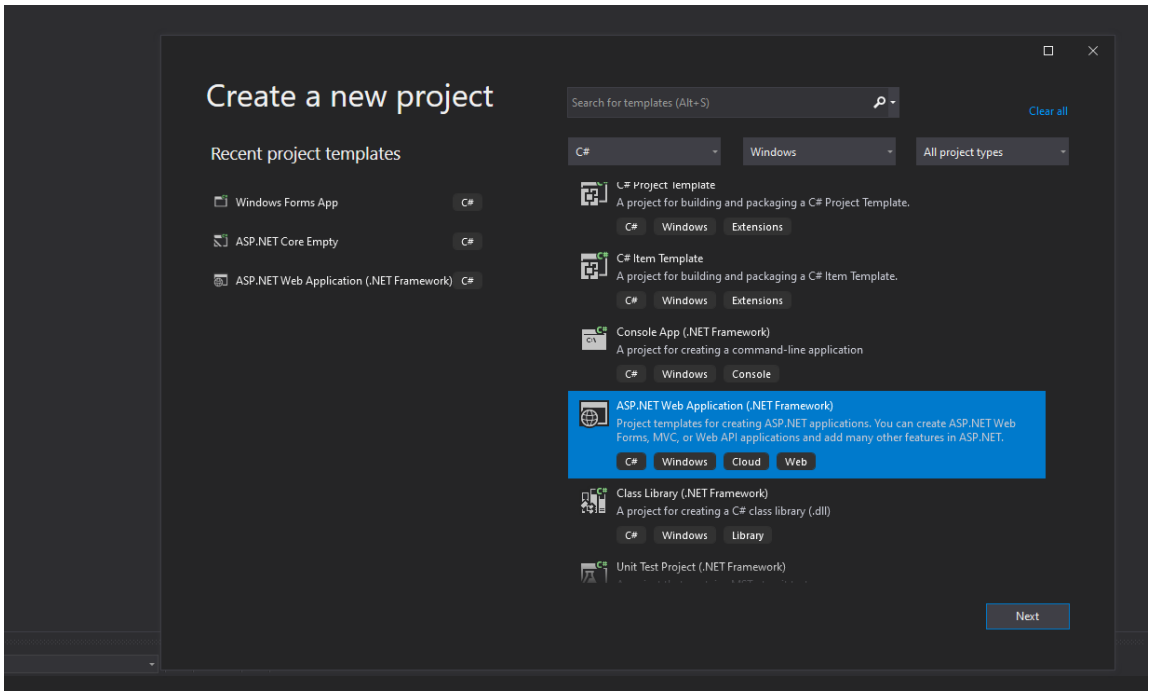
ASP .NET MVC 5 WITH LINQ QUIRES

FROM THE BASICS

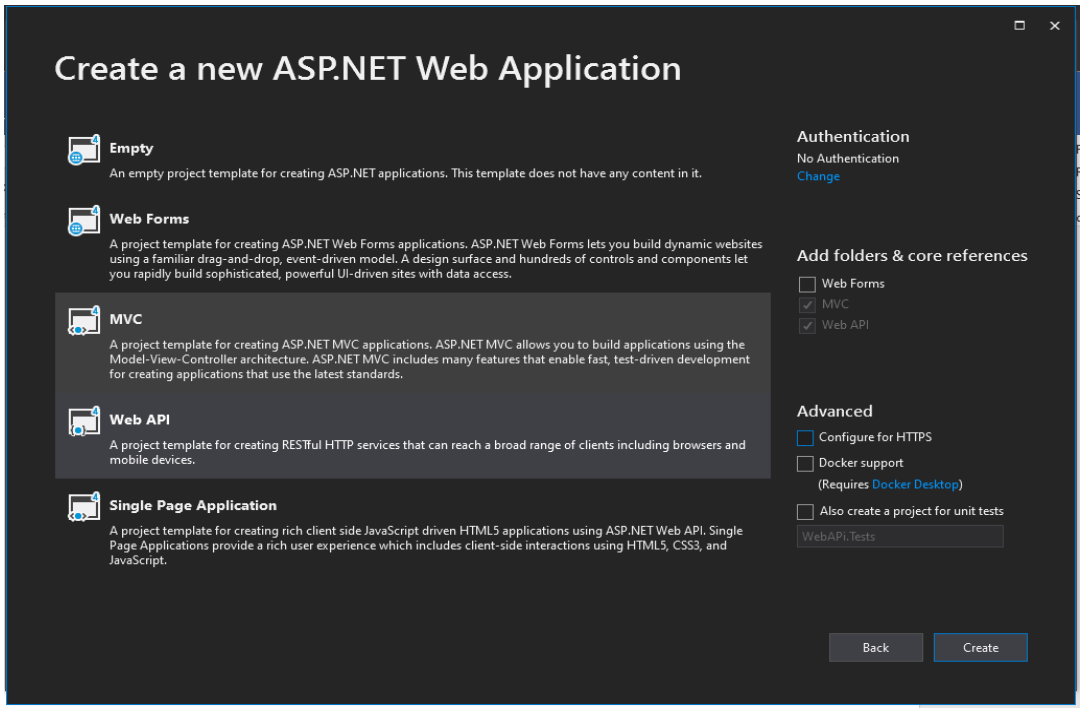
- Download Visual Studio Community Edition 2019 (free version from Microsoft) from https://visualstudio.microsoft.com/downloads/
- Open Installer of Visual Studio Community Edition 2019
- Click next and when you get selection for packages to install
- Select Packages (Must)
  - ASP.Net and Web Development
  - .Net Desktop Development
  - Visual studio extension Development
  - .Net Core – cross platform Development
- Other than above mentioned, Select any package about your requirement
- Click next and Complete your Installation
- After Installation open Visual studio and sign in with your Microsoft account

Usman Mughal (usman.mughal@learninginz.com)

- For Data Base Creation Find the MSSQL Server 2019 Creating 2019 Document
- Open Visual Studio select File tab and select new -> project
- Select ASP .NET Web Application (.Net Framework)

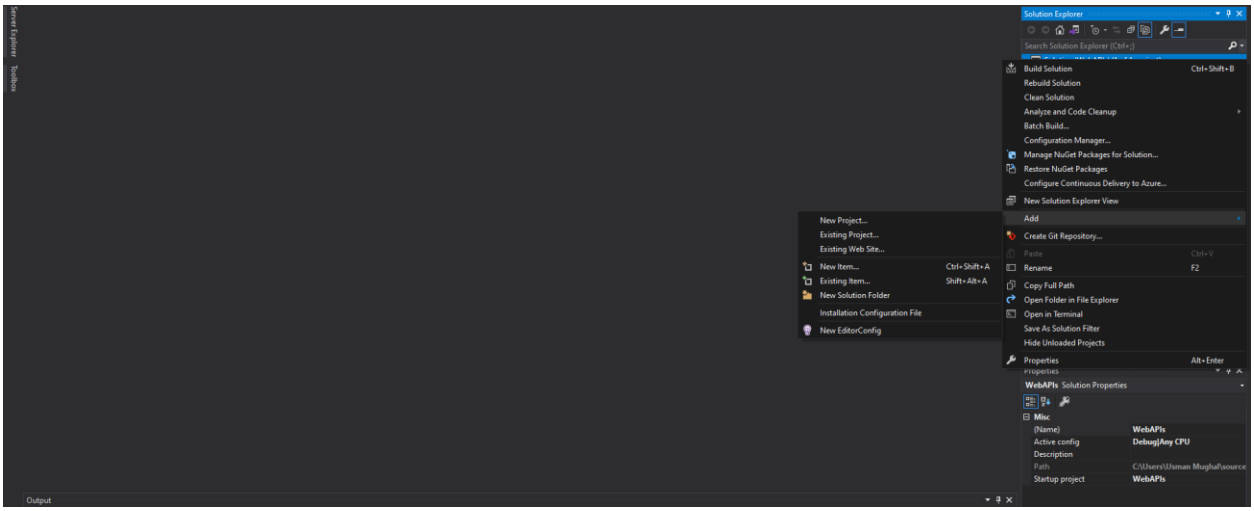


- Write Project Name
- And Click on Create Button
- On the next page Select Web API and shown in image below and Un Check Configure for HTTPS and Check on MVC and WEB API from left Side

