

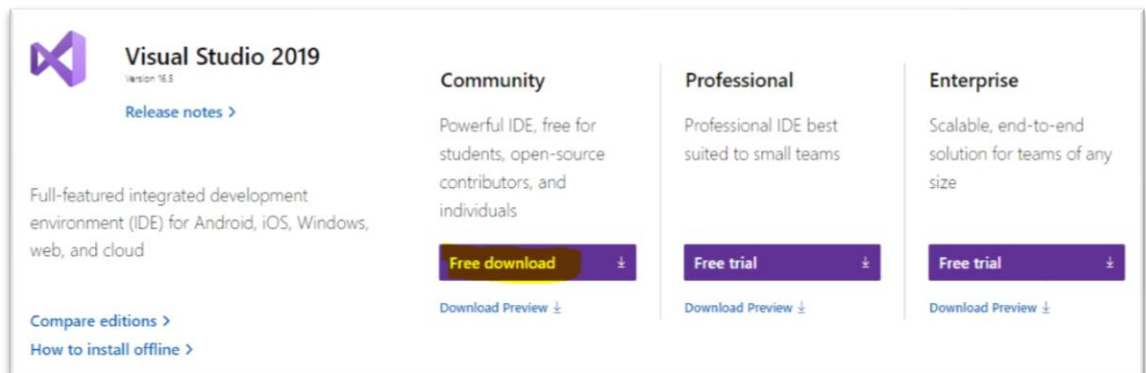
Develop Course Management System Step by Step in .net core

IDE & Tools

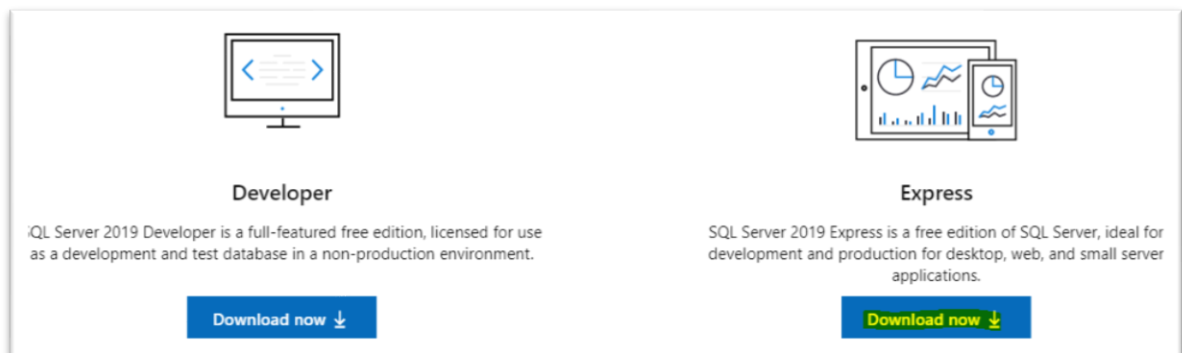
1. Visual Studio 2019 (Community version)
2. MSSQL Server (Express version)
3. SQL Server Management Studio (SSMS)
4. Source Tree

Download Link:

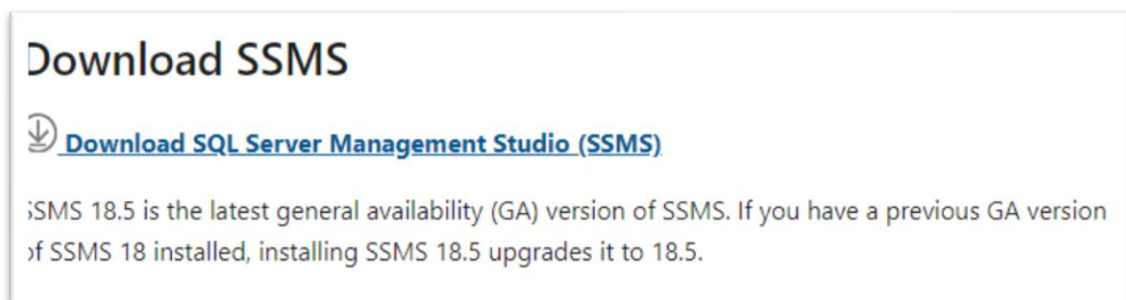
1. Go to this link and download Visual Studio 2019 (Community version) :
<https://visualstudio.microsoft.com/downloads/>



2. Go to this link and download MSSQL Server (Express version)
<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>



3. Go to this link and Download SQL Server Management Studio (SSMS):
<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>



4. Go to this link and Download SourceTree: <https://www.sourcetreeapp.com/>

Develop Course Management System Step by Step in .net core

Installation Process:

Install Visual Studio 2019 (Community)

Step 1 - Make sure your computer is ready for Visual Studio

Before you begin installing Visual Studio:

1. Check the system requirements. These requirements help you know whether your computer supports Visual Studio 2019.
2. Apply the latest Windows updates. These updates ensure that your computer has both the latest security updates and the required system components for Visual Studio.
3. Reboot. The reboot ensures that any pending installs or updates don't hinder the Visual Studio install.
4. Free up space. Remove unneeded files and applications from your %SystemDrive% by, for example, running the Disk Clean-up app.

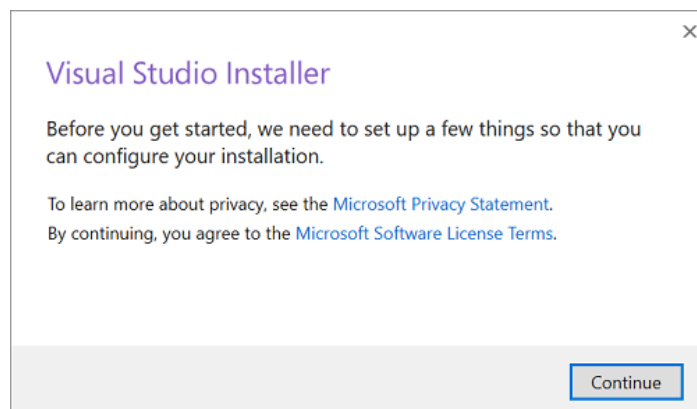
Step 2 - Download Visual Studio

Next, download the Visual Studio bootstrapper file. To do so, choose the following button, choose the edition of Visual Studio that you want, choose Save, and then choose Open folder.

Step 3 - Install the Visual Studio installer

Run the bootstrapper file to install the Visual Studio Installer. This new lightweight installer includes everything you need to both install and customize Visual Studio.

1. From your Downloads folder, double-click the bootstrapper that matches or is like one of the following files:
vs_community.exe for Visual Studio Community
vs_professional.exe for Visual Studio Professional
vs_enterprise.exe for Visual Studio Enterprise
2. We'll ask you to acknowledge the Microsoft License Terms and the Microsoft Privacy Statement. Choose Continue.

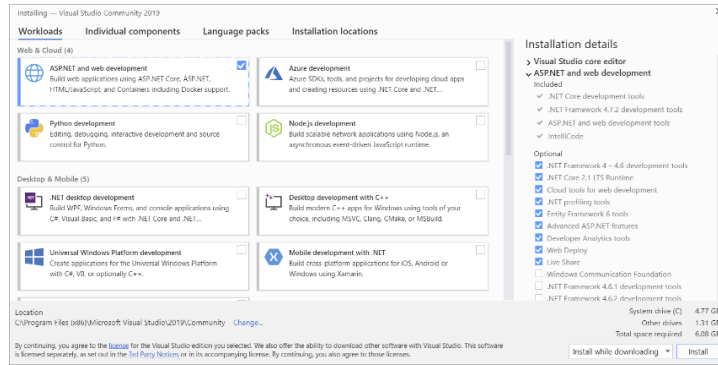


Step 4 - Choose workloads

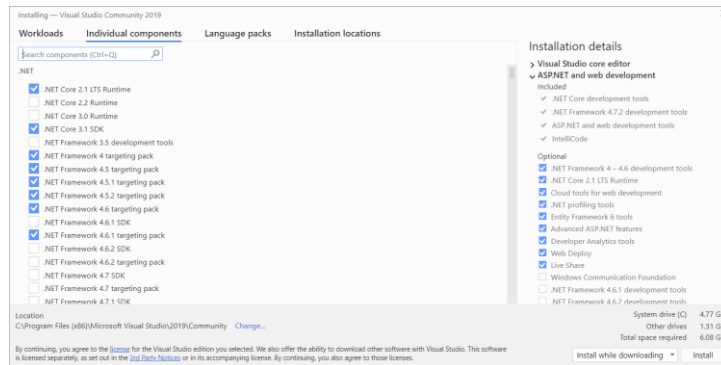
After the installer is installed, you can use it to customize your installation by selecting the feature sets—or workloads—that you want. Here's how.

1. Find the workload you want in the Visual Studio Installer. After you choose the workload(s) you want, choose Install.

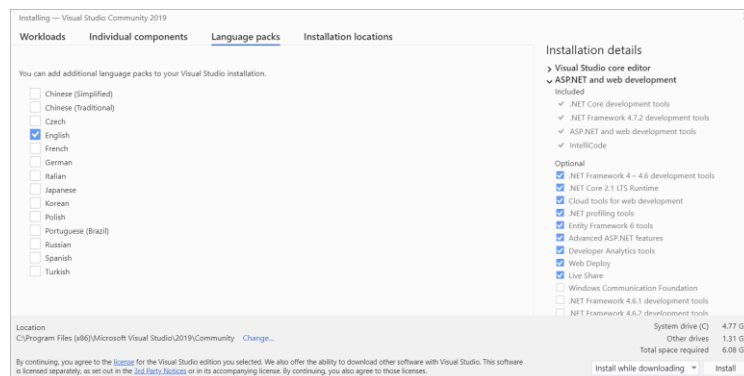
Develop Course Management System Step by Step in .net core



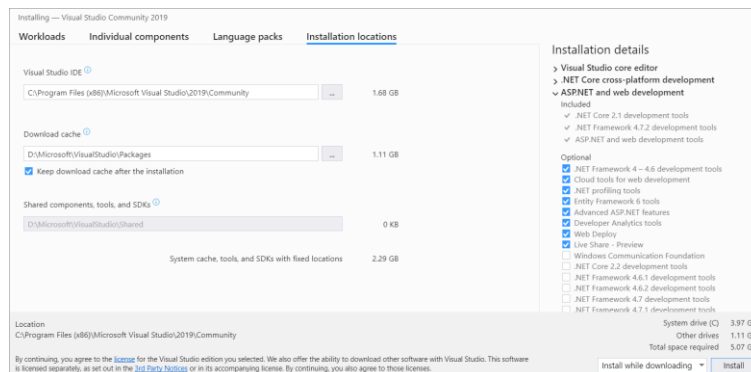
Step 5 - Choose individual components (Optional)



Step 6 - Install language packs (Optional)



Step 7 - Select the installation location (Optional)



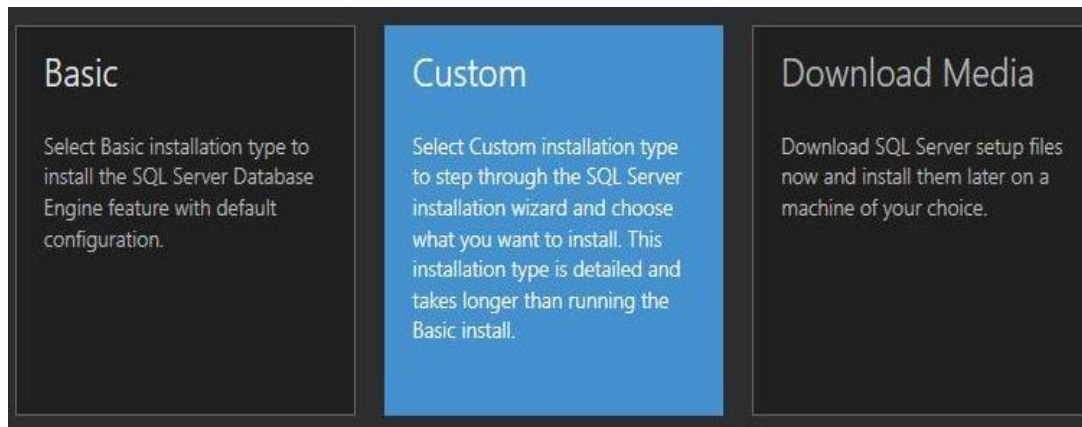
Step 8 - Start developing

After Visual Studio installation is complete, choose the Launch button to get started developing with Visual Studio.

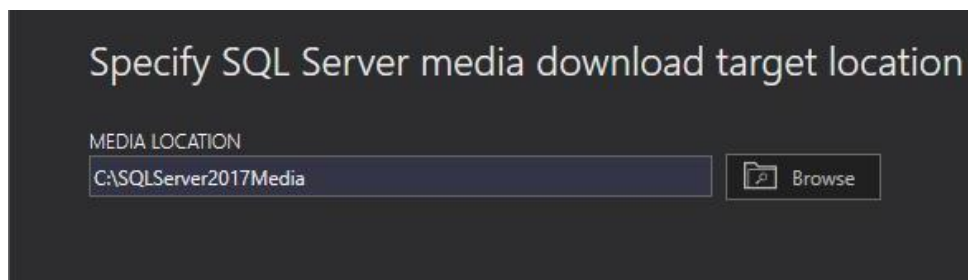
Develop Course Management System Step by Step in .net core

Install Microsoft SQL Server Express

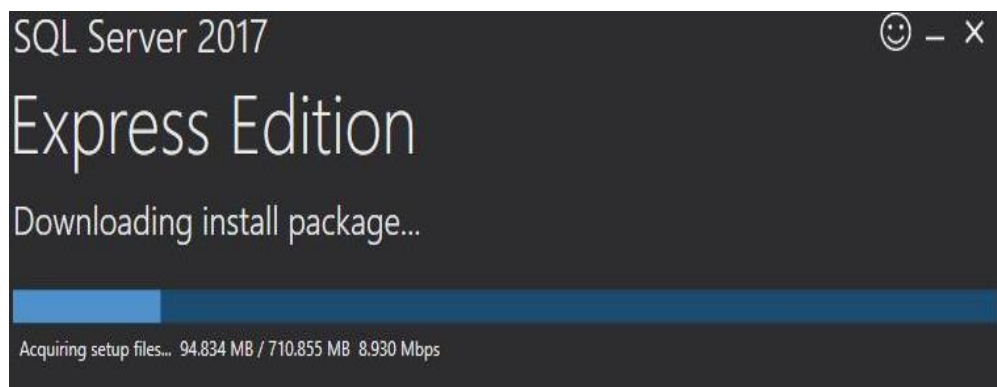
1. Click installation file and select Custom installation, as this allows you to choose what you want to install from the extended tools.



