**Handson**

**GetemployeeList**

@\*

For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

\*@

@{

}

@model List<WebApplication20.Models.Employee>

@{

ViewData["Title"] = "GetEmployeeList";

<div class="row">

<span class="col-md-4">Name</span>

<span class="col-md-4">Salary</span>

<span class="col-md-4">IsPermanent</span>

</div>

}

@foreach (var employee in Model)

{

<div class="row">

@if (employee.IsPermanent.Equals(true))

{

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

<b>@employee.Name</b>

</span>

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

<b>@employee.Salary</b>

</span>

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

<b>@employee.IsPermanent</b>

</span>

}

else

{

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

@employee.Name

</span>

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

@employee.Salary

</span>

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

@employee.IsPermanent

</span>

}

</div>

}

**Home Controller**

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

using WebApplication20.Models;

namespace WebApplication20.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult Index()

{

return View();

}

public IActionResult Privacy()

{

return View();

}

public IActionResult GetEmployeeList()

{

List<Employee> employeeList = new List<Employee>();

employeeList.Add(new Employee()

{

Id = 1,

Name = "John",

Salary = 10000,

IsPermanent = true

});

employeeList.Add(new Employee()

{

Id = 2,

Name = "Smith",

Salary = 5000,

IsPermanent = false

});

employeeList.Add(new Employee()

{

Id = 3,

Name = "Mark",

Salary = 5000,

IsPermanent = false

});

employeeList.Add(new Employee()

{

Id = 4,

Name = "Mary",

Salary = 5000,

IsPermanent = false

});

return View(employeeList);

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

}

}

Output

