**ASP.NET Core Day 63 Handson**

**Handson 1:**

Create a Razor view to list a few Organizations at Hyderabad and highlight the one that has recently build its largest offshore office at India.

Description: Create a new Action method ‘ITOrganizations’ in the Home controller. Add a view to it. Use Razor syntax to create list of data type string. Add few organizations at Hyderabad,

· Microsoft www.microsoft.com

· CTS www.cognizant.com

· Google

· Amazon

· TCS

To the list. Use foreach loop to loop thru the list and use DIV and a span inside it to list the organizations. Amazon is the IT organization that has built its largest office at Hyderabad in 2019.

**Program:**

**Homecontroller.cs**

using mcv.Models;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

namespace mcv.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 ITOrganizations()

{

List<string> organizationList = new List<string>

{

"Microsoft www.microsoft.com","CTS www.cognizant.com", "Google", "Amazon","TCS"

};

return View(organizationList);

}

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

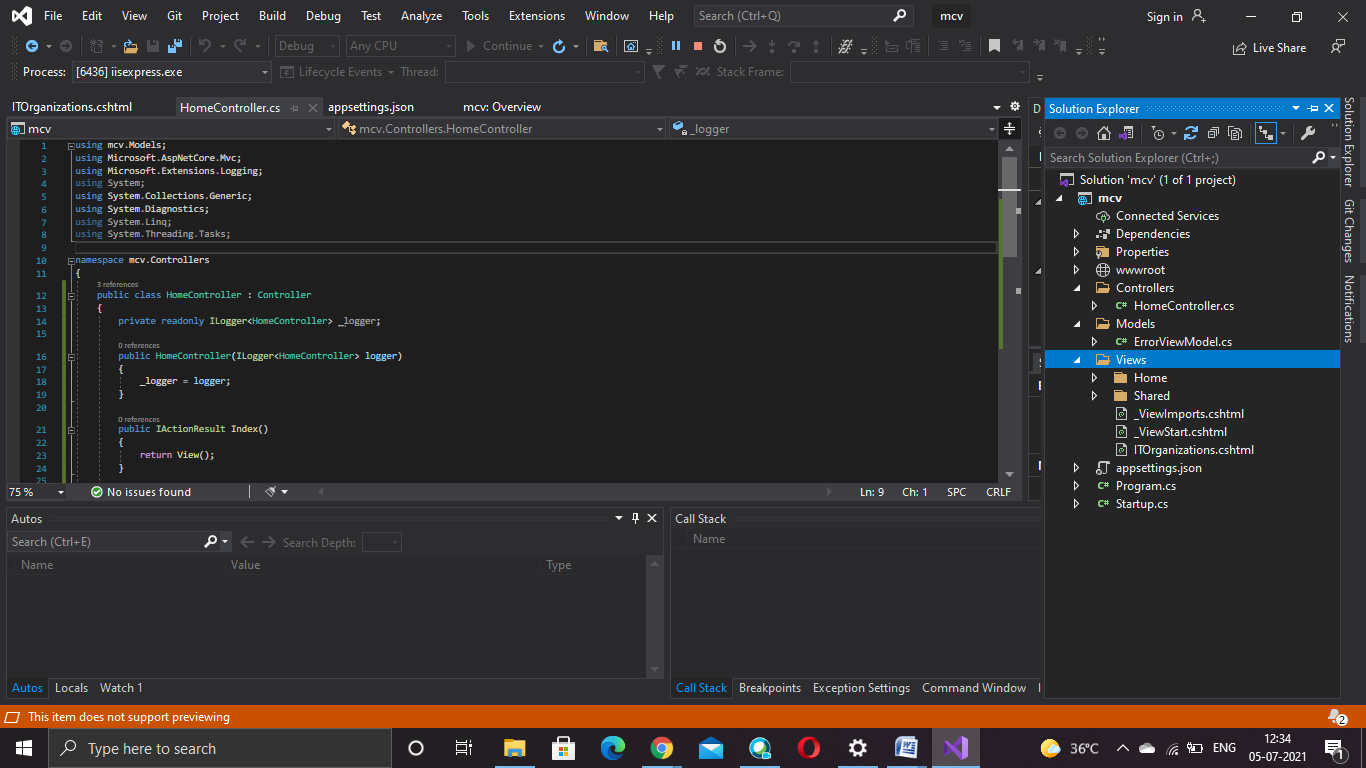
public IActionResult Error()

{

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

}

}

****

**ITOrganizations.cs**

@{

ViewData["Title"] = "ITOrganizations";

}

<h1>ITOrganizations</h1>

@model List<string>

@foreach (var x in Model)