2. On the next page, select a directory for the installation folder. You can leave it as the default or select a custom installation directory.



3. Click the Install button to download the installation packages.

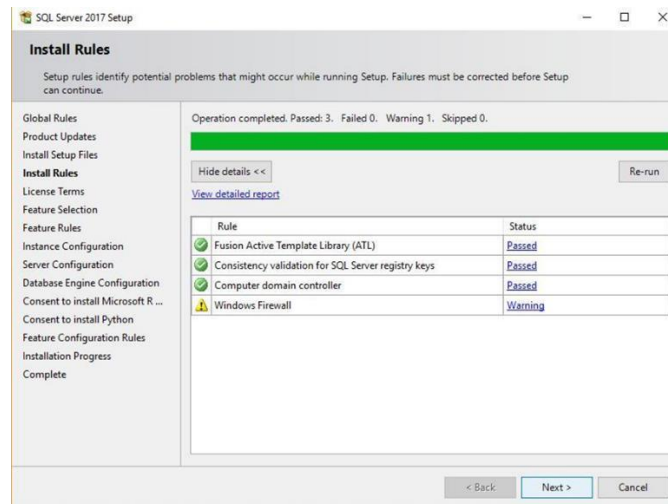


4. After the installation package is downloaded, select New SQL Server stand-alone installation, as shown below

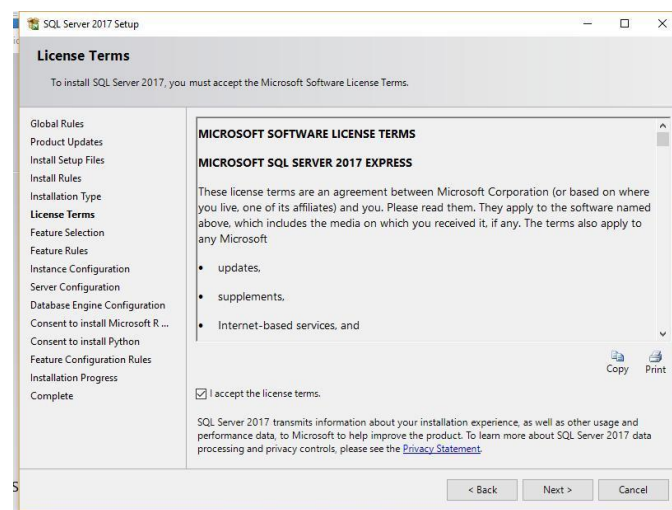


5. On the next screen, you'll see a setup process. Once it completes, click Next.

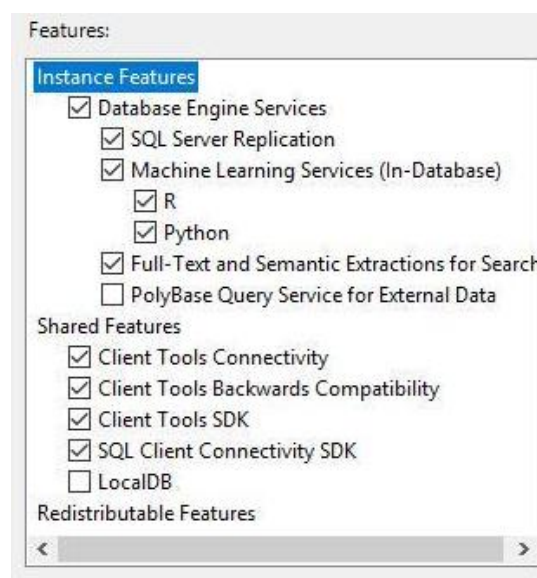
Develop Course Management System Step by Step in .net core



6. You should now see the license terms. Review these if you'd like, and click Accept



7. You'll then be taken to the Feature Selection screen, where you can choose custom SQL Server 2017 features you would like to install.



Develop Course Management System Step by Step in .net core

8. Click the Next button. You can also go with the Default instance name.

The screenshot shows the 'Instance Configuration' step of the SQL Server Enterprise Setup Wizard. On the left is a navigation pane with options: Global Rules, Product Updates, Install Setup Files, Install Rules, License Terms, Feature Selection, Feature Rules, **Instance Configuration**, Server Configuration, Database Engine Configuration, Consent to install Microsoft R..., and Consent to install Python. The main area has the 'Named instance' radio button selected with 'SQLExpress' in the text box. Below it, the 'Instance ID' is 'SQLEXPRESS'. The 'SQL Server directory' is 'C:\Program Files\Microsoft SQL Server\MSSQL14.SQLEXPRESS'. Under 'Installed instances', there is a table with columns: Instance Name, Instance ID, Features, Edition, and Version.

9. Click Next. You'll be taken to the Server Configuration screen. You can leave these settings untouched.

Service	Account Name	Password	Startup Type
SQL Server Database Engine	NT Service\MSSQL\$SQL...		Automatic
SQL Server Launchpad	NT Service\MSSQLLaun...		Automatic
SQL Full-text Filter Daemon Launc...	NT Service\MSSQLFDLa...		Manual
SQL Server Browser	NT AUTHORITY\LOCAL ...		Disabled

10. Click Next. Select windows authentication mode and click next.

11. Click Next to see a summary of the installation process.

The screenshot shows the 'Complete' screen of the SQL Server Enterprise Setup Wizard. It states: 'Your SQL Server 2017 installation completed successfully with product updates.' Below this, there is a section titled 'Information about the Setup operation or possible next steps:' containing a table with columns 'Feature' and 'Status'. The table lists several features that were successfully installed.

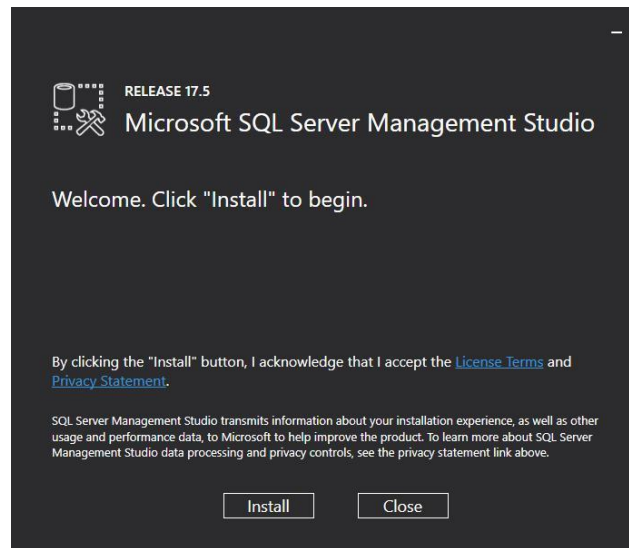
Feature	Status
Full-Text and Semantic Extractions for Search	Succeeded
Python	Succeeded
R	Succeeded
Machine Learning Services (In-Database)	Succeeded
Database Engine Services	Succeeded
SQL Server Replication	Succeeded

12. Once the installation is complete, click Close to finish.

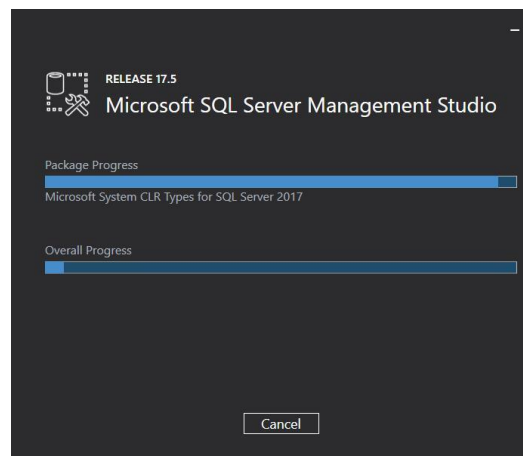
Develop Course Management System Step by Step in .net core

Install Microsoft SQL Server Management Studio

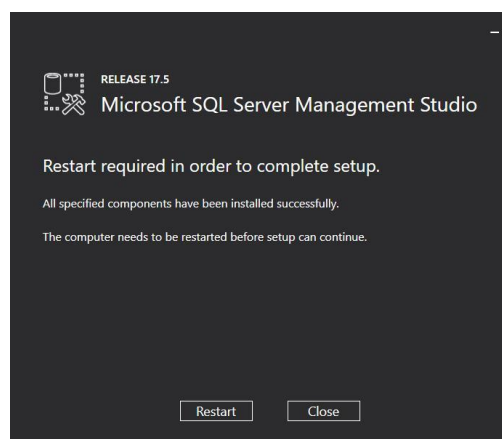
1. Simply run through the steps and click the Install button at the end.



- 2.

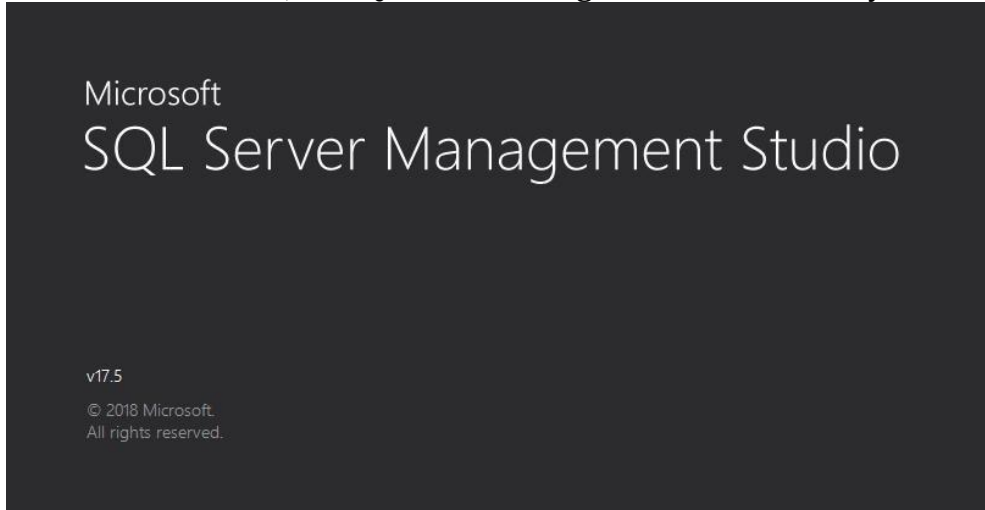


- 3.

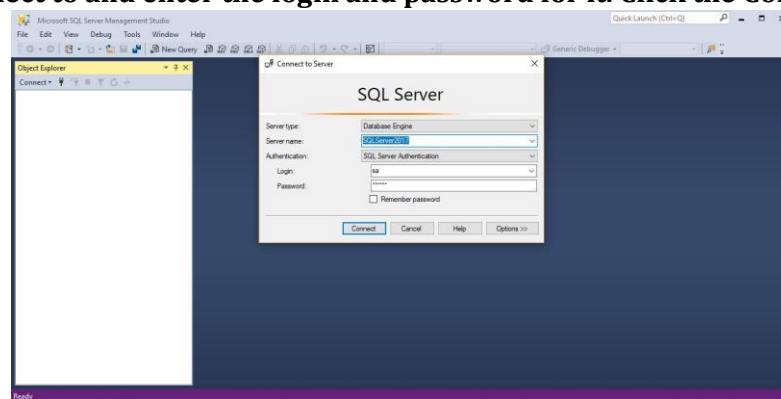


Develop Course Management System Step by Step in .net core

4. After the installation, the SQL Server Management Studio is ready to launch:



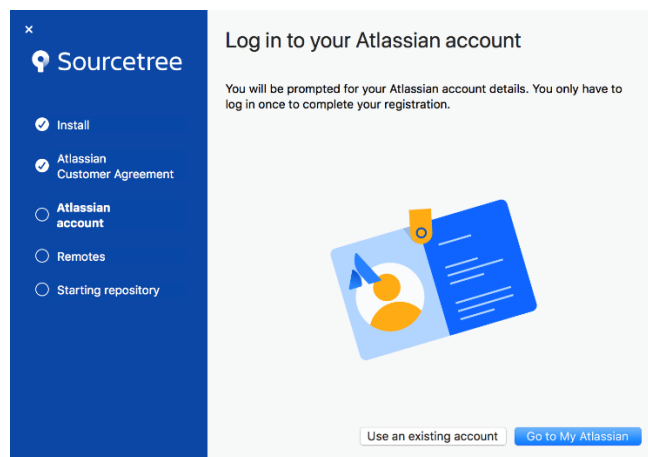
5. First, you'll see the Connect to Server window. Select the name of the server you'd like to connect to and enter the login and password for it. Click the Connect button.



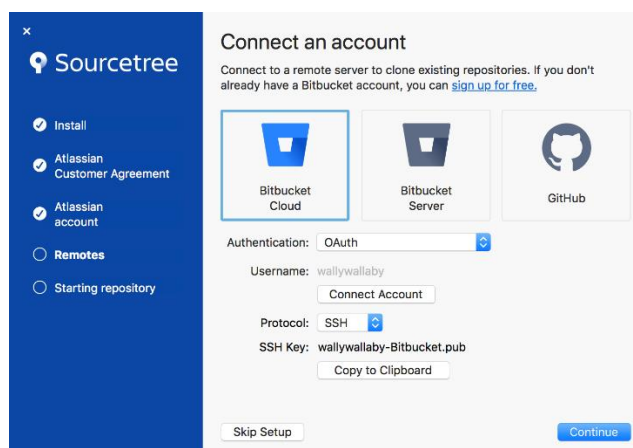
Develop Course Management System Step by Step in .net core

Install SourceTree

1. Go to the [Sourcetree website](#) and click the download button.
2. From the ZIP file you download, click the application file (EXE for Windows or DMG for Mac) to download.
3. After you install, you'll have to agree to the Atlassian Customer Agreement and hit **Continue**.
4. You need an Atlassian account to use Sourcetree. When you get to this screen, click either **Use an existing account** or **Go to My Atlassian** and follow the prompts to [create a new account](#). Once you've got an account, you'll be able to log in with **Use an existing account**.



5. If you want to connect to a remote hosting service, use the default options as you enter your Bitbucket or GitHub credentials. If you can't continue without generating an SSH key, select **HTTPS** to continue or see [Set up an SSH key](#) for more details about setting up SSH with Sourcetree.



