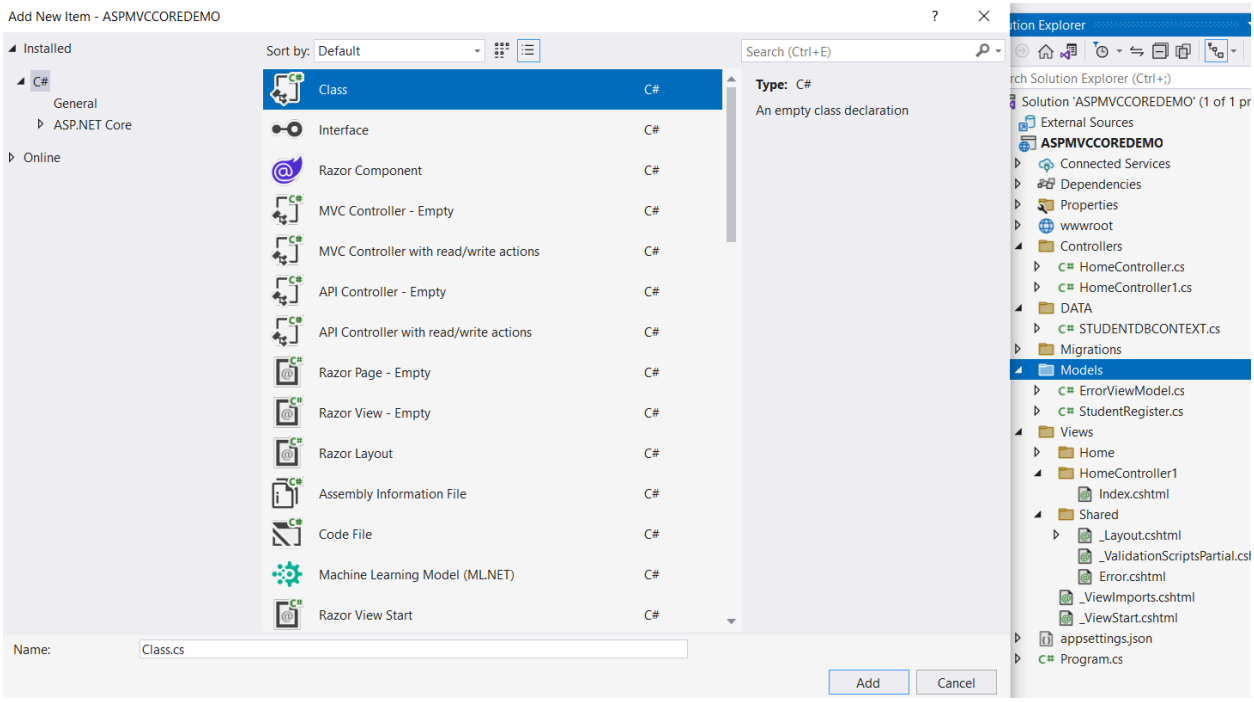
ASP.NET CORE

STEP:1

CREATE A MODEL CLASS

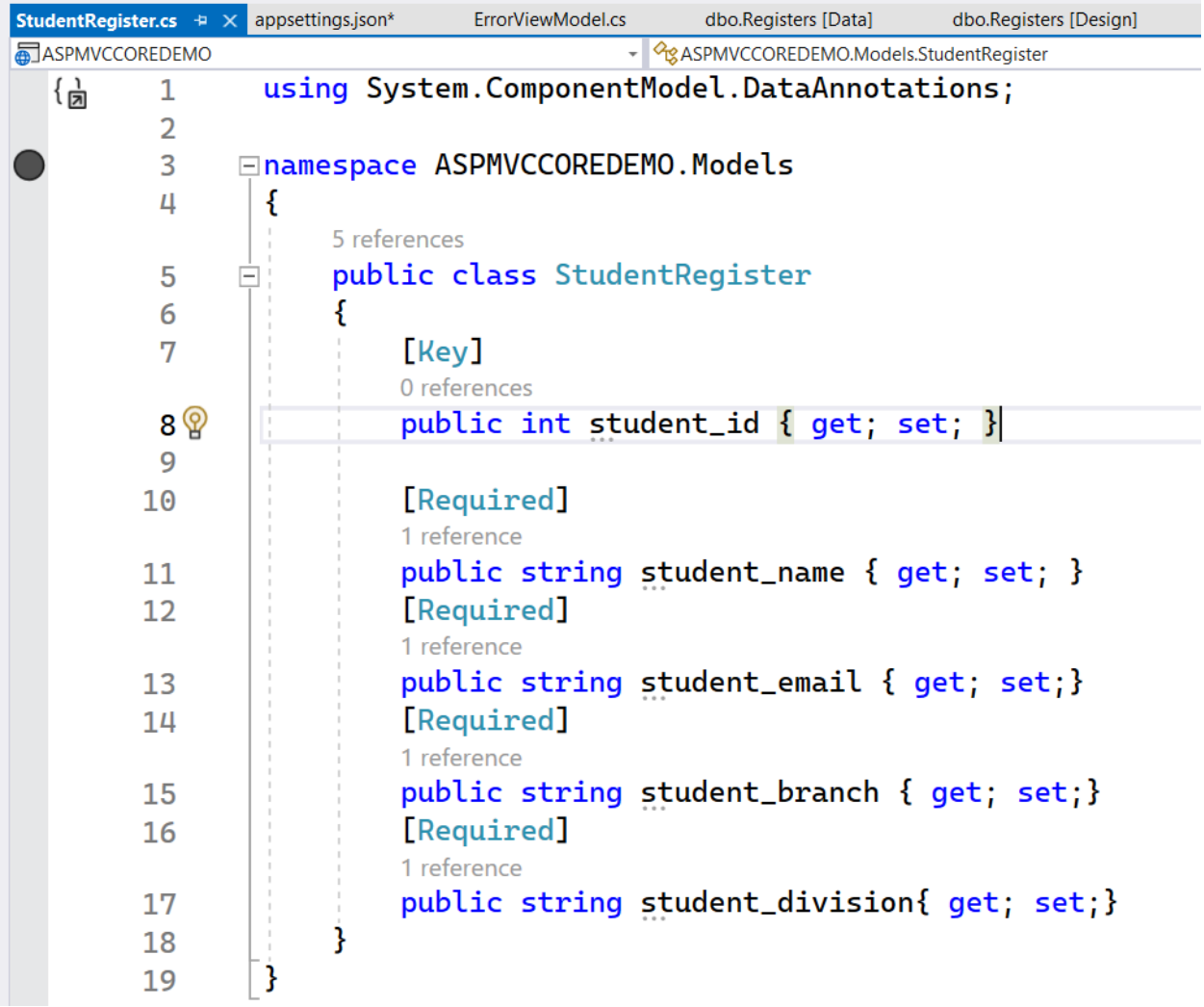


Step:2

In model class you must add properties/attributes in table whatever you needed.

In class write **prop** and do double tab

It will create auto generated code also you can give client side validation also.



Step:3 After creating class, you must make connection string in appsettings .json file.