{

<div>

@if (x.Equals("Amazon"))

{

<span><b>@Html.DisplayFor(o => x)</b></span>

}

else

{

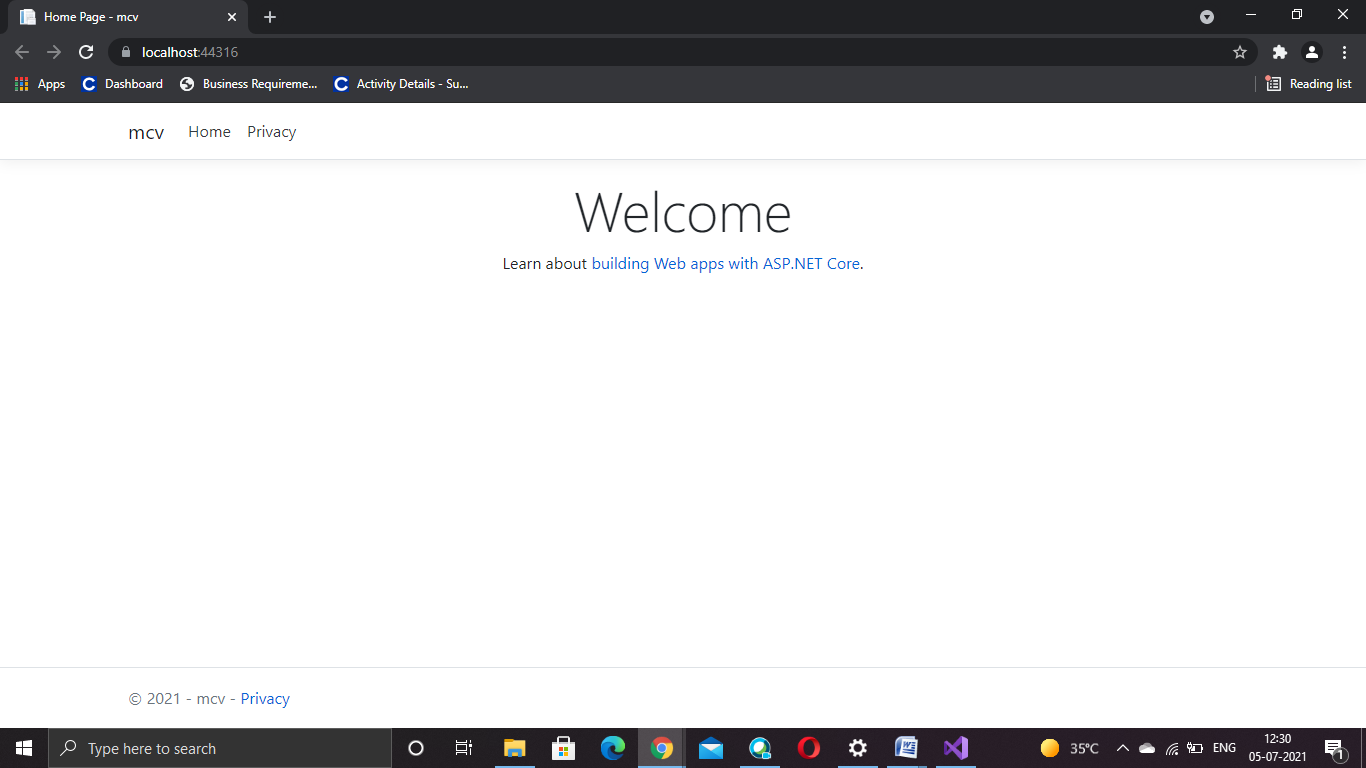
<span>@Html.DisplayFor(o => x)</span>

}

</div>

}

**Output:**

****

**Handson 2:**

Create a Razor view to list few employee details using Model data binding

Description: Create a model class ‘Employee’ to be used for Employee details. Create properties

