

Develop Course Management System step by step in .net core

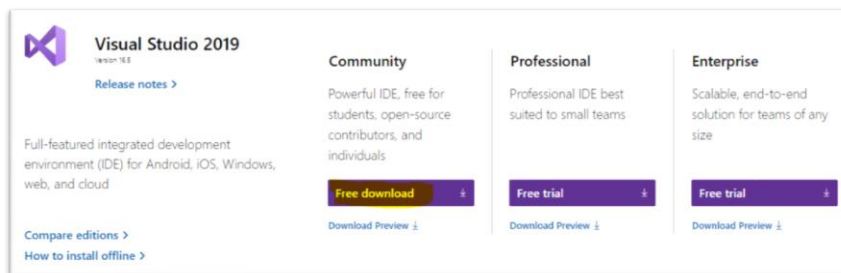
IDE & Tools

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

Download Link:

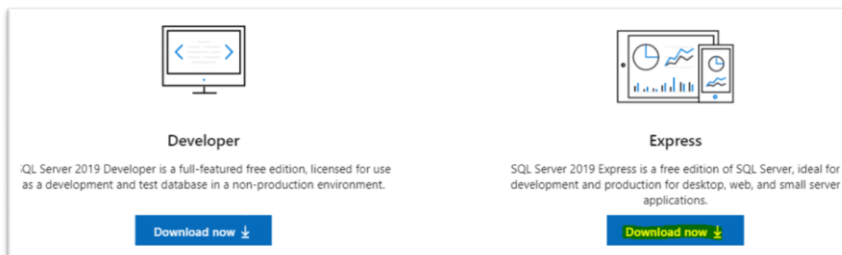
1. Go to this link and download Visual Studio 2019 (Community version) :

<https://visualstudio.microsoft.com/downloads/>



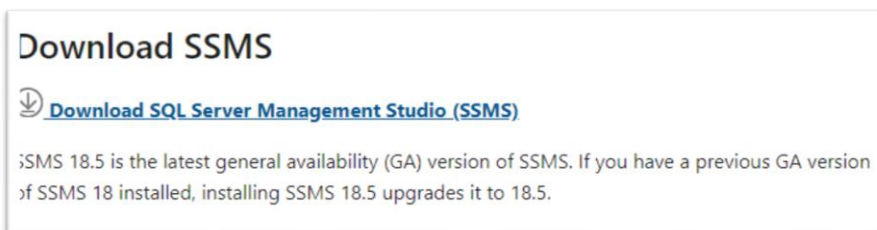
2. Go to this link and download MSSQL Server (Express version)

<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.

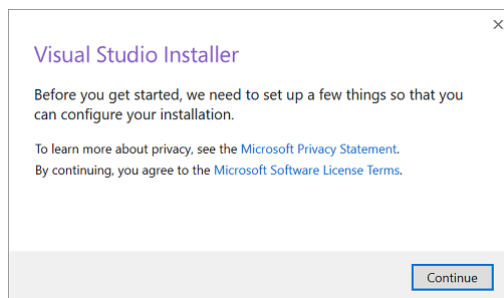
Step 2 - Download Visual Studio

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.

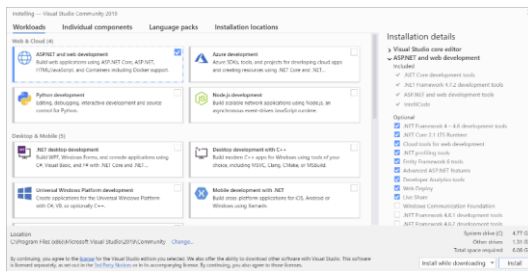


Step 4 - Choose workloads

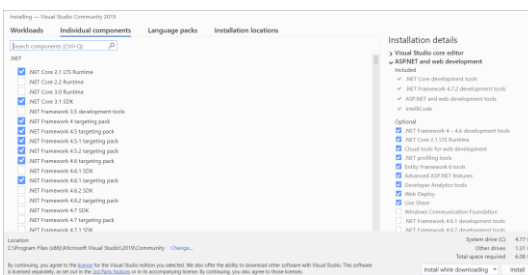
After the installer is installed, you can use it to customize your installation by selecting the feature sets—or workloads—that you want. Here's how.

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

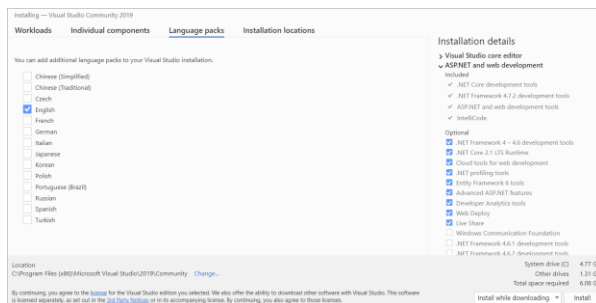
Develop Course Management System step by step in .net core



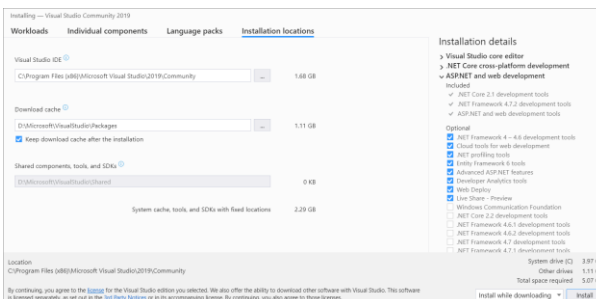
Step 5 - Choose individual components (Optional)



Step 6 - Install language packs (Optional)



Step 7 - Select the installation location (Optional)



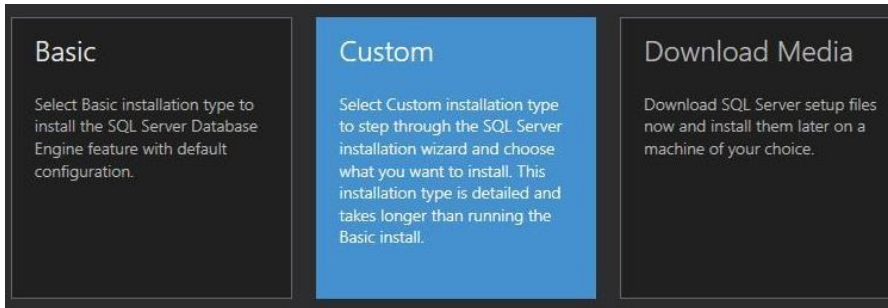
Step 8 - Start developing

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

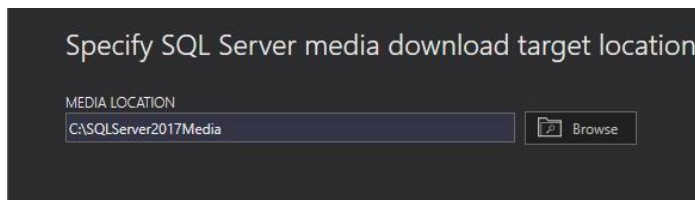
Develop Course Management System step by step in .net core

Install Microsoft SQL Server Express

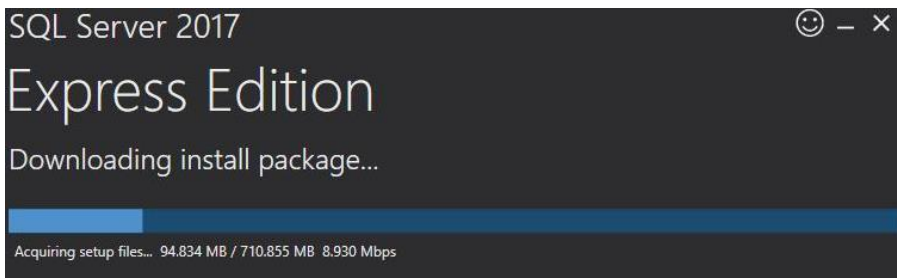
1. Click installation file and select Custom installation, as this allows you to choose what you want to install from the extended tools.



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.