6. Follow the prompts until you're done with the setup. If you aren't ready to clone a repository, click **Skip Setup**.

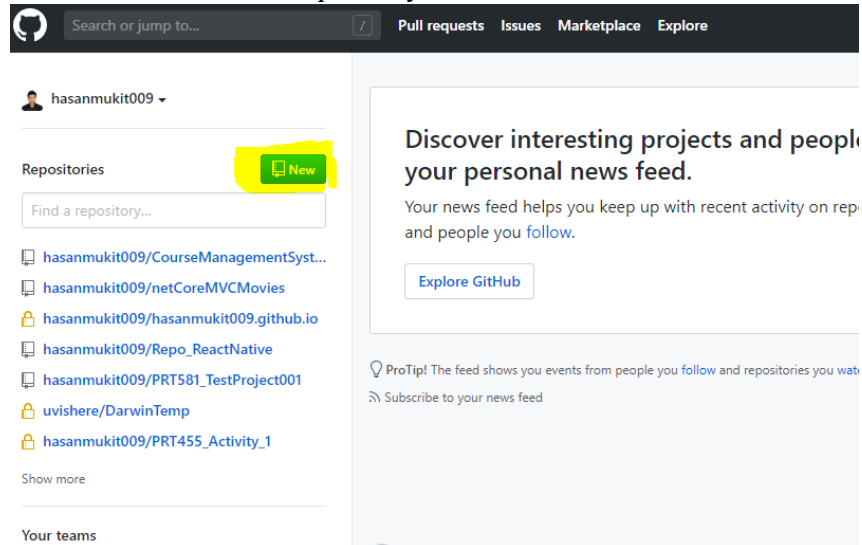
Develop Course Management System Step by Step in .net core

Download the sample project from GitHub

<https://github.com/hasanmukit009/CourseManagementSystem>

GitHub Setup:

1. Go to <https://github.com/> and sign up and open an account
2. Go to GitHub account and create repository from here.



3. Put repository name and create

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner: hasanmukit009

Repository name: ✓

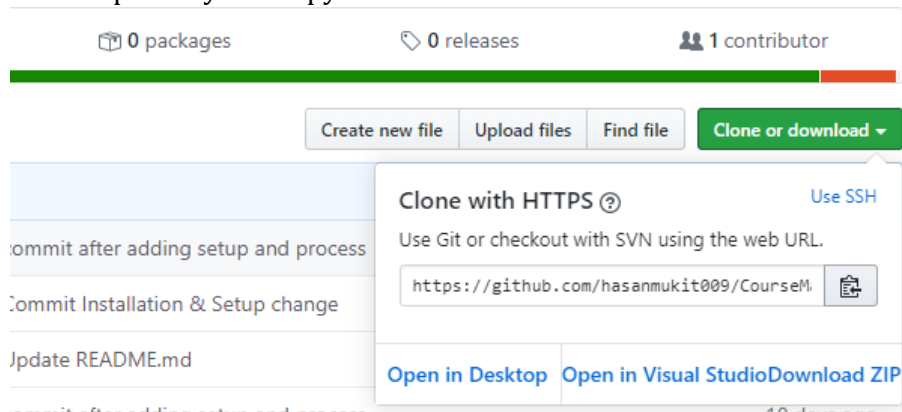
Great repository names are short and memorable. Need inspiration? How about [glowing-octo-fortnight?](#)

Description (optional):

☒ Public
Anyone can see this repository. You choose who can commit.

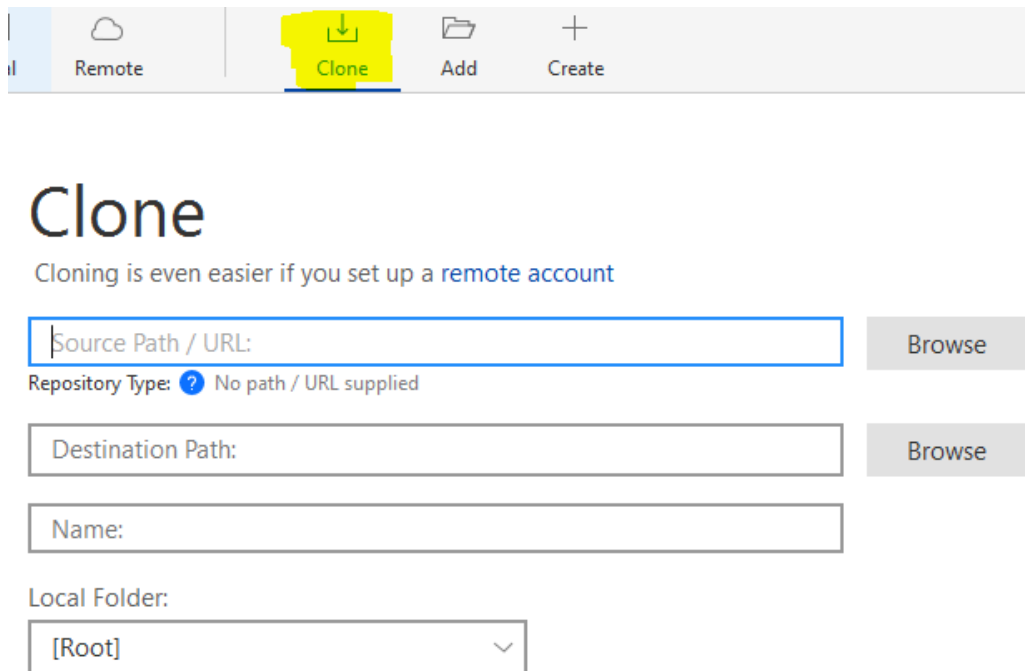
☐ Private

4. Now go to the repository and copy this link



Develop Course Management System Step by Step in .net core

5. Open source tree and click clone



Clone

Cloning is even easier if you set up a [remote account](#)

Source Path / URL: Browse

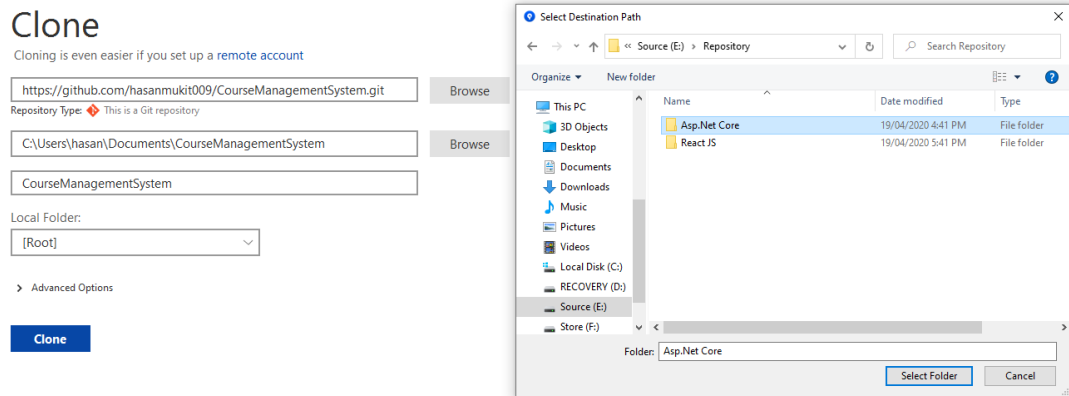
Repository Type: ? No path / URL supplied

Destination Path: Browse

Name:

Local Folder:

6. In the source path put GitHub copied link and in destination path give local pc folder path



Clone

Cloning is even easier if you set up a [remote account](#)

Source Path / URL: Browse

Repository Type: This is a Git repository

Destination Path: Browse

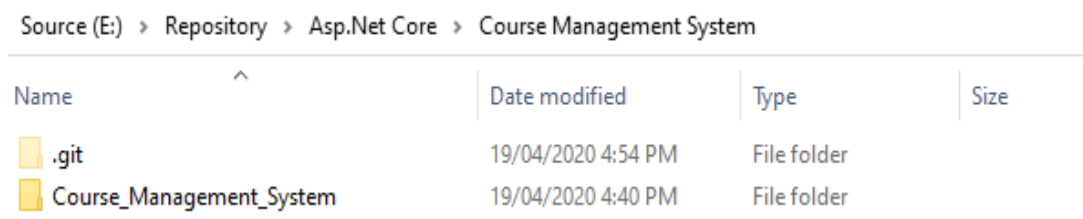
Name:

Local Folder:

> Advanced Options

Clone

7. Now click clone and your GitHub repository and Local PC folder will be connected.

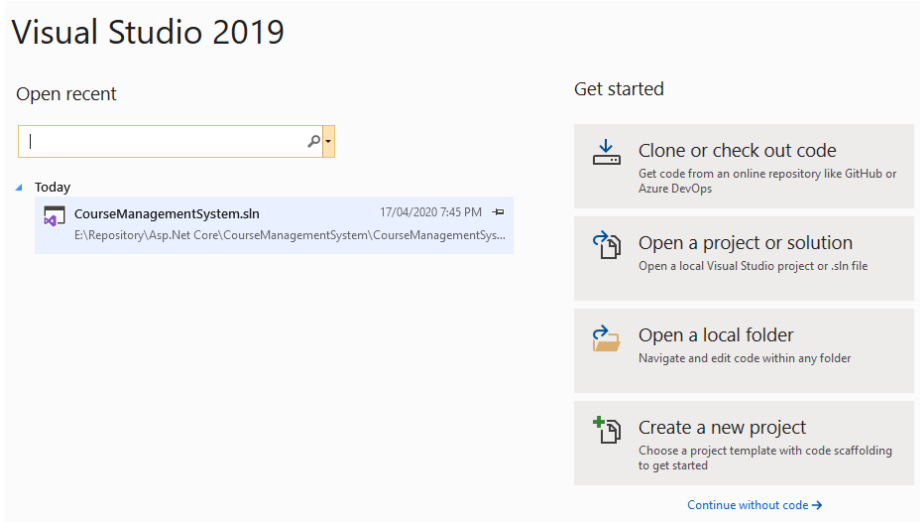


Source (E:) > Repository > Asp.Net Core > Course Management System			
Name	Date modified	Type	Size
.git	19/04/2020 4:54 PM	File folder	
Course_Management_System	19/04/2020 4:40 PM	File folder	

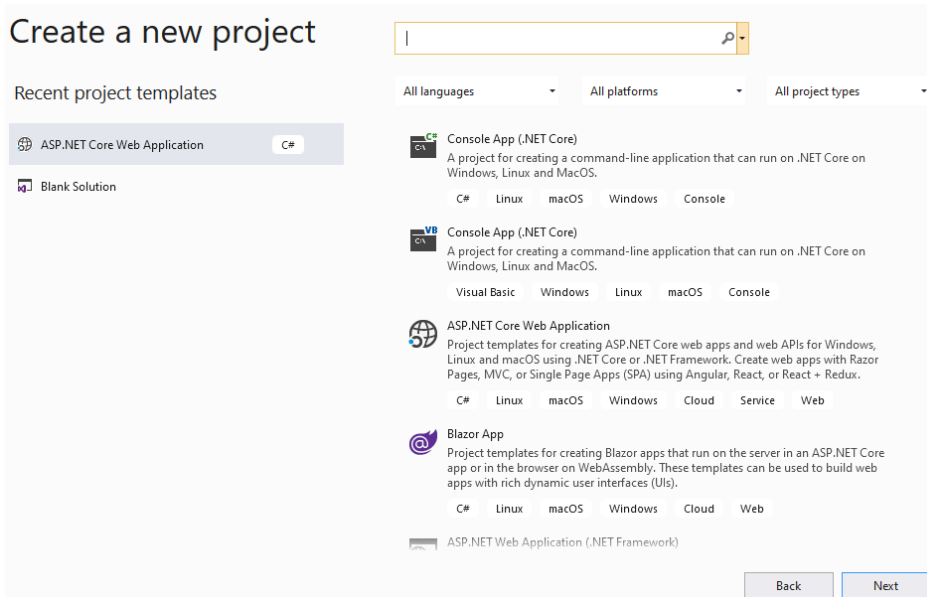
8. Now go to that folder.
9. Create visual studio project here in this folder.

Develop Course Management System Step by Step in .net core

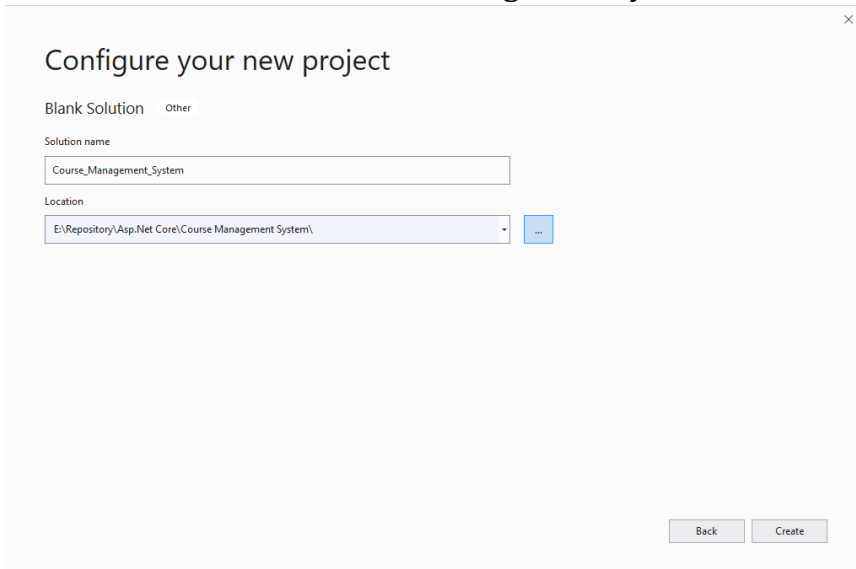
10. Open Visual Studio 2019 and click **Create a new project**