```
"ConnectionStrings": {
  "con": "Server=(localdb)\\MSSQLLocalDB;Database=Student Management System;Trusted_connection=Tr
}
```

Appsettings.json code

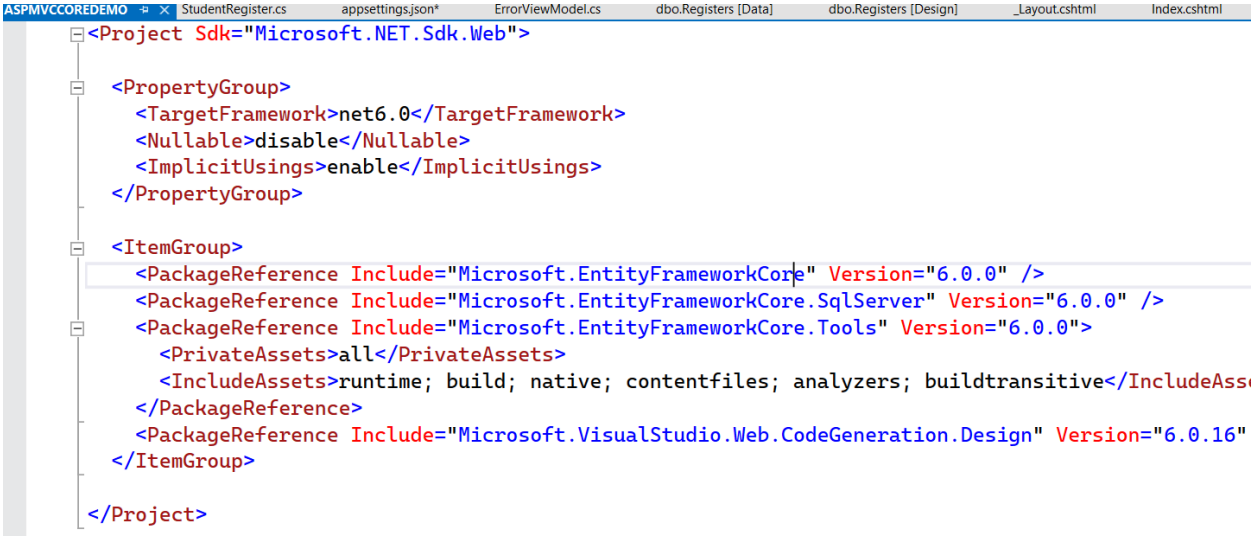
```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
```

```
    "con": "Server=(localdb)\\MSSQLLocalDB;Database=Student Management
System;Trusted_connection=True;MultipleActiveResultSets=True;"
  }
}
```

