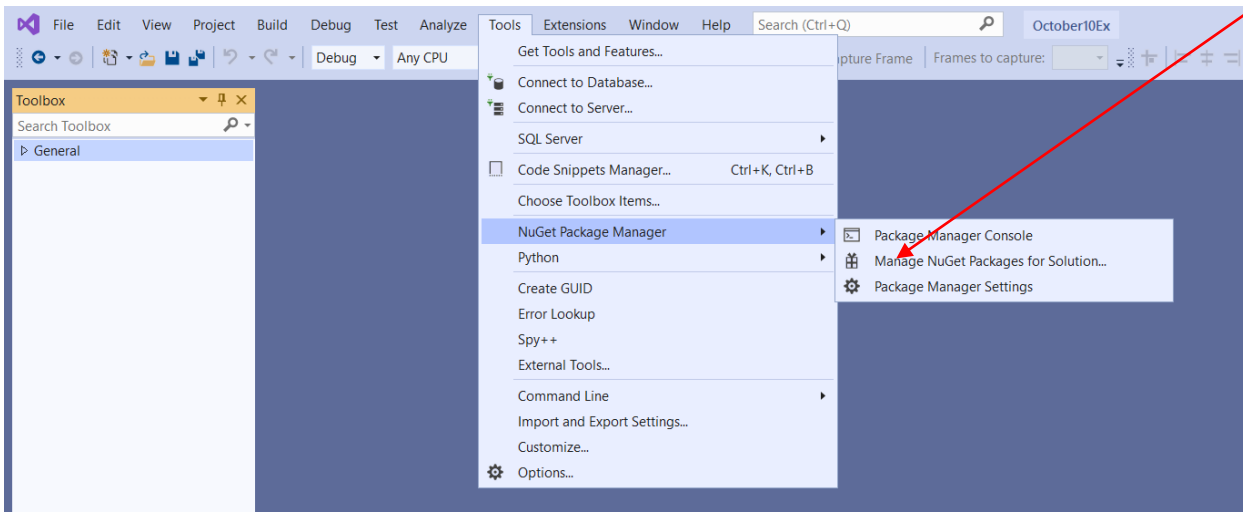


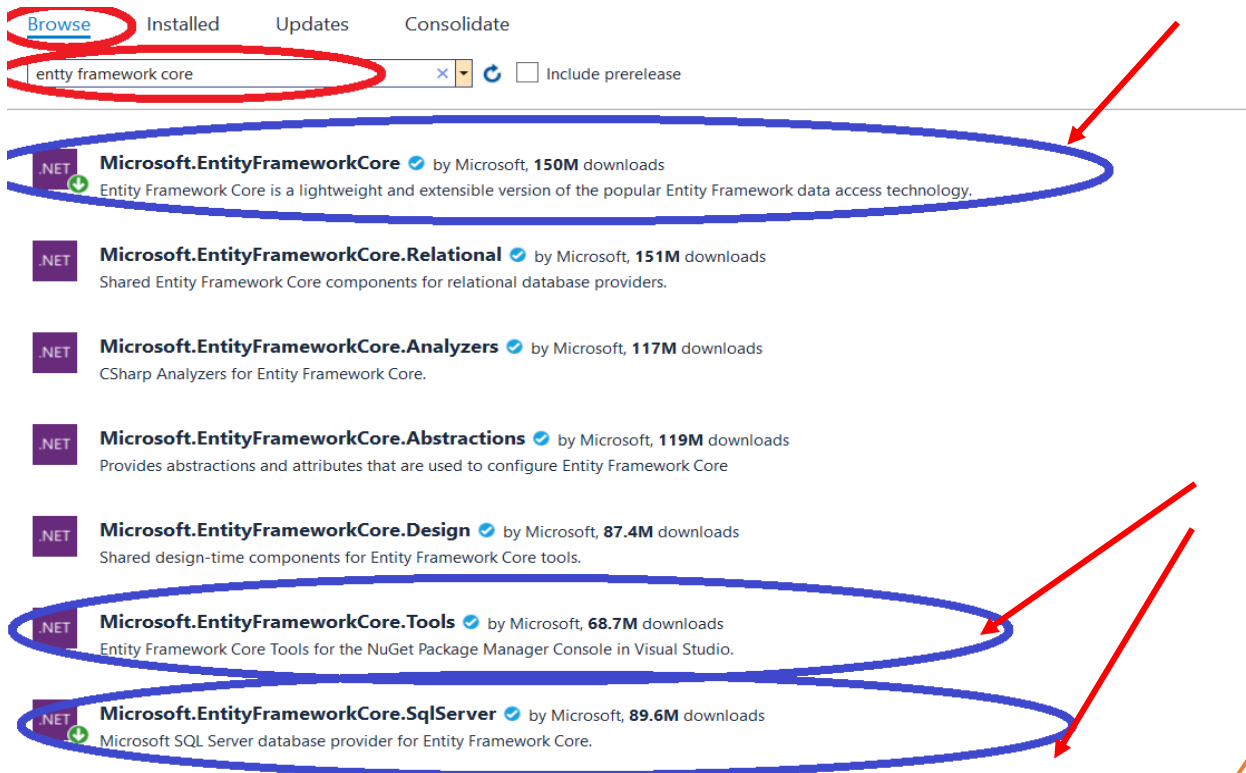
Creating CRUD pages by Reverse Engineering from SQL Server tables

(A) First create your CORE Web application from the template.

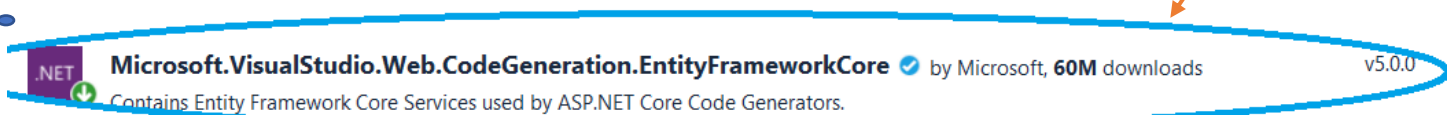
Install the following three packages from the NuGet package Manager. See image below on how to open the NuGet package manager.



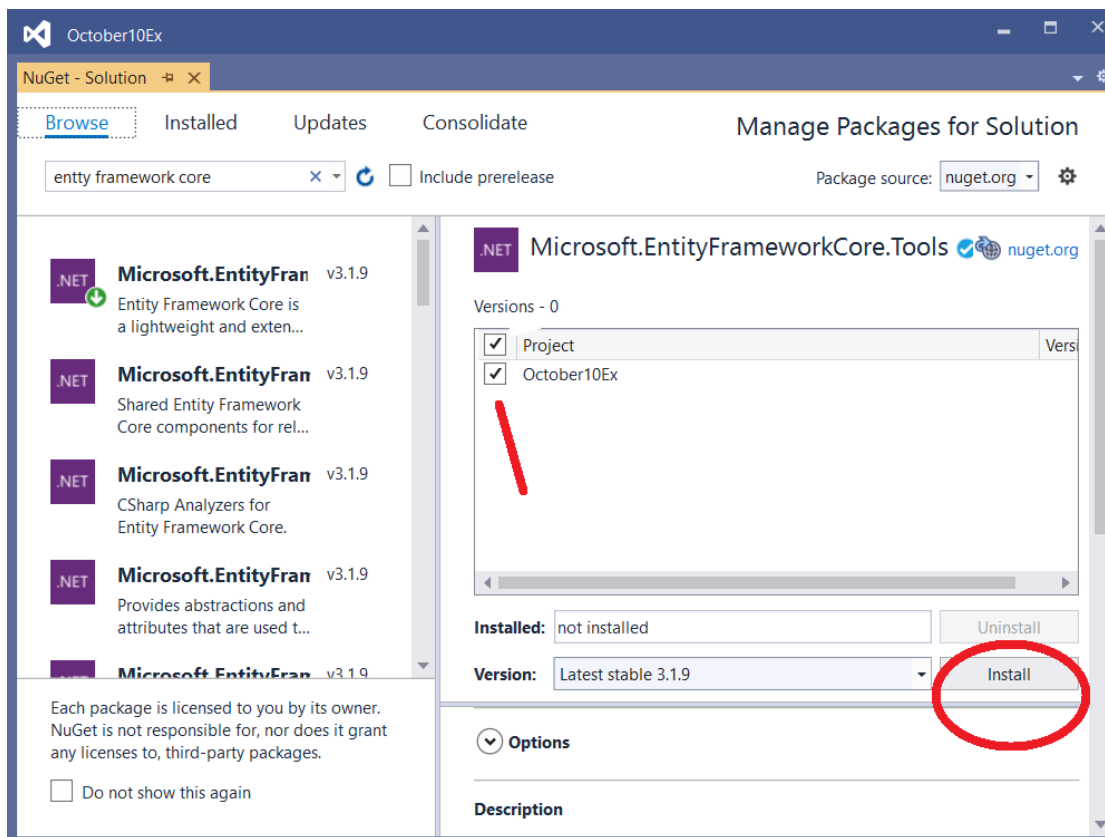
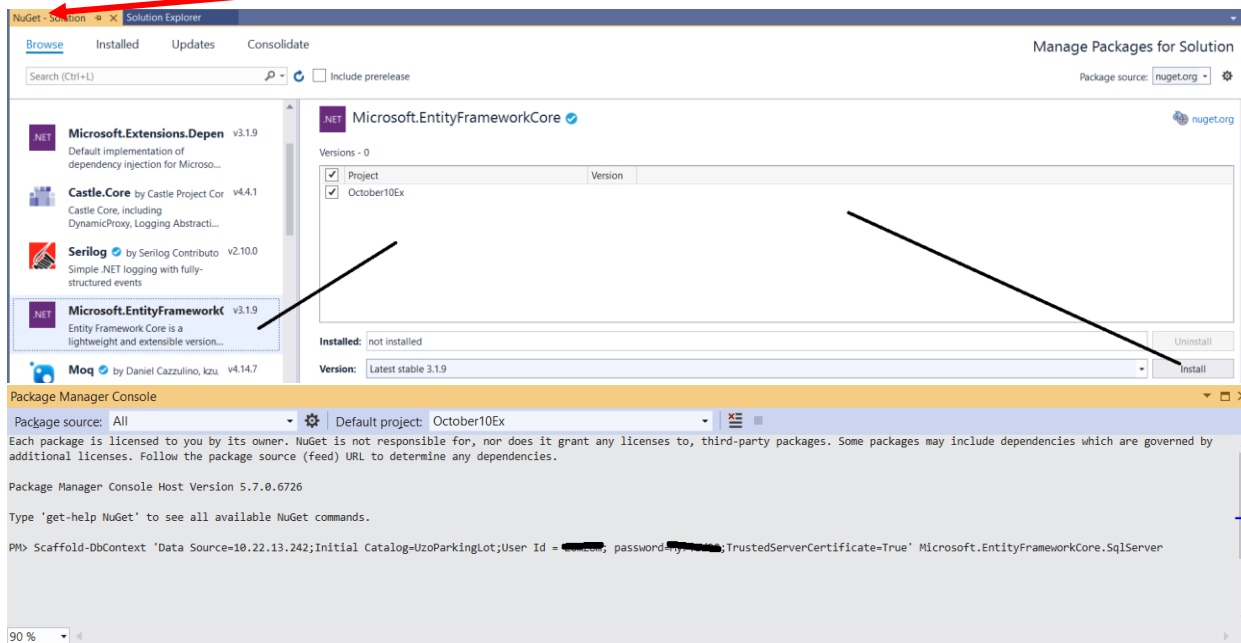
When you select the NuGet package manager, be sure to click on the Browse tab. If you enter “entity framework core” in the search bar, the three packages you are to install pop up. See the three packages encircled with blue ovals below



Also get

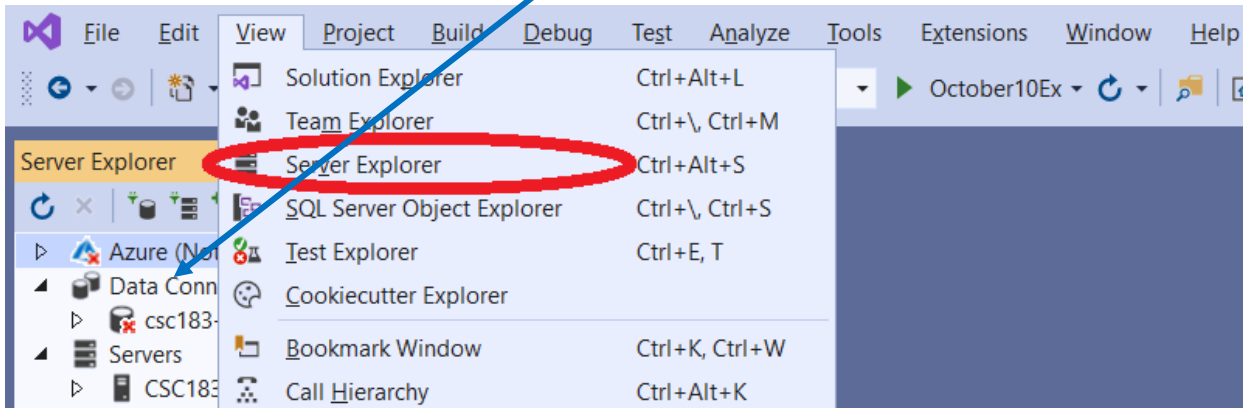


One by one, select each package and install. Click on the check boxes as seen below, and click on the install button seen on the bottom right of the screen. To see properly, click on Nuget Tab and select the FLOAT option and stretch the window.



(B) Set up the database connection

Select the Server Explorer, that will reveal the “Data Connection”. Right click on Data Connection, and click “Add Connection”

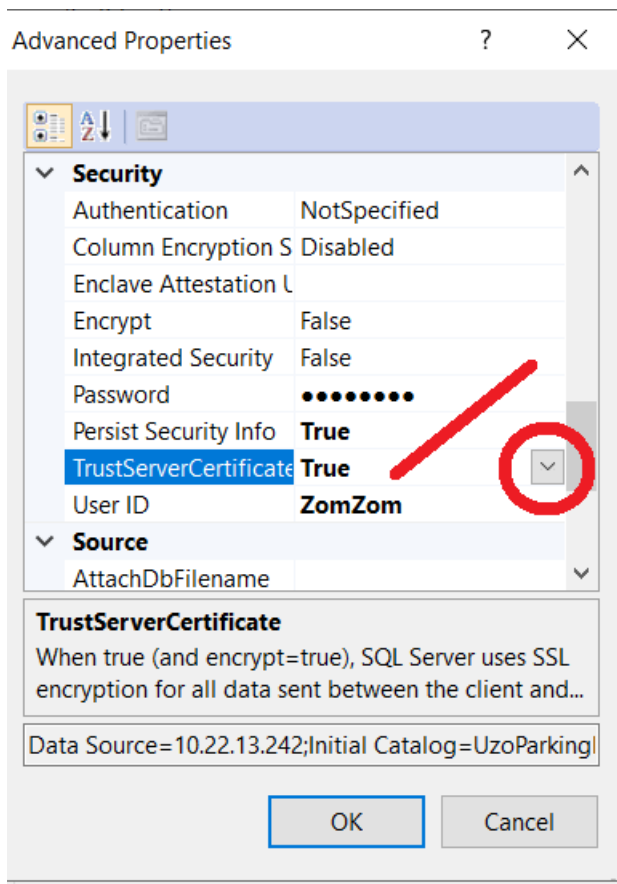
A screenshot of the 'Add Database Connection' dialog box in Visual Studio. Red annotations (arrows and lines) highlight several fields: 'Data source' (Microsoft SQL Server (SqlClient)), 'Server name' (10.22.13.242), 'Authentication' (SQL Server Authentication), 'User name' (SQL Server Authentication), 'Password' (Active Directory Password Authentication), 'Save my password' checkbox, 'Test Connection' button, and 'Advanced...' button. The 'Log on to the server' section is expanded, showing the authentication options. The 'Connect to a database' section has 'Select or enter a database name' selected.

- Select Microsoft SQL Server SQL Client
- Enter Server IP address of 10.22.13.242
- Select SQL Server Authentication
- CLICK SAVE MY PASSWORD
- Click Select database name
- Enter the database name you created in SQLSERVER

To make sure you are on the right track, click on Test Connection.