11. Select **Blank Solution** and click next

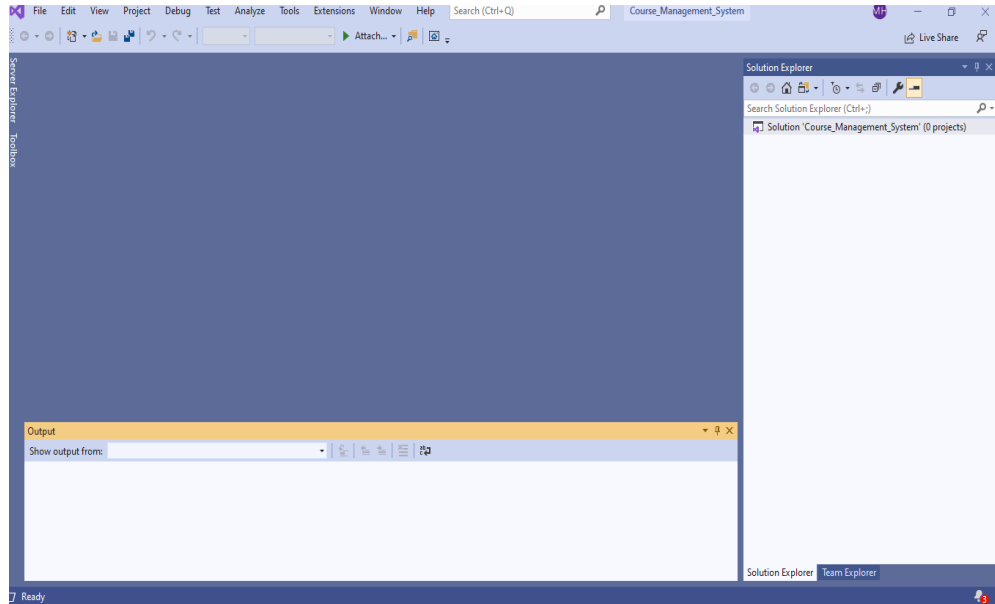


12. Write the solution name “**Course_Management_System**” and click **Create**.

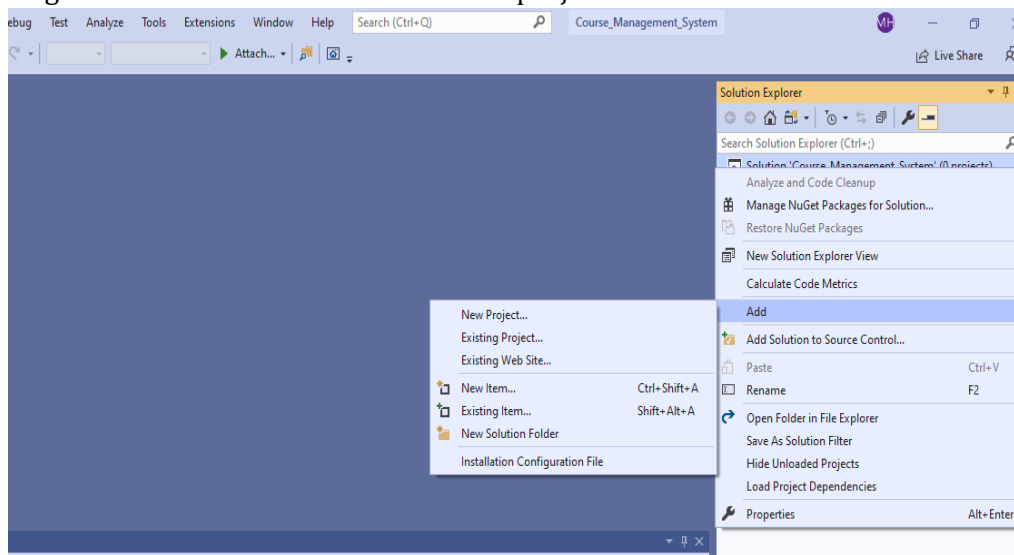


Develop Course Management System Step by Step in .net core

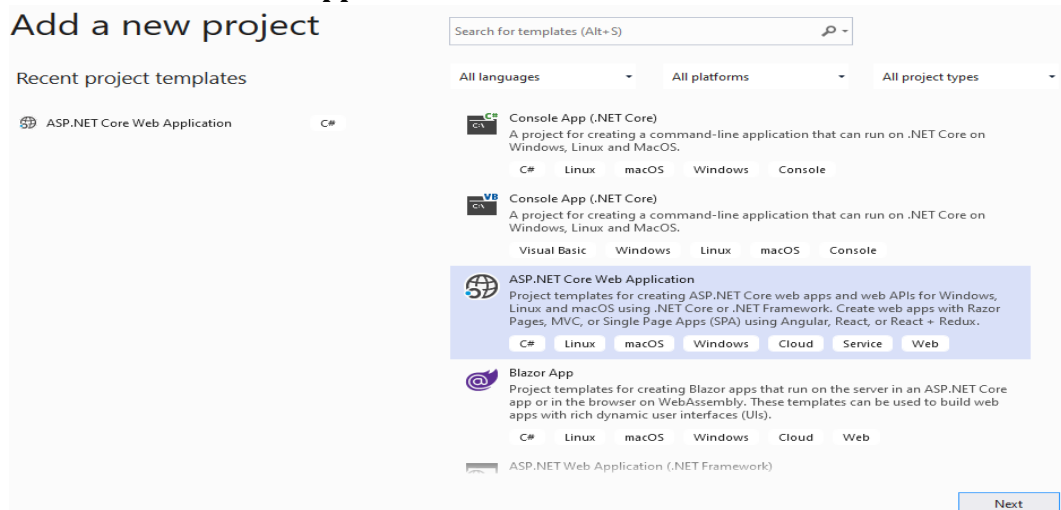
13. Now solution will be like this.



14. Now right click on the solution and add new project

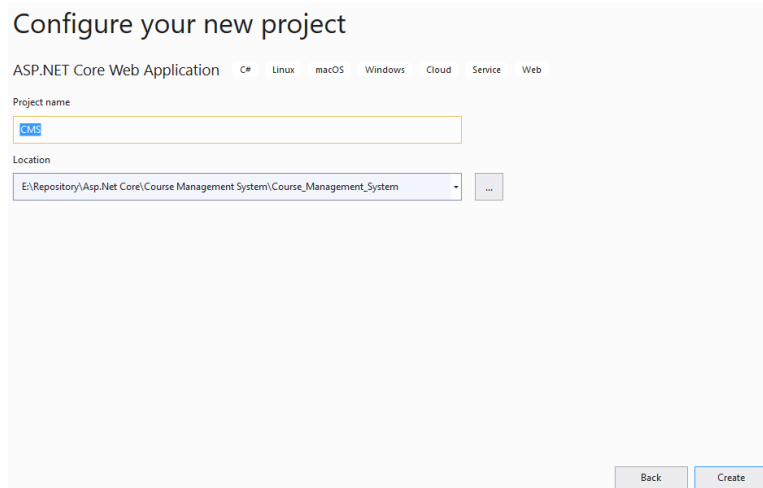


15. Select ASP.Net core web application and click next.

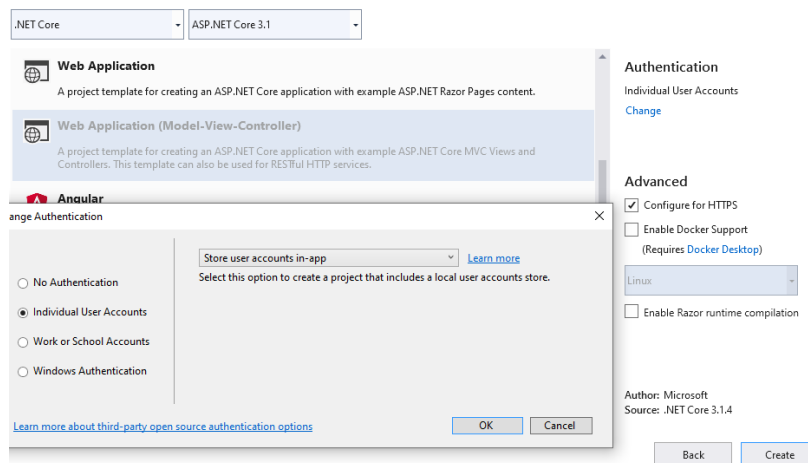


Develop Course Management System Step by Step in .net core

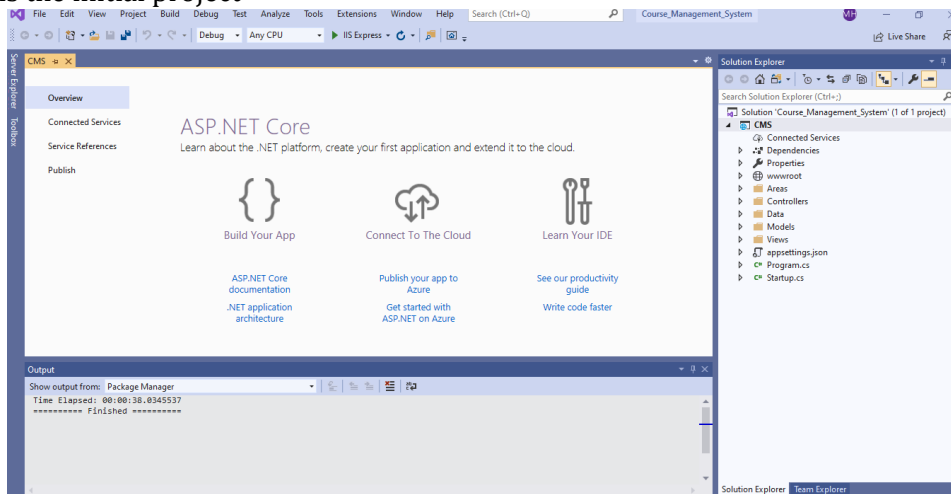
16. Give the **Project Name** and click create.



17. After creating Project select these things and click create
Create a new ASP.NET Core web application



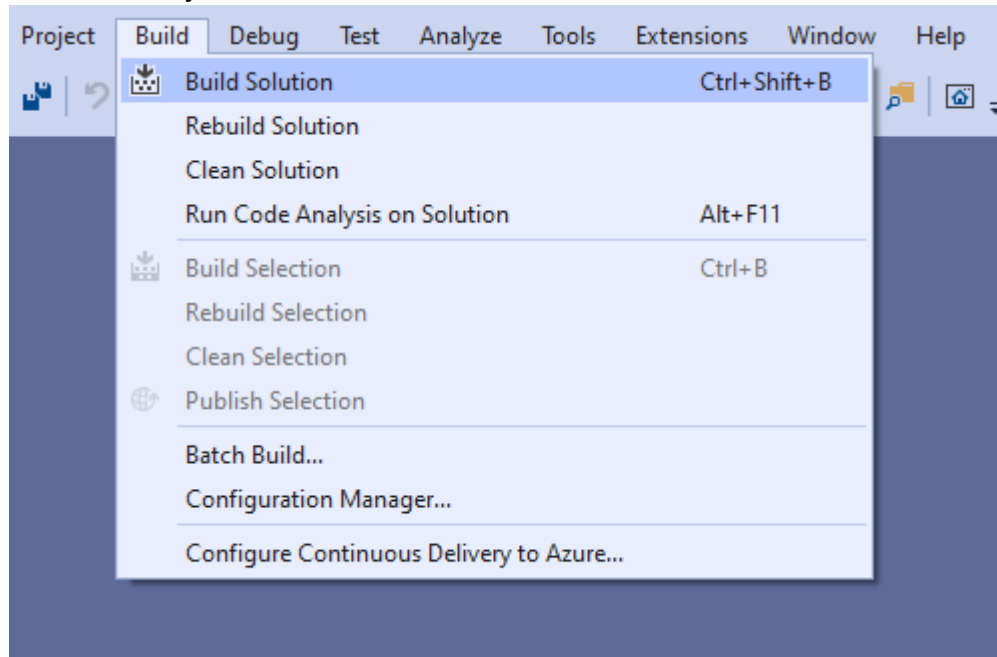
18. So, this the initial project



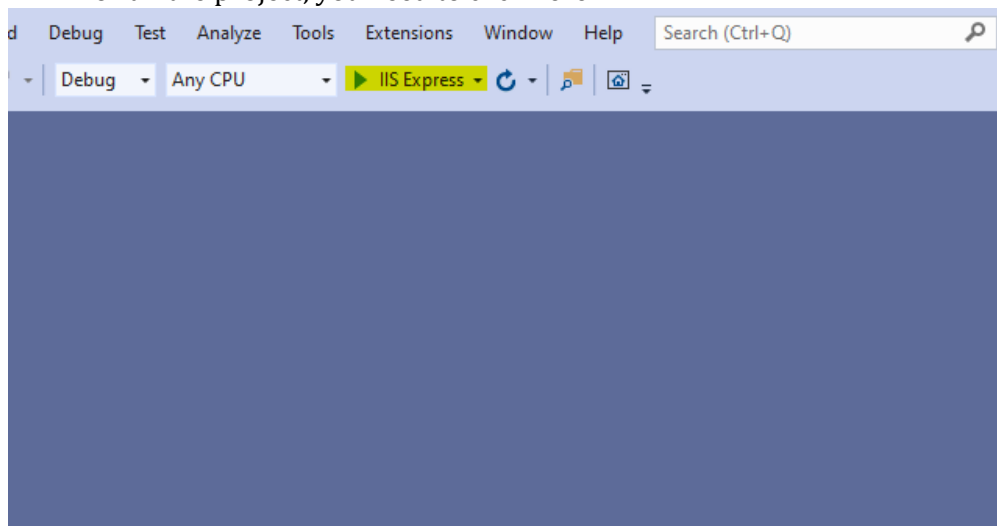
Develop Course Management System Step by Step in .net core

19. Build the project and Click run

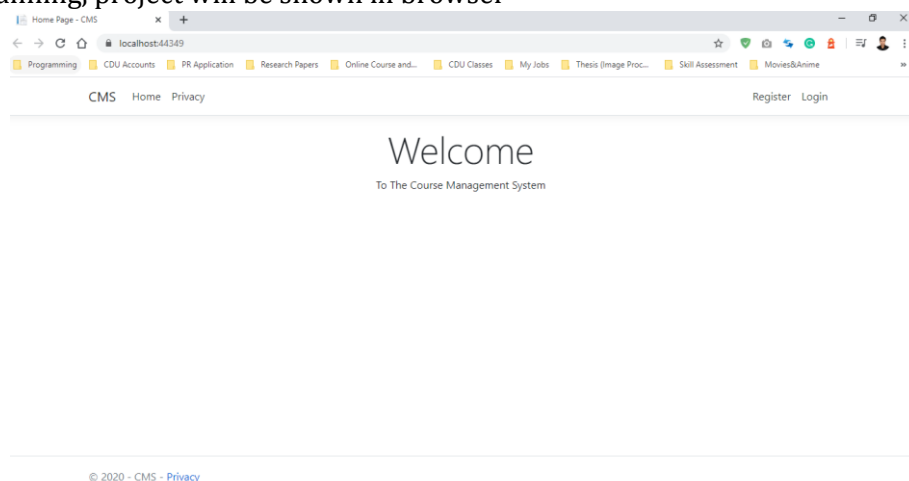
- To build a project you need to right click on the solution and click build or rebuild Or you can click here



- To run the project, you need to click here



After running, project will be shown in browser

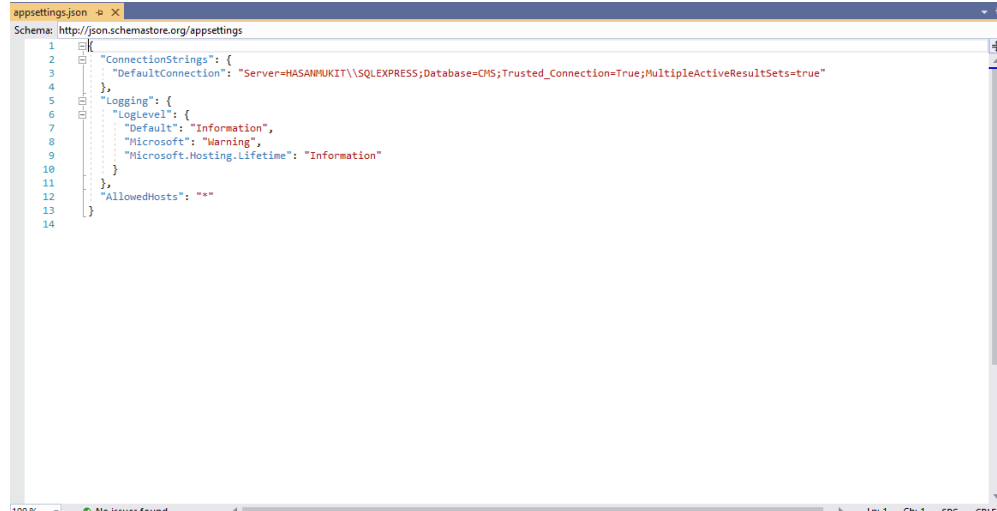


Develop Course Management System Step by Step in .net core

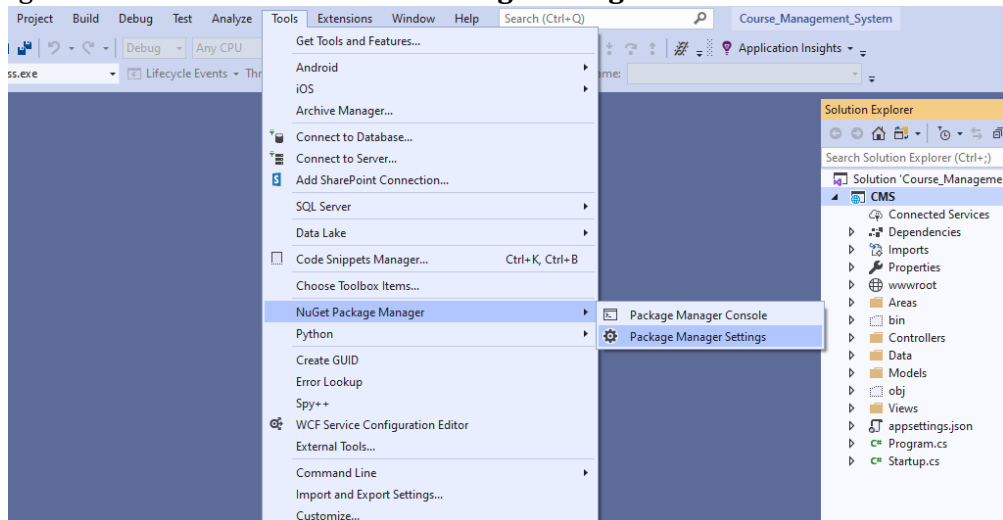
20. Now change the **connection string** here

Copy the code from this link

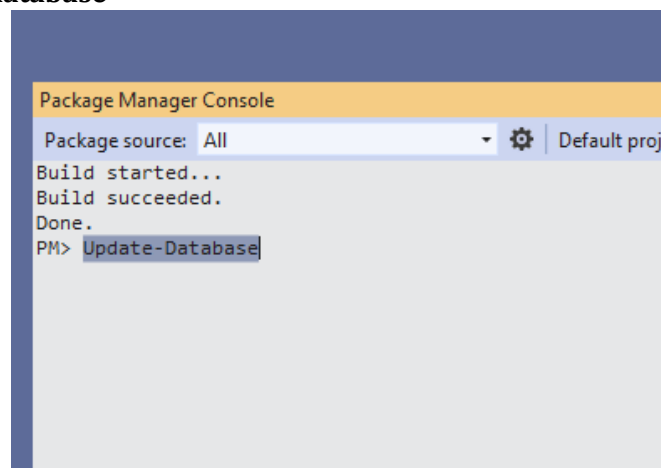
<https://github.com/hasanmukit009/CourseManagementSystem/blob/master/CourseManagementSystem/CMS/appsettings.json>



21. Now go to the visual studio and click **Package Manager Console**

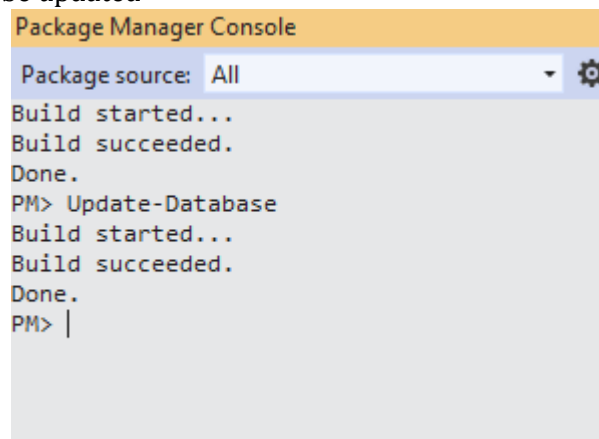


22. Write "update-database"



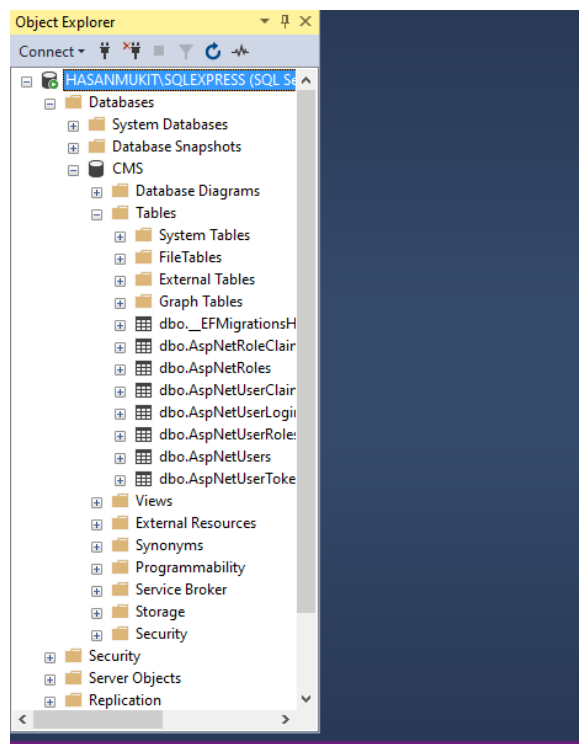
Develop Course Management System Step by Step in .net core

23. Then database will be updated

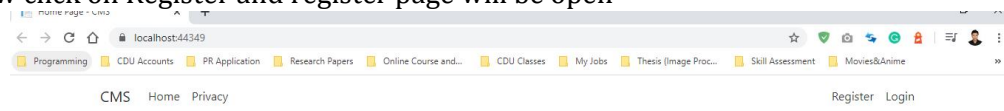


```
Package Manager Console
Package source: All
Build started...
Build succeeded.
Done.
PM> Update-Database
Build started...
Build succeeded.
Done.
PM> |
```

24. When you go to **SQL SERVER MANAGEMENT STUDIO**, you will see the database CMS has been created.



25. Now click on Register and register page will be open



Welcome

To The Course Management System

Develop Course Management System Step by Step in .net core

26. Register page will be like this

Register - CMS

localhost:44349/identity/Account/Register

Programming CDU Accounts PR Application Research Papers Online Course and... CDU Classes My Jobs Thesis (Image Proc... Skill Assessment Movies&Anime

CMS Home Privacy Register Login

Register

Create a new account.

Email

Password

Confirm password

Register

Use another service to register.

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.

27. Give all the information and click register and a new user(hasanmukit009@gmail.com) has been created. Now you can login

CDU Accounts PR Application Research Papers Online Course and... CDU Classes My Jobs Thesis (Image Proc... Skill Assessment Movies&Anime

CMS Home Privacy Hello hasanmukit009@gmail.com! Logout

Welcome

To The Course Management System

28. Click log out and then click login

CMS Home Privacy Register Login

Log in

Use a local account to log in.

Email

hasanmukit009@gmail.com

Password

Remember me?

Log in

Use another service to log in.

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.

[Forgot your password?](#)

[Register as a new user](#)

© 2020 - CMS - [Privacy](#)

Develop Course Management System Step by Step in .net core

29. This page will open

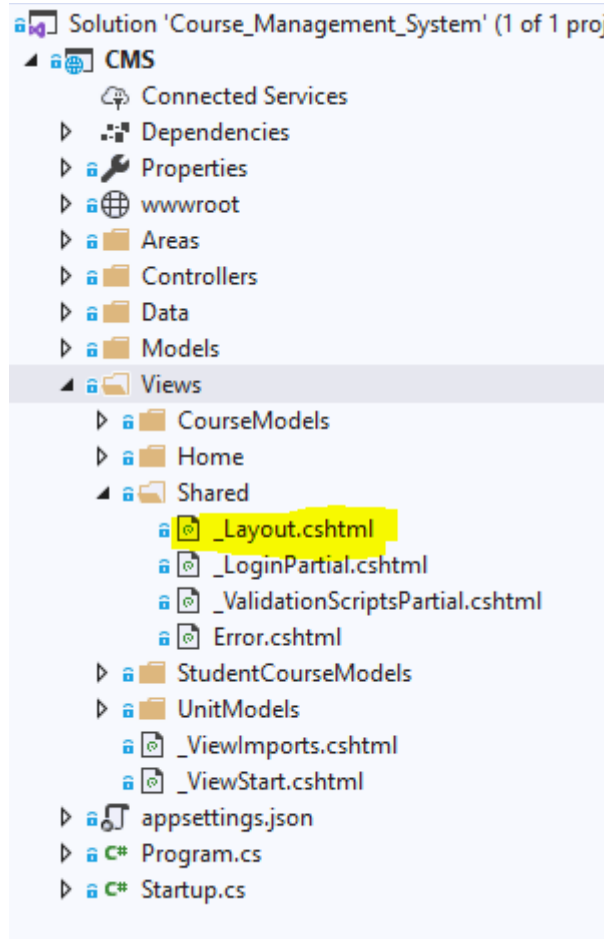
CMS Home

Hello hasanmukit009@gmail.com! Logout

Welcome

To The Course Management System

30. Open layout page (/CMS/Views/Shared/_Layout.cshtml) from project.



Copy the code from this link

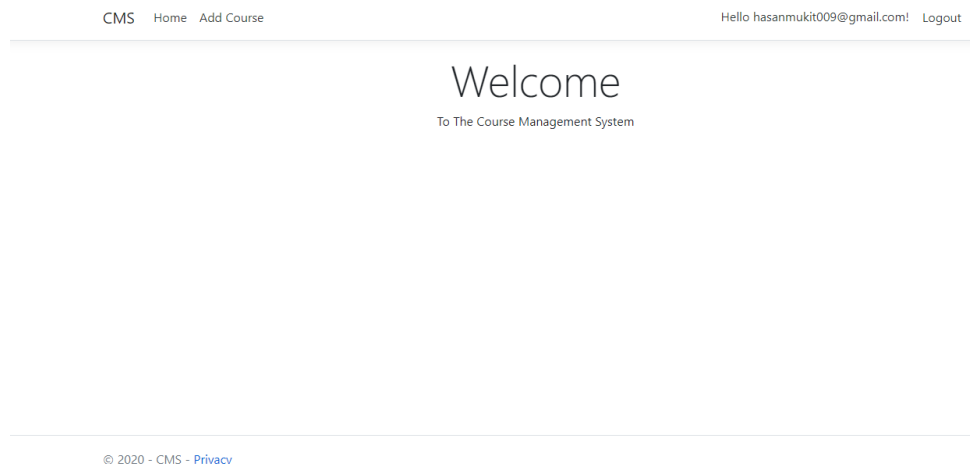
(https://github.com/hasanmukit009/CourseManagementSystem/blob/master/CourseManagementSystem/CMS/Views/Shared/_Layout.cshtml)

Add this code and run

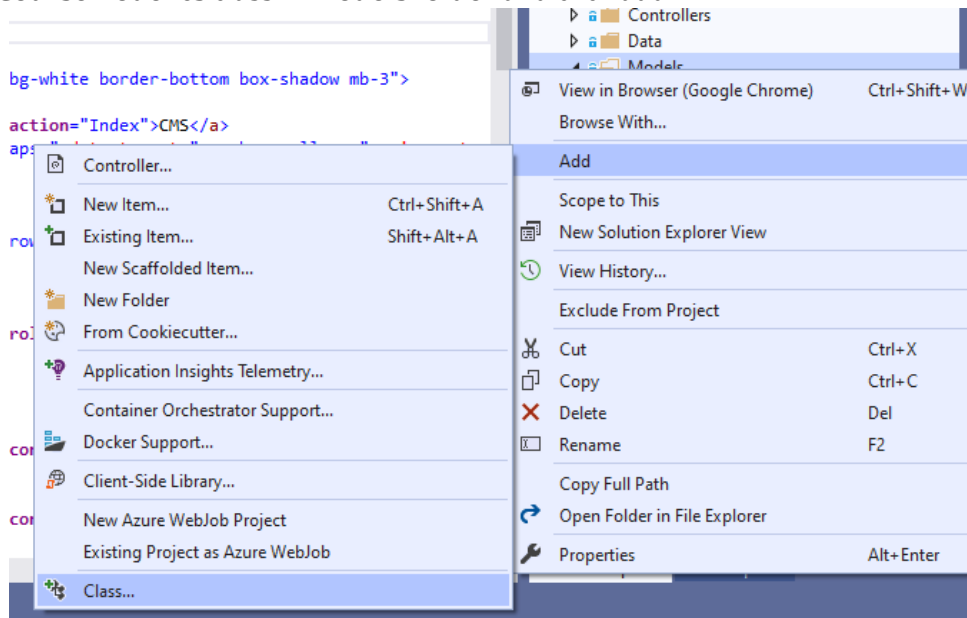
Develop Course Management System Step by Step in .net core

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>@ViewData["Title"] - CMS</title>
<link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
<link rel="stylesheet" href="~/css/site.css" />
</head>
<body>
<header>
<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
<div class="container">
<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">CMS</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse">
<partial name="_LoginPartial" />
<ul class="navbar-nav flex-grow-1">
<li class="nav-item">
<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
</li>
<li class="nav-item">
<a class="nav-link text-dark" asp-area="" asp-controller="Course" asp-action="Index">Add Course</a>
</li>
</ul>
</div>
</div>
</nav>
</header>
<div class="container">
<main role="main" class="pb-3">
@RenderBody()
</main>
</div>
</body>
</html>
```

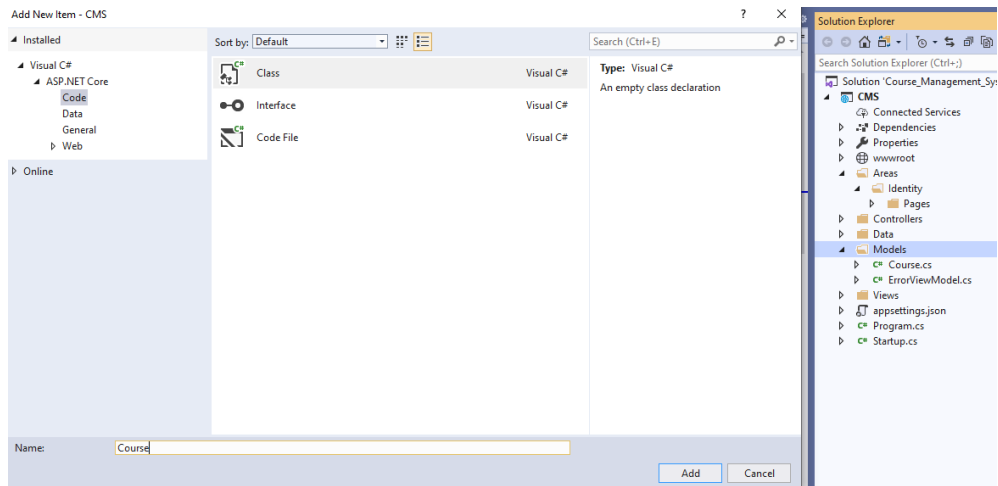
31. Now click run and this will open



32. Add CourseModel.cs class in Models folder and click add



Develop Course Management System Step by Step in .net core



33. Add these codes in **CourseModel.cs** Class

Copy code from this link

([https://github.com/hasanmukit009/CourseManagementSystem/blob/master/CourseManagement System/CMS/Models/CourseModel.cs](https://github.com/hasanmukit009/CourseManagementSystem/blob/master/CourseManagement%20System/CMS/Models/CourseModel.cs))

```
namespace CMS.Models
{
    [Table("Course")]
    22 references
    public class CourseModel
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        12 references
        public int ID { get; set; }
        12 references
        public string CourseName { get; set; }
    }
}
```

34. Now open Package Manager Console from (**Tools>Nuget Manager>Package Manager console**) and write **"Add-Migration AddCourse"** command

```
Package Manager Console
Package source: All Default project: CMS
Package Manager Console Host Version 5.5.0.6473
Type 'get-help NuGet' to see all available NuGet commands.
PM> Add-Migration AddCourse
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
```

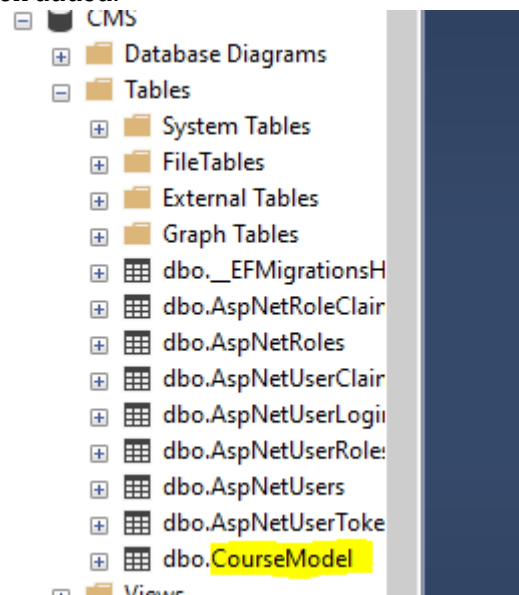
35. Now execute **"Update-Database"** command in Package Manager Console

Develop Course Management System Step by Step in .net core

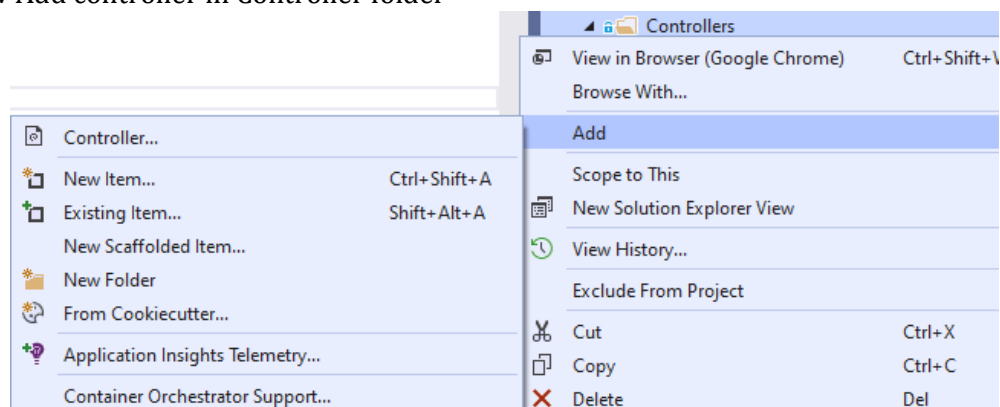
```
PM> Update-Database
Build started...
Build succeeded.
Done.
PM> |
```

100 %

36. Now open **SQL SERVER MANAGEMENT STUDIO** and you will see the new table **"CourseModel"** has been added.

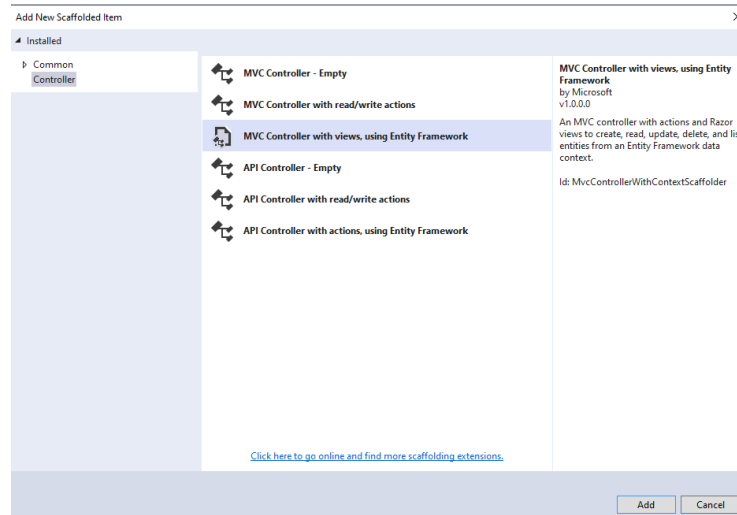


37. Now Add controller in Controller folder

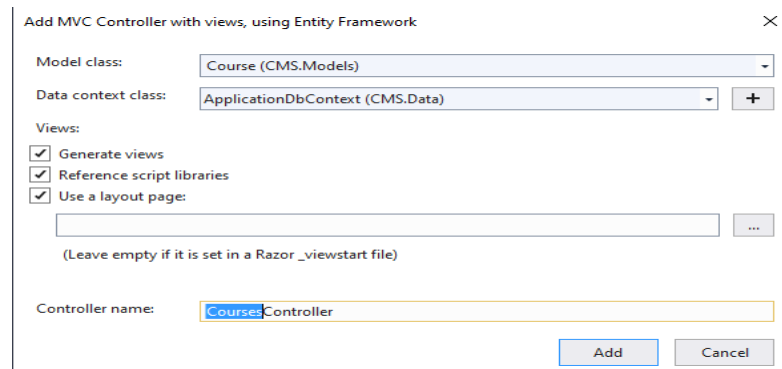


Select this and click add

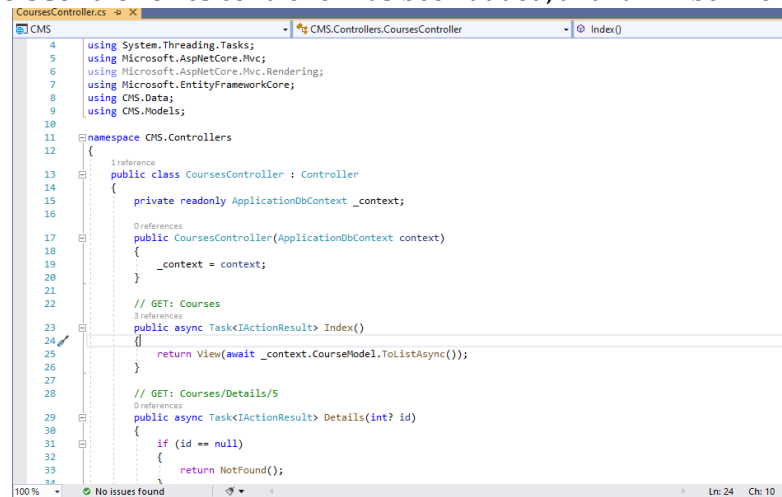
Develop Course Management System Step by Step in .net core



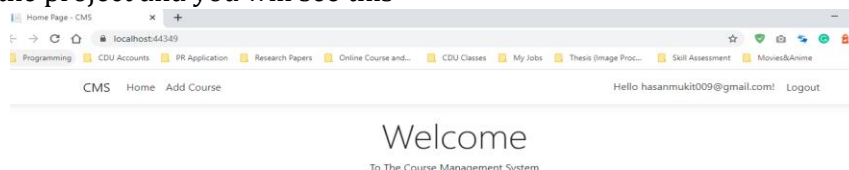
38. Give all this information and click Add



39. CourseModelsController.cs controller has been added, and it will be like this



40. Now run the project and you will see this



Develop Course Management System Step by Step in .net core

41. When you click Add course it will be shown like this and then click **Create New**

CMS Home Add Course Hello hasanmukit009@gmail.com! Find Course Logout

Index

[Create New](#)

CourseName	
Graduate Certificate of Data Science (UDASC1 - 2020)	Edit Details Delete
Graduate Diploma of Data Science (TDASC1 - 2020)	Edit Details Delete
Master of Data Science (SDASC1 - 2020)	Edit Details Delete
Master of Information Technology (Cyber Security) (SITCS2 - 2020)	Edit Details Delete
Master of Information Technology (Information Systems and Data Science) (SITIS2 - 2020)	Edit Details Delete
Master of Information Technology (Software Engineering) (SITSE1 - 2020)	Edit Details Delete
Graduate Certificate of Information Technology (Cyber Security) (UITCS1 - 2020)	Edit Details Delete

42. This page will open and write Course Name in the text box and click **Create**

CMS Home Add Course Hello hasanmukit009@gmail.com! Find Course Logout

CourseName

[Create](#)

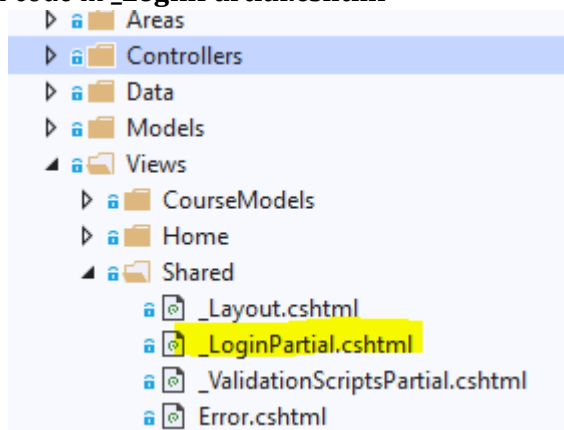
[Back to List](#)

43. New course will be added to the list

CourseName	
Graduate Certificate of Data Science (UDASC1 - 2020)	Edit Details Delete
Graduate Diploma of Data Science (TDASC1 - 2020)	Edit Details Delete
Master of Data Science (SDASC1 - 2020)	Edit Details Delete
Master of Information Technology (Cyber Security) (SITCS2 - 2020)	Edit Details Delete
Master of Information Technology (Information Systems and Data Science) (SITIS2 - 2020)	Edit Details Delete
Master of Information Technology (Software Engineering) (SITSE1 - 2020)	Edit Details Delete
Graduate Certificate of Information Technology (Cyber Security) (UITCS1 - 2020)	Edit Details Delete

44. You can **Edit** and **Delete** and show **Details** from here

45. Now add this piece of code in **_LoginPartial.cshtml**



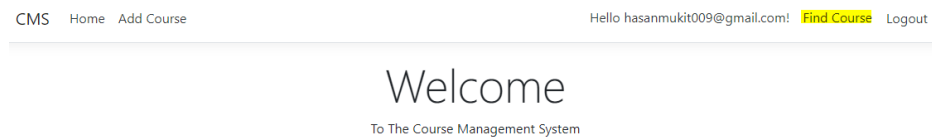
Copy the code from this link

(https://github.com/hasanmukit009/CourseManagementSystem/blob/master/CourseManagementSystem/CMS/Views/Shared/_LoginPartial.cshtml)

Develop Course Management System Step by Step in .net core

```
.LoginPartial.cshtml | _Layout.cshtml | CoursesController.cs
1  @using Microsoft.AspNetCore.Identity
2  @inject SignInManager<IdentityUser> SignInManager
3  @inject UserManager<IdentityUser> UserManager
4
5  <ul class="navbar-nav">
6      @if (SignInManager.IsSignedIn(User))
7      {
8          <li class="nav-item">
9              <a class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Manage/Index" title="Manage">Hello @User.Ident
10             </li>
11             <li class="nav-item">
12                 <a class="nav-link text-dark" asp-area="" asp-controller="Courses" asp-action="FindCourse">Find Course</a>
13             </li>
14             <li class="nav-item">
15                 <form class="form-inline" asp-area="Identity" asp-page="/Account/Logout" asp-route-returnUrl="@Url.Action("Index",
16                     <button type="submit" class="nav-link btn btn-link text-dark">Logout</button>
17                 </form>
18             </li>
19         }
20         else
21         {
22             <li class="nav-item">
23                 <a class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Register">Register</a>
24             </li>
25             <li class="nav-item">
26                 <a class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Login">Login</a>
27             </li>
28         }
29     </ul>
30
```

46. Run the project and it will be like this



47. Add this code in CourseModelsController.cs

Copy code from this link

<https://github.com/hasanmukit009/CourseManagementSystem/blob/master/CourseManagementSystem/CMS/Controllers/CourseModelsController.cs>

```
152
153
154  References
155  public async Task<IActionResult> FindCourse()
156  {
157      return View();
158  }
159
160
```

48. Run the Project and click Find Course link



49. Add this code to _Layout.cshtml

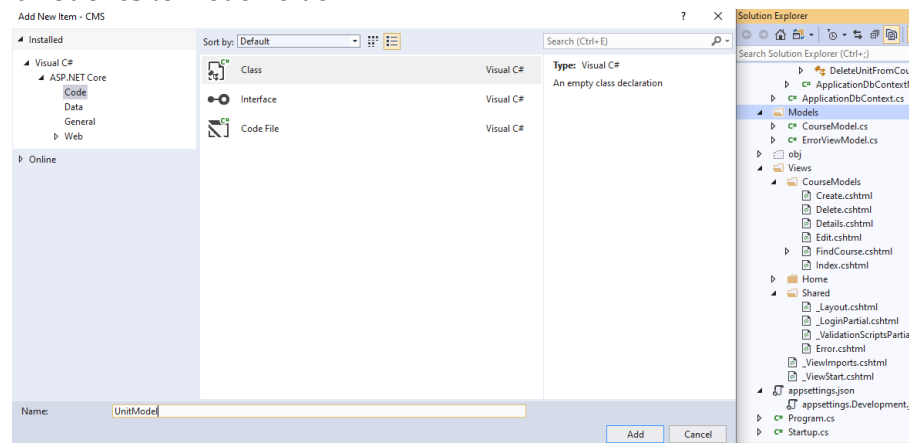
Develop Course Management System Step by Step in .net core

Copy code from this link

(https://github.com/hasanmukit009/CourseManagementSystem/blob/master/CourseManagementSystem/CMS/Views/Shared/_Layout.cshtml)

