
}
```

➔ Services Directory

IDataService.cs

```
using System.Collections.Generic;

namespace EmployeeMySQLEntity.Services
{
    public interface IDataServices<T> where T:class
    {
        bool Save(T TEntity);
        bool Update(T TEntity);
        bool Delete(T TEntity);
        T Get(int Id);
        List<T> Gets(bool include=false);
    }
}
```

IDepartmentService.cs

```
using EmployeeMySQLEntity.Models;

namespace EmployeeMySQLEntity.Services
{
    public interface IDepartmentService:IDataServices<Department>
    {
    }
}
```

IEmployeeServices

```
using EmployeeMySQLEntity.Models;

namespace EmployeeMySQLEntity.Services
{
    public interface IEmployeeService:IDataServices<Employee>
    {
    }
}
```

○ DAL Directory

AppDbContext.cs

```
using EmployeeMySQLEntity.Models;
using Microsoft.EntityFrameworkCore;

namespace EmployeeMySQLEntity.Services.DAL
{
    public class AppDbContext:DbContext
```

សកលវិទ្យាល័យបៀលប្រាស

```
{
    public AppDbContext(DbContextOptions<AppDbContext>
options):base(options)
    {
        }
        public DbSet<Employee> Employees { get; set; }
        public DbSet<Department> Departments { get; set; }
    }
}
```

DepartmentRepository.cs

```
using EmployeeMySQLEntity.Models;
using Microsoft.EntityFrameworkCore;
using System.Collections.Generic;
using System.Linq;

namespace EmployeeMySQLEntity.Services.DAL
{
    public class DepartmentRepository : IDepartmentService
    {
        private readonly AppDbContext _context;
        public DepartmentRepository(AppDbContext context)
        {
            _context = context;
        }
        public bool Delete(Department dep)
        {
            if(dep !=null)
                _context.Departments.Remove(dep);
            return _context.SaveChanges() > 0;
        }

        public Department Get(int Id)
        {
            var dep = _context.Departments.Find(Id);
            return dep;
        }

        public List<Department> Gets(bool include=false)
        {
            var deps = _context.Departments.ToList();
            if (include)
            {
                deps =
                _context.Departments.Include(emp=>emp.Employees).ToList();
            }
            return deps;
        }

        public bool Save(Department TEntity)
        {
            }
```

```
        if (TEntity != null)
            _context.Departments.Add(TEntity);
        return _context.SaveChanges() > 0;
    }

    public bool Update(Department TEntity)
    {
        if (TEntity != null)
            _context.Departments.Update(TEntity);
        return _context.SaveChanges() > 0;
    }
}
```

EmployeeRepository.cs

```
using System.Collections.Generic;
using System.Linq;
using Microsoft.EntityFrameworkCore;
using EmployeeMySQLEntity.Services.DAL;
using EmployeeMySQLEntity.Models;
using EmployeeMySQLEntity.Services;

namespace EmployeeMySQL.Services.DAL
{
    public class EmployeeRepository : IEmployeeService
    {
        private readonly AppDbContext _context;
        public EmployeeRepository(AppDbContext context)
        {
            _context = context;
        }

        public bool Delete(Employee emp)
        {
            if(emp !=null)
                _context.Employees.Remove(emp);
            return _context.SaveChanges() > 0;
        }

        public Employee Get(int Id)
        {
            var emp = _context.Employees.Find(Id);
            return emp;
        }

        public List<Employee> Gets(bool include=false)
        {
            var emps = _context.Employees.ToList();
            if (include)
            {
                emps = _context.Employees.Include(dep =>
dep.Department).ToList();
            }
        }
    }
}
```



```
        return emps;
    }

    public bool Save(Employee TEntity)
    {
        if (TEntity != null)
            _context.Employees.Add(TEntity);
        return _context.SaveChanges() > 0;
    }

    public bool Update(Employee TEntity)
    {
        if (TEntity != null)
            _context.Employees.Update(TEntity);
        return _context.SaveChanges() > 0;
    }
}
```

→ Views Directory

→ Department folder

Create.cshtml

```
@model Department
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="store" asp-controller="department" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <div class="form-group row">
                    <label asp-for="DepartmentName" class="col-form-label col-
md-2"></label>
                    <div class="col-md-10">
                        <input asp-for="DepartmentName" class="form-control" />
                        <span asp-validation-for="DepartmentName" class="text-
danger"></span>
                    </div>
                </div>
            </div>
            <div class="card-footer">
                <button type="submit" class="btn btn-primary">Create</button>
                <a asp-action="List" asp-controller="department">
```

សកលវិទ្យាល័យបៀលប្រាស

`<i class="fas fa-arrow-circle-left ml-2 mr-1"></i>`Back To

List

```
</a>
</div>
</div>
</form>
</div>
```

Detailed.cshtml

```
@model Department
@{
    ViewData["Title"] = "Detailed";
}

<div>
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(d => d.Id)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(d => d.Id)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(d => d.DepartmentName)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(d => d.DepartmentName)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="edit" asp-route-id="@Model.Id">Edit</a>
    |
    <a asp-action="List">Back to List</a>
</div>
```

Delete.cshtml

```
@model Department
@{
    ViewData["Title"] = "Delete";
}

<div class="container mt-4">
    <div class="row">
        <div class="col col-sm-8 offset-2">
            <div class="alert alert-danger" role="alert">
                Do want to delete <strong>@Model.DepartmentName </strong>?
            </div>
            <div>
                <form asp-action="delete" asp-controller="department"
method="post">
                    <input type="hidden" asp-for="Id" />
```

សកលវិទ្យាល័យបៀលប្រាស

```

        <input type="submit" value="Delete" class="btn btn-danger"
/>
        <a asp-action="List" asp-controller="department">
            <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back
To List
        </a>
    </form>
</div>
</div>
</div>
</div>

```

</div>

List.html

```

@model IEnumerable<Department>
@{
    ViewData["Title"] = "List";
}
<div class="card">
    <div class="card-header">
        <strong>All</strong>
        <a asp-action="create" asp-controller="department" class="float-
right">Create</a>
    </div>
    <div class="card-body">
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>@Html.DisplayNameFor(d=>d.DepartmentName)</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody>
                @if (Model.Count() > 0)
                {
                    @foreach (var dep in Model)
                    {
                        <tr>
                            <td>@dep.DepartmentName</td>
                            <td>
                                <a asp-action="detailed" asp-controller="department"
asp-route-id="@dep.Id">
                                    <i class="far fa-eye mr-1"></i>Detail
                                </a> |
                                <a asp-action="edit" asp-controller="department" asp-
route-id="@dep.Id">
                                    <i class="far fa-edit mr-1"></i>Edit
                                </a> |
                                <a asp-action="destroy" asp-controller="department"
asp-route-id="@dep.Id">
                                    <i class="far fa-trash-alt mr-1"></i>
Delete
                                </a>
                            </td>
                        </tr>
                    }
                }
            </tbody>
        </table>
    </div>
</div>

```

សកលវិទ្យាល័យបៀលប្រាស

```
        </td>
      </tr>
    }
  }
</tbody>
</table>
```

```
</div>
</div>
```

Edit.cshtml

```
@model Department
@{
    ViewData["Title"] = "Index";
}

@if(Model != null)
{
    <form asp-action="update" asp-controller="department">
        <div class="card">
            <div class="card-header">
                <b>Edit</b>
            </div>
            <div class="card-body">
                <input type="hidden" asp-for="Id" />
                <div class="form-group frow">
                    <label asp-for="DepartmentName" class="col-form-label-sm
col-sm-2"></label>
                    <div class="col-sm-10">
                        <input asp-for="DepartmentName" class="form-control" />
                        <span asp-validation-for="DepartmentName" class="text-
danger"></span>
                    </div>
                </div>
            </div>
            <div class="card-footer">
                <button type="submit" class="btn btn-warning">Update</button>
                <a asp-action="List" asp-controller="department">
                    <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
                </a>
            </div>
        </div>
    </form>
}
```

- Employee Directory

Create.cshtml

សកលវិទ្យាល័យបៀលប្រាស

```
@model Employee
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="store" asp-controller="employee" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <div class="form-group row">
                    <label asp-for="Name" class="col-form-label col-md-2"></label>
                    <div class="col-md-10">
                        <input asp-for="Name" class="form-control" />
                        <span asp-validation-for="Name" class="text-danger"></span>
                    </div>
                </div>
                <div class="form-group row">
                    <label asp-for="Gender" class="col-form-label col-md-2"></label>
                    <div class="col-md-10">
                        <div>
                            <label asp-for="Gender" class="custom-radio">
                                <input type="radio" asp-for="Gender"
value="female" />
                                Female
                            </label>
                            <label asp-for="Gender">
                                <input type="radio" asp-for="Gender"
value="male" />
                                Male
                            </label>
                        </div>
                        <span asp-validation-for="Gender" class="text-danger"></span>
                    </div>
                </div>
                <div class="form-group row">
                    <label asp-for="DepartmentId" class="col-form-label col-md-2"></label>
                    <div class="col-md-10">
                        <select asp-for="DepartmentId" class="custom-select"
asp-items="@((ViewBag.Department))">
                            <option value="" disabled
selected>Department</option>
                        </select>
                        <span asp-validation-for="DepartmentId" class="text-danger"></span>
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>
```

សកលវិទ្យាល័យហ្សេលាប្រាសាទ

```
<label asp-for="Address" class="col-form-label col-md-2"></label>
<div class="col-md-10">
  <input asp-for="Address" class="form-control" />
  <span asp-validation-for="Address" class="text-danger"></span>
</div>
</div>
<div class="card-footer">
  <button type="submit" class="btn btn-primary">Create</button>
  <a asp-action="Index" asp-controller="employee">
    <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
  </a>
</div>
</div>
</form>
</div>
```

Delete.cshtml

```
@model EmployeeViewModel
@{
  ViewData["Title"] = "Delete";
}

<div class="card">
  <div class="card-header">
    <strong>Confirm?</strong>
  </div>
  <div class="card-body">
    <div class="alert alert-danger" role="alert">
      <p>Do you want to delete
      &nbsp;&nbsp;&nbsp;<strong>@Model.Name</strong>?</p>
      <form asp-action="destroy" method="post" asp-route-id="@Model.Id">
        <button type="submit" class="btn btn-danger">Delete</button>
        <a asp-action="Index" class="ml-2">
          <i class="fas fa-arrow-circle-left"></i>
          Back
        </a>
      </form>
    </div>
  </div>
</div>
</div>
```

Detailed.cshtml

```
@model EmployeeViewModel
@{
  ViewData["Title"] = "Detailed";
}

<h3>Employee information</h3>
```

សកលវិទ្យាល័យបៀលប្រាស

```
<div>
  <dl class="row">
    <dt class="col-sm-2">
      @Html.DisplayNameFor(emp => emp.Id)
    </dt>
    <dd class="col-sm-10">
      @Html.DisplayFor(emp => emp.Id)
    </dd>
    <dt class="col-sm-2">
      @Html.DisplayNameFor(emp => emp.Name)
    </dt>
    <dd class="col-sm-10">
      @Html.DisplayFor(emp => emp.Name)
    </dd>
    <dt class="col-sm-2">
      @Html.DisplayNameFor(emp => emp.Gender)
    </dt>
    <dd class="col-sm-10">
      @Html.DisplayFor(emp => emp.Gender)
    </dd>
    <dt class="col-sm-2">
      @Html.DisplayNameFor(emp => emp.Department)
    </dt>
    <dd class="col-sm-10">
      @Html.DisplayFor(emp => emp.Department)
    </dd>
    <dt class="col-sm-2">
      @Html.DisplayNameFor(emp => emp.Address)
    </dt>
    <dd class="col-sm-10">
      @Html.DisplayFor(emp => emp.Address)
    </dd>
  </dl>
</div>
<div>
  <a asp-action="Create">
    <i class="fas fa-plus-circle"></i>
    Create
  </a>
  |
  <a asp-action="Index">
    <i class="fas fa-arrow-circle-left"></i>
    Back To List
  </a>
</div>
```

Edit.cshtml

```
@model Employee
@{
    ViewData["Title"] = "Create";
```

សកលវិទ្យាល័យបៀលប្រាស

```
}
<div class="container">
  <form asp-action="update" asp-controller="employee" method="post">
    <div class="card">
      <div class="card-header">
        <strong>Create</strong>
      </div>
      <div class="card-body">
        <input type="hidden" asp-for="Id" />
        <div class="form-group row">
          <label asp-for="Name" class="col-form-label col-md-2"></label>
          <div class="col-md-10">
            <input asp-for="Name" class="form-control" />
            <span asp-validation-for="Name" class="text-danger"></span>
          </div>
        </div>
        <div class="form-group row">
          <label asp-for="Gender" class="col-form-label col-md-2"></label>
          <div class="col-md-10">
            <div>
              <label asp-for="Gender" class="custom-radio">
                <input type="radio" asp-for="Gender"
value="female" />
                Female
              </label>
              <label asp-for="Gender">
                <input type="radio" asp-for="Gender"
value="male" />
                Male
              </label>
            </div>
            <span asp-validation-for="Gender" class="text-danger"></span>
          </div>
        </div>
        <div class="form-group row">
          <label asp-for="DepartmentId" class="col-form-label col-md-2"></label>
          <div class="col-md-10">
            <select asp-for="DepartmentId" class="custom-select"
asp-items="@((ViewBag.Departments))">
              <option value="" disabled
selected>Department</option>
            </select>
            <span asp-validation-for="DepartmentId" class="text-danger"></span>
          </div>
        </div>
        <div class="form-group row">
          <label asp-for="Address" class="col-form-label col-md-2"></label>
```


សកលវិទ្យាល័យបៀលប្រាស

```

        <div class="col-md-10">
            <input asp-for="Address" class="form-control" />
            <span asp-validation-for="Address" class="text-
danger"></span>
        </div>
    </div>
</div>
<div class="card-footer">
    <button type="submit" class="btn btn-primary">Create</button>
    <a asp-action="Index" asp-controller="employee">
        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
    </a>
</div>
</div>
</form>
</div>

```

Index.cshtml