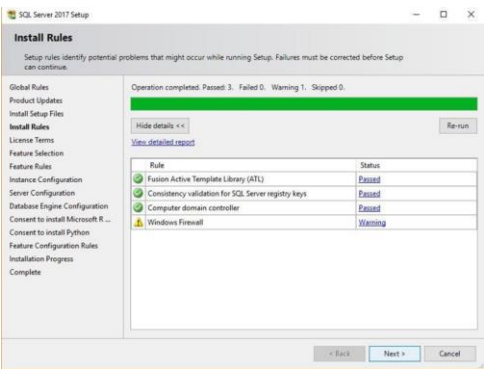


4. After the installation package is downloaded, select New SQL Server stand-alone installation, as shown below

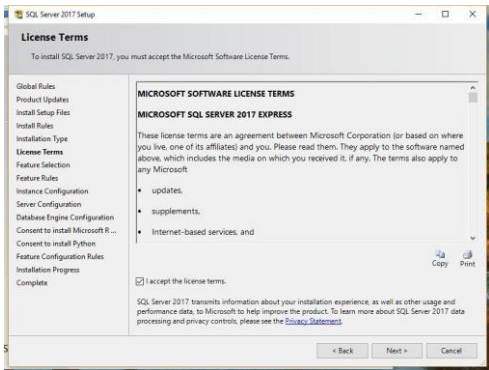


5. On the next screen, you'll see a setup process. Once it completes, click Next.

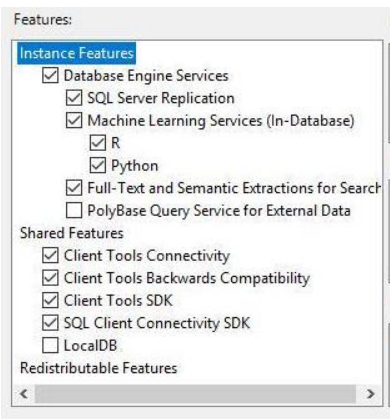
Develop Course Management System step by step in .net core



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.



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

Develop Course Management System step by step in .net core

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 ...
Consent to install Python

☐ Default instance
☒ Named instance:

Instance ID:

SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL14.SQLEXPRESS

Installed instances:

Instance Name	Instance ID	Features	Edition	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\MSSQLSSQL...		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.

Complete

Your SQL Server 2017 installation completed successfully with product updates.

Global Rules
Product Updates
Install Setup Files
Install Rules
License Terms
Feature Selection
Feature Rules
Instance Configuration

Information about the Setup operation or possible next steps:

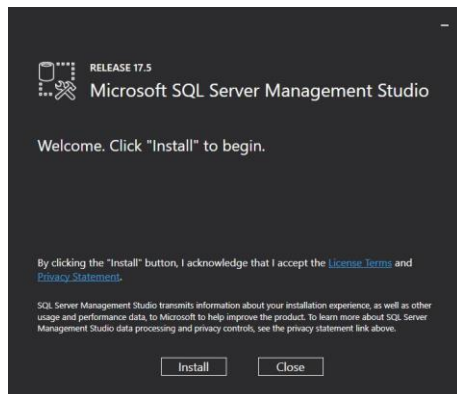
Feature	Status
Full-Text and Semantic Extractions for Search	Succeeded
Python	Succeeded
R	Succeeded
Machine Learning Services (In-Database)	Succeeded
Database Engine Services	Succeeded
SQL Server Replication	Succeeded

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

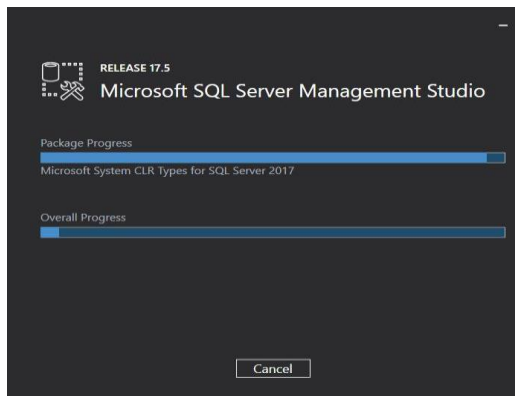
Develop Course Management System step by step in .net core

Install Microsoft SQL Server Management Studio

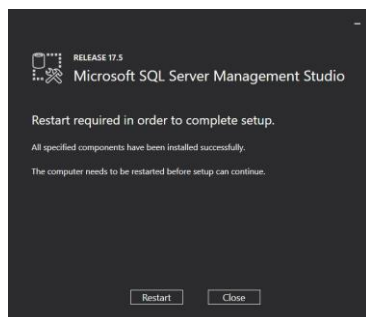
1. Simply run through the steps and click the Install button at the end.



- 2.

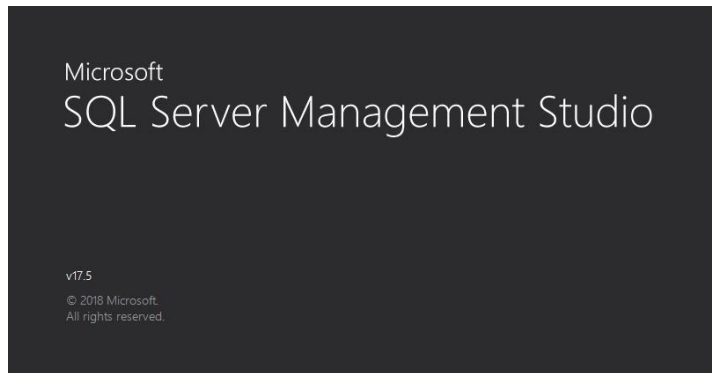


- 3.

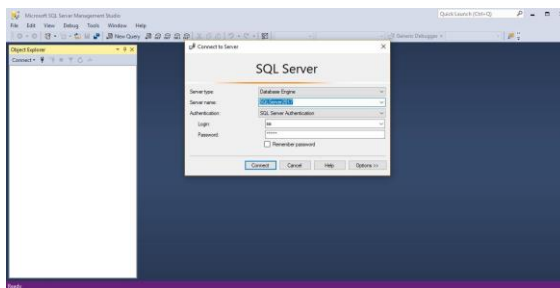


4. After the installation, the SQL Server Management Studio is ready to launch:

Develop Course Management System step by step in .net core



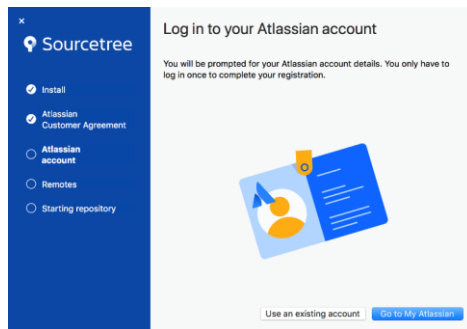
5. First, you'll see the Connect to Server window. Select the name of the server you'd like to connect to and enter the login and password for it. Click the Connect button.



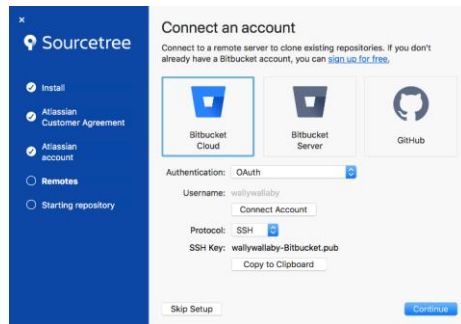
Develop Course Management System step by step in .net core

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**.

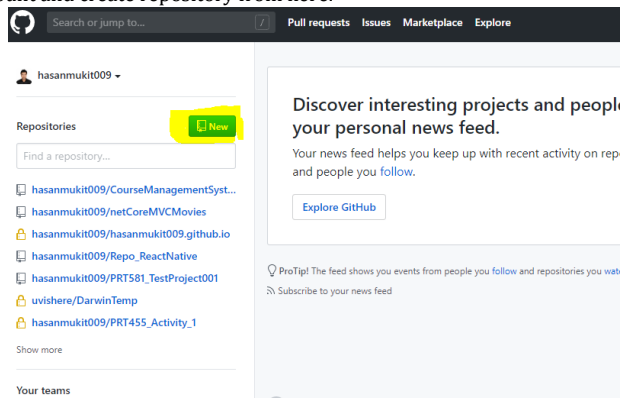
Develop Course Management System step by step in .net core

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: Repository name:

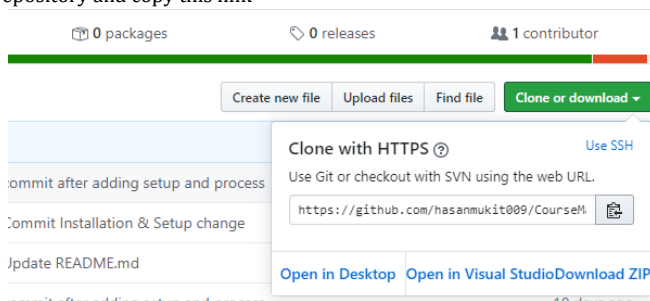
Great repository names are short and memorable. Need inspiration? How about [glowing-octo-fortnight](#)?

