

TASK-6

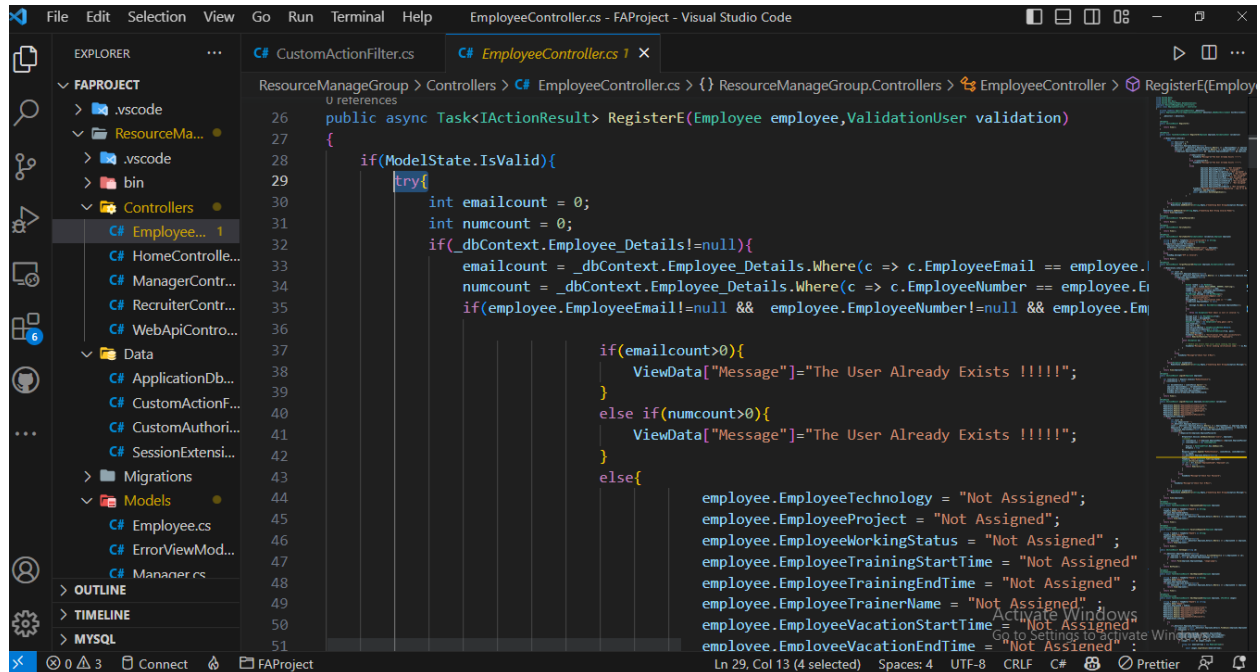
System Exception:

```
SystemException > C# Program.cs > Program > Main()
7      try
8      {
9          int[] numbers = { 1, 2, 3 };
10         int index = 4;
11
12         int result = numbers[index]; // Throws IndexOutOfRangeException
13         Console.WriteLine("Result: " + result);
14     }
15     catch (IndexOutOfRangeException exception)
16     {
17         Console.WriteLine("Error: Index is out of range.");
18         // Handles the IndexOutOfRangeException
19     }
20     catch (DivideByZeroException exception)
21     {
22         Console.WriteLine("Error: Cannot divide by zero.");
23         // Handles the DivideByZeroException
24     }
25     catch (InvalidOperationException exception)
26     {
27         Console.WriteLine("Error: Invalid operation.");
28         // Handles the InvalidOperationException
29     }
30     catch (Exception exception)
31     {
32         Console.WriteLine("An error occurred: " + exception.Message);
```

Error: Index is out of range.

```
F:\Aspire Training\Tasks\25-May\SystemException>dotnet run
F:\Aspire Training\Tasks\25-May\SystemException\Program.cs(15,41): warning CS0168: The variable 'exception' is declared but never used [F:\Aspire Training\Tasks\25-May\SystemException\SystemException.csproj]
F:\Aspire Training\Tasks\25-May\SystemException\Program.cs(20,38): warning CS0168: The variable 'exception' is declared but never used [F:\Aspire Training\Tasks\25-May\SystemException\SystemException.csproj]
F:\Aspire Training\Tasks\25-May\SystemException\Program.cs(25,42): warning CS0168: The variable 'exception' is declared but never used [F:\Aspire Training\Tasks\25-May\SystemException\SystemException.csproj]
Error: Index is out of range.
```