```

@model IEnumerable<EmployeeViewModel>

@{
    ViewData["Title"] = "Index";
}
<div class="card">
    <div class="card-header">
        <strong>Employee Information</strong>
        <a asp-action="Create" class="float-right">Create</a>
    </div>
    <div class="card-body">
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>@Html.DisplayNameFor(emp => emp.Name)</th>
                    <th>@Html.DisplayNameFor(emp => emp.Gender)</th>
                    <th>@Html.DisplayNameFor(emp => emp.Department)</th>
                    <th>@Html.DisplayNameFor(emp => emp.Address)</th>
                    <th>@Html.DisplayName("Actions")</th>
                </tr>
            </thead>
            <tbody>
                @if (Model.Count() > 0 || Model != null)
                {
                    @foreach (var emp in Model)
                    {
                        <tr>
                            <td>@Html.DisplayFor(em => emp.Name)</td>
                            <td>@Html.DisplayFor(em => emp.Gender)</td>
                            <td>@Html.DisplayFor(em => emp.Department)</td>
                            <td>@Html.DisplayFor(em => emp.Address)</td>
                            <td>
                                <a asp-action="Detailed" asp-route-id="@emp.Id">
                                    <i class="far fa-eye mr-1"></i>
                                </a>
                            </td>
                        </tr>
                    }
                }
            </tbody>
        </table>
    </div>
</div>

```

```

        Detail
    </a> |
    <a asp-action="Edit" asp-route-id="@emp.Id">
        <i class="far fa-edit mr-1"></i>Edit
    </a> |
    <a asp-action="Delete" asp-route-id="@emp.Id">
        <i class="far fa-trash-alt mr-1"></i>
        Delete
    </a>
</td>
</tr>
}
}
else
{
    <tr>
        <td colspan="5">
            <h4>there is no data</h4>
        </td>
    </tr>
}
</tbody>
</table>
</div>
</div>
```

- Share Directory

__Layout.cshtml

```
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>
    <link href="~/bootstrap/css/bootstrap.css" rel="stylesheet" />
    <link href="~/font-awesome/css/all.css" rel="stylesheet" />
</head>
<body>
    <partial name="_Navbar" />
    <div class="container-fluid">
        @RenderBody()
    </div>
    <script src="~/jquery/jquery.js"></script>
    <script src="~/bootstrap/js/bootstrap.js"></script>
    <script src="~/font-awesome/js/all.js"></script>
</body>
</html>
```

__NavBar.cshtml

សកលវិទ្យាល័យបៀលប្រាស

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-
only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link"
          asp-action="Index" asp-controller="employee">Employee</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" asp-action="list" asp-
controller="department">Department</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search"
placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success my-2 my-sm-0"
type="submit">Search</button>
    </form>
  </div>
</nav>
```

_ViewImports.cshtml

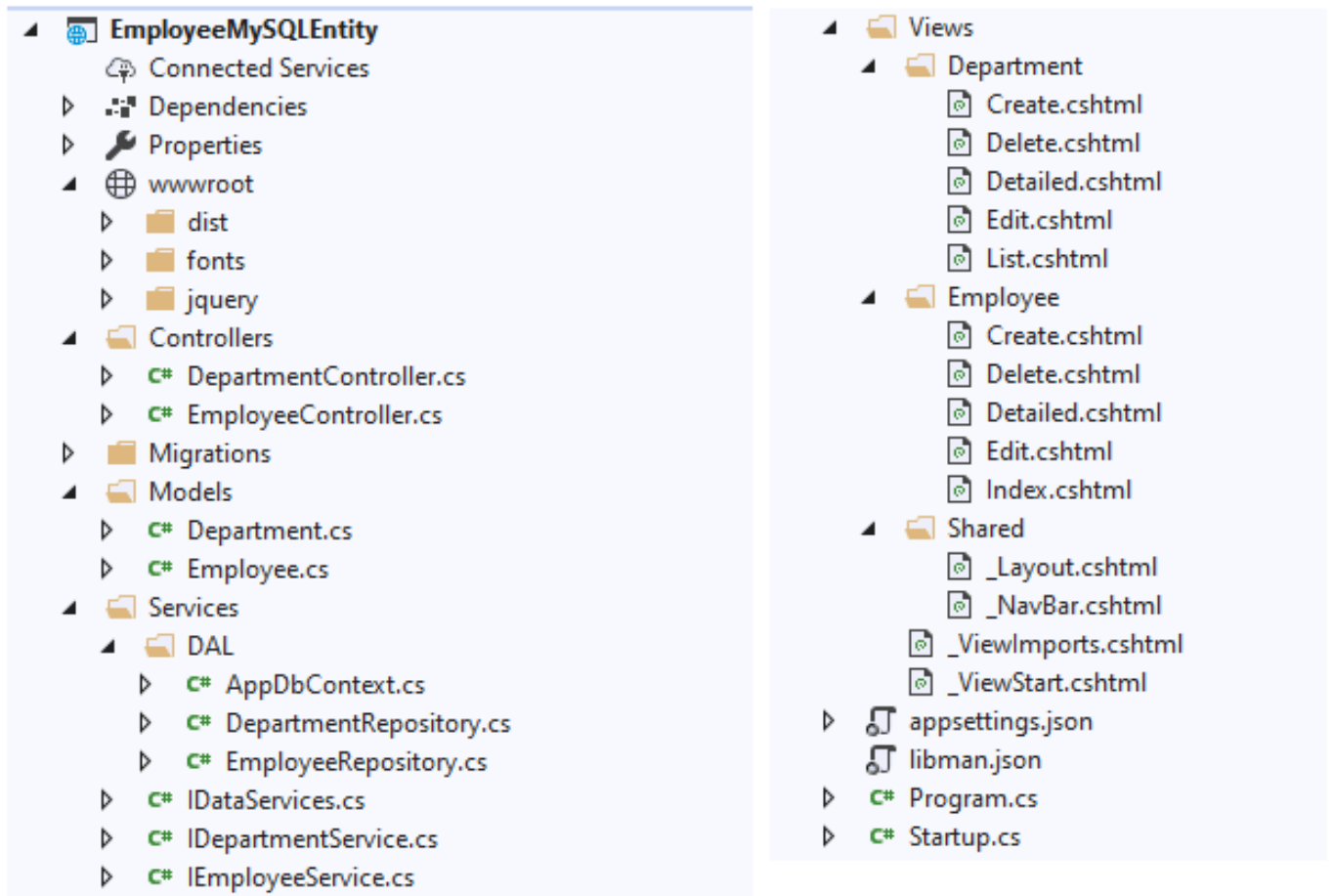
```
@using EmployeeSQL.Models;
@using EmployeeSQL.ViewModels;
@addTagHelper *,Microsoft.AspNetCore.Mvc.TagHelpers
```