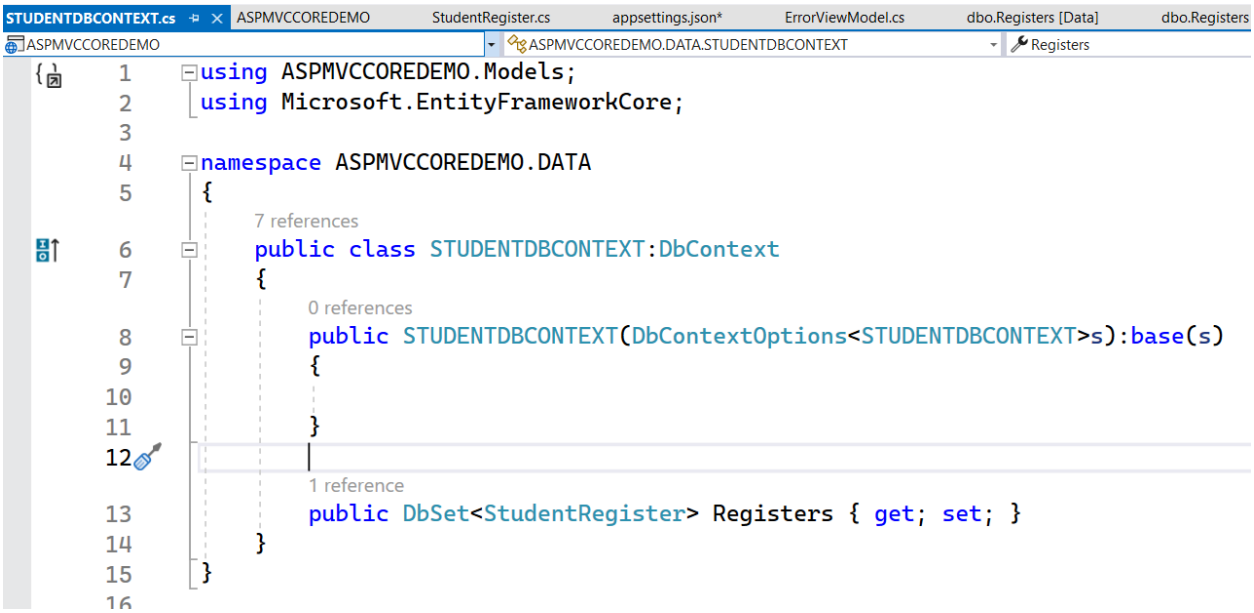
Step:4 After creating connection string you have to make one data folder for DBCONTEXT file

In solution explorer right click add new folder add class file.

Now before writing code for class file, you have to install Entity framework and sql server package of your appropriate version of your .net framework version.