If it succeeded then

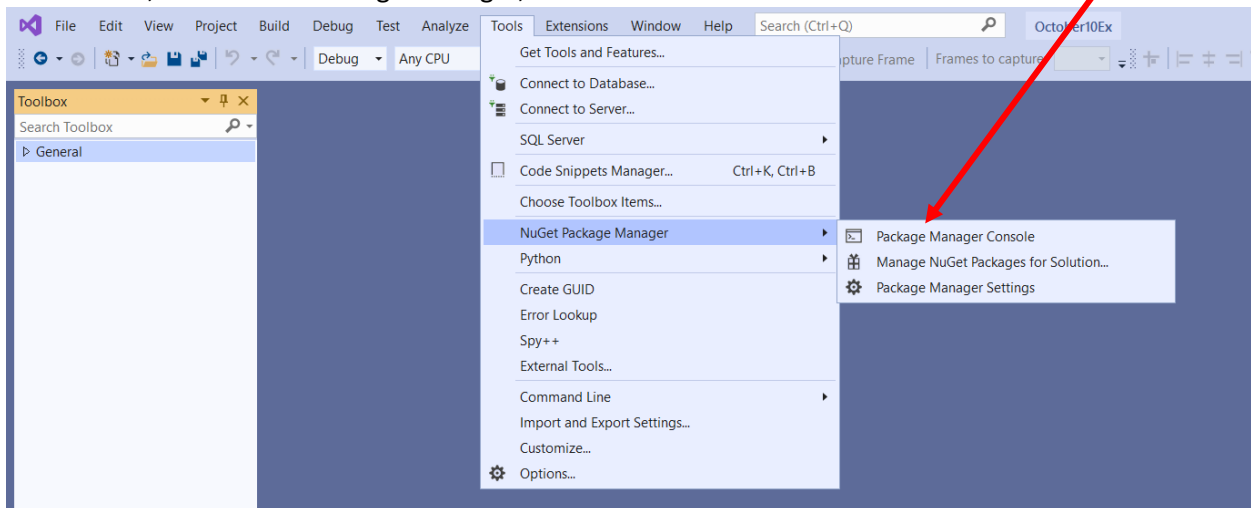
- Click ADVANCED and see the image on the next page



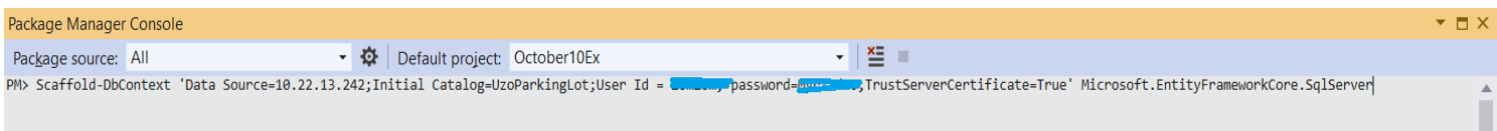
- TrustServerCertificate, be sure to Select TRUE
- Select OK

(C) Scaffold the Database

Select Tools, then NuGet Package Manager, then PACKAGE MANAGER CONSOLE



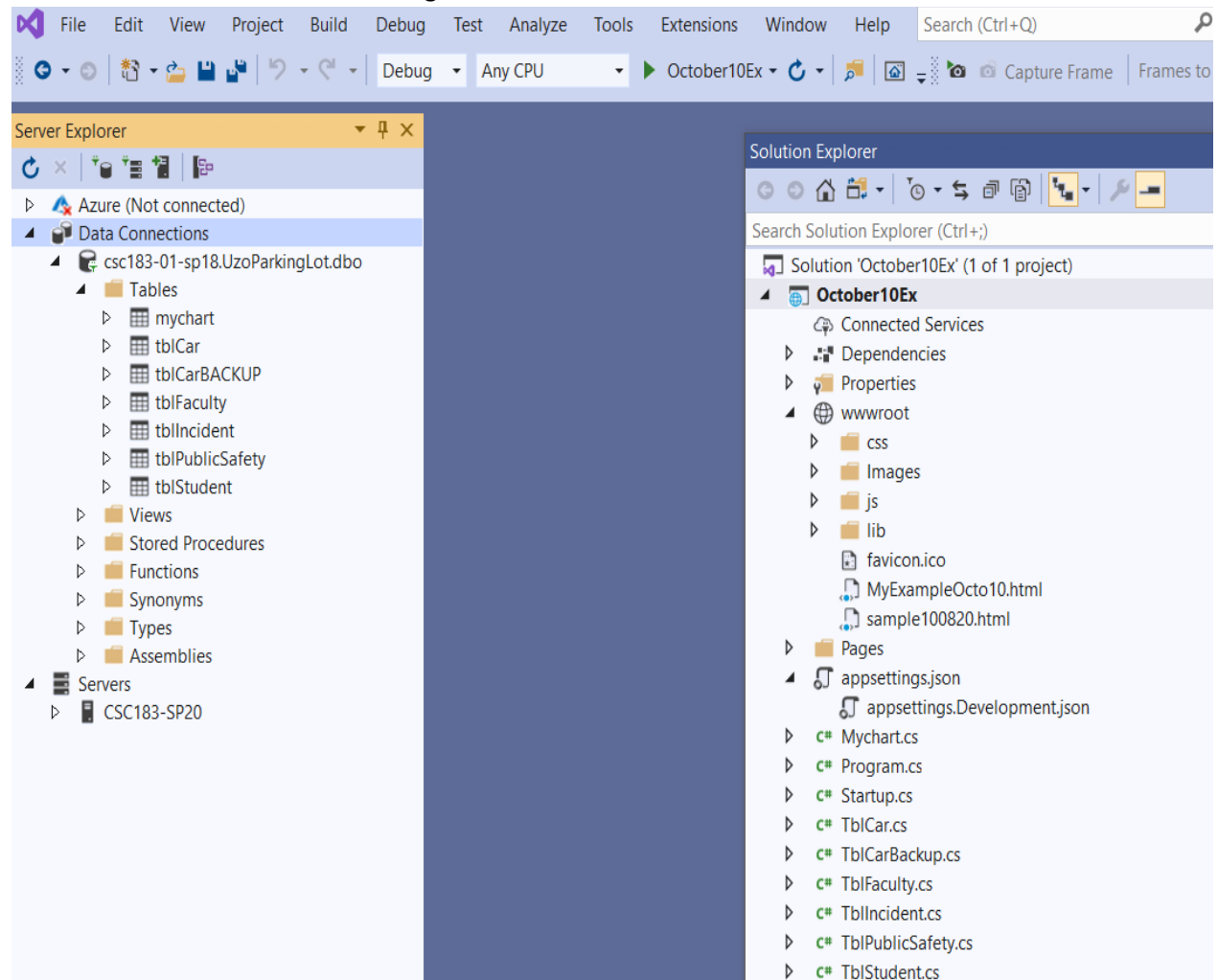
In the package manager console window, shown below, do the following:



Paste the following command in the console window, put **your database name** (the one you made in SQL SERVER), your **user id**, and **password** in the shaded areas below. Do exactly as shown:

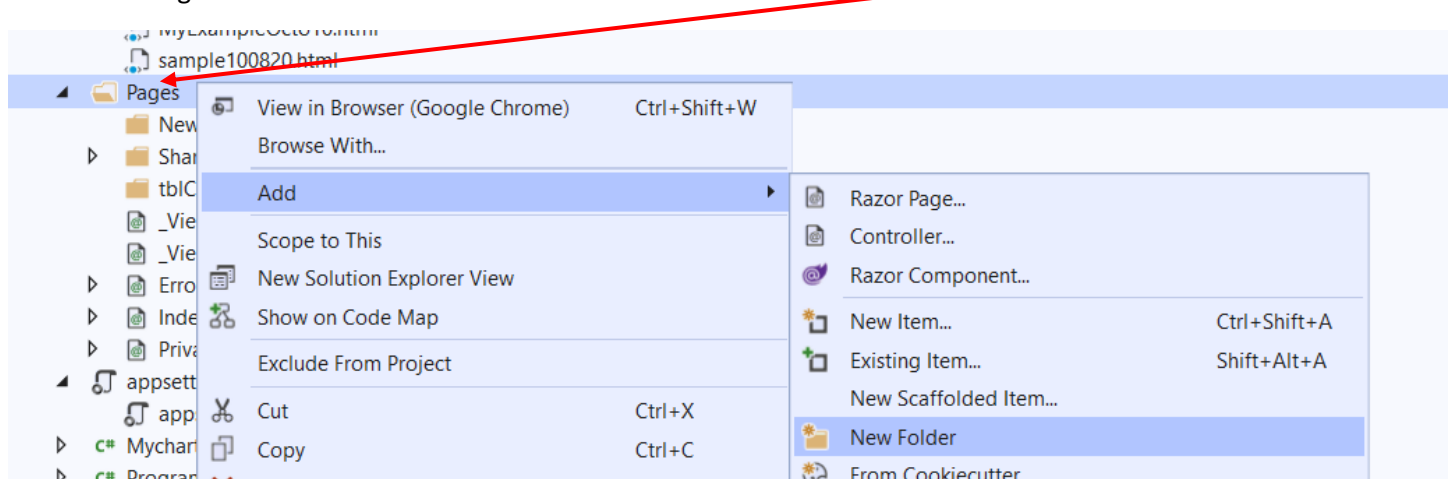
Scaffold-DbContext 'Data Source=10.22.13.242; Initial Catalog= yourDataBase ; User Id = yourUserId ; password= yourPwd ; TrustServerCertificate=True' Microsoft.EntityFrameworkCore.SqlServer

After running the above command, A C# (.cs) page will be made for each table in your specified database. Your screen should look like the image below:

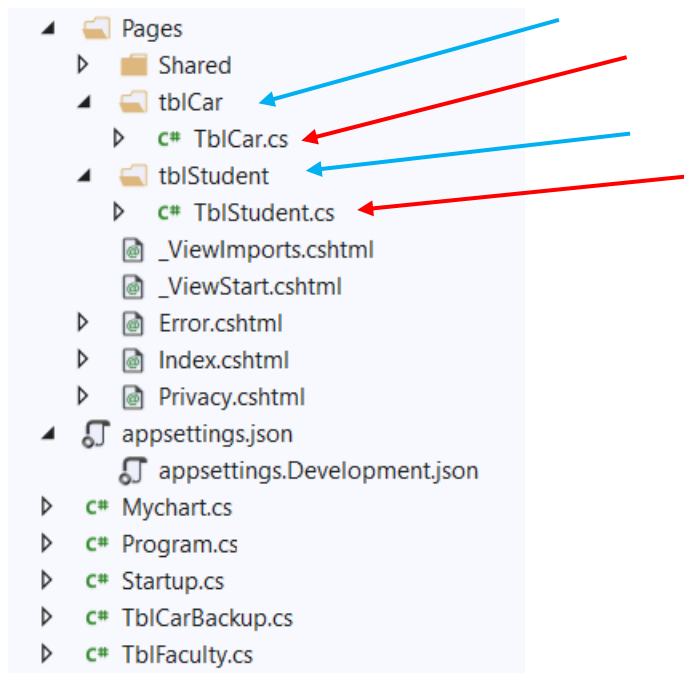


(D) Reengineering the Database and Make the CRUD Pages

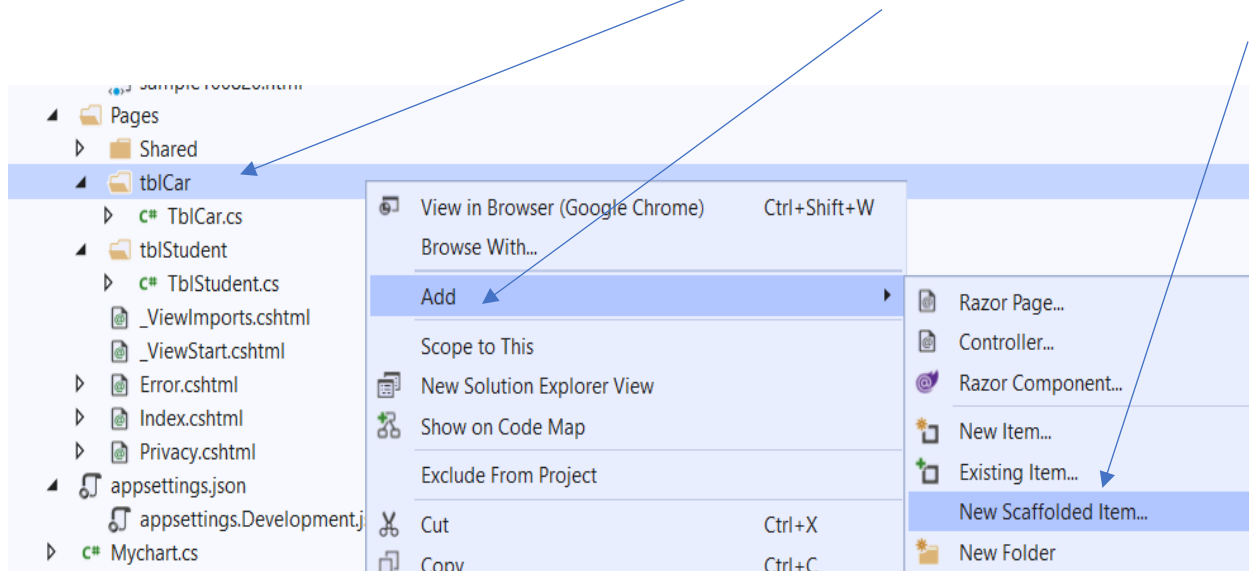
- 1) Start by creating a folder for each table you would like to scaffold. Be sure to select Pages before right clicking and Adding new folder.



2) Move the controller file for that table into the appropriate folder. Just drag the respective .cs file into the folder.

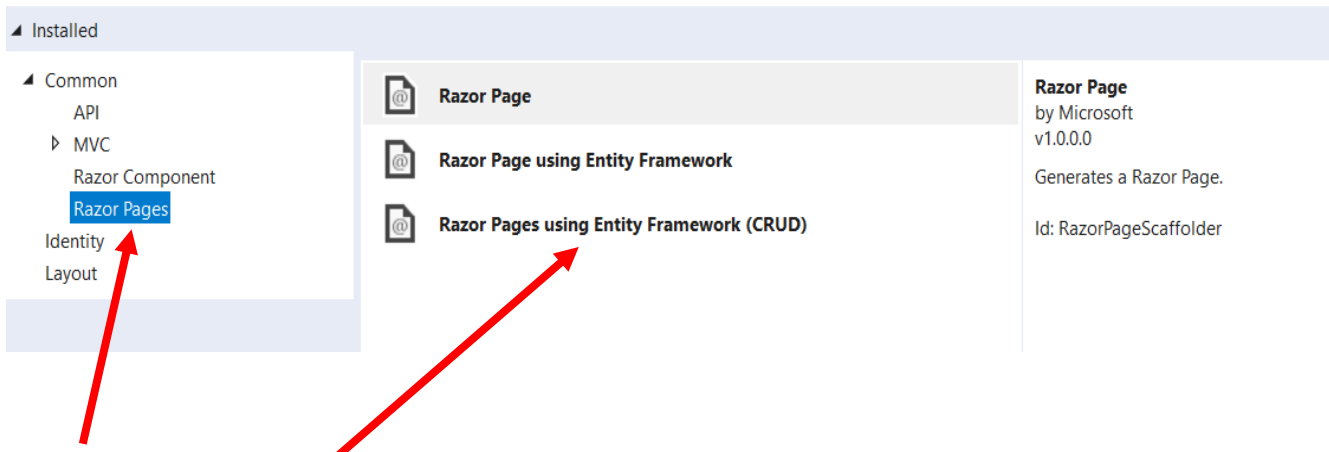


3) Make the Scaffolding

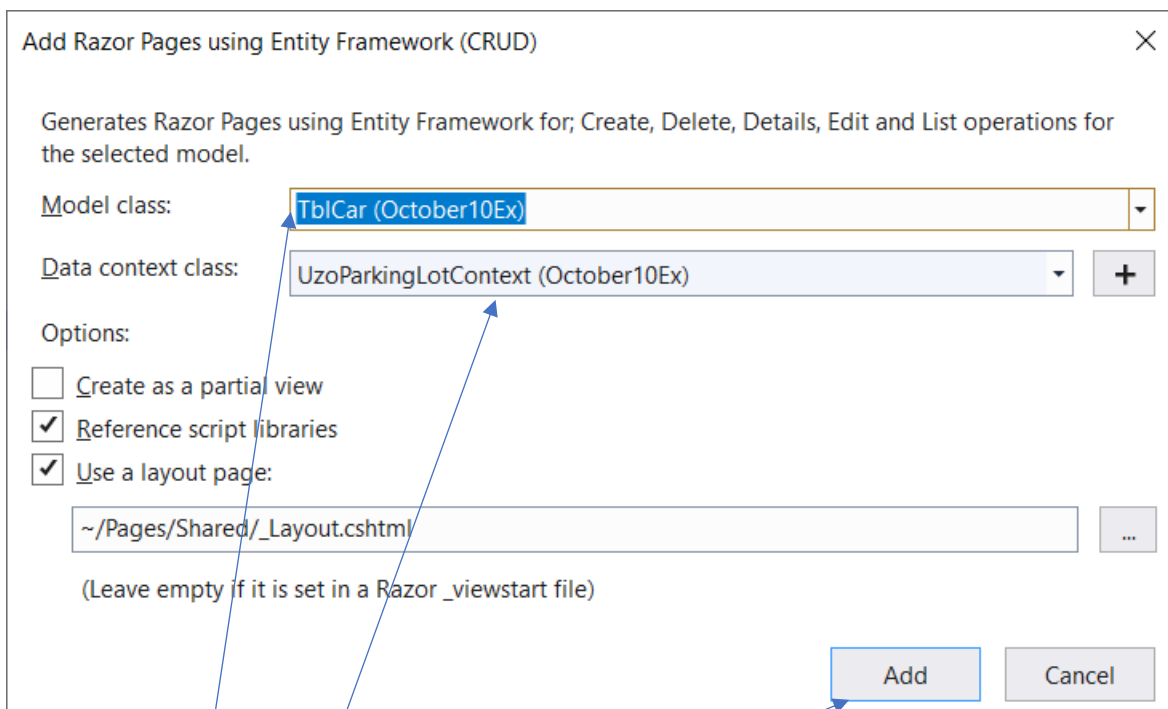


4) Create the Scaffolded CRUD pages

Add New Scaffolded Item

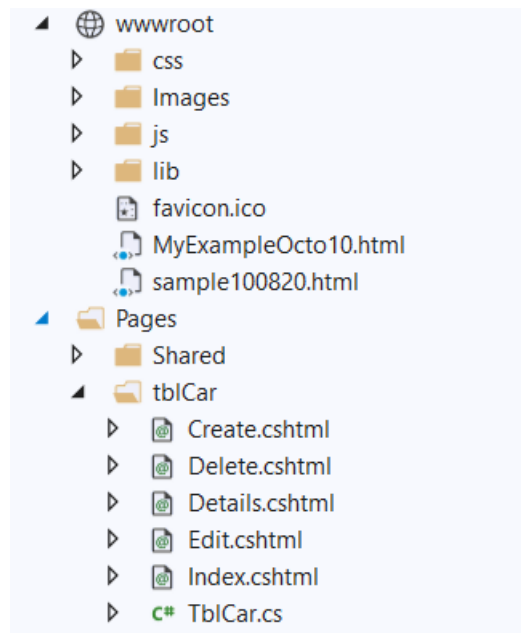


- 5) After selecting Razor Pages using Entity Framework CRUD, click the ADD button at the bottom right of the screen.



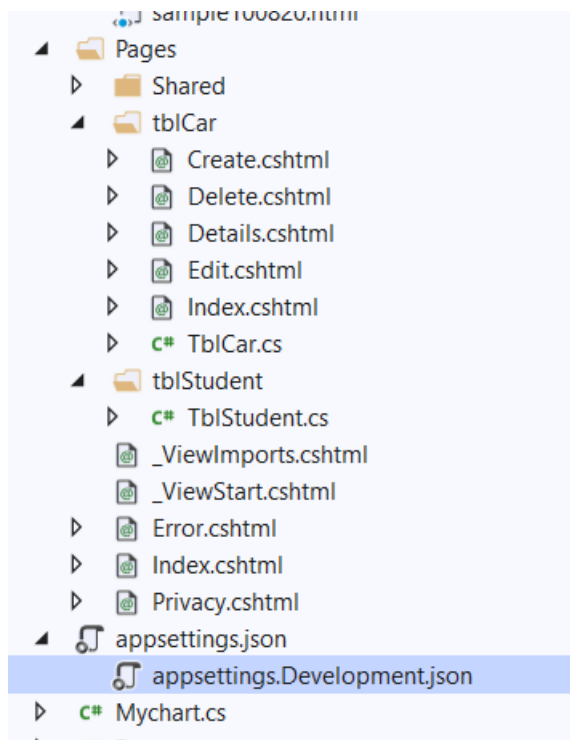
- 6) Select which entity (model class) you wish to create CRUD pages on. Your database context which should already be selected will be listed. Then click ADD

Your screen will look like the following:



E) Establish application settings in the appsettings.json file.

1) In your **appsettings.development.json** file seen below



Enter the following as a new json object in the file, replacing your database name, user id, and password as indicated by your sqlserver credentials:

"ConnectionStrings": {

"give your connection string a valid name": "Data Source=10.22.13.242; Initial Catalog =yourDatabase ; User ID = yourUSERId ; Password = yourPassword ; TrustServerCertificate = True ;"

}

For example "parkingDBConnection"
Include the quotes

Be sure to replace the Initial Catalog value with your database name, the user id and password with your sqlserver credentials. Also replace "give your connection string a name" with the name of your choice e.g.

"parkingDBConnection". Keep the SAME connection string name for any prompt requiring a connection to your database from this web application.