Description (optional):

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

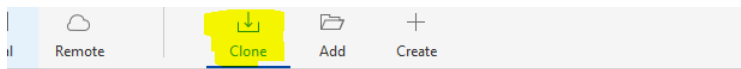
☐ Private

4. Now go to the repository and copy this link



5. Open source tree and click clone

Develop Course Management System step by step in .net core



Clone

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

Source Path / URL:

Repository Type: No path / URL supplied

Destination Path:

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:

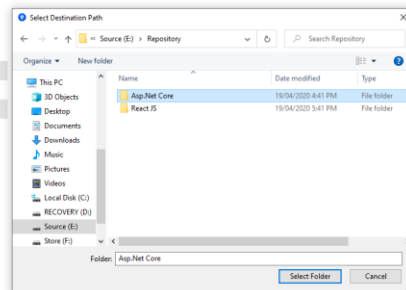
Repository Type: This is a Git repository

Destination Path:

Name:

Local Folder:

[Advanced Options](#)



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

Source (E:) > Repository > Asp.Net Core > Course Management System

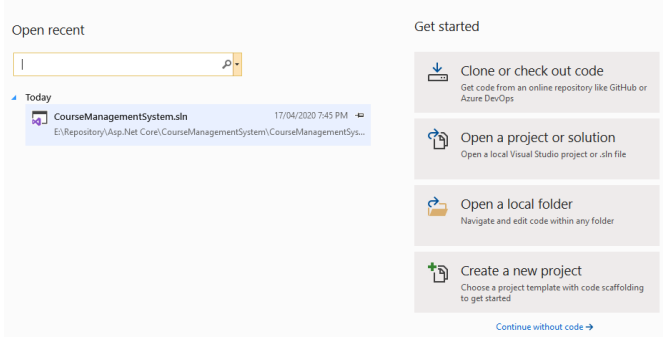
Name	Date modified	Type	Size
.git	19/04/2020 4:54 PM	File folder	
Course_Management_System	19/04/2020 4:40 PM	File folder	

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

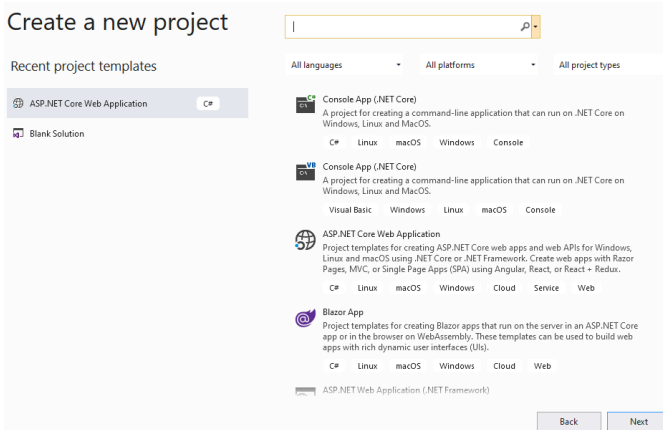
10. Open Visual Studio 2019 and click Create a new project

Visual Studio 2019

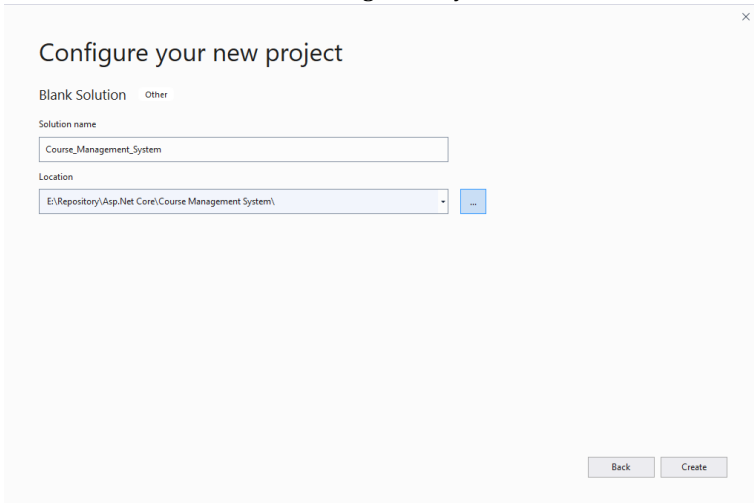


11. Select Blank Solution and click next

Create a new project

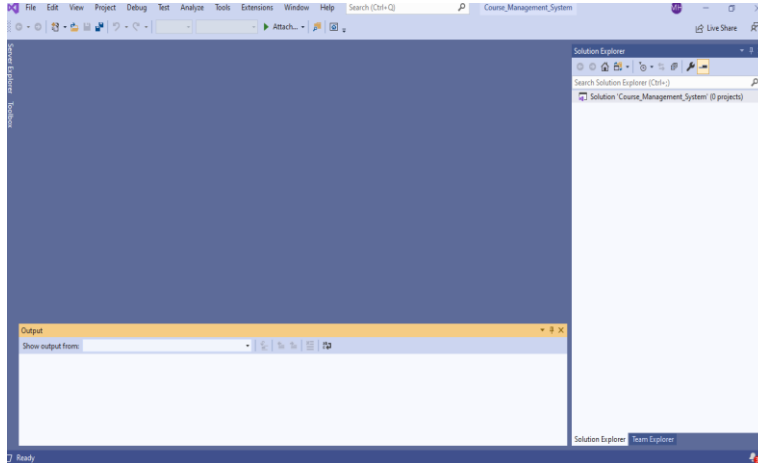


12. Write the solution name "Course_Management_System" and click Create.

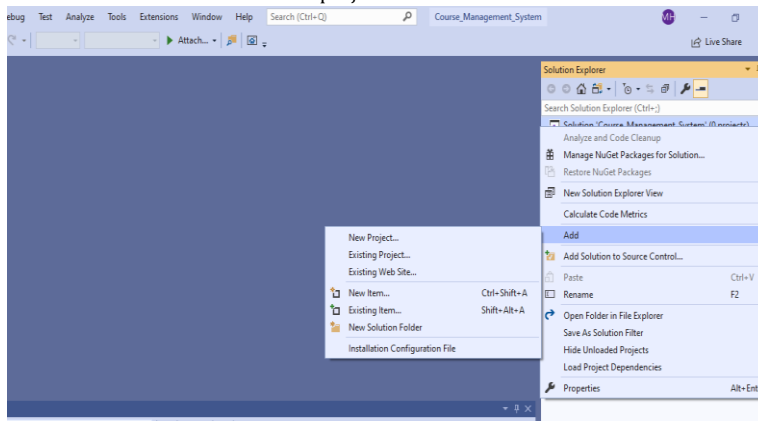


Develop Course Management System step by step in .net core

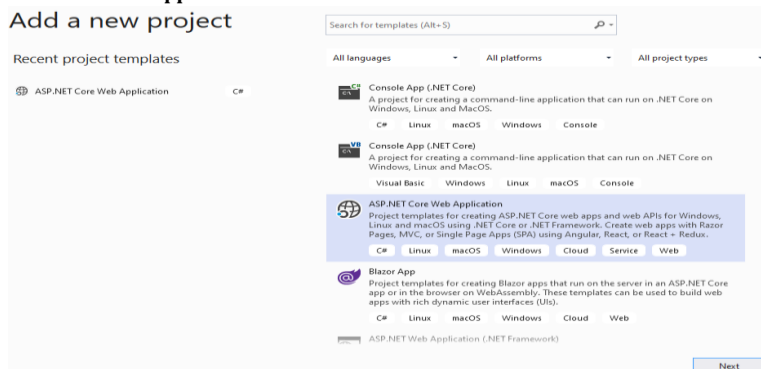
13. Now solution will be like this.



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

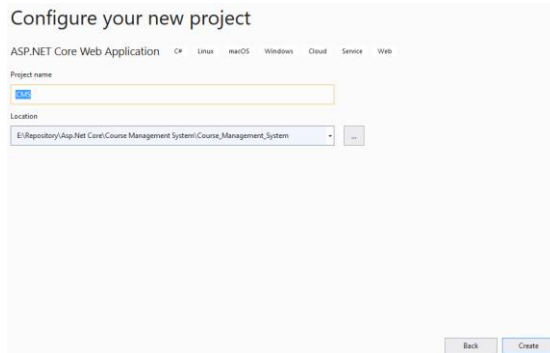


15. Select ASP.Net core web application and click next.



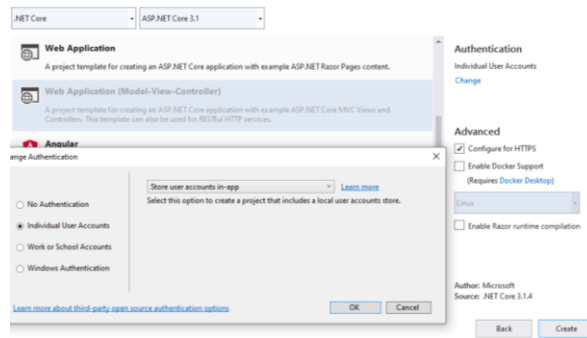
Develop Course Management System step by step in .net core

16. Give the Project Name and click create.

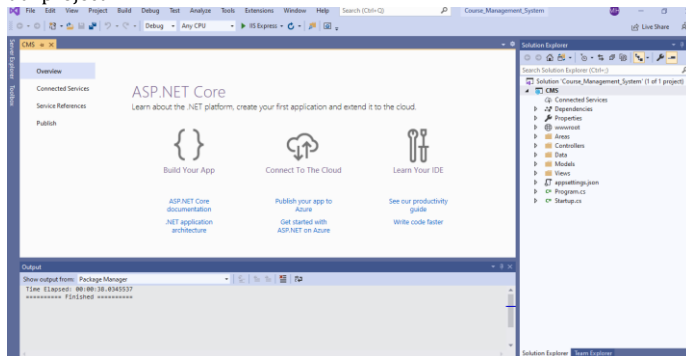


17. After creating Project select these things and click create

Create a new ASP.NET Core web application



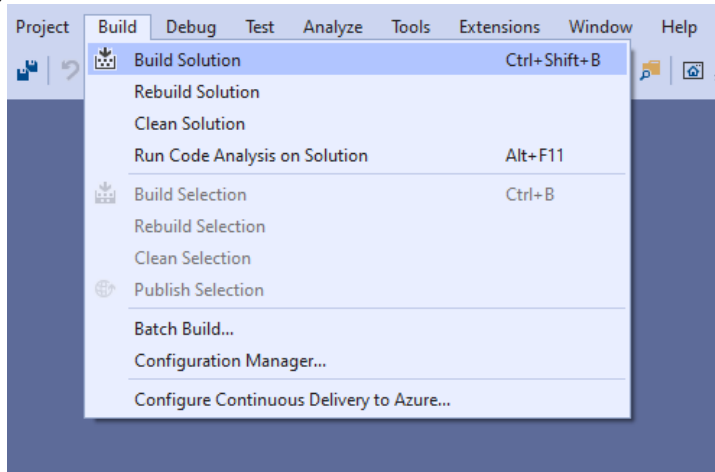
18. So, this the initial project



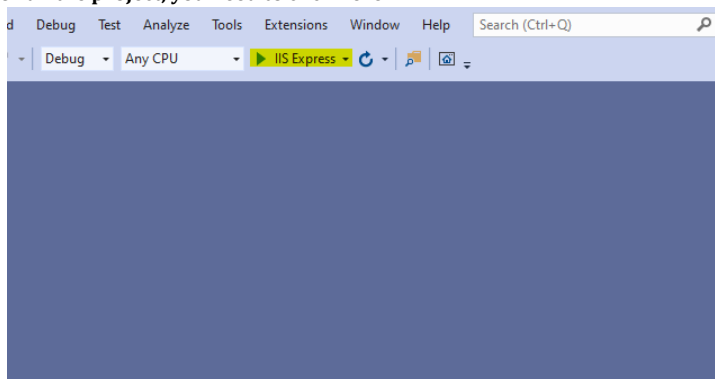
Develop Course Management System step by step in .net core

19. Build the project and Click run

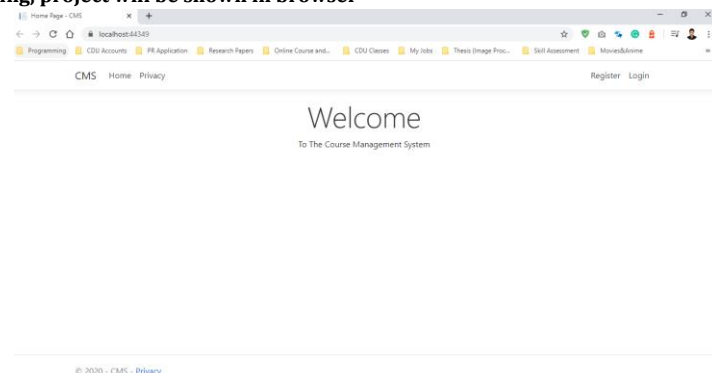
- To build a project you need to right click on the solution and click build or rebuild Or you can click here



- To run the project, you need to click here



After running, project will be shown in browser

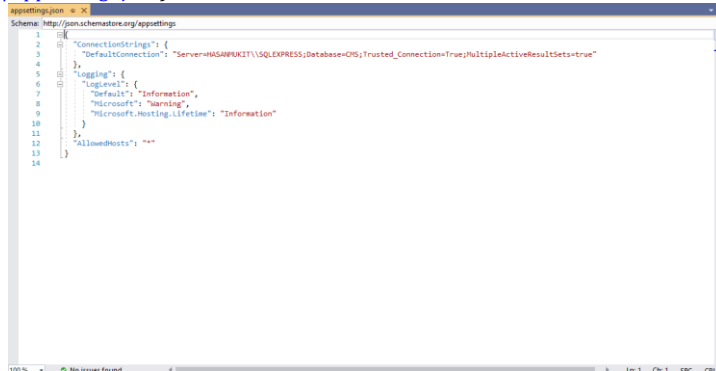


Develop Course Management System step by step in .net core

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

Copy the code from this link

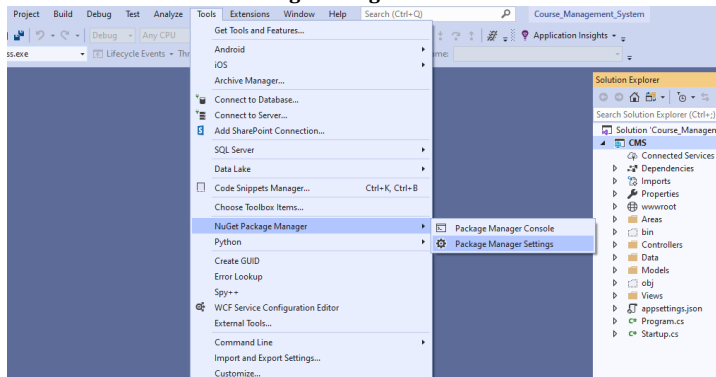
([https://github.com/hasanmukit009/CourseManagementSystem/blob/master/Course Management System/CMS/appsettings.json](https://github.com/hasanmukit009/CourseManagementSystem/blob/master/Course%20Management%20System/CMS/appsettings.json))



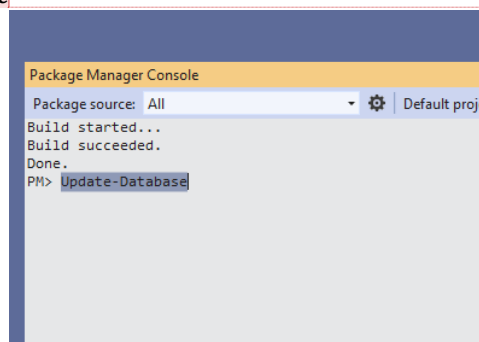
```
1 {
2   "ConnectionStrings": {
3     "DefaultConnection": "Server=HASANMUKIT\\SQLSERVER;Database=CMS;Trusted_Connection=True;MultipleActiveResultSets=true"
4   },
5   "Logging": {
6     "LogLevel": {
7       "Default": "Information",
8       "Microsoft": "Warning",
9       "Microsoft.Hosting.Lifetime": "Information"
10    }
11  },
12  "AllowedHosts": "*"
13 }
14 }
```

Commented [MH1]: You can connect your C# application to data in a SQL Server database using the connection string

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



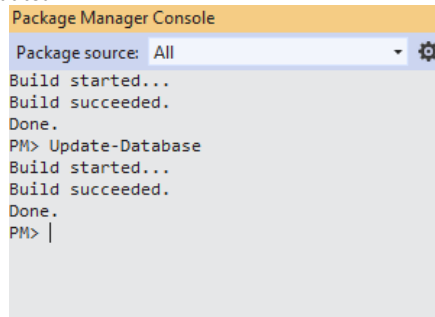
22. Write **"update-database"**



Commented [MH2]: This is a migration command. It will create database.

