**EmployeeMVC:**

Startup.cs:

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.EntityFrameworkCore;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using SecondNVCApplication.Models;

using SecondNVCApplication.Services;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace SecondNVCApplication

{

public class Startup

{

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

services.AddControllersWithViews();

services.AddDbContext<CompanyContext>(options =>

{

options.UseSqlServer(Configuration["ConnectionStrings:conCompany"]);

}

);

services.AddScoped<IRepo<Employee>, EmployeeRepo>();

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

}

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

});

}

}

}

EmployeeController.cs:

using Microsoft.AspNetCore.Mvc;

using SecondNVCApplication.Models;

using SecondNVCApplication.Services;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace SecondNVCApplication.Controllers

{

public class EmployeeController : Controller

{

private readonly IRepo<Employee> \_repo;

public EmployeeController(IRepo<Employee> repo)

{

\_repo = repo;

}

public IActionResult Index()

{

return View(\_repo.GetAll());

}

public IActionResult Details(int id)

{

return View(\_repo.Get(id));

}

public IActionResult Create()

{

return View();

}

[HttpPost]

public IActionResult Create(Employee employee)

{

\_repo.Add(employee);

return RedirectToAction("Index");

}

public IActionResult Edit(int id)

{

return View(\_repo.Get(id));

}

[HttpPost]

public IActionResult Edit(int id,Employee employee)

{

\_repo.Update(employee);

return RedirectToAction("Index");

}

public IActionResult Delete(int id)

{

return View(\_repo.Get(id));

}

[HttpPost]

public IActionResult Delete(int id, Employee employee)

{

\_repo.Delete(id);

return RedirectToAction("Index");

}

}

}

VIEW:

Create.CsHtml:

@model SecondNVCApplication.Models.Employee

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Create</title>

</head>

<body>

<h4>Employee</h4>

<hr />

<div class="row">

<div class="col-md-4">

@using (Html.BeginForm("Create", "Employee", FormMethod.Post))

{

<div **asp-validation-summary**="ModelOnly" class="text-danger"></div>

<div class="form-group">

@Html.LabelFor(m => m.Name)

@Html.EditorFor(m => m.Name)

</div>

<div class="form-group">

@Html.LabelFor(m => m.Age)

@Html.EditorFor(m => m.Age)

</div>

<div class="form-group">

@Html.LabelFor(m => m.DepartmentId)

@Html.EditorFor(m => m.DepartmentId)

</div>

<div class="form-group">

<input type="submit" value="Create" class="btn btn-primary" />

</div>}

</div>

</div>

<div>

<a **asp-action**="Index">Back to List</a>

</div>

</body>

</html>

Delete.CsHtml:

@model SecondNVCApplication.Models.Employee

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Delete</title>

</head>

<body>

<h3>Are you sure you want to delete this?</h3>

<div>

<h4>Employee</h4>

<hr />

<dl class="row">

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Name)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Name)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Age)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Age)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.DepartmentId)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.DepartmentId)

</dd class>

</dl>

<form **asp-action**="Delete">

<input **type**="hidden" **asp-for**="Id" />

<input type="submit" value="Delete" class="btn btn-danger" /> |

<a **asp-action**="Index">Back to List</a>

</form>

</div>

</body>

</html>

Details.CsHtml:

@model SecondNVCApplication.Models.Employee

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Details</title>

</head>

<body>

<div>

<h4>Employee</h4>

<hr />

<dl class="row">

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Name)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Name)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Age)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Age)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.DepartmentId)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.DepartmentId)

</dd>

</dl>

</div>

<div>

@Html.ActionLink("Edit", "Edit", new {id=Model.Id})|

<a **asp-action**="Index">Back to List</a>

</div>

</body>

</html>

Edit.CsHtml:

@model SecondNVCApplication.Models.Employee

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Edit</title>

</head>

<body>

<h4>Employee</h4>

<hr />

<div class="row">

<div class="col-md-4">

@using (Html.BeginForm("Edit", "Employee", FormMethod.Post))

{

<div **asp-validation-summary**="ModelOnly" class="text-danger"></div>

<div class="form-group">

@Html.LabelFor(m => m.Name)

@Html.EditorFor(m => m.Name)

</div>

<div class="form-group">

@Html.LabelFor(m => m.Age)

@Html.EditorFor(m => m.Age)

</div>

<div class="form-group">

@Html.LabelFor(m => m.DepartmentId)

@Html.EditorFor(m => m.DepartmentId)

</div>

<div class="form-group">

<input type="submit" value="Edit" class="btn btn-primary" />

</div>

}

</div>

</div>

<div>

<a **asp-action**="Index">Back to List</a>

</div>

</body>

</html>

Index.CsHtml:

@model IEnumerable<SecondNVCApplication.Models.Employee>

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Index</title>

</head>

<body>

<p>

<a **asp-action**="Create">Create New</a>

</p>

<table class="table">

<thead>

<tr>

<th>

@Html.DisplayNameFor(model => model.Name)

</th>

<th>

@Html.DisplayNameFor(model => model.Age)

</th>

<th>

@Html.DisplayNameFor(model => model.DepartmentId)

</th>

<th></th>

</tr>

</thead>

<tbody>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Name)

</td>

<td>

@Html.DisplayFor(modelItem => item.Age)

</td>

<td>

@Html.DisplayFor(modelItem => item.DepartmentId)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id = item.Id }) |

@Html.ActionLink("Details", "Details", new { id = item.Id }) |

@Html.ActionLink("Delete", "Delete", new { id = item.Id })

</td>

</tr>

}

</tbody>

</table>

</body>

</html>

Models:

Employee.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations.Schema;

using System.Linq;

using System.Threading.Tasks;

namespace SecondNVCApplication.Models

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Age { get; set; }

public int DepartmentId { get; set; }

[ForeignKey("DepartmentId")]

public Department department { get; set; }

}

}

Department.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace SecondNVCApplication.Models

{

public class Department

{

public int Id { get; set; }

public string Name { get; set; }

public ICollection<Employee> employees { get; set; }

}

}

OUTPUT:







