**Sompalli Rajesh**

**Step by step Crud Operations Project through the Code First Approach in ASP.NET MVC CORE with SQL Server:**

**Example Project:**

1. **Tools: NuGet Packge Manager: Manage NuGet Packages for Solution:**

**Browse:**

Microsoft.EntityFrameworkCore

Microsoft.EntityFrameworkCore.SqlServer

Microsoft.EntityFrameworkCore.Design

Microsoft.EntityFrameworkCore.Tools

1. **DB Connection:**

**appsettings.json:**

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*",

"ConnectionStrings": {

"DefaultConnection": "Server=DESKTOP-54HLCQ0\\SQLEXPRESS;Database=EMP;Trusted\_Connection=True;TrustServerCertificate=True;"

}

}

**Models:**

**ApplictionContext.cs:**

public class ApplicationContext : DbContext

{

public ApplicationContext(DbContextOptions<ApplicationContext> options) : base(options)

{

}

**Program.cs:**

builder.Services.AddDbContext<ApplicationContext>(options => options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

**Tools:NuGet Packge Manager: Package Manager Console:**

Update-Database

1. **Models class for Properties:**

namespace EmployeeDataManagement.Models

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public string Job { get; set; }

public int? Salary { get; set; }

}

}

1. **DbSet:**

**Models:**

**ApplicationContext.cs:**

using Microsoft.EntityFrameworkCore;

namespace EmployeeDataManagement.Models

{

public class ApplicationContext:DbContext

{

public ApplicationContext(DbContextOptions<ApplicationContext> options) : base(options)

{

}

**public DbSet<Employee> Employees { get; set;}**

}

}

Note : Add-Migration and Update-Database

1. **Controller:**

using EmployeeDataManagement.Models;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

namespace EmployeeDataManagement.Controllers

{

public class EmployeeController : Controller

{

private readonly ApplicationContext \_applicationContext;

public EmployeeController(ApplicationContext applicationContext)

{

this.\_applicationContext = applicationContext;

}

public IActionResult Index()

{

var employees = \_applicationContext.Employees.ToList();

return View(employees);

}

[HttpGet]

public IActionResult Create()

{

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult Create(Employee employee)

{

//if (ModelState.IsValid)

{

\_applicationContext.Employees.Add(employee);

\_applicationContext.SaveChanges();

return RedirectToAction("Index");

}

return View();

}

// GET: Employee/Edit/5

public ActionResult Edit(int id)

{

var employee = \_applicationContext.Employees.Find(id);

return View(employee);

}

// POST: Employee/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(Employee employee)

{

//if (ModelState.IsValid)

{

\_applicationContext.Entry(employee).State = EntityState.Modified;

\_applicationContext.SaveChanges();

return RedirectToAction("Index");

}

return View(employee);

}

// GET: Employee/Delete/5

public ActionResult Delete(int id)

{

var employee = \_applicationContext.Employees.Find(id);

return View(employee);

}

// POST: Employee/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

var employee = \_applicationContext.Employees.Find(id);

\_applicationContext.Employees.Remove(employee);

\_applicationContext.SaveChanges();

return RedirectToAction("Index");

}

}

}

**Note: Create “Razor View Empty” to Index Action method:**

**Html:**

@model List<Employee>

<!DOCTYPE html>

<html>

<head>

<title>Employee List</title>

</head>

<body>

<h2>Employee List</h2>

<table class="table table-bordered table-striped">

<tr>

<th>Id</th>

<th>Name</th>

<th>Job</th>

<th>Salary</th>

</tr>

@foreach (var employee in Model)

{

<tr>

<td>@employee.Id</td>

<td>@employee.Name</td>

<td>@employee.Job</td>

<td>@employee.Salary</td>

<td>

<a href="@Url.Action("Edit", "Employee", new { id = employee.Id })" class="btn btn-primary">Edit</a>

<a href="@Url.Action("Delete", "Employee", new { id = employee.Id })" class="btn btn-danger">Delete</a>

</td>

</tr>

}

</table>

</body>

</html>

**Note: Create “Razor Views” to Create, Edit and Delete action methods.**

**Note: Create Labels in Layouts to Index and Create action methods:**

<li class="nav-item">

<**a** class="nav-link text-primary" **asp-area**="" **asp-controller**="Employee" **asp-action**="Index">EmployeeBioData</**a**>

</li>

<li class="nav-item">

<**a** class="nav-link text-primary" **asp-area**="" **asp-controller**="Employee" **asp-action**="Create">EmployeeDetails</**a**>

</li>

Note: Compile and Run Project.