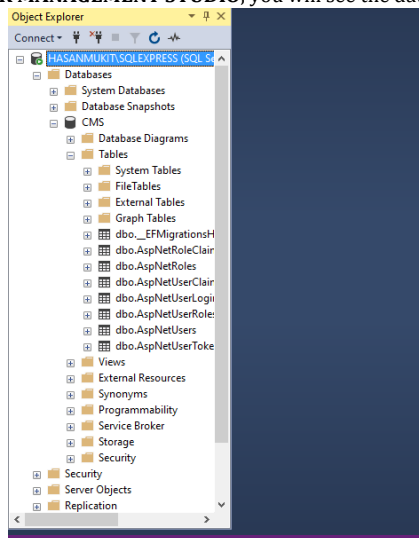
Develop Course Management System step by step in .net core

23. Then database will be updated

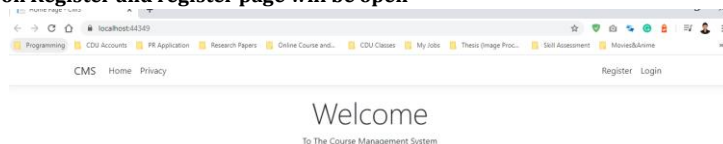


```
Package Manager Console
Package source: All
Build started...
Build succeeded.
Done.
PM> Update-Database
Build started...
Build succeeded.
Done.
PM> |
```

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



Develop Course Management System step by step in .net core

26. Register page will be like this

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

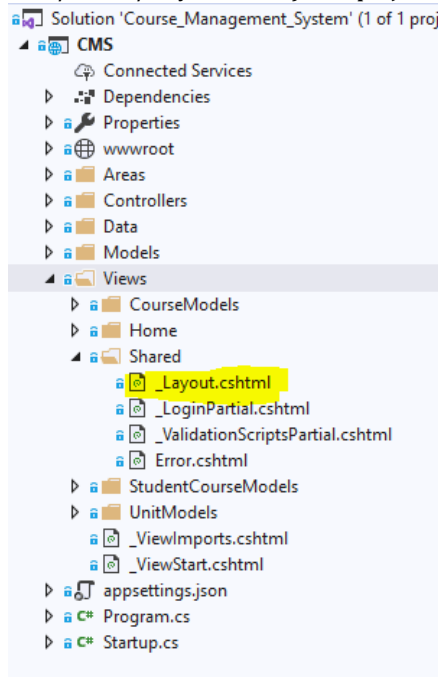
28. Click log out and then click login

Develop Course Management System step by step in .net core

29. This page will open



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



Copy the code from this link

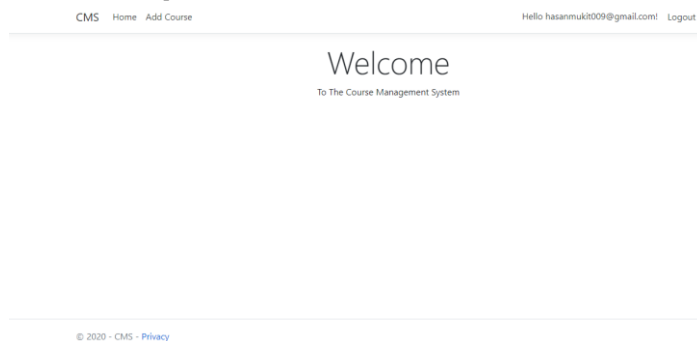
(https://github.com/hasanmukit009/CourseManagementSystem/blob/master/Course_Management_System/CMS/Views/Shared/_Layout.cshtml)

Add this code and run

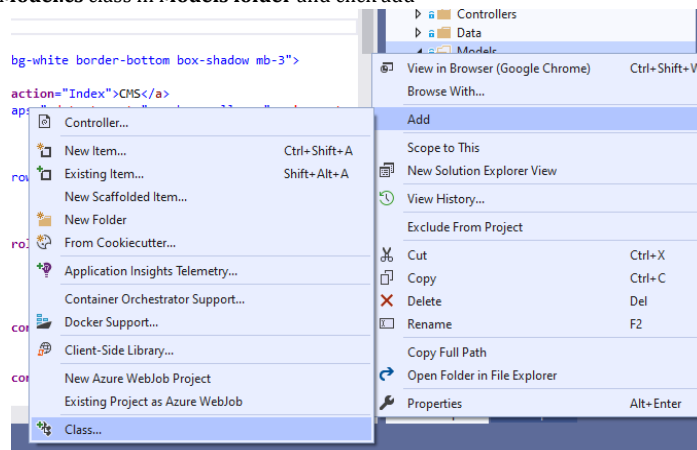
Develop Course Management System step by step in .net core

```
<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-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">
    <div class="main" role="main" class="pb-3">
      @RenderBody()
    </div>
  </div>
</body>
</html>
```

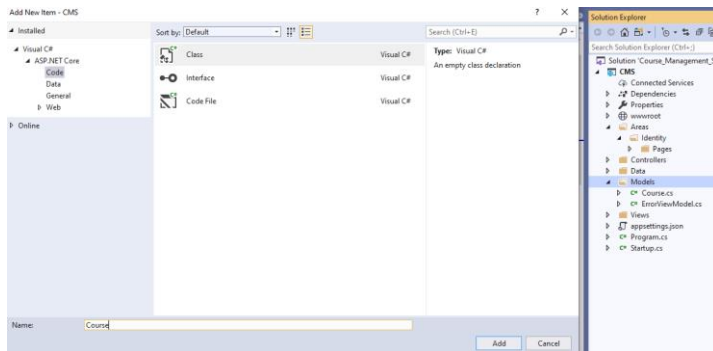
31. Now click run and this will open



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/Course Management System/CMS/Models/CourseModel.cs](https://github.com/hasanmukit009/CourseManagementSystem/blob/master/Course%20Management%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.
```

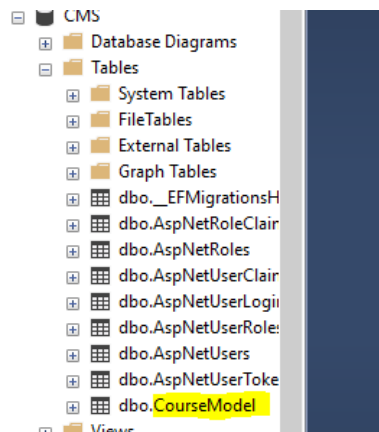
35. Now execute **"Update-Database"** command in Package Manager Console

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

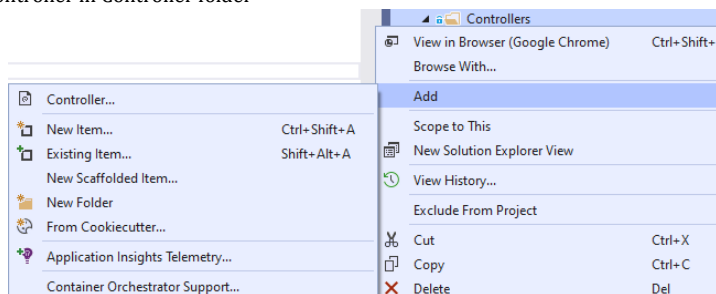
100 %
```

Develop Course Management System step by step in .net core

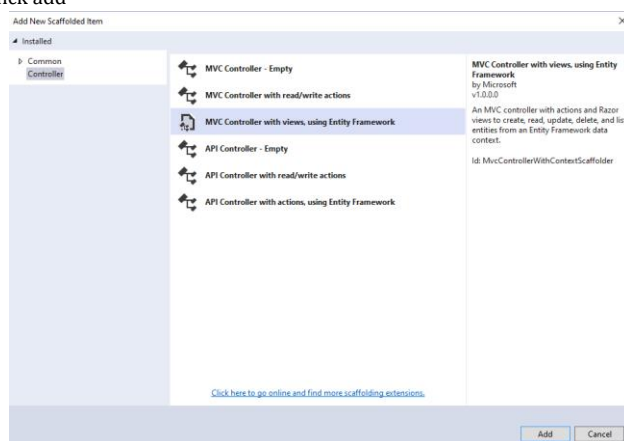
36. Now open **SQL SERVER MANAGEMENT STUDIO** and you will see the new table “**CourseModel**” has been added.