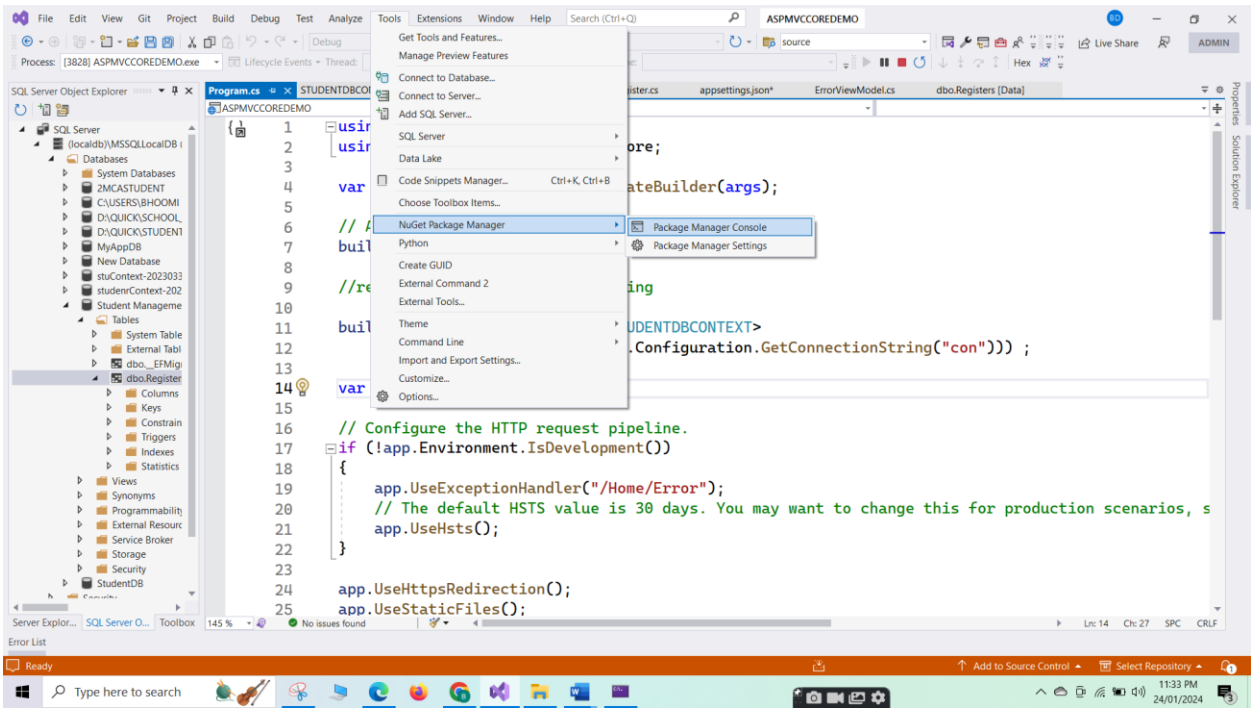
Now comes to your data folder for dbcontext file and add dbcontext class



Step:5 now you must register your connection string in program.cs file

```
using ASPMVCCOREDEMO.DATA;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();

//registering the connection string

builder.Services.AddDbContext<STUDENTDBCONTEXT>
    ( s =>s.UseSqlServer(builder.Configuration.GetConnectionString("con"))) ;

var app = builder.Build();
```

Step 6: build your project, if it's successfully then open your package manager console from tools



In console write two commands,

1)add-migration "file name"

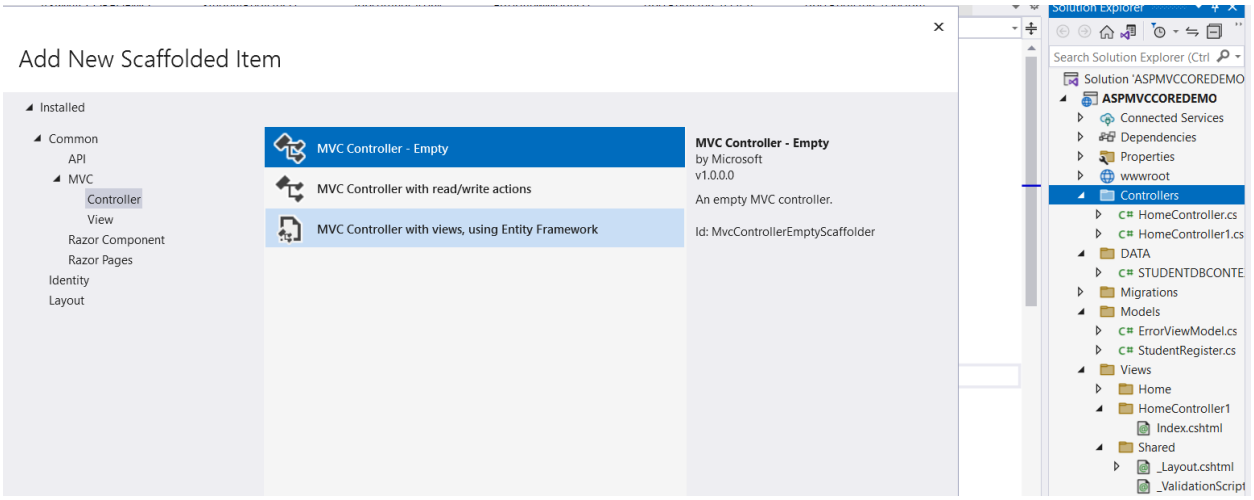
2)update-database

After that you can see you database and tables from sql server object explorer.

