TASK - 11

'using' keyword:

Importing Namespaces:

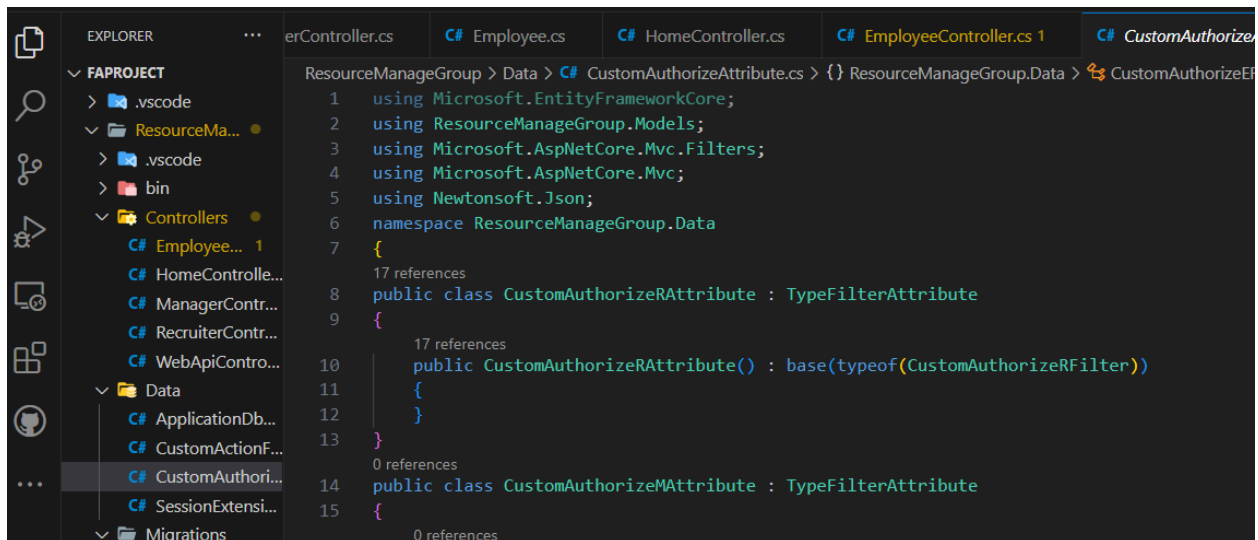   The using keyword is commonly used at the top of a file to import namespaces.

Alias Declaration:

    The using keyword can also be used to create an alias for a namespace or a type.

Resource Management:

    The using keyword is used for managing resources,

EXPLORER   rController.cs   C# Employee.cs   C# HomeController.cs   C# EmployeeController.cs 1 ✕   C# CustomAuthorizeAttribute.cs

FAPROJECT

ResourceManageGroup › Controllers › EmployeeController.cs › …

```csharp
1    using System.Diagnostics;
2    using Microsoft.AspNetCore.Mvc;
3    using ResourceManageGroup.Models;
4    using ResourceManageGroup.Data;
5    using Microsoft.EntityFrameworkCore;
6    using System.Data;
7    using System.Net;
8    using System.Net.Mail;
9    using System.ComponentModel.DataAnnotations;
10   namespace ResourceManageGroup.Controllers;
     0 references
11   public class EmployeeController : Controller
12   {
         27 references
         private readonly ApplicationDbContext _dbContext;
         0 references
14       public EmployeeController(ApplicationDbContext dbContext,IWebHostEnvironment hostEnvironment)
15       {
16           _dbContext = dbContext;
```

EXPLORER   rController.cs   C# Employee.cs   C# HomeController.cs   C# EmployeeController.cs 1 ✕   C# CustomAuthorizeAttribute.cs

FAPROJECT

ResourceManageGroup › Controllers › EmployeeController.cs › {} ResourceManageGroup.Controllers › EmployeeController › Edit

```csharp
294      {
295          employees.EmployeeName = employee.EmployeeName;
296          employees.EmployeeNumber = employee.EmployeeNumber;
297          employees.EmployeeEmail = employee.EmployeeEmail;
298
299          if (image1 != null && image1.Length > 0)
300          {
301              using (var memoryStream = new MemoryStream())
302              {
303                  await image1.CopyToAsync(memoryStream);
304                  employees.EmployeeImage = memoryStream.ToArray();
305              }
306          }
307          await _dbContext.SaveChangesAsync();
308          var url = Url.Action("EmployeeViewE", "Employee", new { id = employee.EmployeeId });
309          if (url != null)
310          {
311              return Redirect(url);
312          }
313      }
314  }
315  else
316  {
317      ModelState.AddModelError(string.Empty, "Employee not found");
```