37. Now Add controller in Controller folder

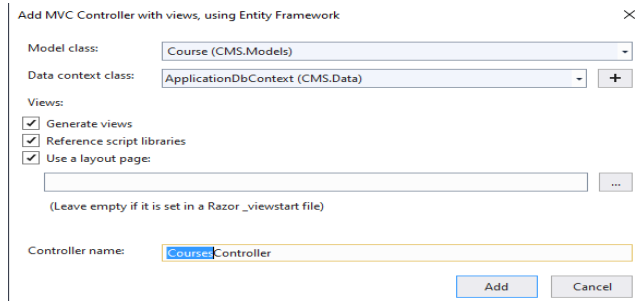


Select this and click add

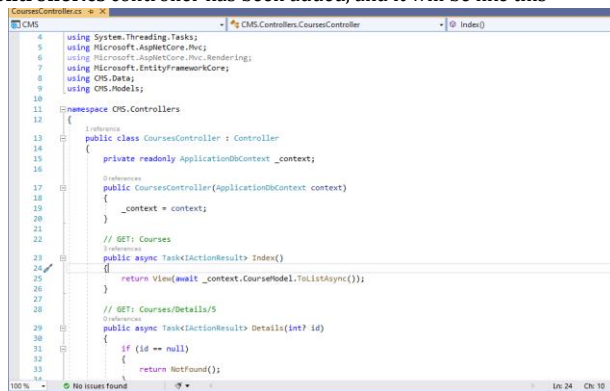


Develop Course Management System step by step in .net core

38. Give all this information and click Add



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



```
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using CMS.Models;
using CMS.Data;

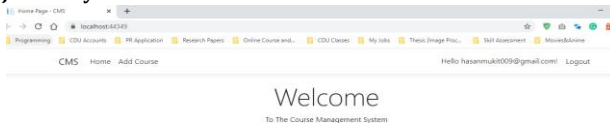
namespace CMS.Controllers
{
    public class CoursesController : Controller
    {
        private readonly ApplicationDbContext _context;

        public CoursesController(ApplicationDbContext context)
        {
            _context = context;
        }

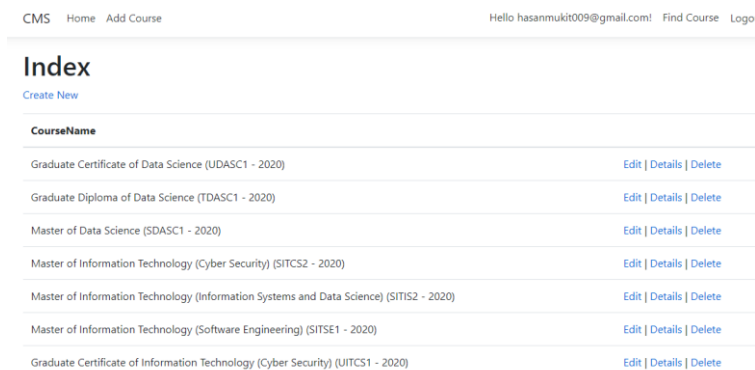
        // GET: Courses
        public async Task Index()
        {
            return View(await _context.CourseModel.ToListAsync());
        }

        // GET: Courses/Details/5
        public async Task Details(int? id)
        {
            if (id == null)
            {
                return NotFound();
            }
        }
    }
}
```

40. Now run the project and you will see this



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

Develop Course Management System step by step in .net core

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

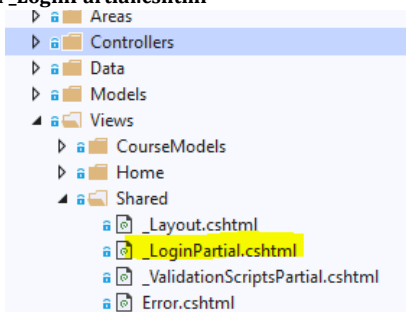
[Back to List](#)

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/Course_Management_System/CMS/Views/Shared/_LoginPartial.cshtml)

```
1 using Microsoft.AspNetCore.Identity
2 [inject] SignInManager<IdentityUser> SignInManager
3 [inject] UserManager<IdentityUser> UserManager
4
5 <ul class="navbar-nav">
6     <li class="nav-item">
7         <a class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Manage/Index" title="Manage">Hello @User.Identity.Name</a>
8     </li>
9     <li class="nav-item">
10        <a class="nav-link text-dark" asp-area="" asp-controller="Courses" asp-action="FindCourse">Find Course</a>
11    </li>
12    <li class="nav-item">
13        <form class="form-inline" asp-area="Identity" asp-page="/Account/Logout" asp-route-returnurl="@Url.Action("Index", "ValidationScriptsPartial")">
14            <button type="submit" class="nav-link btn btn-link text-dark">Logout</button>
15        </form>
16    </li>
17    <li class="nav-item">
18        <a class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Register">Register</a>
19    </li>
20    <li class="nav-item">
21        <a class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Login">Login</a>
22    </li>
23 </ul>
```

46. Run the project and it will be like this

Develop Course Management System step by step in .net core



47. Add this code in **CourseModelsController.cs**

Copy code from this link

([https://github.com/hasanmukt009/CourseManagementSystem/blob/master/Course Management System/CMS/Controllers/CourseModelsController.cs](https://github.com/hasanmukt009/CourseManagementSystem/blob/master/Course%20Management%20System/CMS/Controllers/CourseModelsController.cs))

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

48. Run the Project and click **Find Course** link



49. Add this code to **_Layout.cshtml**

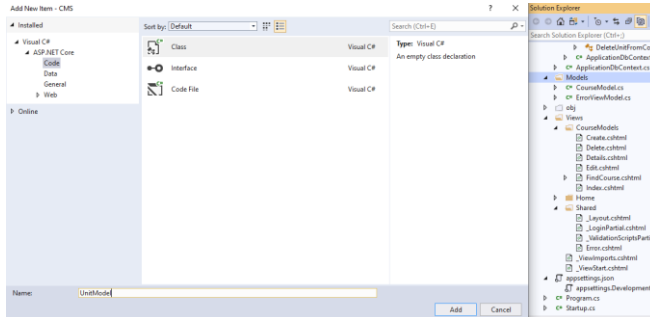
Copy code from this link

([https://github.com/hasanmukt009/CourseManagementSystem/blob/master/Course Management System/CMS/Views/Shared/_Layout.cshtml](https://github.com/hasanmukt009/CourseManagementSystem/blob/master/Course%20Management%20System/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
10
11 <header>
12     <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
13         <div class="container">
14             <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">CMS</a>
15             <button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-expanded="false" aria-label="Toggle navigation">
16                 <span class="navbar-toggler-icon"></span>
17             </button>
18             <div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse">
19                 <partial name="_LoginPartial" />
20                 <ul class="navbar-nav flex-grow-1">
21                     <li class="nav-item">
22                         <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
23                     </li>
24                     <li class="nav-item">
25                         <a class="nav-link text-dark" asp-area="" asp-controller="CourseModels" asp-action="Index">Add Course</a>
26                     </li>
27                     <li class="nav-item">
28                         <a class="nav-link text-dark" asp-area="" asp-controller="Unit" asp-action="Index">Add Unit</a>
29                     </li>
30                 </ul>
31             </div>
32         </div>
33     </nav>
34 </header>
35 <div class="container">
36     <main role="main" class="pb-3">
37         @RenderBody()
38     </main>
39 </div>
```

Develop Course Management System step by step in .net core

50. Add UnitModel.cs to Model folder



51. Add these codes to UnitModel.cs

Copy code from this link

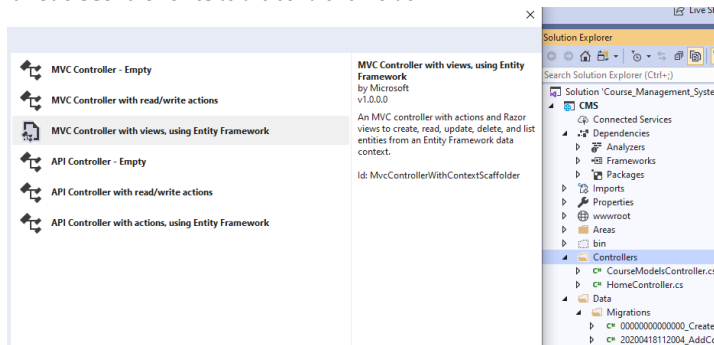
([https://github.com/hasanmukit009/CourseManagementSystem/blob/master/Course Management System/CMS/Models/UnitModel.cs](https://github.com/hasanmukit009/CourseManagementSystem/blob/master/Course%20Management%20System/CMS/Models/UnitModel.cs))