_ViewStart.cshtml

```
@{
    Layout = "_Layout";
}
```

V. ASP.NET និង Entity Framework(MySQL Server)

- បង្កើត Project ដែលមានឈ្មោះនិងទំរង់ដូចខាងក្រោម៖

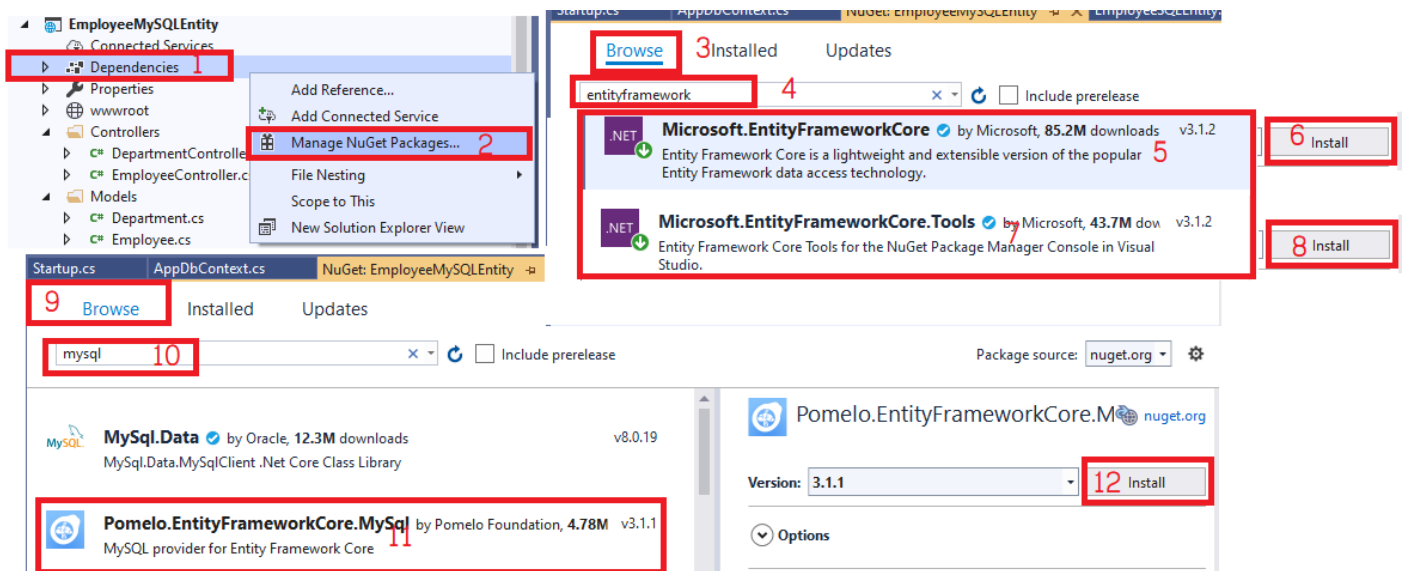


→ Configure connectionString ក្នុង appsetting.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "Connection": "Server=localhost;database=empMgts;user=root;password="
  }
}
```

សកលវិទ្យាល័យបៀលប្រាស

→install Package(right click on dependency)



→Startup.cs

```
using EmployeeMySQL.Services.DAL;
using EmployeeMySQLEntity.Services;
using EmployeeMySQLEntity.Services.DAL;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;

namespace EmployeeSQLEntity
{
    public class Startup
    {
        private readonly IConfiguration configuration;

        public Startup(IConfiguration configuration)
        {
            this.configuration = configuration;
        }

        public void ConfigureServices(IServiceCollection services)
        {
            services.AddControllersWithViews();
            services.AddDbContext<AppDbContext>(
                options =>
                options.UseMySQL(configuration.GetConnectionString("Connection"))
            );
            services.AddScoped<IEmployeeService, EmployeeRepository>();
            services.AddScoped<IDepartmentService, DepartmentRepository>();
        }
    }
}
```

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseStaticFiles();

    app.UseRouting();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute("default",
            "{controller}/{action}/{id?}",
            new { controller = "Employee", action = "Index" });
    });
}
}
```

→Model Directory

Department.cs

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace EmployeeMySQLEntity.Models
{
    public class Department
    {
        public Department()
        {
            Employees = new HashSet<Employee>();
        }
        public int Id { get; set; }

        [Required]
        [Display(Name = "Department")]
        public string DepartmentName { get; set; }

        public ICollection<Employee> Employees { get; set; }
    }
}
```

Employee.cs

```
using System.ComponentModel.DataAnnotations;

namespace EmployeeMySQLEntity.Models
{
    public class Employee
```

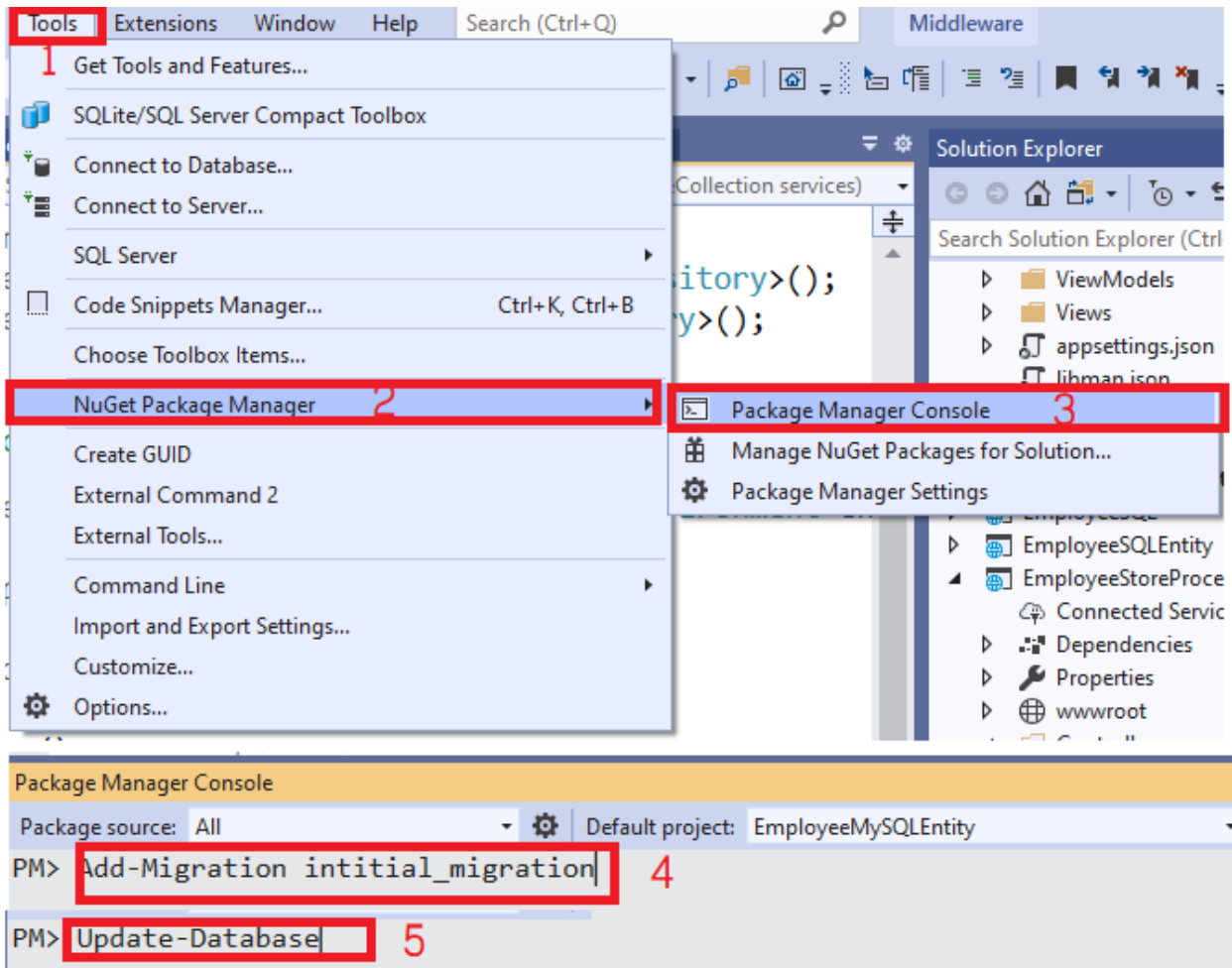
សកលវិទ្យាល័យបៀលប្រាស

```
{  
    public int Id { get; set; }  
    [Required]  
    public string Name { get; set; }  
    [Required]  
    public string Gender { get; set; }  
    [Required]  
    [Display(Name = "Department")]  
    public int DepartmentId { get; set; }  
    [Required]  
    public string Address { get; set; }  
  
    public Department Department { get; set; }  
}
```

→បង្កើត migrationនិងដាំតាបេស

- ចូល tools→NuGet Package Manager→Package manager console
- ក្នុងconsole វាយ command ដូចខាងក្រោម៖
 - o Add-Migration <your_migration_name> រួច enter
 - o Update-database រួចenter

សកលវិទ្យាល័យបៀលប្រាស



→ Services Directory

IDataService.cs

```
using System.Collections.Generic;

namespace EmployeeMySQLEntity.Services
{
    public interface IDataServices<T> where T: class
    {
        bool Save(T TEntity);
        bool Update(T TEntity);
        bool Delete(T TEntity);
        T Get(int Id);
        List<T> Gets(bool include=false);
    }
}
```

IDepartmentService.cs

```
using EmployeeMySQLEntity.Models;
```


សកលវិទ្យាល័យបៀលប្រាស

```
namespace EmployeeMySQLEntity.Services
{
    public interface IDepartmentService:IDataServices<Department>
    {
    }
}
```

IEmployeeServices

```
using EmployeeMySQLEntity.Models;
```

```
namespace EmployeeMySQLEntity.Services
{
    public interface IEmployeeService:IDataServices<Employee>
    {
    }
}
```

- DAL Directory

AppDbContext.cs

```
using EmployeeMySQLEntity.Models;
using Microsoft.EntityFrameworkCore;

namespace EmployeeMySQLEntity.Services.DAL
{
    public class AppDbContext:DbContext
    {
        public AppDbContext(DbContextOptions<AppDbContext>
options):base(options)
        {
        }
        public DbSet<Employee> Employees { get; set; }
        public DbSet<Department> Departments { get; set; }
    }
}
```

DepartmentRepository.cs

```
using EmployeeMySQLEntity.Models;
using Microsoft.EntityFrameworkCore;
using System.Collections.Generic;
using System.Linq;

namespace EmployeeMySQLEntity.Services.DAL
{
    public class DepartmentRepository : IDepartmentService
    {
    }
}
```

```
private readonly AppDbContext _context;
public DepartmentRepository(AppDbContext context)
{
    _context = context;
}
public bool Delete(Department dep)
{
    if(dep !=null)
        _context.Departments.Remove(dep);
    return _context.SaveChanges() > 0;
}

public Department Get(int Id)
{
    var dep = _context.Departments.Find(Id);
    return dep;
}

public List<Department> Gets(bool include=false)
{
    var deps = _context.Departments.ToList();
    if (include)
    {
        deps =
        _context.Departments.Include(emp=>emp.Employees).ToList();
    }
    return deps;
}

public bool Save(Department TEntity)
{
    if (TEntity != null)
        _context.Departments.Add(TEntity);
    return _context.SaveChanges() > 0;
}

public bool Update(Department TEntity)
{
    if (TEntity != null)
        _context.Departments.Update(TEntity);
    return _context.SaveChanges() > 0;
}
}
```

EmployeeRepository.cs

```
using System.Collections.Generic;
using System.Linq;
using Microsoft.EntityFrameworkCore;
using EmployeeMySQLEntity.Services.DAL;
using EmployeeMySQLEntity.Models;
using EmployeeMySQLEntity.Services;
```

សកលវិទ្យាល័យបៀលប្រាស

```
namespace EmployeeMySQL.Services.DAL
{
    public class EmployeeRepository : IEmployeeService
    {
        private readonly AppDbContext _context;
        public EmployeeRepository(AppDbContext context)
        {
            _context = context;
        }
        public bool Delete(Employee emp)
        {
            if(emp !=null)
                _context.Employees.Remove(emp);
            return _context.SaveChanges() > 0;
        }

        public Employee Get(int Id)
        {
            var emp = _context.Employees.Find(Id);
            return emp;
        }

        public List<Employee> Gets(bool include=false)
        {
            var emps = _context.Employees.ToList();
            if (include)
            {
                emps = _context.Employees.Include(dep =>
dep.Department).ToList();
            }
            return emps;
        }