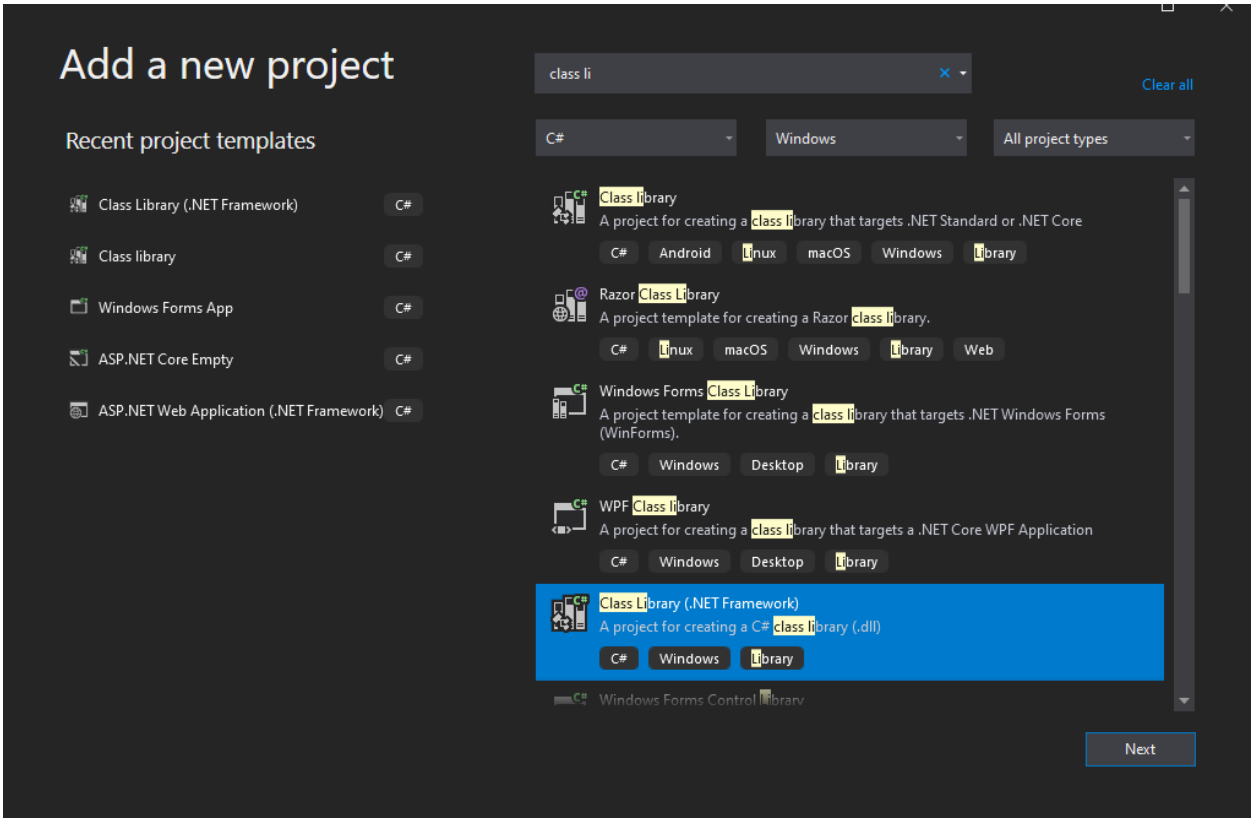


- Click on Create Button to complete project creation
- After creation, Dummy project will open image attached below

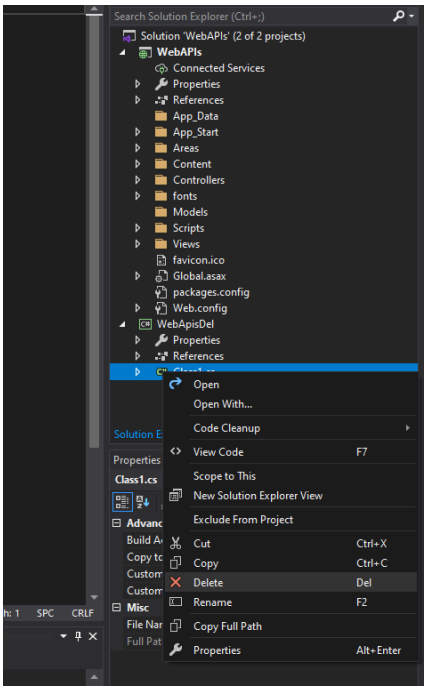
- Left click on **Solution** and select **Add -> New Project**


Server Explorer
Toolbox
Output
Solution Explorer
Search Solution Explorer (Ctrl+;)
Build Solution Ctrl+Shift+B
Rebuild Solution
Clean Solution
Analyze and Code Cleanup
Batch Build...
Configuration Manager...
Manage NuGet Packages for Solution...
Restore NuGet Packages
Configure Continuous Delivery to Azure...
New Solution Explorer View
Add
Create Git Repository...
Paste Ctrl+V
Rename F2
Copy Full Path
Open Folder in File Explorer
Open in Terminal
Save As Solution Filter
Hide Unloaded Projects
Properties Alt+Enter
New Project...
Existing Project...
Existing Web Site...
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Solution Folder
Installation Configuration File
New EditorConfig
WebAPIs Solution Properties
Misc
(Name) WebAPIs
Active config Debug|Any CPU
Description
Path C:\Users\Usman Mughal\source
Startup project WebAPIs

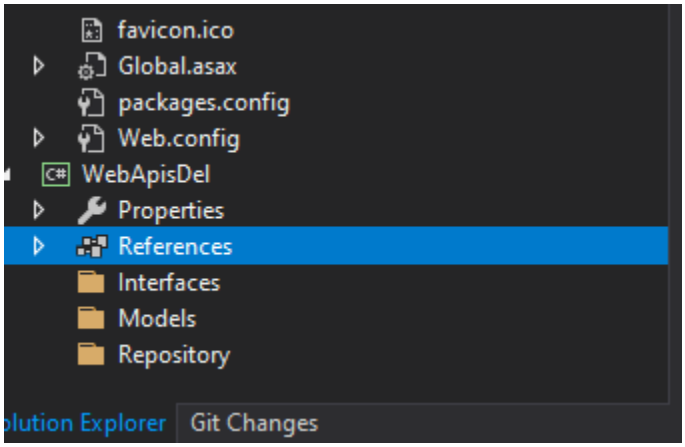
- Select **Class Library** (.Net Framework) and click next



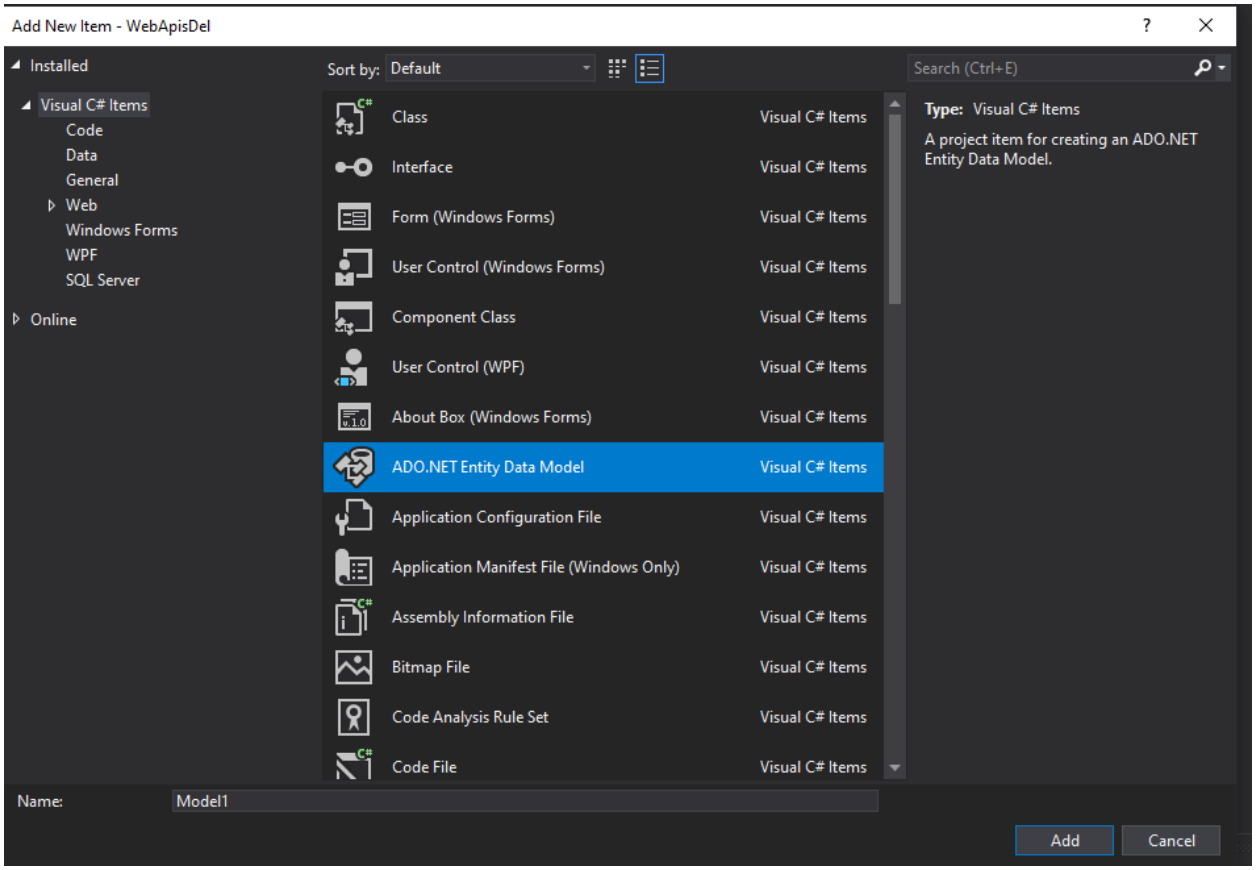
- Write **project name**
- Select Framework version (latest) and click create Project
- After Creation **Delete Class1.cs** file (don't required anymore)



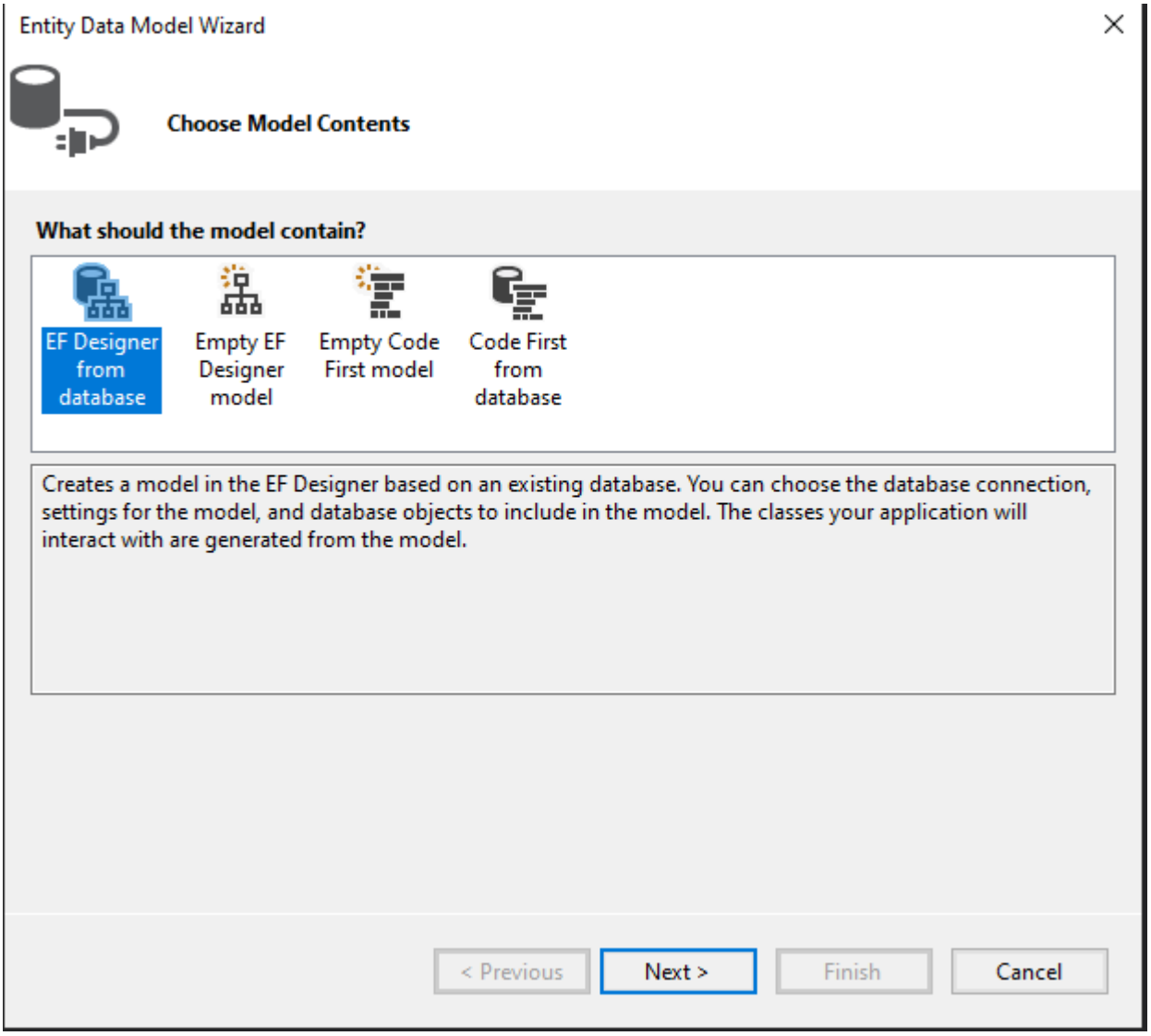
- Add Folders in **Project** (as per requirement)