```
7 link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
8 link rel="stylesheet" href="~/css/site.css" />
9 </head>
10 <body>
11 <div class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
12 <div class="container">
13 <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">CMS</a>
14 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-expanded="false" aria-label="Toggle navigation">
15 <span class="navbar-toggler-icon"></span>
16 </button>
17 <div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse">
18 <partial name="_LoginPartial" />
19 <ul class="navbar-nav flex-grow-1">
20 <li class="nav-item">
21 <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
22 </li>
23 <li class="nav-item">
24 <a class="nav-link text-dark" asp-area="" asp-controller="CourseModels" asp-action="Index">Add Course</a>
25 </li>
26 <li class="nav-item">
27 <a class="nav-link text-dark" asp-area="" asp-controller="Unit" asp-action="Index">Add Unit</a>
28 </li>
29 </ul>
30 </div>
31 </div>
32 </div>
33 </div>
34 </div>
35 <div class="container">
36 <main role="main" class="pb-3">
37 @RenderBody()
38 </main>
39 </div>
40 </body>
```

50. Add UnitModel.cs to Model folder



51. Add these codes to UnitModel.cs

Copy code from this link

(<https://github.com/hasanmukit009/CourseManagementSystem/blob/master/CourseManagementSystem/CMS/Models/UnitModel.cs>)

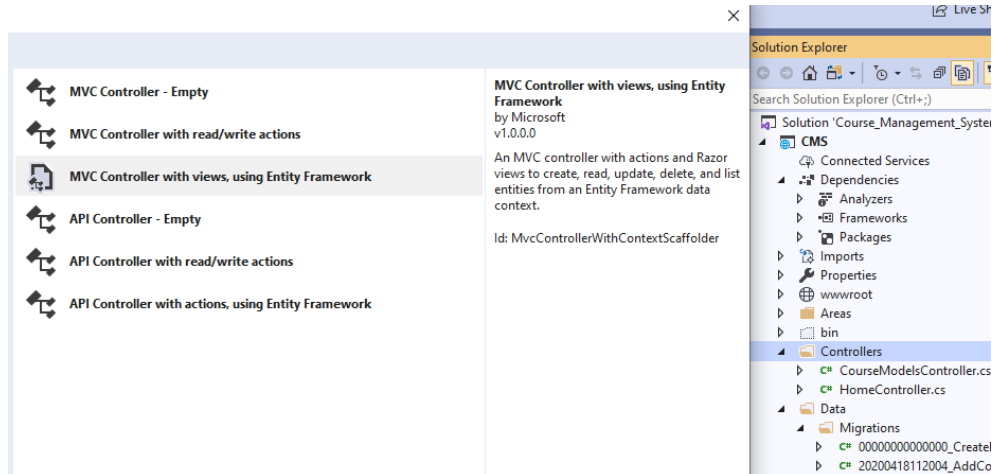
```
public class UnitModel
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    public int ID { get; set; }
    public string UnitName { get; set; }
    public string UnitCode { get; set; }
    public int CourseID { get; set; }
}
```

52. Now Run these commands in Package Manager Console

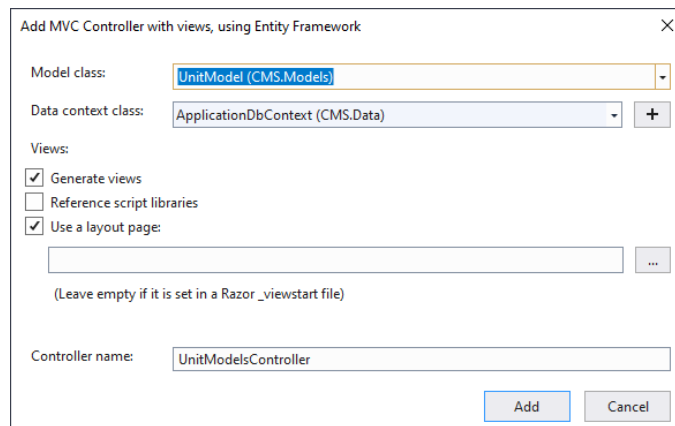
```
PM> Add-Migration AddUnitToDB
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM> Update-Database AddUnitToDB
Build started...
Build succeeded.
Done.
PM> |
```

Develop Course Management System Step by Step in .net core

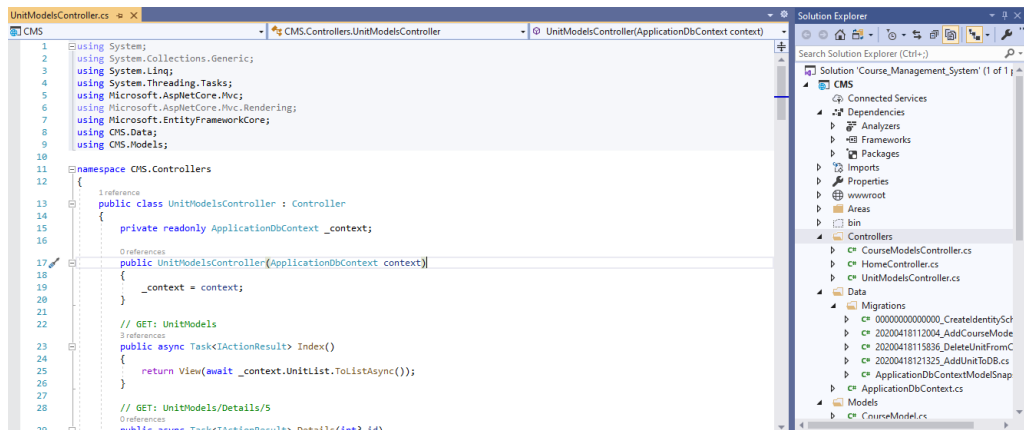
53. Now add **UnitModelsController.cs** to the controller folder



54. After selecting this information click **Add**



55. UnitModelsController has been added to the controller



56. Now run project and click **Add Unit**

CMS Home Add Course Add Unit

Hello hasanmukit009@gmail.com! Find Course Logout

[Create New](#)

CourseName	UnitName	UnitCode
------------	----------	----------

Develop Course Management System Step by Step in .net core

57. Click **Create New** and this page will be opened

CMS Home Add Course Add Unit Hello hasanmukit009@gmail.com! Find Course Logout

Create

UnitModel

CourseName

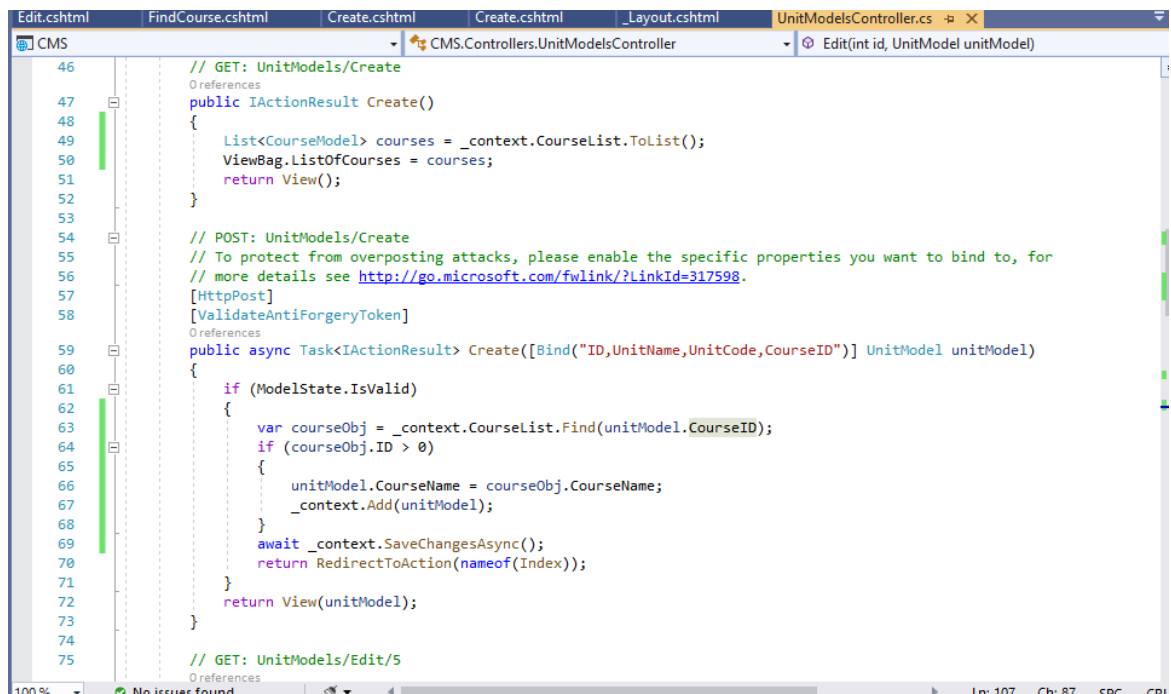
UnitCode

UnitName

[Back to List](#)

58. Now go to the **UnitModelsController.cs** and add these codes which will create Unit
Copy code from this link and paste in **UnitModelsController.cs**

(https://github.com/hasanmukit009/CourseManagementSystem/blob/master/CourseManagement_System/CMS/Controllers/UnitModelsController.cs)



```
46 // GET: UnitModels/Create
47 public IActionResult Create()
48 {
49     List<CourseModel> courses = _context.CourseList.ToList();
50     ViewBag.ListOfCourses = courses;
51     return View();
52 }
53
54 // POST: UnitModels/Create
55 // To protect from overposting attacks, please enable the specific properties you want to bind to, for
56 // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
57 [HttpPost]
58 [ValidateAntiForgeryToken]
59 public async Task<IActionResult> Create([Bind("ID,UnitName,UnitCode,CourseID")] UnitModel unitModel)
60 {
61     if (ModelState.IsValid)
62     {
63         var courseObj = _context.CourseList.Find(unitModel.CourseID);
64         if (courseObj.ID > 0)
65         {
66             unitModel.CourseName = courseObj.CourseName;
67             _context.Add(unitModel);
68         }
69         await _context.SaveChangesAsync();
70         return RedirectToAction(nameof(Index));
71     }
72     return View(unitModel);
73 }
74
75 // GET: UnitModels/Edit/5
```

Run the project then **click Add Unit > Click Create New > Fill up the details and click Create**

Develop Course Management System Step by Step in .net core

[CMS](#) [Home](#) [Add Course](#) [Add Unit](#) Hello hasanmukit009@gmail.com! [Find Course](#) [Logout](#)

Create

CourseName

Master of Data Science (SDASC1 - 2020) ▼

UnitCode

PRT581

UnitName

Principle of Software System

Create

[Back to List](#)

© 2020 - CMS - Privacy

59. Now Click **Create** and new unit will be added.

[CMS](#) [Home](#) [Add Course](#) [Add Unit](#) Hello hasanmukit009@gmail.com! [Find Course](#) [Logout](#)

[Create New](#)

CourseName	UnitName	UnitCode	
Graduate Certificate of Data Science (UDASC1 - 2020)	Principle of Software System	PRT582	Edit Details Delete

60. It also can be edit, details and delete if you check the code in that **UnitModelController.cs** file

61. Now its time for the Find Course Functionality for students

62. So first create a model: **StudentCourseModel.cs**

Copy code from this link

(<https://github.com/hasanmukit009/CourseManagementSystem/blob/master/CourseManagementSystem/CMS/Models/StudentCourseModel.cs>) and paste the code in **StudentCourseModel.cs**

```
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Threading.Tasks;