CONTROLLER

Step:1

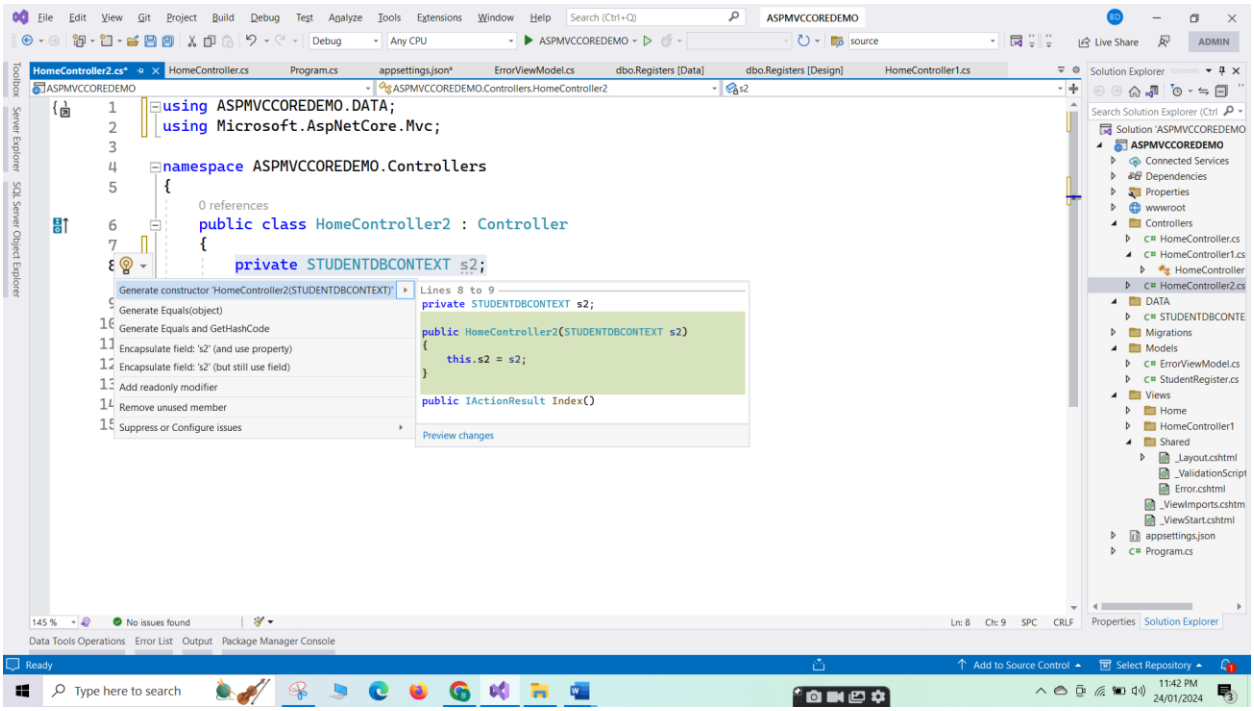
On controller folder do right click and add controller and select MVC controller-empty




```
using Microsoft.AspNetCore.Mvc;

namespace ASPMVCCOREDEMO.Controllers
{
    0 references
    public class HomeController2 : Controller
    {
        0 references
        public IActionResult Index()
        {
            return View();
        }
    }
}
```