```
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

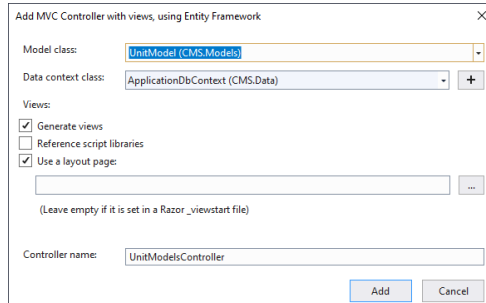
```
PM> Add-Migration AddUnitToDB
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM> Update-Database AddUnitToDB
Build started...
Build succeeded.
Done.
PM> |
```

53. Now add UnitModelsController.cs to the controller folder

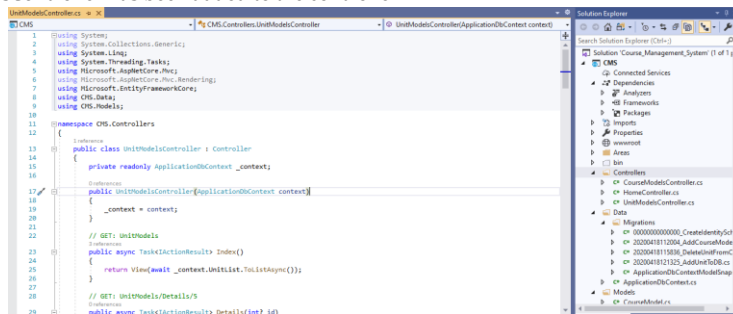


Develop Course Management System step by step in .net core

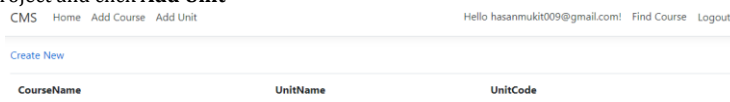
54. After selecting this information click **Add**



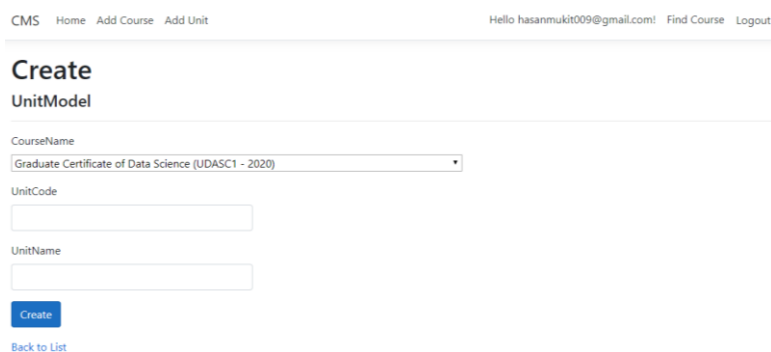
55. UnitModelsController has been added to the controller



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



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



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/Course Management System/CMS/Controllers/UnitModelsController.cs](https://github.com/hasanmukit009/CourseManagementSystem/blob/master/Course%20Management%20System/CMS/Controllers/UnitModelsController.cs))

Develop Course Management System step by step in .net core

```
46 // GET: UnitModels/Create
47 // To protect from overposting attacks, please enable the specific properties you want to bind to, for
48 // more details see http://go.microsoft.com/fwlink/?linkid=317598.
49 public IActionResult Create()
50 {
51     List<CourseModel> courses = _context.CourseList.ToList();
52     ViewBag.ListOfCourses = courses;
53     return View();
54 }
55
56 // POST: UnitModels/Create
57 // To protect from overposting attacks, please enable the specific properties you want to bind to, for
58 // more details see http://go.microsoft.com/fwlink/?linkid=317598.
59 [HttpPost]
60 [ValidateAntiForgeryToken]
61 public async Task<IActionResult> Create([Bind("ID,UnitName,UnitCode,CourseID")] UnitModel unitModel)
62 {
63     if (ModelState.IsValid)
64     {
65         var courseObj = _context.CourseList.Find(unitModel.CourseID);
66         if (courseObj.ID > 0)
67         {
68             unitModel.CourseName = courseObj.CourseName;
69             _context.Add(unitModel);
70         }
71         await _context.SaveChangesAsync();
72         return RedirectToAction(nameof(Index));
73     }
74     return View(unitModel);
75 }
76
77 // GET: UnitModels/Edit/5
78 // To protect from overposting attacks, please enable the specific properties you want to bind to, for
79 // more details see http://go.microsoft.com/fwlink/?linkid=317598.
80 public IActionResult Edit(int id, UnitModel unitModel)
81 {
82     if (id != unitModel.ID)
83     {
84         return NotFound();
85     }
86     if (ModelState.IsValid)
87     {
88         try
89         {
90             _context.Update(unitModel);
91             await _context.SaveChangesAsync();
92         }
93         catch (DbUpdateConcurrencyException)
94         {
95             if (!UnitModelExists(id))
96             {
97                 return NotFound();
98             }
99             else
100             {
91                 return RedirectToAction(nameof(Index));
92             }
93         }
94     }
95     return View(unitModel);
96 }
97
98 private bool UnitModelExists(int id)
99 {
100     return _context.UnitModels.Any(e => e.ID == id);
101 }
```

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

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

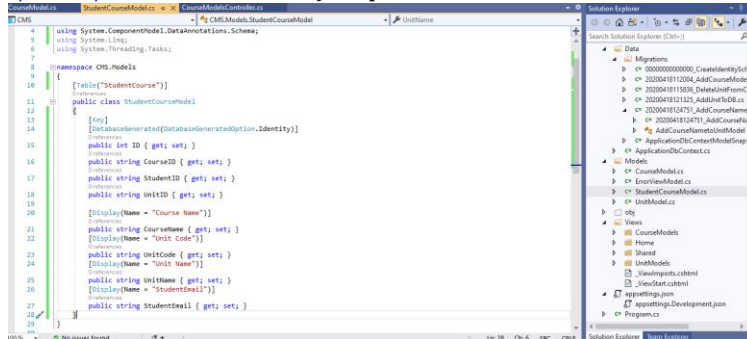
61. Now its time for the Find Course Functionality for students

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

Develop Course Management System step by step in .net core

Copy code from this link

([https://github.com/hasanmukit009/CourseManagementSystem/blob/master/Course Management System/CMS/Models/StudentCourseModel.cs](https://github.com/hasanmukit009/CourseManagementSystem/blob/master/Course%20Management%20System/CMS/Models/StudentCourseModel.cs)) and paste the code in **StudentCourseModel.cs**



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

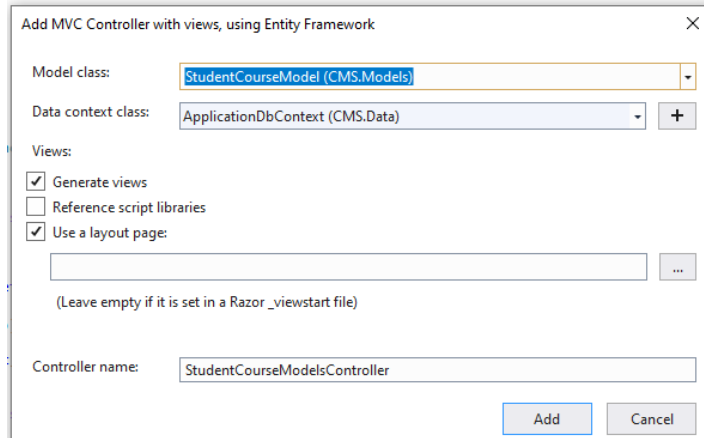
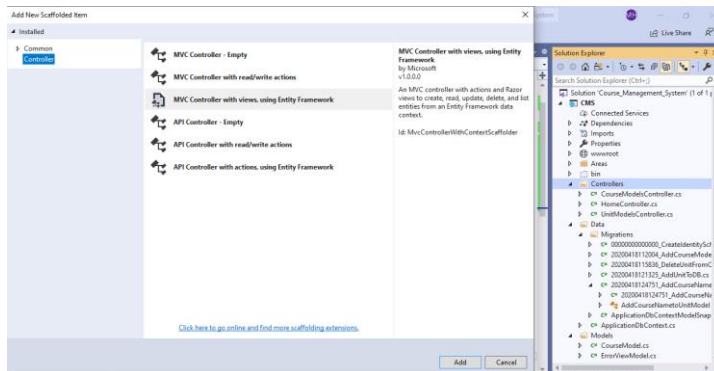
namespace CMS.Models
{
    [Table("StudentCourse")]
    public class StudentCourseModel
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int ID { get; set; }
        public string CourseID { get; set; }
        public string StudentID { get; set; }
        public string UnitCode { get; set; }
        public string UnitName { get; set; }
        public string StudentEmail { get; set; }