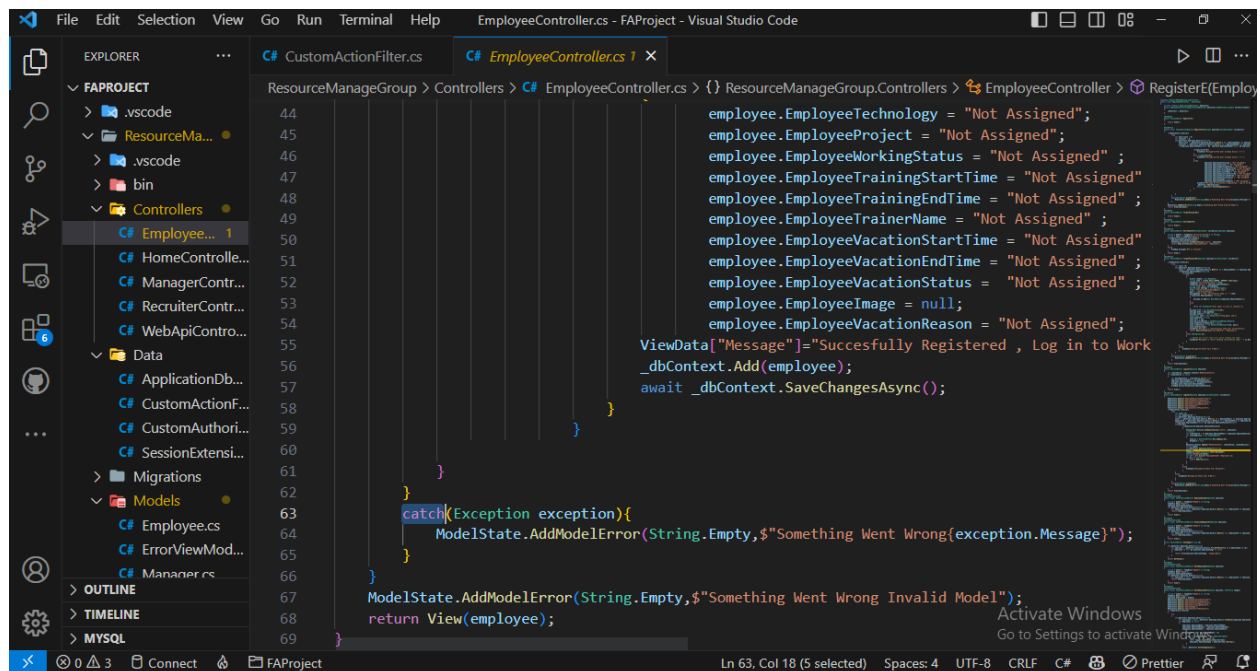
Application Exception:



```

26 public async Task<IActionResult> RegisterE(Employee employee, ValidationUser validation)
27 {
28     if (ModelState.IsValid)
29     {
30         try
31         {
32             int emailcount = 0;
33             int numcount = 0;
34             if (_dbContext.Employee_Details != null)
35             {
36                 emailcount = _dbContext.Employee_Details.Where(c => c.EmployeeEmail == employee.EmployeeEmail).Count();
37                 numcount = _dbContext.Employee_Details.Where(c => c.EmployeeNumber == employee.EmployeeNumber).Count();
38                 if (emailcount > 0)
39                 {
40                     ViewData["Message"] = "The User Already Exists !!!!!";
41                 }
42                 else if (numcount > 0)
43                 {
44                     ViewData["Message"] = "The User Already Exists !!!!!";
45                 }
46                 else
47                 {
48                     employee.EmployeeTechnology = "Not Assigned";
49                     employee.EmployeeProject = "Not Assigned";
50                     employee.EmployeeWorkingStatus = "Not Assigned";
51                     employee.EmployeeTrainingStartTime = "Not Assigned";
52                     employee.EmployeeTrainingEndTime = "Not Assigned";
53                     employee.EmployeeTrainerName = "Not Assigned";
54                     employee.EmployeeVacationStartTime = "Not Assigned";
55                     employee.EmployeeVacationEndTime = "Not Assigned";
56                     employee.EmployeeVacationStatus = "Not Assigned";
57                     employee.EmployeeImage = null;
58                     employee.EmployeeVacationReason = "Not Assigned";
59                     ViewData["Message"] = "Successfully Registered , Log in to Work";
60                     _dbContext.Add(employee);
61                     await _dbContext.SaveChangesAsync();
62                 }
63             }
64         }
65         catch (Exception exception)
66         {
67             ModelState.AddModelError(String.Empty, $"Something Went Wrong {exception.Message}");
68         }
69         ModelState.AddModelError(String.Empty, $"Something Went Wrong Invalid Model");
70         return View(employee);
71     }
72 }

```



```

44     employee.EmployeeTechnology = "Not Assigned";
45     employee.EmployeeProject = "Not Assigned";
46     employee.EmployeeWorkingStatus = "Not Assigned";
47     employee.EmployeeTrainingStartTime = "Not Assigned";
48     employee.EmployeeTrainingEndTime = "Not Assigned";
49     employee.EmployeeTrainerName = "Not Assigned";
50     employee.EmployeeVacationStartTime = "Not Assigned";
51     employee.EmployeeVacationEndTime = "Not Assigned";
52     employee.EmployeeVacationStatus = "Not Assigned";
53     employee.EmployeeImage = null;
54     employee.EmployeeVacationReason = "Not Assigned";
55     ViewData["Message"] = "Successfully Registered , Log in to Work";
56     _dbContext.Add(employee);
57     await _dbContext.SaveChangesAsync();
58 }
59 }
60 }
61 }
62 }
63 catch (Exception exception)
64 {
65     ModelState.AddModelError(String.Empty, $"Something Went Wrong {exception.Message}");
66 }
67 ModelState.AddModelError(String.Empty, $"Something Went Wrong Invalid Model");
68 return View(employee);
69 }

```