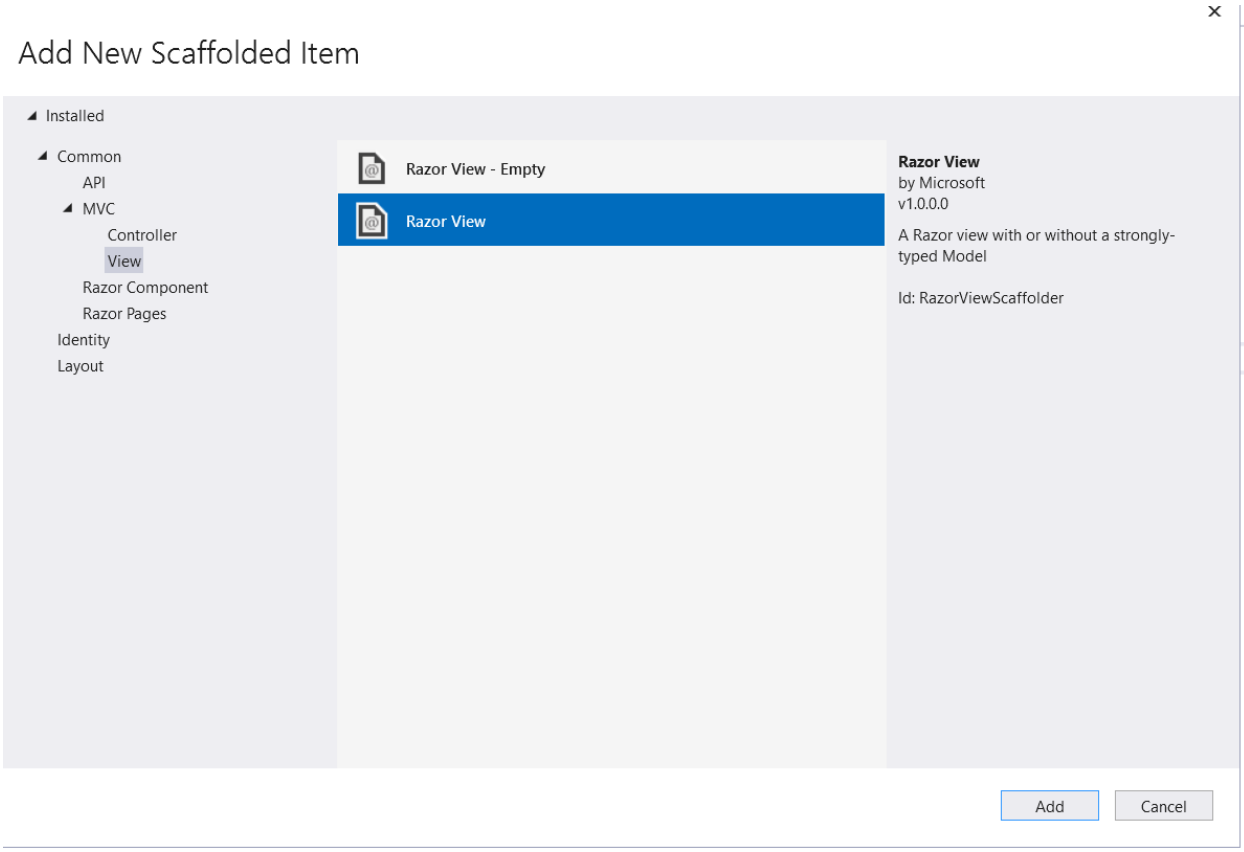
Step 2:you can see by default code now in this controller you have to make one object of your dbcontext file object

Then you can select above line do alt+enter and from suggestions you can select



Step 3:

Now in index method right click then add view and select razor view



After adding view, you have to do design for retrieving the data from the database.

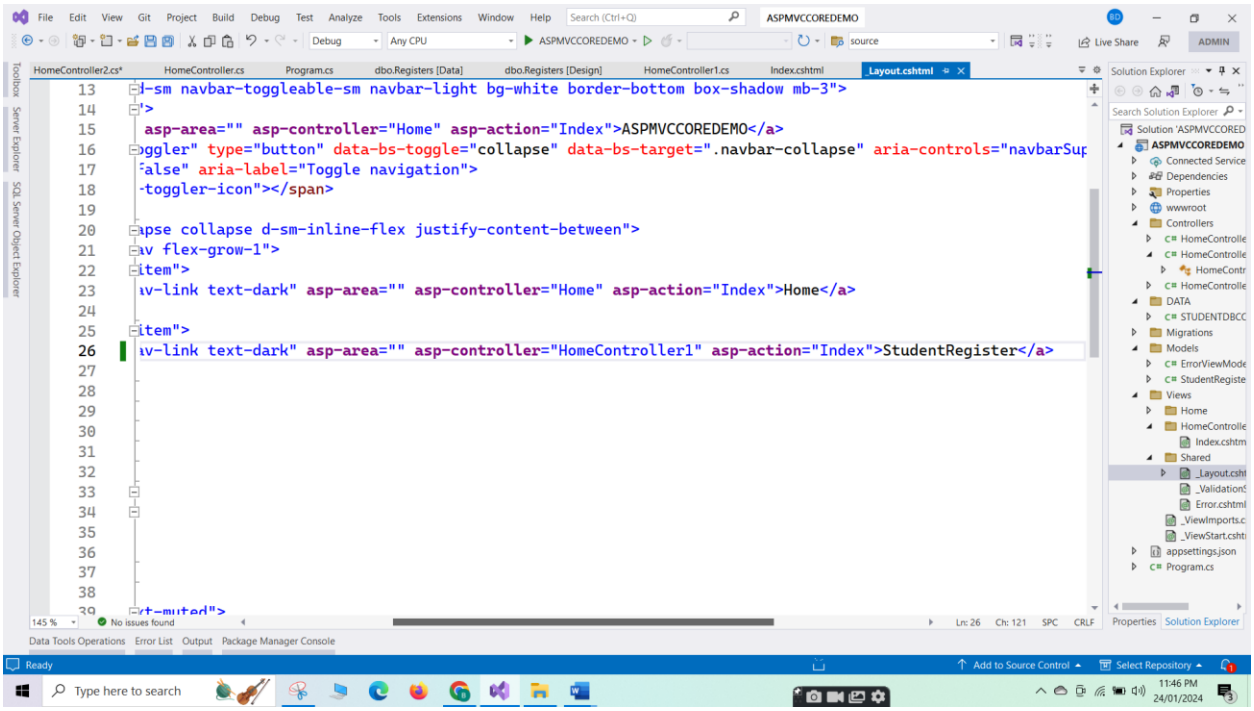
In database add some data on table so we can retrieve the data