· Id – datatype Int

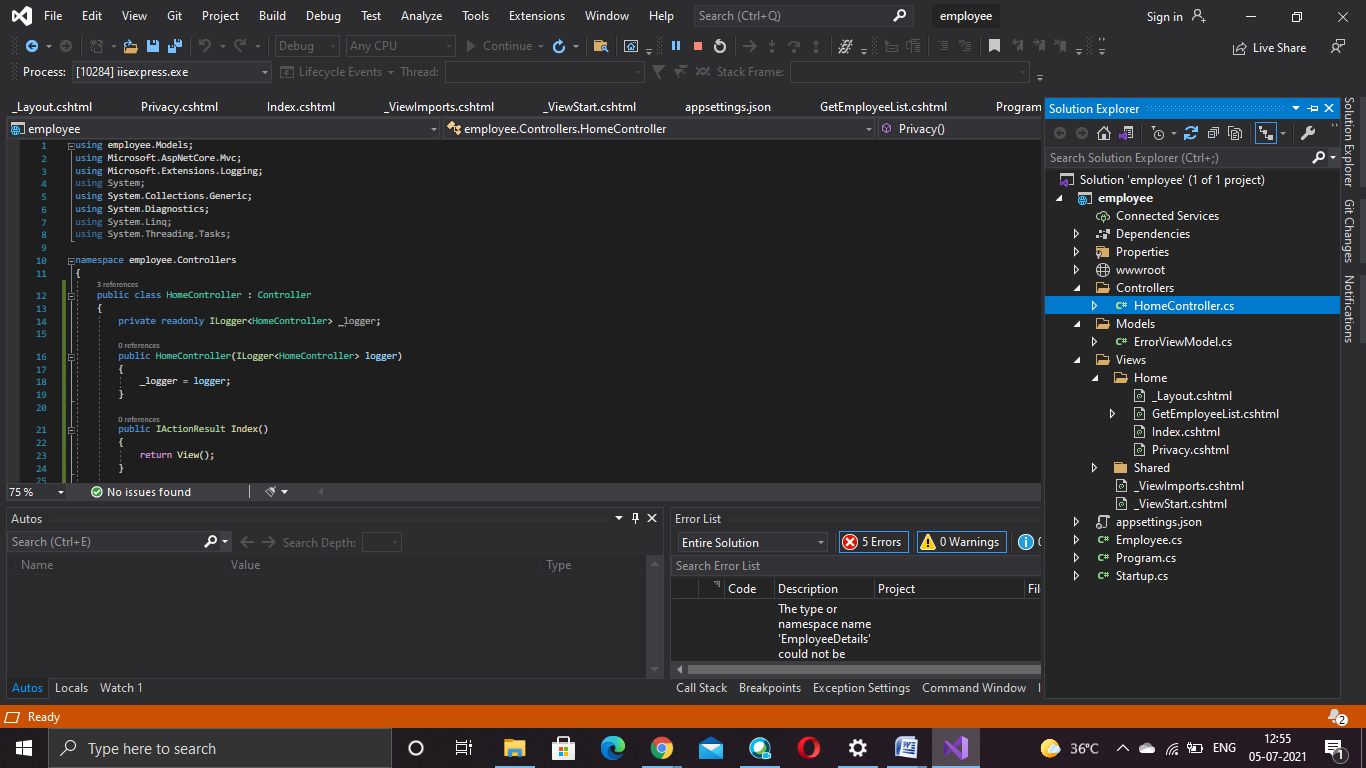
· Name – datatype string

· Salary – datatype decimal

· IsPermanent – datatype bool

Create a new Action method ‘GetEmployeeList’ in the Home controller and add a new View. Use the data provided below to feed the data to a list of Employee model class

**Program:**

****

**HomeController.cs**

using employee.Models;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

namespace employee.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 });

}

}

}

**EmployeeList.cshtml**

@{

ViewData["Title"] = "GetEmployeeList";

}

<h1>GetEmployeeList</h1>

@model List<EmployeeDetails.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>