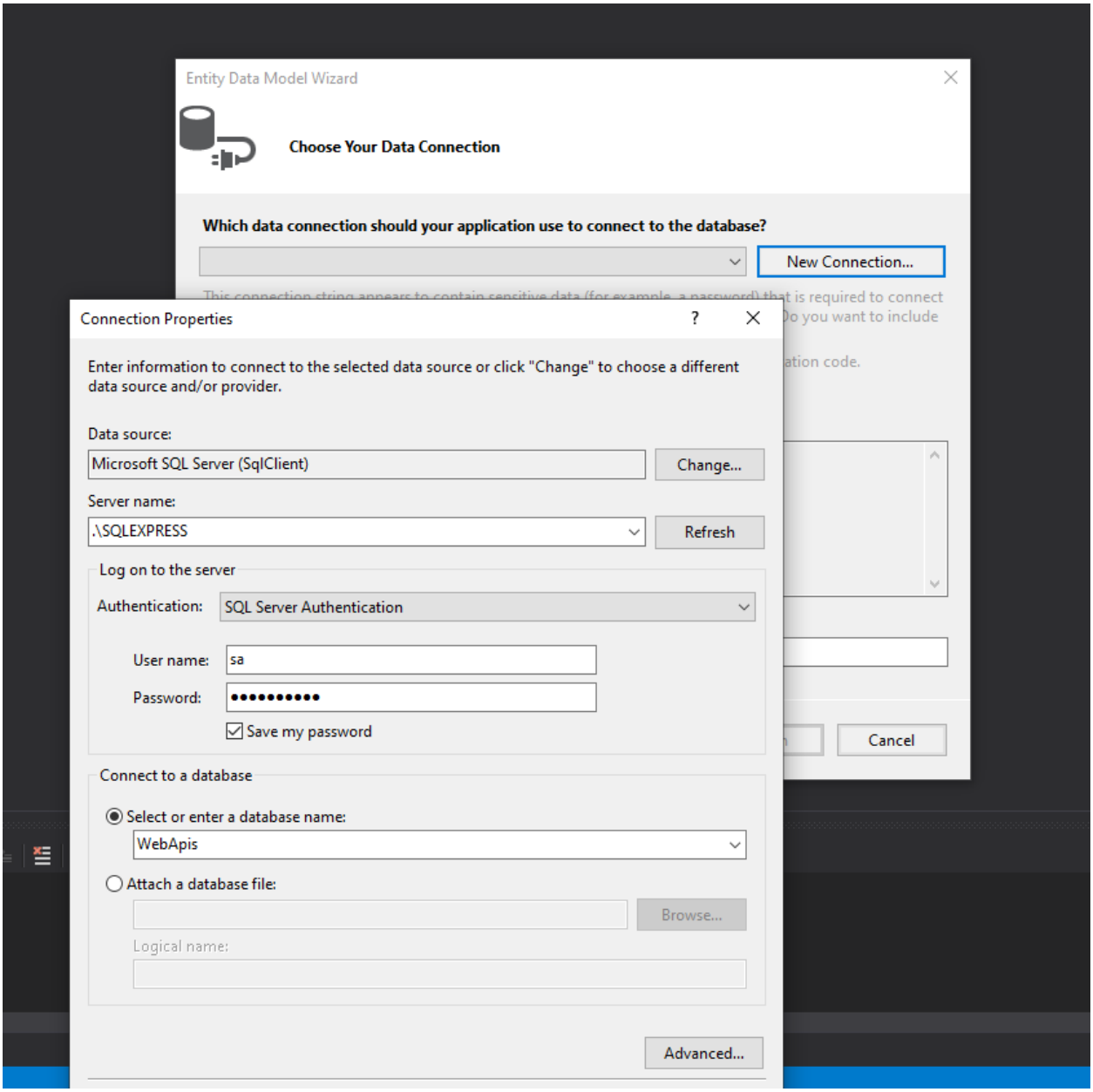
- Left Click on **Models** and **add New Item**
- Select **ADO.NET Entity Data Model**



- Select **EF Designer from Database**

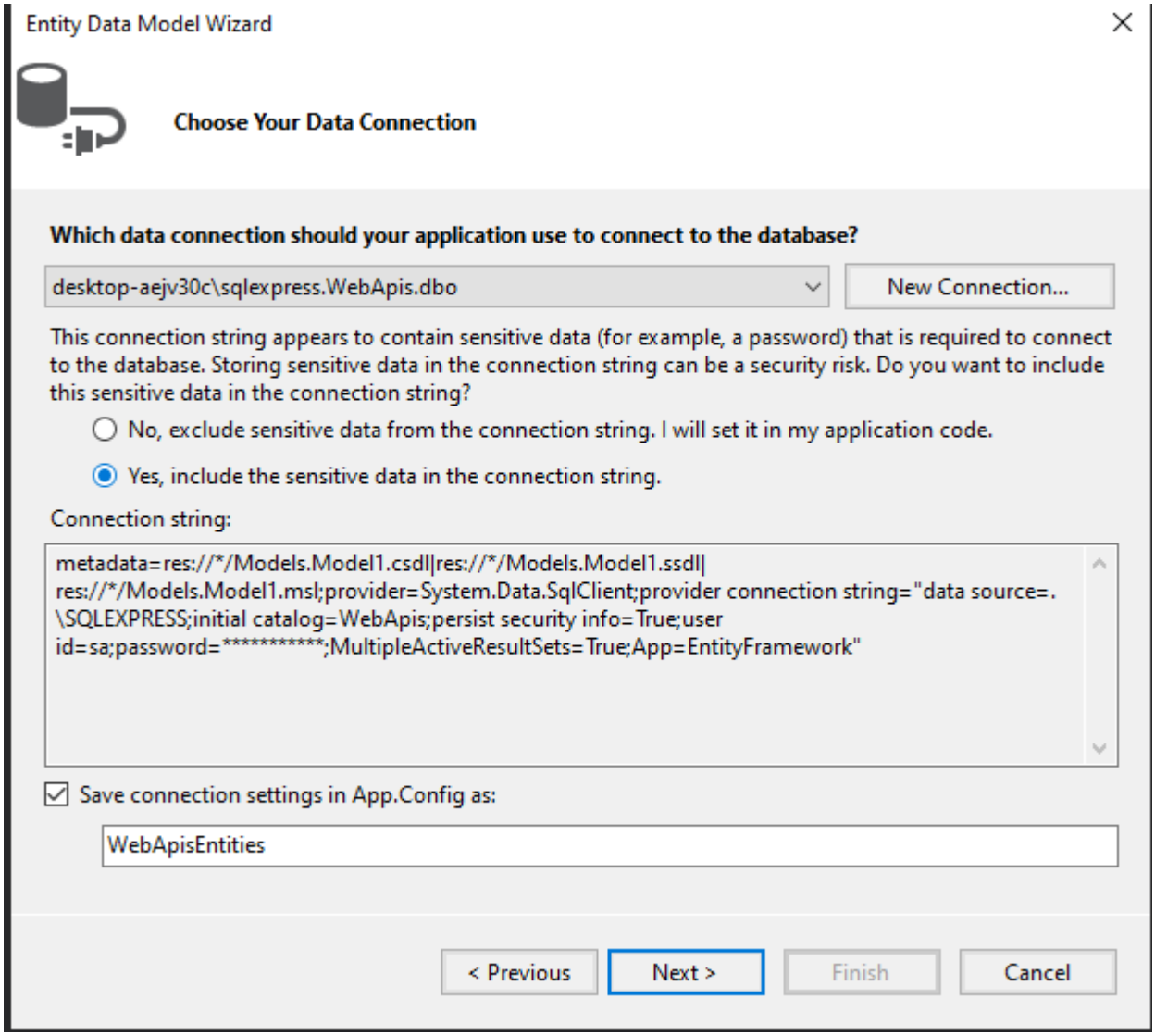


- Click **Next**
- Click on **New Connection**
- Write **Server Name**
- Select **Authentication** SQL Server Authentication
- Write **User Name** and **Password**
- Select **Data Base** Name


Entity Data Model Wizard
Choose Your Data Connection
Which data connection should your application use to connect to the database?
New Connection...
Connection Properties
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server (SqlClient)
Change...
Server name:
.\SQLEXPRESS
Refresh
Log on to the server
Authentication:
SQL Server Authentication
User name:
sa
Password:
Save my password
Connect to a database
Select or enter a database name:
WebApis
Attach a database file:
Browse...
Logical name:
Advanced...
Cancel

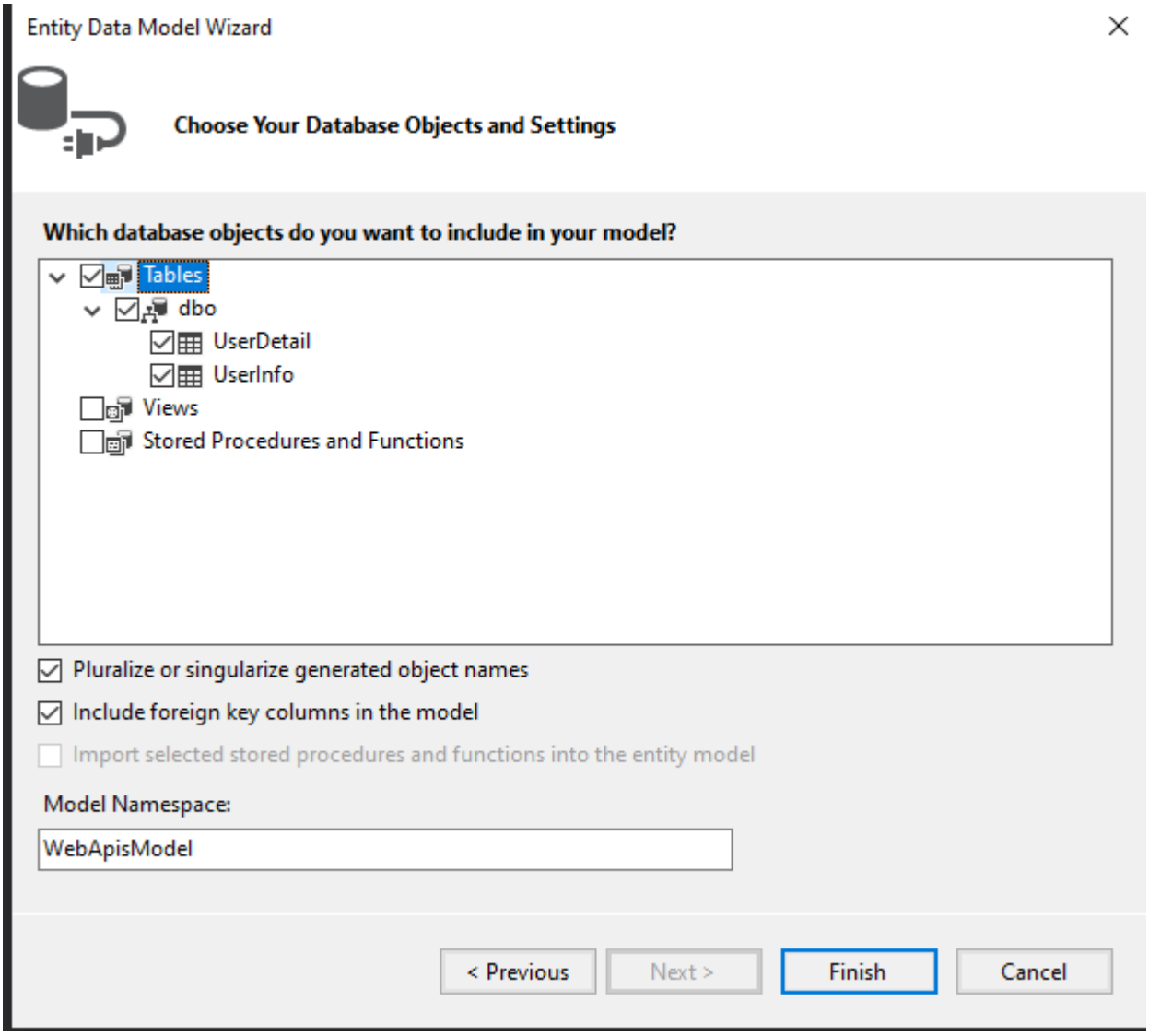
- Click **Ok**

- Select Check Box **Yes, Include the sensitive……**
-



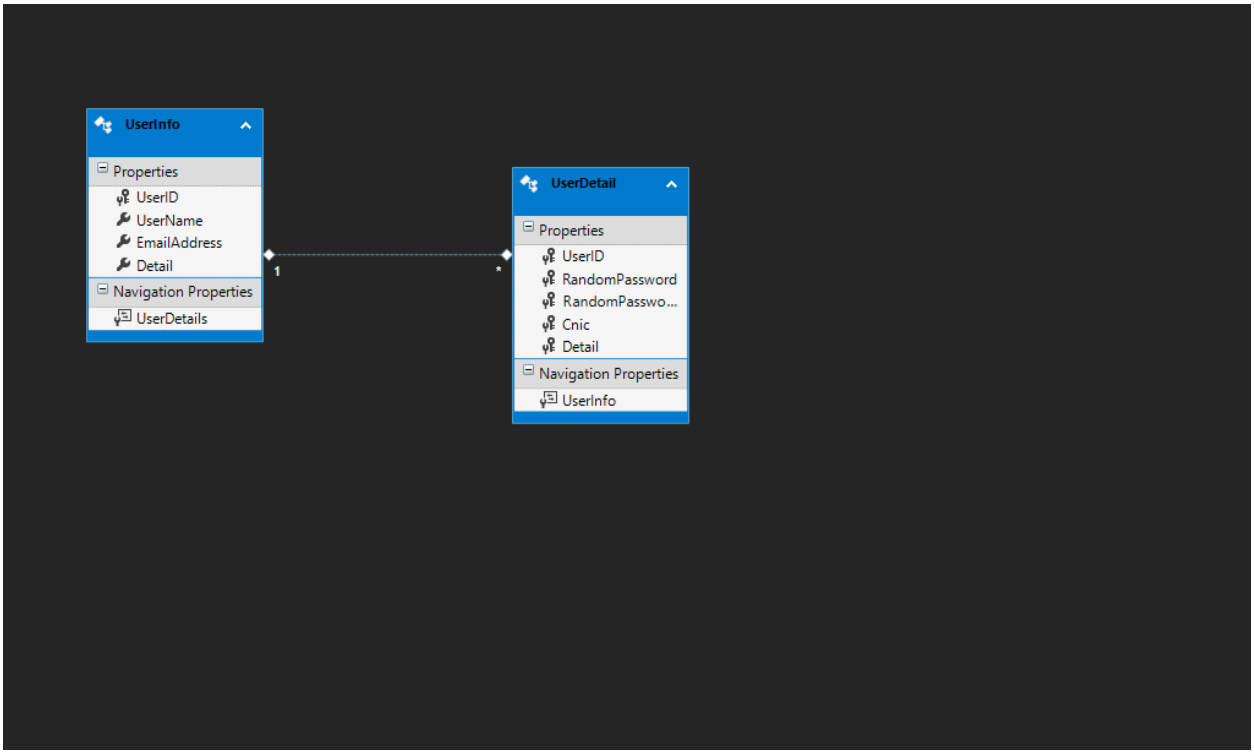
- Click **Next**
- Select **Entity Frame Work Version (latest)**
- Click **Next**

- Select tables which you want to connect

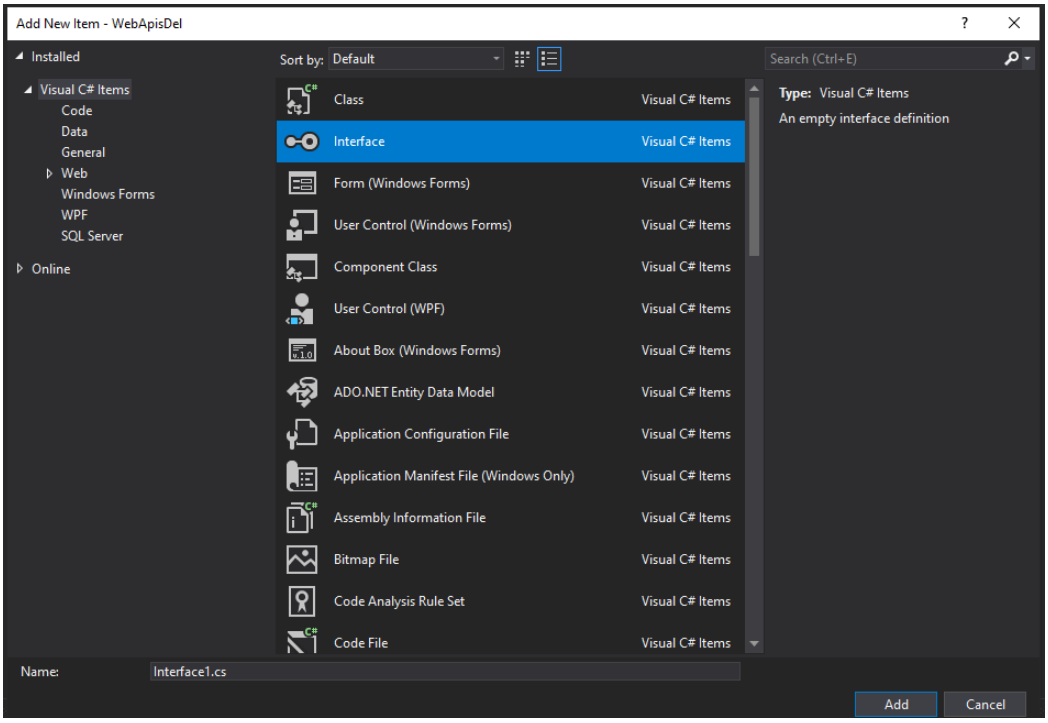


- Click Finish
- Model will create In form of Diagram
- Must Press (Ctrl+s) to save changes and retrieve Database

- Looks some thing like this



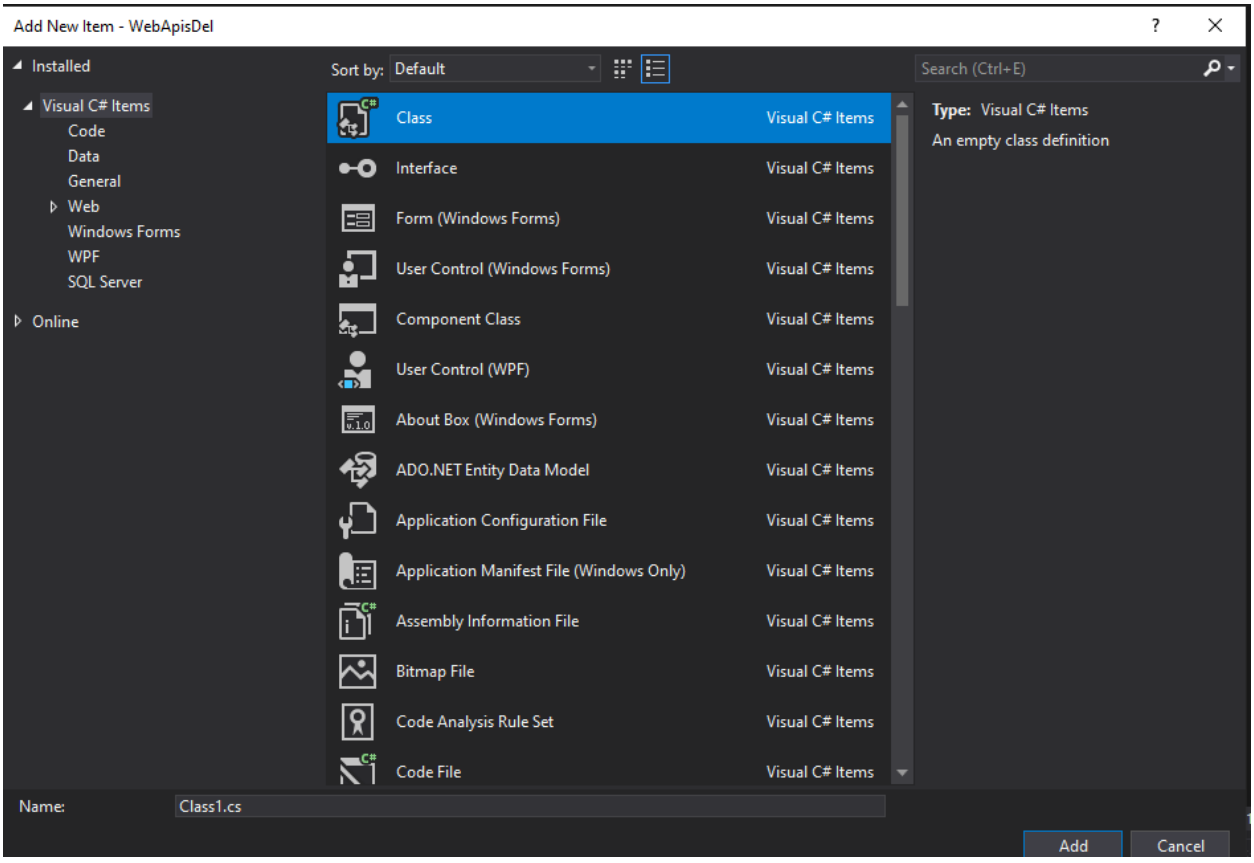
- Right click on **interface**
- Add **new item**
- Add **interface**