Index.cshtml

```
@model IEnumerable<StudentRegister>
@{
    ViewData["Title"] = "Index";
}

<h1>Students List</h1>
<div class="container p-4">
    <div class="row p-3">
        <div class="col-6">
            <h2>  Students </h2>
        </div>
        <div class="col-6">
             <a class=" btn btn-primary" asp-action="ADD" asp-
controller="HomeController1">Add Students</a>
        </div>
    </div>
</div>

<table class="table table-bordered table-striped" style="width:auto">
    <thead>
        <tr>
```

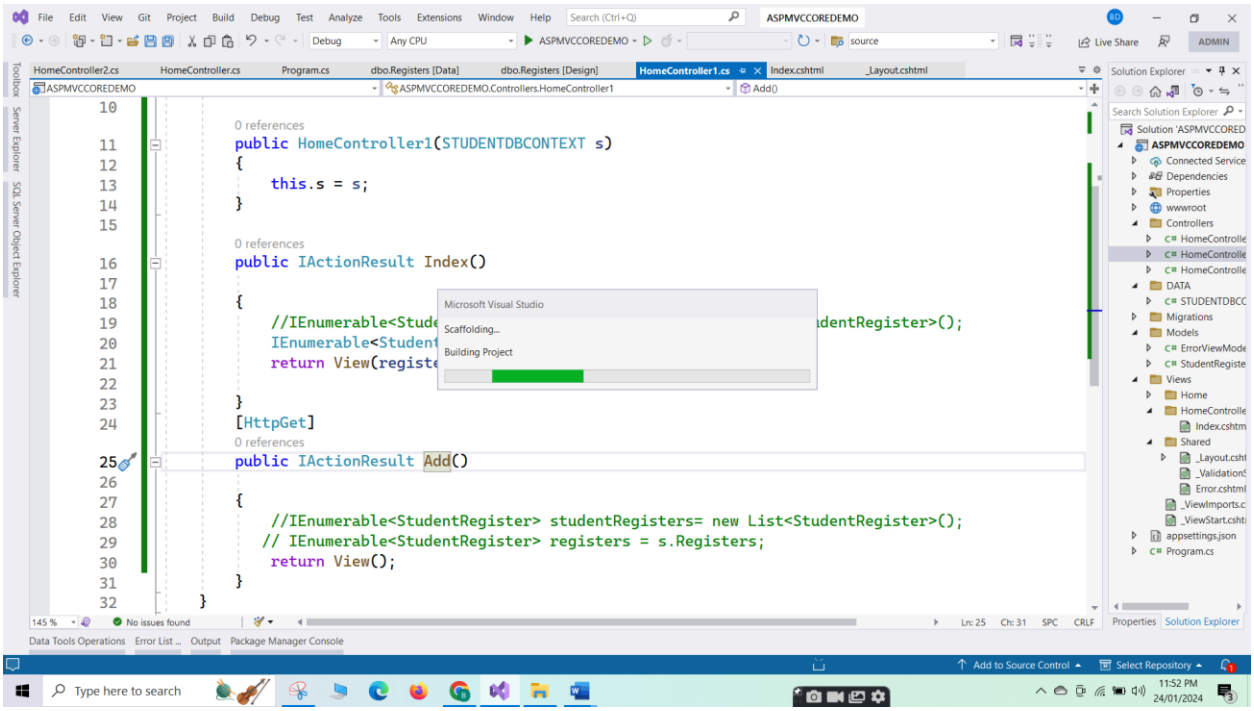
```
            <th>
                Student_Name
            </th>
            <th>
                Student_Email
            </th>
            <th>
                Student_Branch
            </th>
            <th>
                Student_Division
            </th>
        </tr>
    </thead>
    <tbody>
        @foreach(var item in Model){
        <tr>
            <td> @item.student_name</td>
            <td>@item.student_email </td>
            <td>@item.student_branch </td>
            <td>@item.student_division </td>


        </tr>
        }
    </tbody>

</table>
```

Then do changes in layout file.