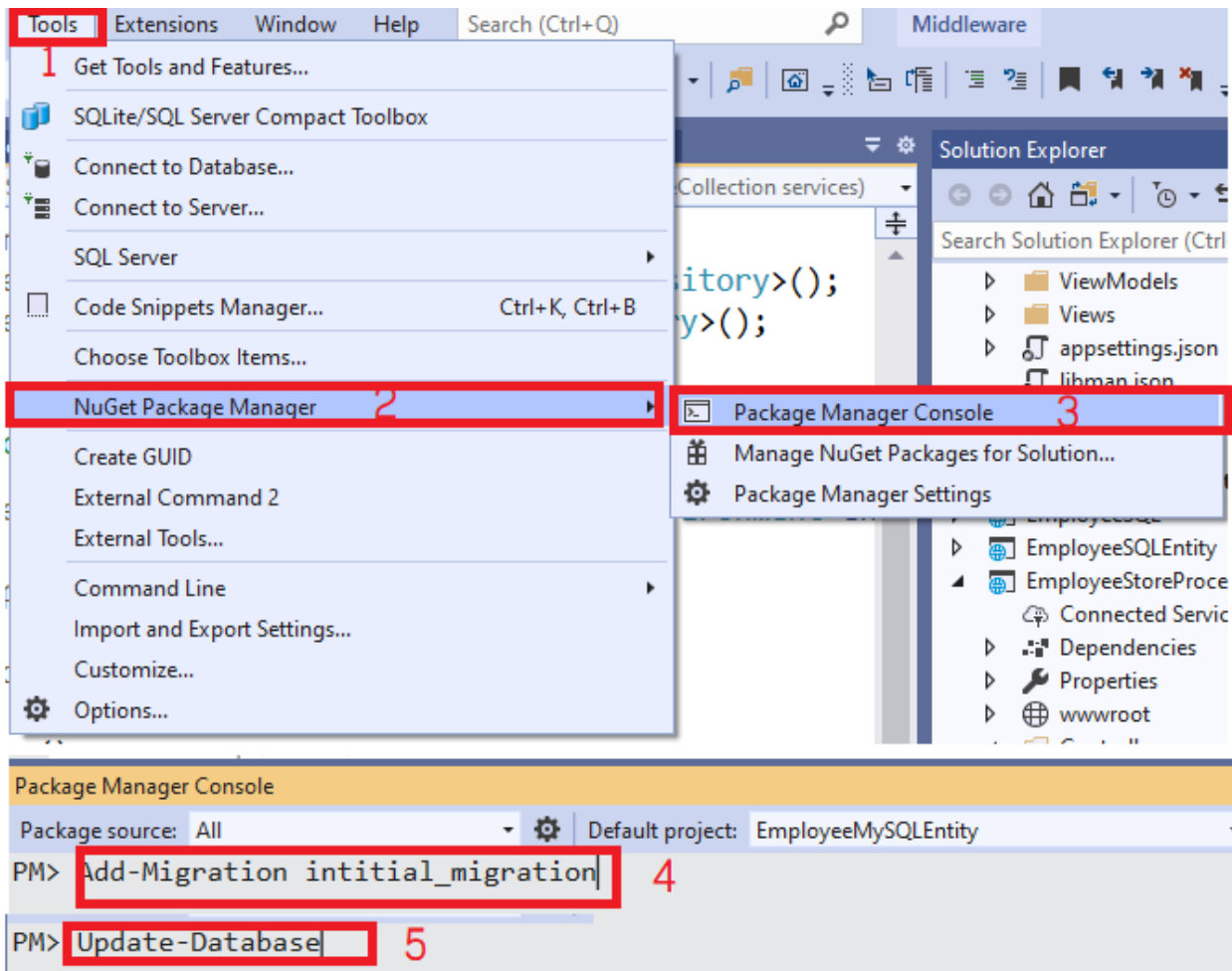
        public bool Save(Employee TEntity)
        {
            if (TEntity != null)
                _context.Employees.Add(TEntity);
            return _context.SaveChanges() > 0;
        }

        public bool Update(Employee TEntity)
        {
            if (TEntity != null)
                _context.Employees.Update(TEntity);
            return _context.SaveChanges() > 0;
        }
    }
}
```

→បង្កើត migrationនិងជាតារាងស

សកលវិទ្យាល័យបៀលប្រាស

- ចូល tools→NuGet Package Manager→Package manager console
- ក្នុងconsole វាយ command ដូចខាងក្រោម៖
 - Add-Migration <your_migration_name> រួច enter
 - Update-database រួចenter



→Controllers Directory

DepartmentController.cs

```
using EmployeeMySQL.Models;  
using EmployeeMySQL.Services;  
using Microsoft.AspNetCore.Mvc;  
  
namespace EmployeeMySQL.Controllers  
{  
    public class DepartmentController : Controller  
    {  
        private readonly IDepartmentService _departmentService;
```

```
public DepartmentController(IDepartmentService departmentService)
{
    this._departmentService = departmentService;
}
[ActionName("List")]
public IActionResult Index()
{
    var deps = _departmentService.Get();
    return View("List", deps);
}
public IActionResult Create()
{
    return View();
}

[HttpPost]
public IActionResult store([Bind("DepartmentName")] Department dep)
{
    if (ModelState.IsValid)
    {
        if (_departmentService.Save(dep))
        {
            return RedirectToAction("List");
        }
    }
    return View("create", dep);
}

public IActionResult Edit(int? id)
{
    var dep = _departmentService.Get(id.Value);
    if (dep != null)
    {
        return View(dep);
    }
    else
    {
        return NotFound();
    }
}

[HttpPost]
public IActionResult Update([Bind("Id", "DepartmentName")]
Department dep)
{
    if (ModelState.IsValid)
    {
        if (_departmentService.Update(dep))
        {
            return RedirectToAction("List");
        }
    }
    return View("create", dep);
}

[HttpPost]
public IActionResult Destroy(int? id)
{

```

```
        if (_departmentService.Delete(id.Value))
        {
            return RedirectToAction("List");
        }
        return NotFound();
    }

    [HttpGet]
    public IActionResult Delete(int? id)
    {
        var dep = _departmentService.Get(id.Value);
        if (dep != null)
        {
            return View("Delete", dep);
        }
        return NotFound();
    }

    public IActionResult Detailed(int? id)
    {
        var dep = _departmentService.Get(id.Value);
        if (dep != null)
        {
            return View("Detailed", dep);
        }
        return NotFound();
    }
}
}
```

EmployeeController.cs

```
using System.Linq;
using EmployeeMySQL.Models;
using EmployeeMySQL.Services;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;

namespace EmployeeMySQL.Controllers
{
    public class EmployeeController : Controller
    {
        private readonly IEmployeeService _empService;
        private readonly IDepartmentService _departmentService;
        public EmployeeController(IEmployeeService iempService,
IDepartmentService departmentService)
        {
            this._empService = iempService;
            this._departmentService = departmentService;
        }
        public IActionResult Index()
        {
            var emps = _empService.AllEmployees();
```

សកលវិទ្យាល័យបៀលប្រាស

```
        return View("Index", emps);
    }

    [HttpGet]
    public IActionResult Create()
    {
        ViewBag.Departments = _departmentService.Get().Select(dep => new
SelectListItem
    {
        Text = dep.DepartmentName,
        Value = dep.Id.ToString()
    });
        return View();
    }

    [HttpPost]
    public IActionResult Store(Employee emp)
    {
        if (ModelState.IsValid)
        {
            if (_empService.Save(emp))
            {
                return RedirectToAction("Index");
            }
        }
        ViewBag.Departments = _departmentService.Get().Select(dep => new
SelectListItem
    {
        Text = dep.DepartmentName,
        Value = dep.Id.ToString()
    });
        return View("Create", emp);
    }

    public IActionResult Edit(int Id)
    {
        var emp = _empService.Get(Id);
        if (emp != null)
        {
            ViewBag.Departments = _departmentService.Get().Select(dep =>
new
                SelectListItem
                {
                    Text = dep.DepartmentName,
                    Value = dep.Id.ToString(),
                });
            return View(emp);
        }
        else
            return NotFound();
    }

    [HttpPost]
    public IActionResult Update(Employee emp)
    {
        if (ModelState.IsValid)
```

សកលវិទ្យាល័យបៀលប្រាស