- Write Code in Interface
  - ```
    using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Text;
    using System.Threading.Tasks;

    namespace WebApisDel.Interfaces
    {
        interface IUsers
        {
            object GetAllUsers(String UserName);
        }
    }
    ```

- Right Click on **Repository**
- Add new **Item**
- Select **Class**


    ```

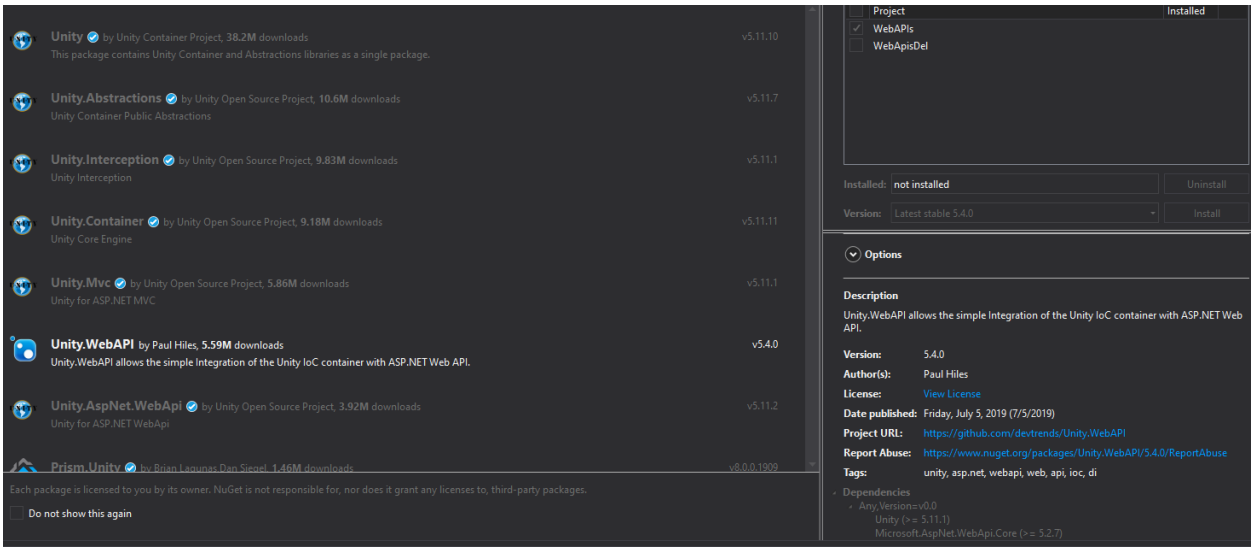
- Writ code in **Class**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using WebApisDel.Interfaces;
using WebApisDel.Models;

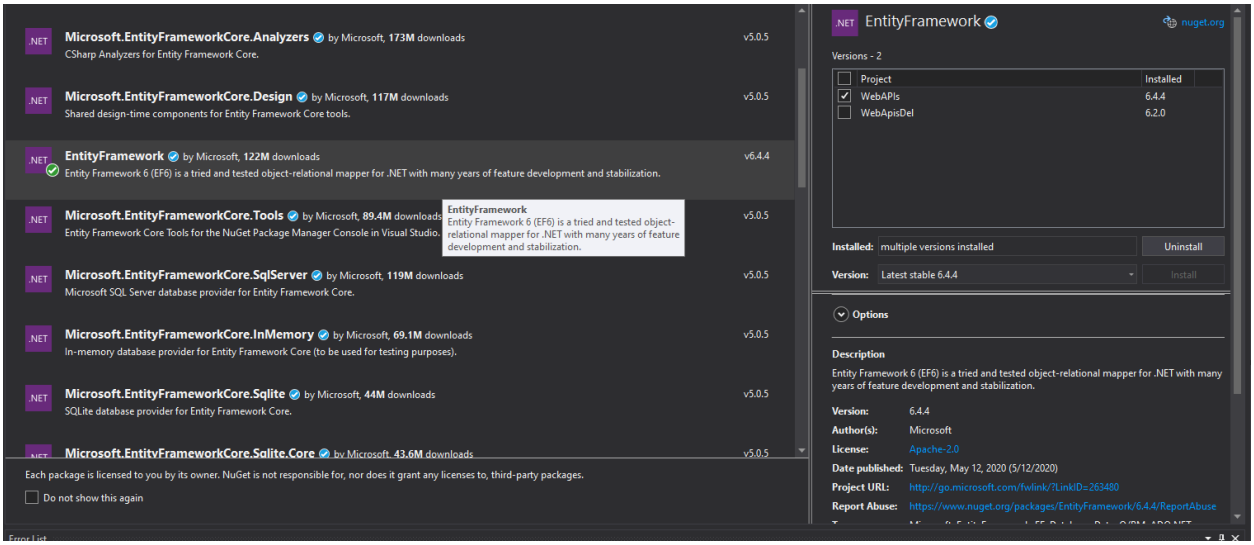
namespace WebApisDel.Repository
{
    public class UserRepo : IUsers
    {
        private readonly WebApisEntities context = new WebApisEntities();
        public object GetAllUsers(string UserName)
        {
       return context.UserDetails.Select(x=> new {
               x.Cnic,
              UserInfoSec = new {
                     x.UserInfo.EmailAddress
         }
            }).ToList();        }
    }
}
```

- **Repository** with **data base** is ready
- Now we have to write code for creating **API's** in our first project (**WebAPIs**)

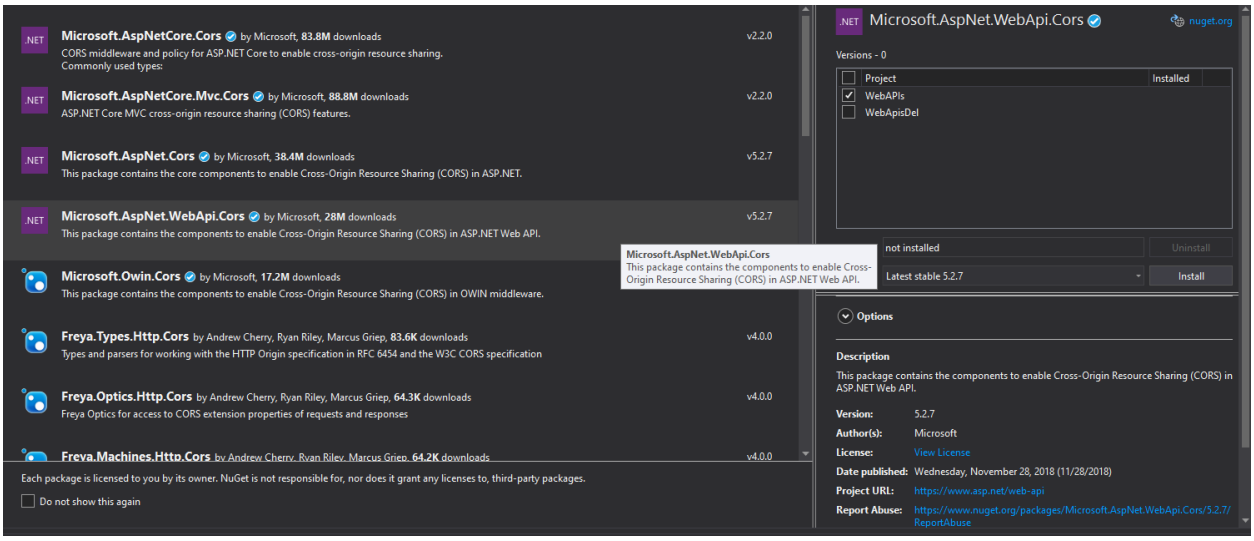
- Right Click on **Solution** and select **'Manage nudget packages for solution'**
- Search for Package **Unity**
- Select **Unity Web APIs** then select project where you want to install and install it