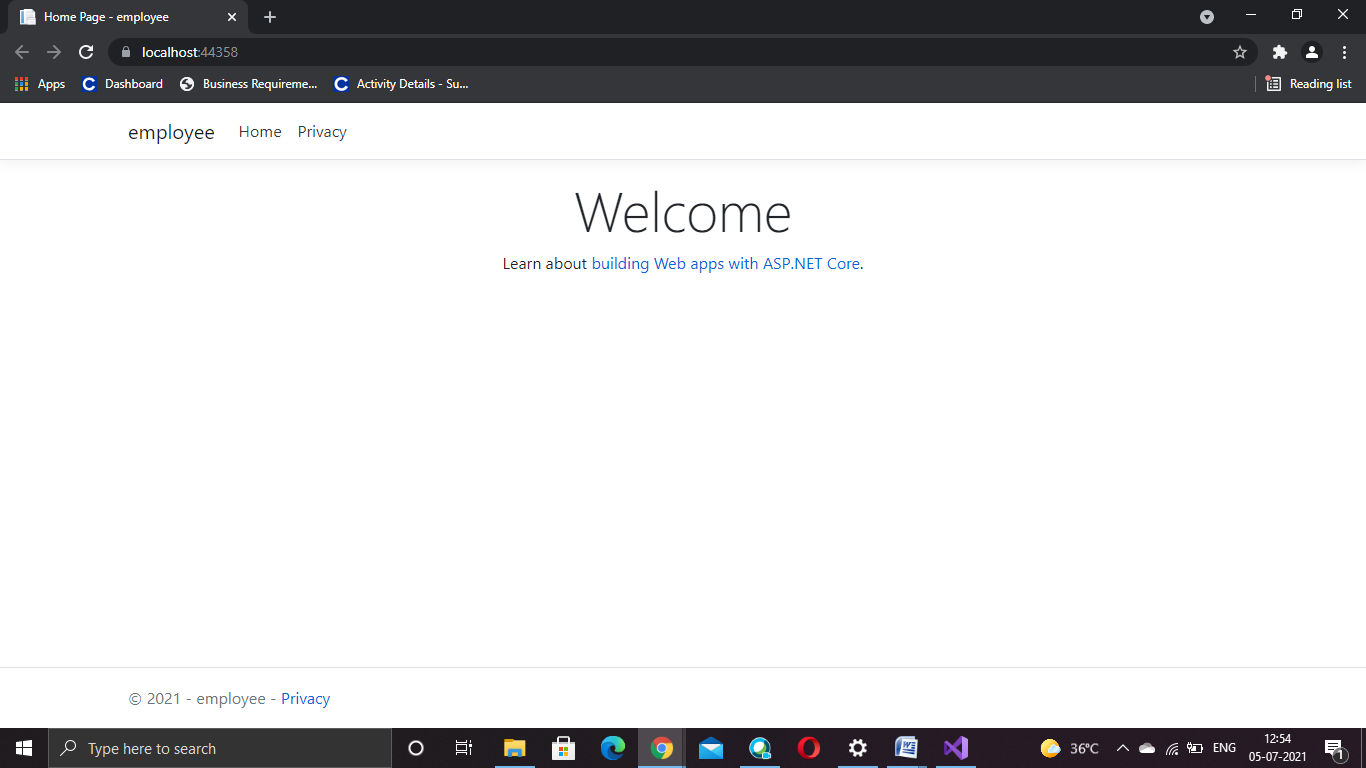
<span class="col-md-4">@employee.IsPermanent</span>

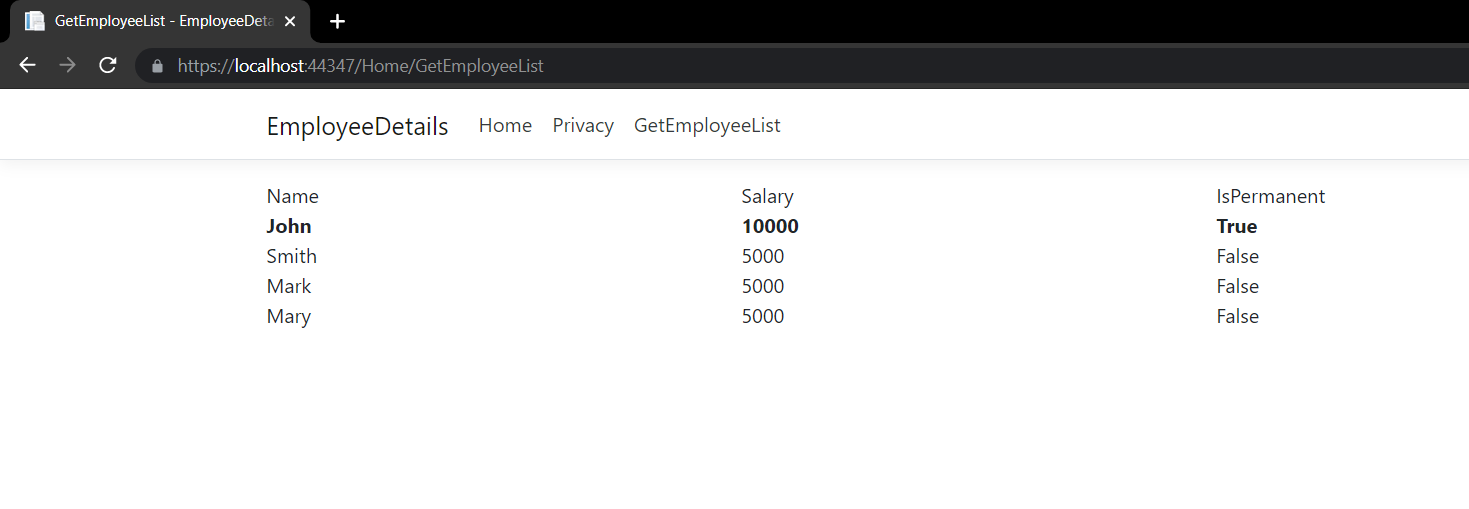
}

</div>

}

**Output:**

****

****