        [Display(Name = "Course Name")]
        public string CourseName { get; set; }
        [Display(Name = "Unit Code")]
        public string UnitCode { get; set; }
        [Display(Name = "Unit Name")]
        public string UnitName { get; set; }
        [Display(Name = "Student Email")]
        public string StudentEmail { get; set; }
    }
}
```

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

- Add-Migration AddStudentCourseModelToDB
- Update-Database AddStudentCourseModelToDB

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



Add MVC Controller with views, using Entity Framework

Model class: **StudentCourseModel (CMS.Models)**

Data context class: **ApplicationDbContext (CMS.Data)**

Views:

☒ Generate views

☐ Reference script libraries

☒ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

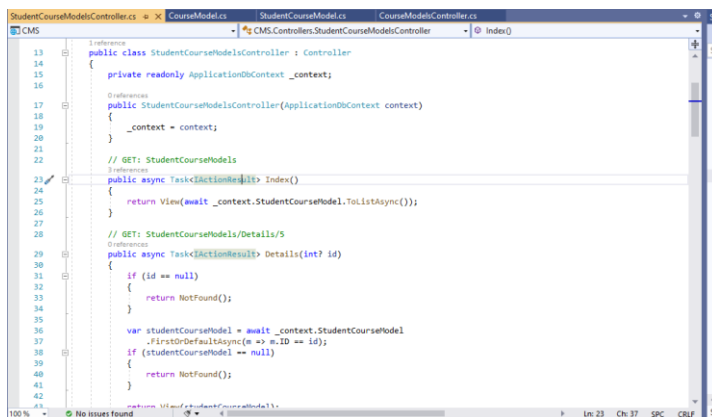
Controller name: **StudentCourseModelsController**

Add **Cancel**

Develop Course Management System step by step in .net core

Copy code from this link

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



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

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!

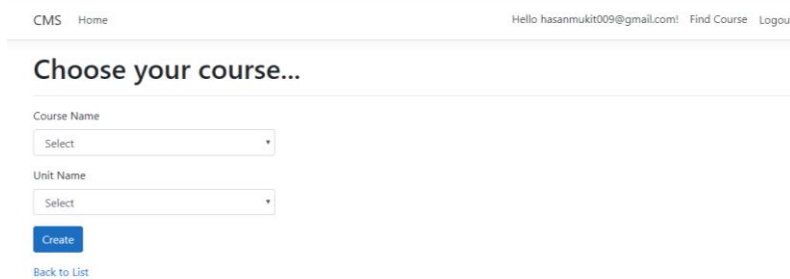
Find Course

Logout

Create New

Course Name	Unit Code	Unit Name	Email
Graduate Certificate of Data Science (UDASC1 - 2020)	PRTS82	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.

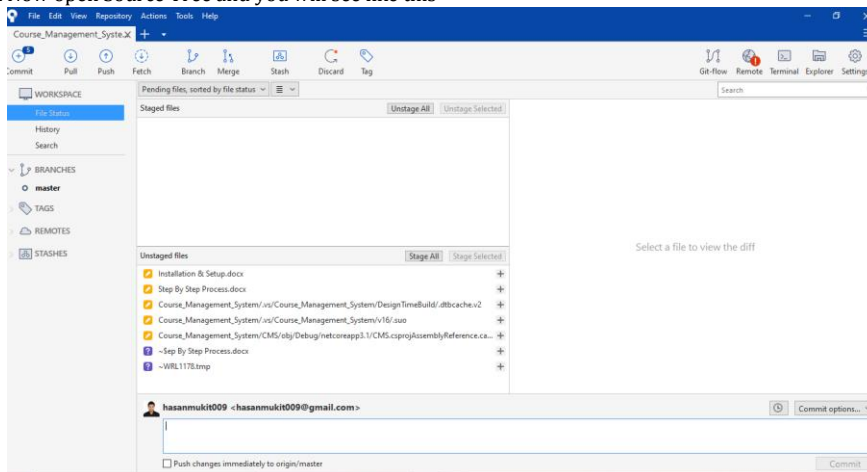
CMS Home

Hello hasanmukit009@gmail.com! Find Course Logout

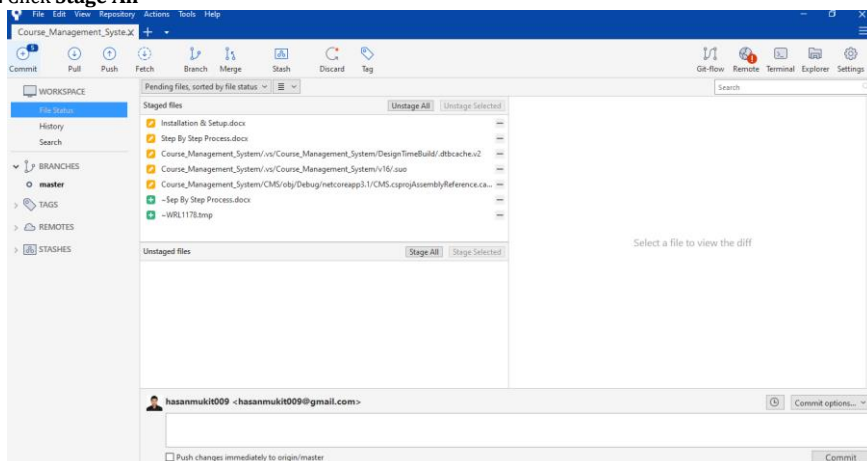
Create New

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

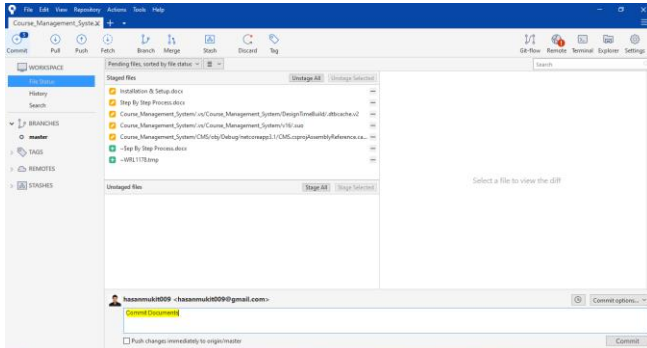


69. Click Stage All

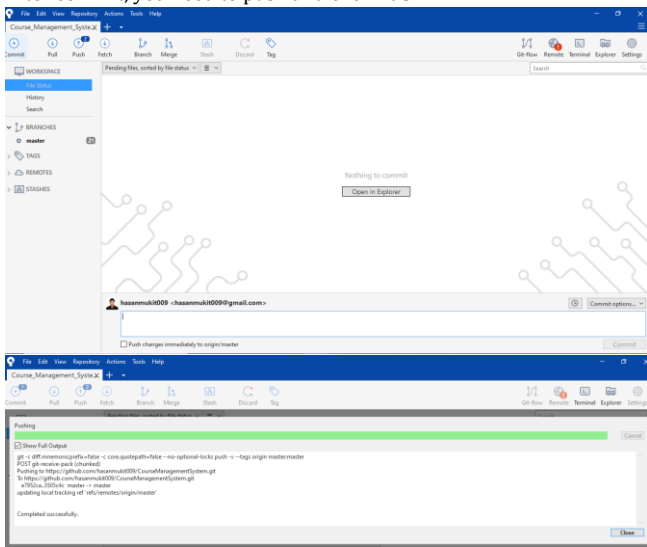


Develop Course Management System step by step in .net core

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.