namespace CMS.Models
{
    [Table("StudentCourse")]
    public class StudentCourseModel
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int ID { get; set; }
        [References]
        public string CourseID { get; set; }
        [References]
        public string StudentID { get; set; }
        [References]
        public string UnitID { get; set; }
        [Display(Name = "Course Name")]
        [References]
        public string CourseName { get; set; }
        [Display(Name = "Unit Code")]
        [References]
        public string UnitCode { get; set; }
        [Display(Name = "Unit Name")]
        [References]
        public string UnitName { get; set; }
        [Display(Name = "StudentEmail")]
        [References]
        public string StudentEmail { get; set; }
    }
}
```

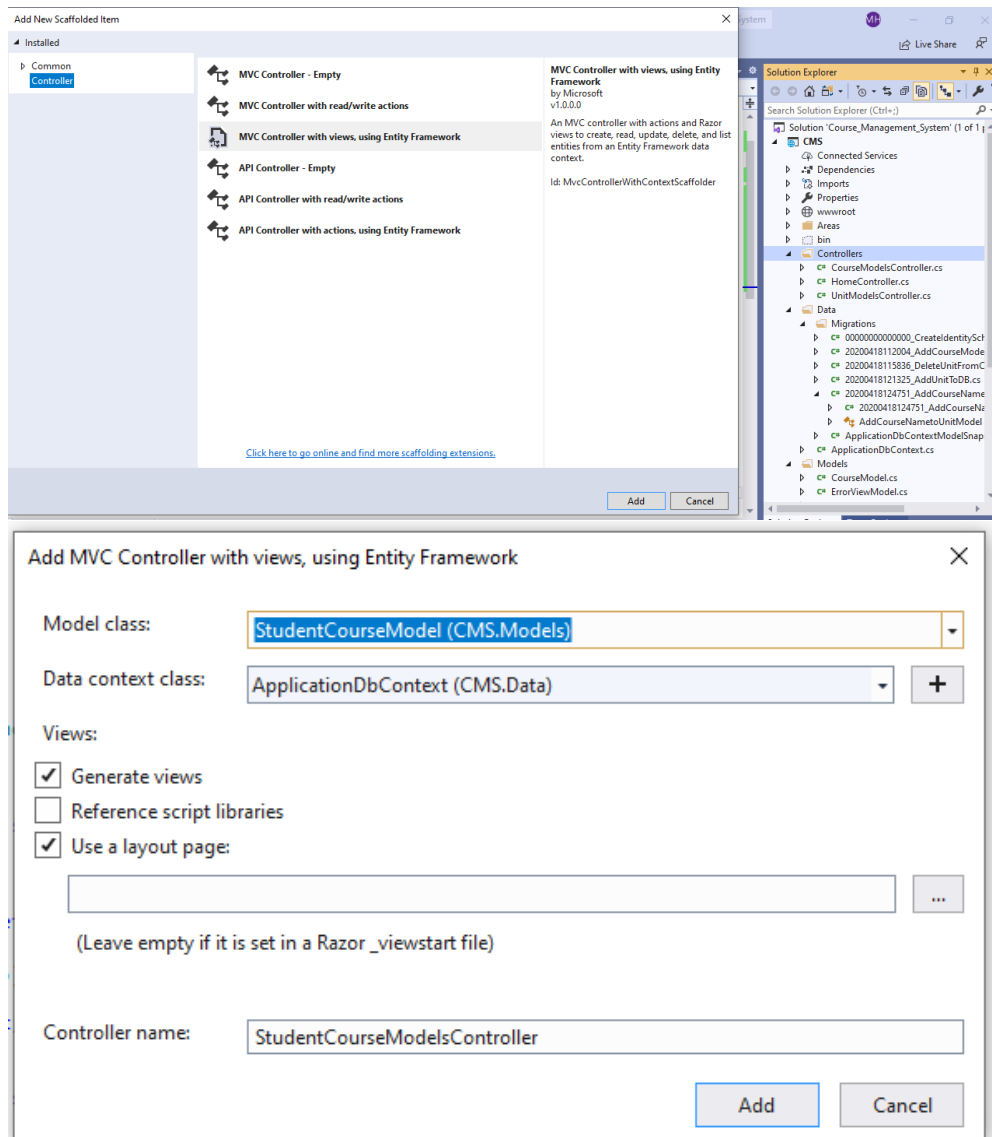
The Solution Explorer on the right shows the project structure with folders for Data, Migrations, Models, Views, and obj, and files for appsettings.json and Program.cs.

63. Now Run these commands in **Package Manager Console**

- **Add-Migration AddStudentCourseModelToDB**
- **Update-Database AddStudentCourseModelToDB**

64. Now add **StudentCourseModelsController.cs** controller in Controller folder

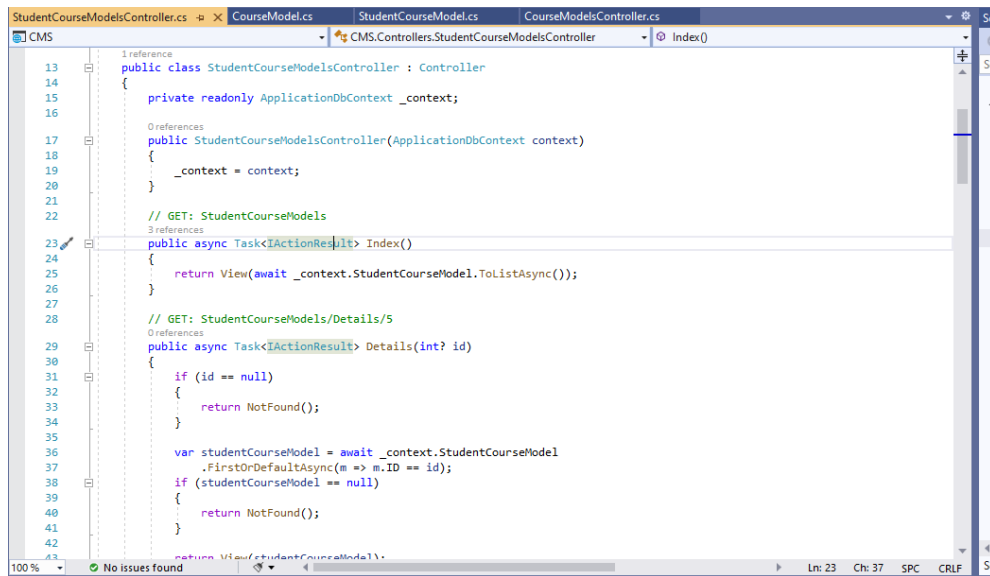
Develop Course Management System Step by Step in .net core



Copy code from this link

(<https://github.com/hasanmukit009/CourseManagementSystem/blob/master/CourseManagementSystem/CMS/Controllers/StudentCourseModelsController.cs>) and paste it to **StudentCourseModelsController.cs**

Develop Course Management System Step by Step in .net core



```
13 public class StudentCourseModelsController : Controller
14 {
15     private readonly ApplicationDbContext _context;
16
17     // GET: StudentCourseModels
18     public StudentCourseModelsController(ApplicationDbContext context)
19     {
20         _context = context;
21     }
22
23     // GET: StudentCourseModels
24     public async Task<IActionResult> Index()
25     {
26         return View(await _context.StudentCourseModel.ToListAsync());
27     }
28
29     // GET: StudentCourseModels/Details/5
30     public async Task<IActionResult> Details(int? id)
31     {
32         if (id == null)
33         {
34             return NotFound();
35         }
36
37         var studentCourseModel = await _context.StudentCourseModel
38             .FirstOrDefaultAsync(m => m.ID == id);
39         if (studentCourseModel == null)
40         {
41             return NotFound();
42         }
43         return View(studentCourseModel);
44     }
45 }
```

65. Now Add code to give functionality for Add, Edit and Delete and Details. This is the page when you click Find Course

CMS Home Hello hasanmukit009@gmail.com! Find Course Logout

Create New

Course Name	Unit Code	Unit Name	Email
Graduate Certificate of Data Science (UDASC1 - 2020)	PRT582	Principle of Software System	Details Delete

66. Run the project > Login > Click **Find Course** > Select Course Name and Unit Name > Click **Create**

CMS Home Hello hasanmukit009@gmail.com! Find Course Logout

Choose your course...

Course Name

Select

Unit Name

Select

Create

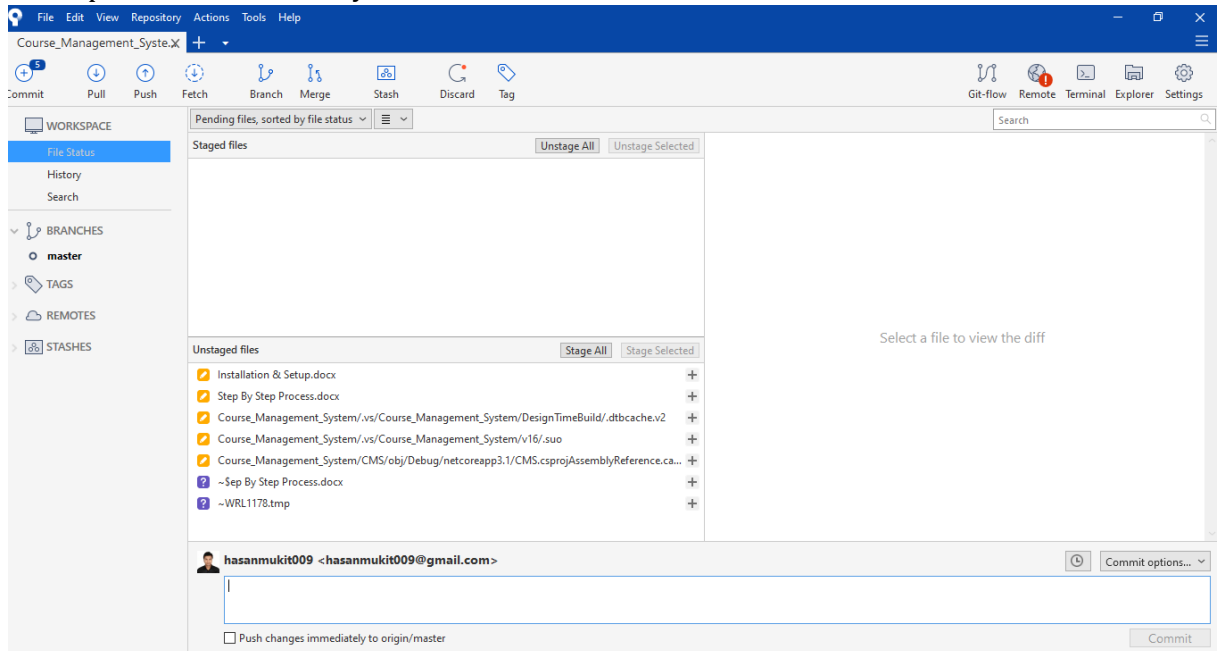
[Back to List](#)

Develop Course Management System Step by Step in .net core

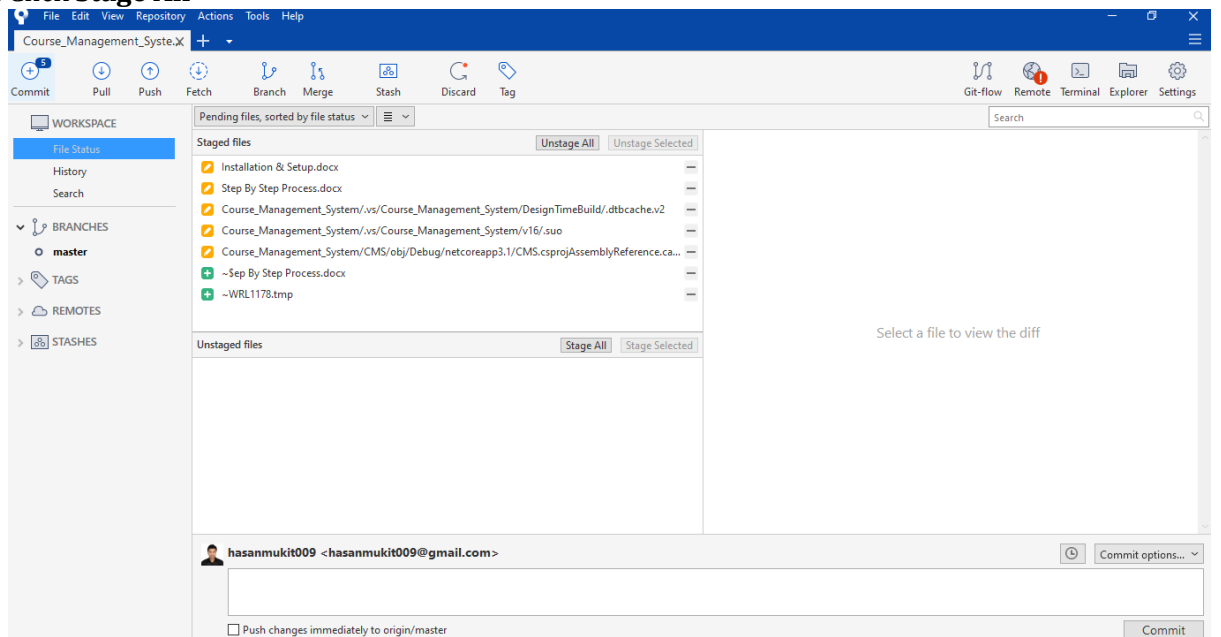
67. New course has been to the user's account.

CMS	Home	Hello hasanmukit009@gmail.com!	Find Course	Logout
Create New				
Course Name	Unit Code	Unit Name	Email	
Graduate Certificate of Data Science (UDASC1 - 2020)	PRT582	Principle of Software System	Details Delete	

68. Now open Source Tree and you will see like this

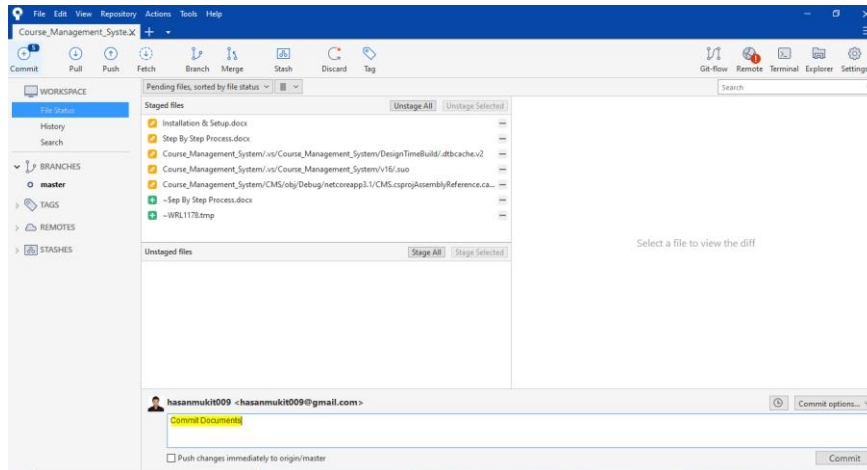


69. Click Stage All

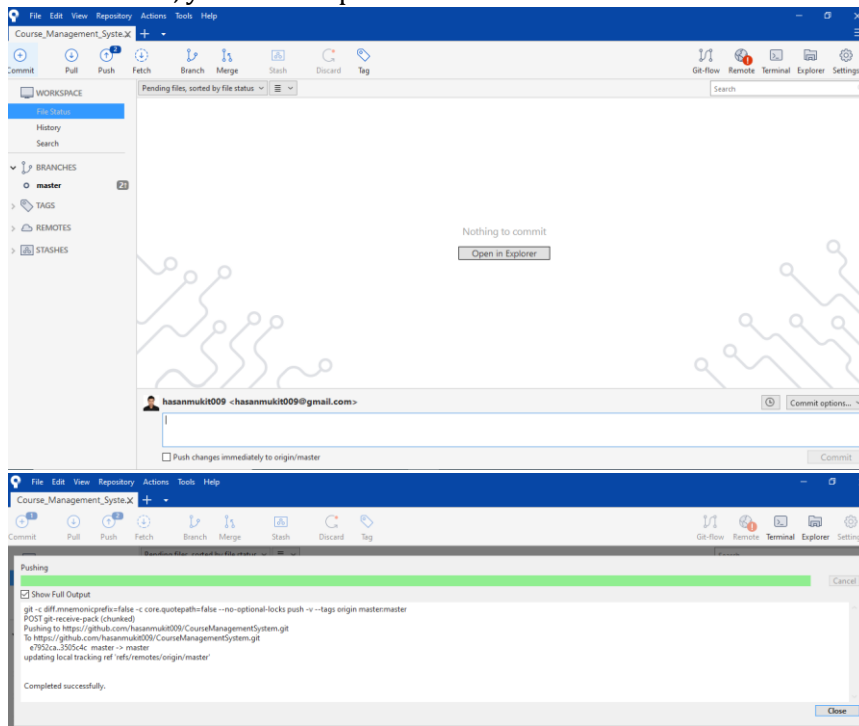


Develop Course Management System Step by Step in .net core

70. Write comment in the comment section and click commit



71. After commit, you need to push and click **Push**



72. After **Push** go to the GitHub repository. And you will see the changes.