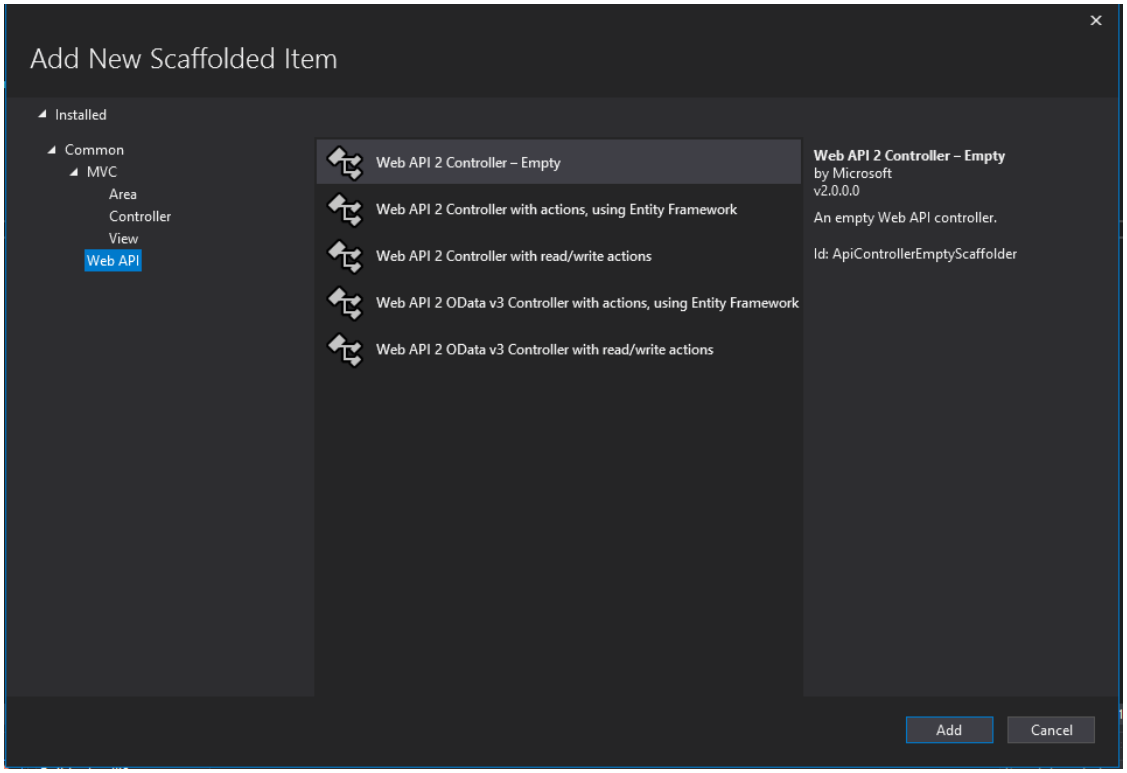
- Search for Package **Entity Frame Work**
- Select **Entity Framework** then select project where you want to install and install it



- Search for Package **CORS**
- Select **Microsoft ASPNet.WebAPi. Cors** then select project where you want to install and install it



- After this Right click on **controller** in **WebAPI's** Project
- Select **Add -> Controller**
- Select **Web Api's 2 Controller Empty**



- Write Code in **API Controller Class**

```
using System.Web.Http;
using WebApisDel.Interfaces;

namespace WebAPIs.Controllers
{
    public class UsersController : ApiController
    {
        private readonly IUsers userRepo;
        public UsersController(IUsers _userRepo)
        {
            userRepo = _userRepo;
        }

        [HttpGet]
        [Route("api/Users/GetAllUser")]
        public IHttpActionResult GetAllUser(string userName = "")
        {
            var result = userRepo.GetAllUsers(userName);
            return Json(result);

        }
    }
}
```

- Go to **Web API's Project**
- Select **App_Start**
- Open **WebApiConfig.cs**
- Replace all the **code** from old to given below

```
using System.Net.Http;
using System.Web.Http;
using System.Web.Http.Routing;

namespace WebAPIs
{
    public static class WebApiConfig
    {
        public static void Register(HttpConfiguration config)
        {
            // Web API configuration and services

            // Web API routes
            config.EnableCors();

            config.MapHttpAttributeRoutes();

            config.Routes.MapHttpRoute("DefaultApi", "api/{controller}/{id}", new { id =
            RouteParameter.Optional }, new { id = @"\d+" });

            config.Routes.MapHttpRoute("DefaultApiWithAction", "api/{controller}/{action}");

            config.Routes.MapHttpRoute("DefaultApiGet", "api/{controller}", new { action =
            "Get" }, new { httpMethod = new HttpMethodConstraint(HttpMethod.Get) });

            config.Routes.MapHttpRoute("DefaultApiPost", "api/{controller}", new { action =
            "Post" }, new { httpMethod = new HttpMethodConstraint(HttpMethod.Post) });

            config.Routes.MapHttpRoute("DefaultApiPut", "api/{controller}", new { action =
            "Put" }, new { httpMethod = new HttpMethodConstraint(HttpMethod.Put) });

            config.Routes.MapHttpRoute("DefaultApiDelete", "api/{controller}", new { action =
            "Delete" }, new { httpMethod = new HttpMethodConstraint(HttpMethod.Delete) });
        }
    }
}
```

- Go to **Web API's Project**
- Select **App_Start**
- Open **UnityConfig.cs**
- Replace the **method** from old to given below
    -

```
public static void RegisterComponents()
    {
            var container = new UnityContainer();
      // register all your components with the container here
     // it is NOT necessary to register your controllers
     // e.g. container.RegisterType<ITestService, TestService>();
     container.RegisterType<IUsers, UserRepo>();

     GlobalConfiguration.Configuration.DependencyResolver = new
     UnityDependencyResolver(container);
     }
```

- Note
  - we Added **container.RegisterType** and Pass it Interface and Repository

- Go to **Web API's Project**
- Open **Global.asax**
- Write below mentioned code in **Application_Start** Method
  - **UnityConfig.RegisterComponents();**

- Go to **Web API's Project**
- Open **Web.Config**
- Write below mentioned code in **<configuration> Tab** Method

```
<connectionStrings>
      <add name="WebApisEntities"
connectionString="metadata=res://*/Models.Model1.csdl|res://*/Models.Model1.ssdl|res://*/
Models.Model1.msl;provider=System.Data.SqlClient;provider connection string=&quot;data
source=.\SQLEXPRESS;initial catalog=WebApis;persist security info=True;user
id=sa;password=Welcome123;MultipleActiveResultSets=True;App=EntityFramework&quot;"
providerName="System.Data.EntityClient" />
</connectionStrings>
```