Note the curly braces used and the comma separating the previous object from the connection string object

```
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft": "Warning",
6       "Microsoft.Hosting.Lifetime": "Information"
7     }
8   },
9   "ConnectionStrings": {
10    "give your connection string a valid name": "Data Source=10.22.13.242;Initial Catalog=yourDatabase;user ID=yourUSERId;Password=yourPassword;TrustServerCertificate=True;"
11  }
12 }
13
14
15
```

- 2) Do the same thing for the appsettings.json file, slightly different from the development .json file. Note the comma after allowed hosts, and the balancing of the curly braces.

```
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft": "Warning",
6       "Microsoft.Hosting.Lifetime": "Information"
7     }
8   },
9   "AllowedHosts": "*",
10  "ConnectionStrings": {
11    "give your connection string a valid name": "Data Source=10.22.13.242;Initial Catalog=yourDatabase;user ID=yourUSERId;Password=yourPassword;TrustServerCertificate=True;"
12  }
13 }
14
15
```

3) Adjust **Startup.cs** file you will need the name of the context.cs page created in (C) and the name of your connection string created in (E1). At the top of the page add the following:

Add this →>

using Microsoft.EntityFrameworkCore;

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using Microsoft.AspNetCore.Builder;
6 using Microsoft.AspNetCore.Hosting;
7 using Microsoft.AspNetCore.HttpsPolicy;
8 using Microsoft.Extensions.Configuration;
9 using Microsoft.Extensions.DependencyInjection;
10 using Microsoft.Extensions.Hosting;
11 using Microsoft.EntityFrameworkCore;
12 namespace October10Ex
13 {
14     2 references
15     public class Startup
16     {
17         0 references
18         public Startup(IConfiguration configuration)
19         {
20             Configuration = configuration;
21         }
22         2 references
23         public IConfiguration Configuration { get; }
24
25         // This method gets called by the runtime. Use this method to add services to the container.
26         0 references
27         public void ConfigureServices(IServiceCollection services)
28         {
29             services.AddDbContextPool<UzoParkingLotContext>(options => options.UseSqlServer(Configuration.GetConnectionString(
30                 ("UzoParkingDBString"))));
31             services.AddRazorPages();
32         }
33
34         // This method gets called by the runtime. Use this method to configure the HTTP request pipeline
```

Replace the contents of the public void ConfigureServices function with your specified connection string, instead of leaving your user id and password exposed.

Put the following statement **Before** the services.AddRazorPages() statement

services.AddDbContextPool< replace with the name of your context file >(options =>
options.UseSqlServer(Configuration.GetConnectionString("name of your connection string")));

F) Modify contents of context file to utilize connection string as opposed to exposing user id and password

- ▷ C# Program.cs
- ▷ C# Startup.cs
- ▷ C# TblCarBackup.cs
- ▷ C# TblFaculty.cs
- ▷ C# TblIncident.cs
- ▷ C# TblPublicSafety.cs
- ▷ C# UzoParkingLotContext.cs

Your context file will have a name that has your databasename with the word Context and file extension of .cs

The screenshot shows the Visual Studio IDE with the 'UzoParkingLotContext.cs' file open. The file is part of the 'October10Ex' solution. The code is as follows:

```
1 using System;
2 using Microsoft.EntityFrameworkCore;
3 using Microsoft.EntityFrameworkCore.Metadata;
4 using Microsoft.Extensions.Configuration; //ADD THIS
5
6 namespace October10Ex
7 {
8     13 references
9     public partial class UzoParkingLotContext : DbContext
10     {
11         //REMOVE
12         // public UzoParkingLotContext()
13         // {
14         // }
15
16         0 references
17         public UzoParkingLotContext(DbContextOptions<UzoParkingLotContext> options)
18             : base(options)
19         {
20         }
21
22         0 references
23         public virtual DbSet<Mychart> Mychart { get; set; }
24
25         8 references
26         public virtual DbSet<TblCar> TblCar { get; set; }
27
28         0 references
29         public virtual DbSet<TblCarBackup> TblCarBackup { get; set; }
30
31         0 references
32         public virtual DbSet<TblFaculty> TblFaculty { get; set; }
33
34         0 references
35         public virtual DbSet<TblIncident> TblIncident { get; set; }
36
37         0 references
38         public virtual DbSet<TblPublicSafety> TblPublicSafety { get; set; }
39
40         0 references
41         public virtual DbSet<TblStudent> TblStudent { get; set; }
42
43         0 references
44         public virtual DbSet<VwFacultyCar> VwFacultyCar { get; set; }
45
46         0 references
47         public virtual DbSet<VwIncidents> VwIncidents { get; set; }
48
49         0 references
50         public virtual DbSet<VwStudentBirthday> VwStudentBirthday { get; set; }
51
52         1 reference
53         public IConfiguration myconfig { get; } //ADD THIS
54
55         0 references
56         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
57         {
58             if (!optionsBuilder.IsConfigured)
59             {
60                 optionsBuilder.UseSqlServer(myconfig.GetConnectionString("UzoParkingDBString")); //ADD THIS
61             }
62         }
63
64         0 references
65         protected override void OnModelCreating(ModelBuilder modelBuilder)
66         {
67             modelBuilder.Entity<Mychart>(entity =>
68             {
69                 entity.HasNoKey();
70             });
71         }
72     }
73 }
```

Annotations in the image include:

- An arrow pointing to line 4: "Add this sentence 'using Microsoft.Extensions.Configuration;'"
- An arrow pointing to the empty constructor (lines 11-14): "Remove the function public contextname which is an empty constructor. Be sure to remove the braces {}"
- An arrow pointing to line 53: "Add the public IConfiguration declaration for myconfig"
- An arrow pointing to line 60: "Add this sentence replacing my connection string name with your"

Save everything by clicking on double blue diskette icon on menu bar.

Test your CRUD pages by right-clicking on the index.cshtml of the table you scaffolded.and select View in Browser