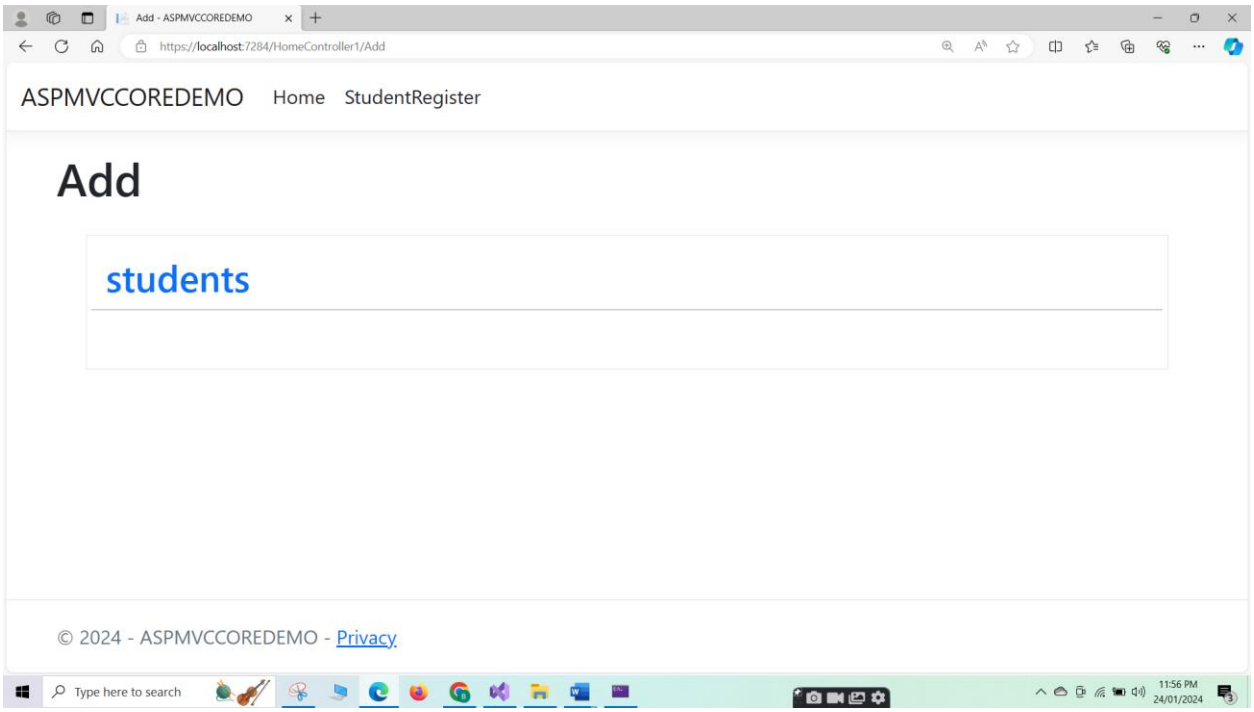
Now I am creating add method for add students in controller.

Again with httpget method you are creating add method



Then in add right click and add razor view named as Add.

```
@model StudentRegister
@{
    ViewData["Title"] = "Add";
}

<h1>Add</h1>
<form method="post">
    <div class=" border p-3 m-4">
        <div class="row pb-3">
            <h2 class="text-primary">students</h2>
            <hr/>
        </div>

    </div>

</form>
```



AFTER THAT

```
@model StudentRegister
@{
    ViewData["Title"] = "Add";
}

<h1>Add</h1>
<form method="post">
    <div class=" border p-3 m-4">
        <div class="row pb-3">
            <h2 class="text-primary">students</h2>
            <hr/>
        </div>
```

```
        <div class="row mb-3">
            <label asp-for="student_name"></label>
            <input asp-for="student_name" class="form-control" />

        </div>
        <div class="row mb-3">
            <label asp-for="student_email"></label>
            <input asp-for="student_email" class="form-control" />

        </div>
        <div class="row mb-3">
            <label asp-for="student_branch"></label>
            <input asp-for="student_branch" class="form-control" />

        </div> <div class="row mb-3">
            <label asp-for="student_division"></label>
            <input asp-for="student_division" class="form-control" />

        </div>
        <button class="btn btn-primary"type="submit">ADD</button>

    </div>
    <a asp-action="Index" asp-controller="studentregister">back to register </a>

</form>
```