```
{
    if (_empService.Update(emp))
    {
        return RedirectToAction("Index");
    }
}
return View(emp);
}

public IActionResult Detailed(int Id)
{
    var emp = _empService.GetEmployee(Id);
    if (emp != null)
        return View(emp);
    else
        return NotFound();
}

public IActionResult Delete(int Id)
{
    var emp = _empService.GetEmployee(Id);
    if (emp != null)
        return View(emp);
    else
        return NotFound();
}

public IActionResult Destroy(int Id)
{
    var emp = _empService.Delete(Id);
    if (emp)
        return RedirectToAction("Index");
    else
        return NotFound();
}
}
}
```

➔ Views Directory

➔ _Department folder

Create.cshtml

```
@model Department
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="store" asp-controller="department" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
        </div>
    </form>
</div>
```


សកលវិទ្យាល័យបៀលប្រាស

```

    </div>
    <div class="card-body">
        <div class="form-group row">
            <label asp-for="DepartmentName" class="col-form-label col-
md-2"></label>
            <div class="col-md-10">
                <input asp-for="DepartmentName" class="form-control" />
                <span asp-validation-for="DepartmentName" class="text-
danger"></span>
            </div>
        </div>
    </div>
    <div class="card-footer">
        <button type="submit" class="btn btn-primary">Create</button>
        <a asp-action="List" asp-controller="department">
            <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
        </a>
    </div>
</div>
</form>
</div>

```

Detailed.cshtml

```

@model Department
@{
    ViewData["Title"] = "Detailed";
}

<div>
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(d => d.Id)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(d => d.Id)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(d => d.DepartmentName)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(d => d.DepartmentName)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="edit" asp-route-id="@Model.Id">Edit</a>
    |
    <a asp-action="List">Back to List</a>
</div>

```

សកលវិទ្យាល័យបៀលប្រាស

Delete.cshtml

```
@model Department
@{
    ViewData["Title"] = "Delete";
}
<div class="container mt-4">
    <div class="row">
        <div class="col col-sm-8 offset-2">
            <div class="alert alert-danger" role="alert">
                Do want to delete <strong>@Model.DepartmentName </strong>?
            </div>
            <div>
                <form asp-action="delete" asp-controller="department"
method="post">
                    <input type="hidden" asp-for="Id" />
                    <input type="submit" value="Delete" class="btn btn-danger"
/>

                    <a asp-action="List" asp-controller="department">
                        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back
To List
                </a>
            </form>
        </div>
    </div>
</div>
```

List.html

```
@model IEnumerable<Department>
@{
    ViewData["Title"] = "List";
}
<div class="card">
    <div class="card-header">
        <strong>All</strong>
        <a asp-action="create" asp-controller="department" class="float-
right">Create</a>
    </div>
    <div class="card-body">
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>@Html.DisplayNameFor(d=>d.DepartmentName)</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody>
                @if (Model.Count() > 0)
                {
                    @foreach (var dep in Model)
```

សកលវិទ្យាល័យបៀលប្រាស

```

        {
            <tr>
                <td>@dep.DepartmentName</td>
                <td>
                    <a asp-action="detailed" asp-controller="department"
asp-route-id="@dep.Id">
                        <i class="far fa-eye mr-1"></i>Detail
                    </a> |
                    <a asp-action="edit" asp-controller="department" asp-
route-id="@dep.Id">
                        <i class="far fa-edit mr-1"></i>Edit
                    </a> |
                    <a asp-action="destroy" asp-controller="department"
asp-route-id="@dep.Id">
                        <i class="far fa-trash-alt mr-1"></i>
                        Delete
                    </a>
                </td>
            </tr>
        }
    </tbody>
</table>

</div>
</div>

```

Edit.cshtml

```

@model Department
@{
    ViewData["Title"] = "Index";
}

@if (Model != null)
{
    <form asp-action="update" asp-controller="department">
        <div class="card">
            <div class="card-header">
                <b>Edit</b>
            </div>
            <div class="card-body">
                <input type="hidden" asp-for="Id" />
                <div class="form-group frow">
                    <label asp-for="DepartmentName" class="col-form-label-sm
col-sm-2"></label>
                    <div class="col-sm-10">
                        <input asp-for="DepartmentName" class="form-control" />
                        <span asp-validation-for="DepartmentName" class="text-
danger"></span>
                    </div>
                </div>
            </div>
        </div>
    </form>
}

```

```
        <div class="card-footer">
            <button type="submit" class="btn btn-warning">Update</button>
            <a asp-action="List" asp-controller="department">
                <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
            </a>
        </div>
    </div>
</div>

</form>
}
```

- Employee Directory

Create.cshtml

```
@model Employee
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="store" asp-controller="employee" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <div class="form-group row">
                    <label asp-for="Name" class="col-form-label col-md-
2"></label>
                    <div class="col-md-10">
                        <input asp-for="Name" class="form-control" />
                        <span asp-validation-for="Name" class="text-
danger"></span>
                    </div>
                </div>
                <div class="form-group row">
                    <label asp-for="Gender" class="col-form-label col-md-
2"></label>
                    <div class="col-md-10">
                        <div>
                            <label asp-for="Gender" class="custom-radio">
                                <input type="radio" asp-for="Gender"
value="femlae" />
                                Female
                            </label>
                            <label asp-for="Gender">
                                <input type="radio" asp-for="Gender"
value="male" />
                                Male
                            </label>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>
```

សកលវិទ្យាល័យបៀលប្រាស

```

        <span asp-validation-for="Gender" class="text-
danger"></span>
    </div>
</div>
<div class="form-group row">
    <label asp-for="DepartmentId" class="col-form-label col-md-
2"></label>
    <div class="col-md-10">
        <select asp-for="DepartmentId" class="custom-select"
asp-items="@ (ViewBag.Department)">
            <option value="" disabled
selected>Department</option>
        </select>
        <span asp-validation-for="DepartmentId" class="text-
danger"></span>
    </div>
</div>
<div class="form-group row">
    <label asp-for="Address" class="col-form-label col-md-
2"></label>
    <div class="col-md-10">
        <input asp-for="Address" class="form-control" />
        <span asp-validation-for="Address" class="text-
danger"></span>
    </div>
</div>
</div>
<div class="card-footer">
    <button type="submit" class="btn btn-primary">Create</button>
    <a asp-action="Index" asp-controller="employee">
        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
    </a>
</div>
</div>
</form>
</div>
Delete.cshtml

@model EmployeeViewModel
@{
    ViewData["Title"] = "Delete";
}

<div class="card">
    <div class="card-header">
        <strong>Confirm?</strong>
    </div>
    <div class="card-body">
        <div class="alert alert-danger role="alert">
            <p>Do you want to delete
            &nbsp;&nbsp;&nbsp;<strong>@Model.Name</strong>?</p>
            <form asp-action="destroy" method="post" asp-route-id="@Model.Id">

```

```
        <button type="submit" class="btn btn-danger">Delete</button>
        <a asp-action="Index" class="ml-2">
            <i class="fas fa-arrow-circle-left"></i>
            Back
        </a>
    </form>
</div>
</div>
</div>
```

Detailed.cshtml

```
@model EmployeeViewModel
@{
    ViewData["Title"] = "Detailed";
}

<h3>Employee information</h3>
<div>
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Id)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Id)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Name)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Name)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Gender)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Gender)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Department)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Department)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(emp => emp.Address)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(emp => emp.Address)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="Create">
```

សកលវិទ្យាល័យបៀលប្រាស

```
        <i class="fas fa-plus-circle"></i>
        Create
    </a>
    |
    <a asp-action="Index">
        <i class="fas fa-arrow-circle-left"></i>
        Back To List
    </a>
</div>
```

Edit.cshtml

```
@model Employee
@{
    ViewData["Title"] = "Create";
}
<div class="container">
    <form asp-action="update" asp-controller="employee" method="post">
        <div class="card">
            <div class="card-header">
                <strong>Create</strong>
            </div>
            <div class="card-body">
                <input type="hidden" asp-for="Id" />
                <div class="form-group row">
                    <label asp-for="Name" class="col-form-label col-md-2"></label>
                    <div class="col-md-10">
                        <input asp-for="Name" class="form-control" />
                        <span asp-validation-for="Name" class="text-danger"></span>
                    </div>
                </div>
                <div class="form-group row">
                    <label asp-for="Gender" class="col-form-label col-md-2"></label>
                    <div class="col-md-10">
                        <div>
                            <label asp-for="Gender" class="custom-radio">
                                <input type="radio" asp-for="Gender"
                                value="femlae" />
                                Female
                            </label>
                            <label asp-for="Gender">
                                <input type="radio" asp-for="Gender"
                                value="male" />
                                Male
                            </label>
                        </div>
                        <span asp-validation-for="Gender" class="text-danger"></span>
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>
```

សកលវិទ្យាល័យបៀលប្រាស

```

        </div>
    </div>
    <div class="form-group row">
        <label asp-for="DepartmentId" class="col-form-label col-md-
2"></label>
        <div class="col-md-10">
            <select asp-for="DepartmentId" class="custom-select"
asp-items="@ (ViewBag.Departments)">
                <option value="" disabled
selected>Department</option>
            </select>
            <span asp-validation-for="DepartmentId" class="text-
danger"></span>
        </div>
    </div>
    <div class="form-group row">
        <label asp-for="Address" class="col-form-label col-md-
2"></label>
        <div class="col-md-10">
            <input asp-for="Address" class="form-control" />
            <span asp-validation-for="Address" class="text-
danger"></span>
        </div>
    </div>
</div>
<div class="card-footer">
    <button type="submit" class="btn btn-primary">Create</button>
    <a asp-action="Index" asp-controller="employee">
        <i class="fas fa-arrow-circle-left ml-2 mr-1"></i>Back To
List
    </a>
</div>
</div>
</form>
</div>
Index.cshtml
```

```
@model IEnumerable<EmployeeViewModel>
```

```
@{
    ViewData["Title"] = "Index";
}
<div class="card">
    <div class="card-header">
        <strong>Employee Information</strong>
        <a asp-action="Create" class="float-right">Create</a>
    </div>
    <div class="card-body">
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>@Html.DisplayNameFor(emp => emp.Name)</th>
                    <th>@Html.DisplayNameFor(emp => emp.Gender)</th>
                </tr>
            </thead>
            <tbody>
                @foreach (var emp in Model)
                {
                    <tr>
                        <td>@emp.Name</td>
                        <td>@emp.Gender</td>
                    </tr>
                }
            </tbody>
        </table>
    </div>
</div>
```


សកលវិទ្យាល័យបៀលប្រាស

```
<th>@Html.DisplayNameFor(emp => emp.Department)</th>
<th>@Html.DisplayNameFor(emp => emp.Address)</th>
<th>@Html.DisplayName("Actions")</th>
</tr>
</thead>
<tbody>
    @if (Model.Count() > 0 || Model != null)
    {
        @foreach (var emp in Model)
        {
            <tr>
                <td>@Html.DisplayFor(em => emp.Name)</td>
                <td>@Html.DisplayFor(em => emp.Gender)</td>
                <td>@Html.DisplayFor(em => emp.Department)</td>
                <td>@Html.DisplayFor(em => emp.Address)</td>
                <td>
                    <a asp-action="Detailed" asp-route-id="@emp.Id">
                        <i class="far fa-eye mr-1"></i>
                        Detail
                    </a> |
                    <a asp-action="Edit" asp-route-id="@emp.Id">
                        <i class="far fa-edit mr-1"></i>Edit
                    </a> |
                    <a asp-action="Delete" asp-route-id="@emp.Id">
                        <i class="far fa-trash-alt mr-1"></i>
                        Delete
                    </a>
                </td>
            </tr>
        }
    }
    else
    {
        <tr>
            <td colspan="5">
                <h4>there is no data</h4>
            </td>
        </tr>
    }
</tbody>
</table>
</div>
</div>
```

- Share Directory

_Layout.cshtml

```
<!DOCTYPE html>

<html>
<head>
```

```
<meta name="viewport" content="width=device-width" />
<title>@ViewBag.Title</title>
<link href="~/bootstrap/css/bootstrap.css" rel="stylesheet" />
<link href="~/font-awesome/css/all.css" rel="stylesheet" />
</head>
<body>
  <partial name="_Navbar" />
  <div class="container-fluid">
    @RenderBody()
  </div>
  <script src="~/jquery/jquery.js"></script>
  <script src="~/bootstrap/js/bootstrap.js"></script>
  <script src="~/font-awesome/js/all.js"></script>
</body>
</html>
```

_Navbar.cshtml

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-
only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link"
          asp-action="Index" asp-controller="employee">Employee</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" asp-action="list" asp-
controller="department">Department</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search"
placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success my-2 my-sm-0"
type="submit">Search</button>
    </form>
  </div>
</nav>
```

_ViewImports.cshtml

សកលវិទ្យាល័យបៀលប្រាស

```
@using EmployeeSQL.Models;  
@using EmployeeSQL.ViewModels;  
@addTagHelper *,Microsoft.AspNetCore.Mvc.TagHelpers
```

_ViewStart.cshtml

```
@{  
    Layout = "_Layout";  
}
```

លទ្ធផល

Create

Name

Gender ☐ Female ☐ Male

Department

Address

[Create](#) [Back To List](#)

Index

Employee Information [Create](#)

Name	Gender	Department	Address	Actions
ly ty	male	staff	siem reap	Detail Edit Delete
rachana	femlae	HR	siem reap	Detail Edit Delete
thearak	femlae	Admin	kandal	Detail Edit Delete

សកលវិទ្យាល័យបៀលប្រាស

Detailed

https://localhost:44351/Employee/Detailed/1

Navbar Home Employee Department

Search Search

Employee information

Id	1
Name	ly ty
Gender	male
Department	
Address	siem reap

[Create](#) | [Back To List](#)

Create

https://localhost:44351/Employee/Edit/1

Navbar Home Employee Department

Search Search

Create

Name	<input type="text" value="ly ty"/>
Gender	<input type="radio"/> Female <input checked="" type="radio"/> Male
Department	<input type="text" value="staff"/>
Address	<input type="text" value="siem reap"/>

[Create](#) [Back To List](#)

Delete

https://localhost:44351/Employee/Delete/1003

Navbar Home Employee Department

Search Search

Confirm?

Do you want to delete **thearak**?

[Delete](#) [Back](#)