

Sample task: 08312021\_Empite\_Co\_Solution

Project Documentation.

To develop this sample application I used below resources,

IDE: Visual studio community edition 2019

Database server: SQL server management studio

3<sup>rd</sup> party integrations: these packages are installed to current visual studio solution through “Nutget Package manager”

1. "Microsoft.AspNetCore.Identity.EntityFrameworkCore" Version="5.0.9"
2. "Microsoft.AspNetCore.Identity.UI" Version="5.0.9"
3. "Microsoft.EntityFrameworkCore.Sqlite" Version="5.0.8"
4. "Microsoft.EntityFrameworkCore.SqlServer" Version="5.0.8"
5. "Microsoft.EntityFrameworkCore.SqlServer.Design" Version="1.1.6"
6. "Microsoft.EntityFrameworkCore.Tools" Version="5.0.8"

This was the concept and procedure I followed!

First of all I have created ASP.Net core Web Application with MVC. After that I have two sub folders called “Data” and “Models”

Data:

This folder contains all context classes that I used to develop this sample application.

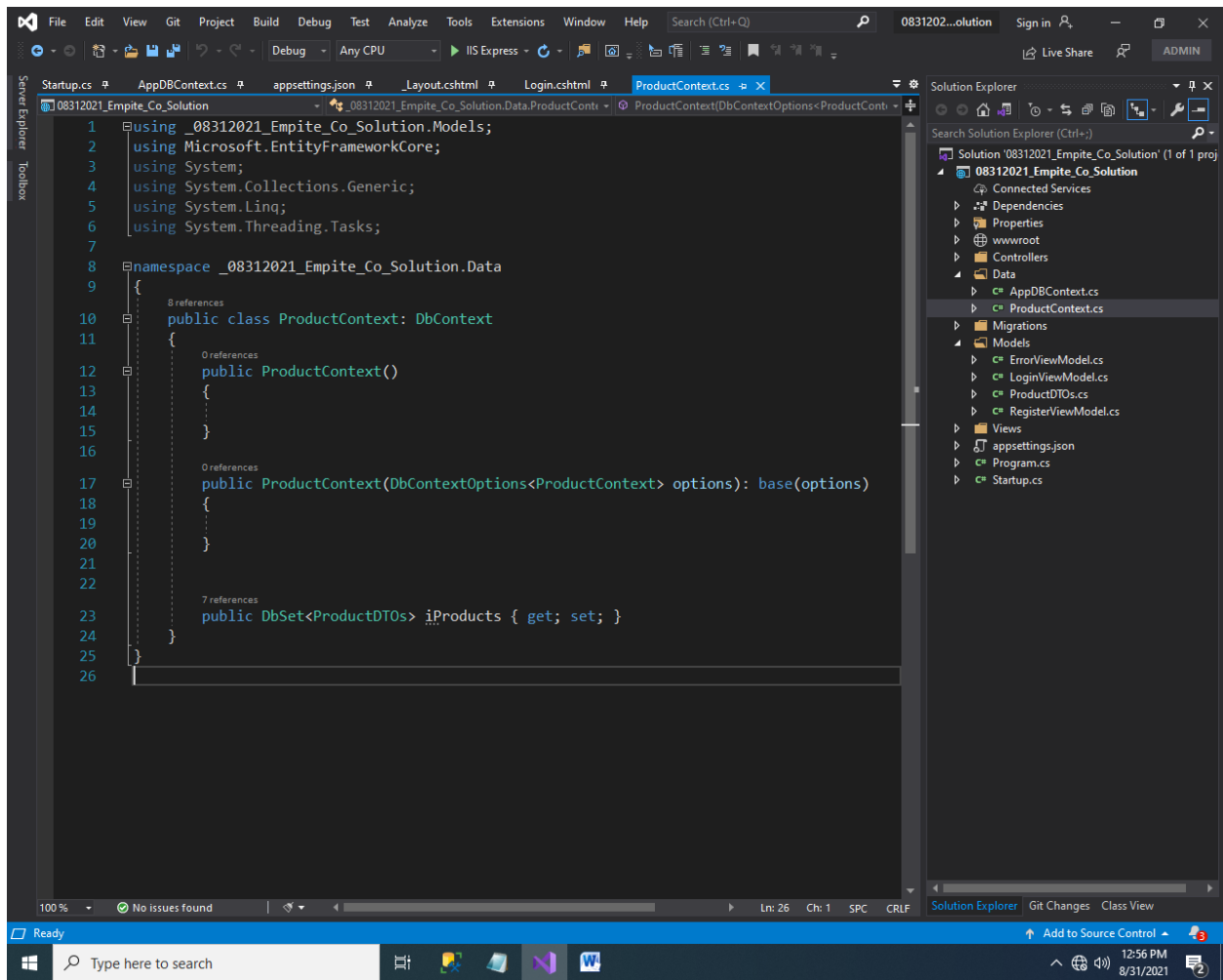
I have used EF core for create this,

This contain 2 context classes

- ProductContext
- AppDBContext

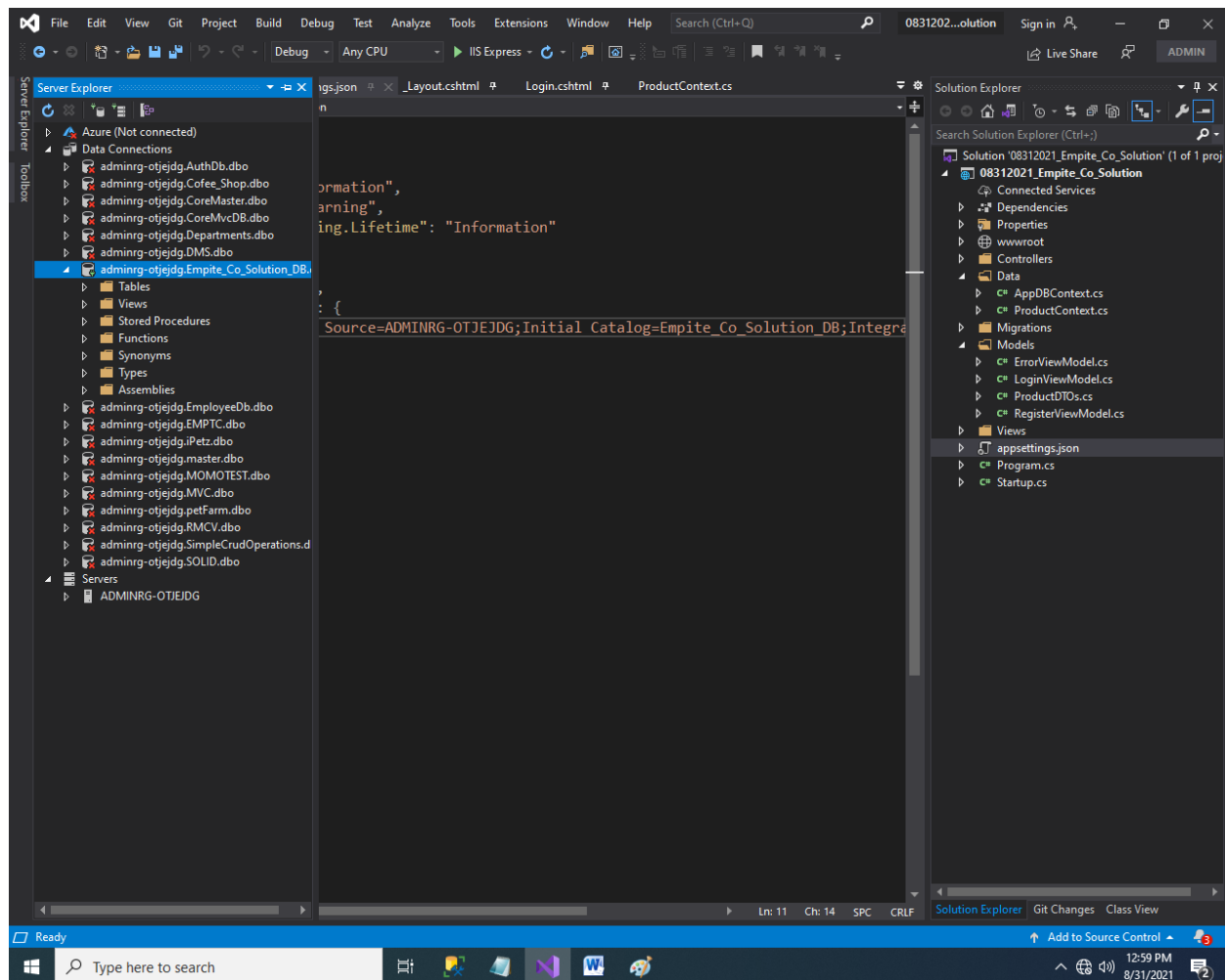
ProductContext:

That is used for initializing default constructor and added parameterized constructor with data set,

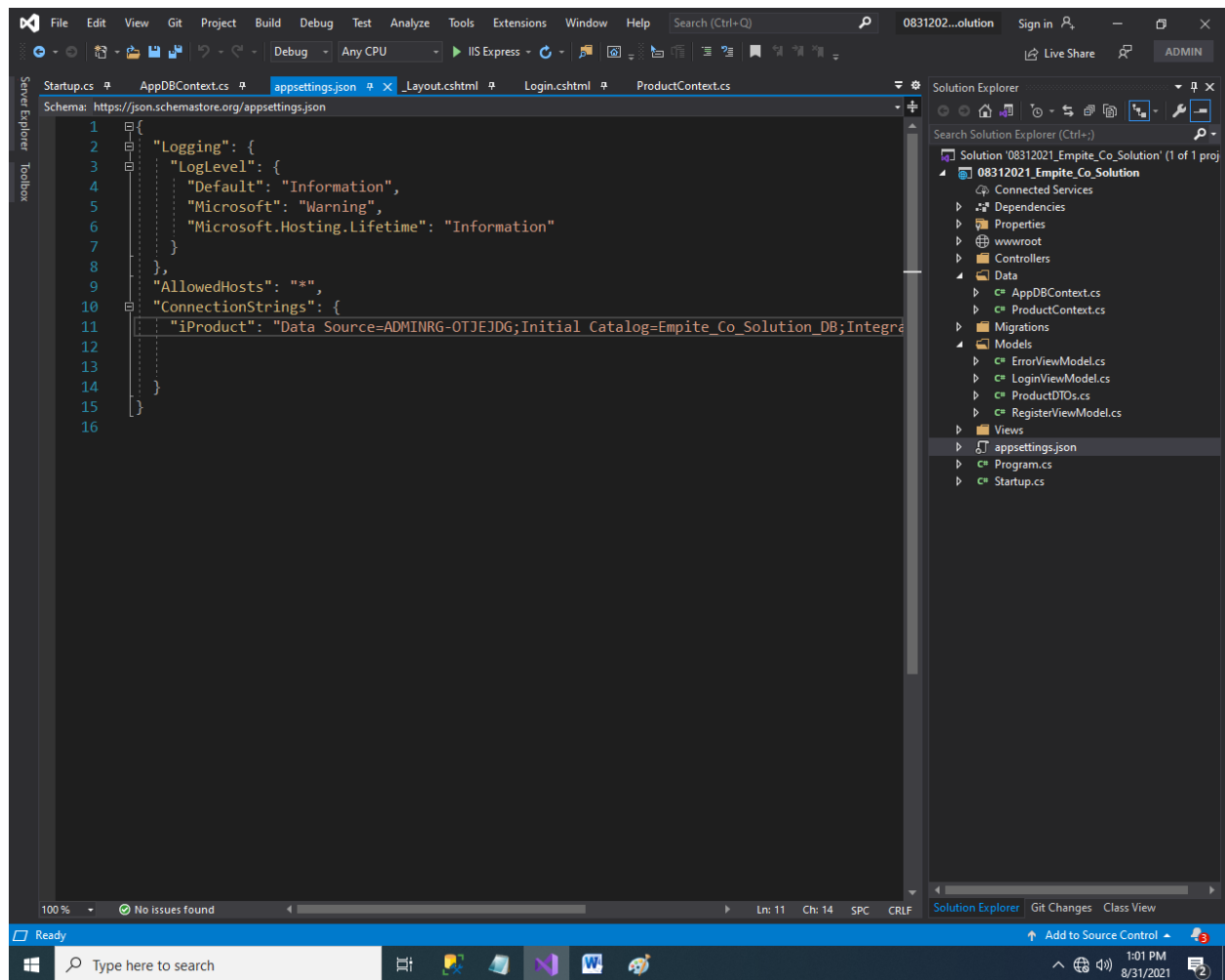


Here “iProducts ” is the data table that will created for my application,

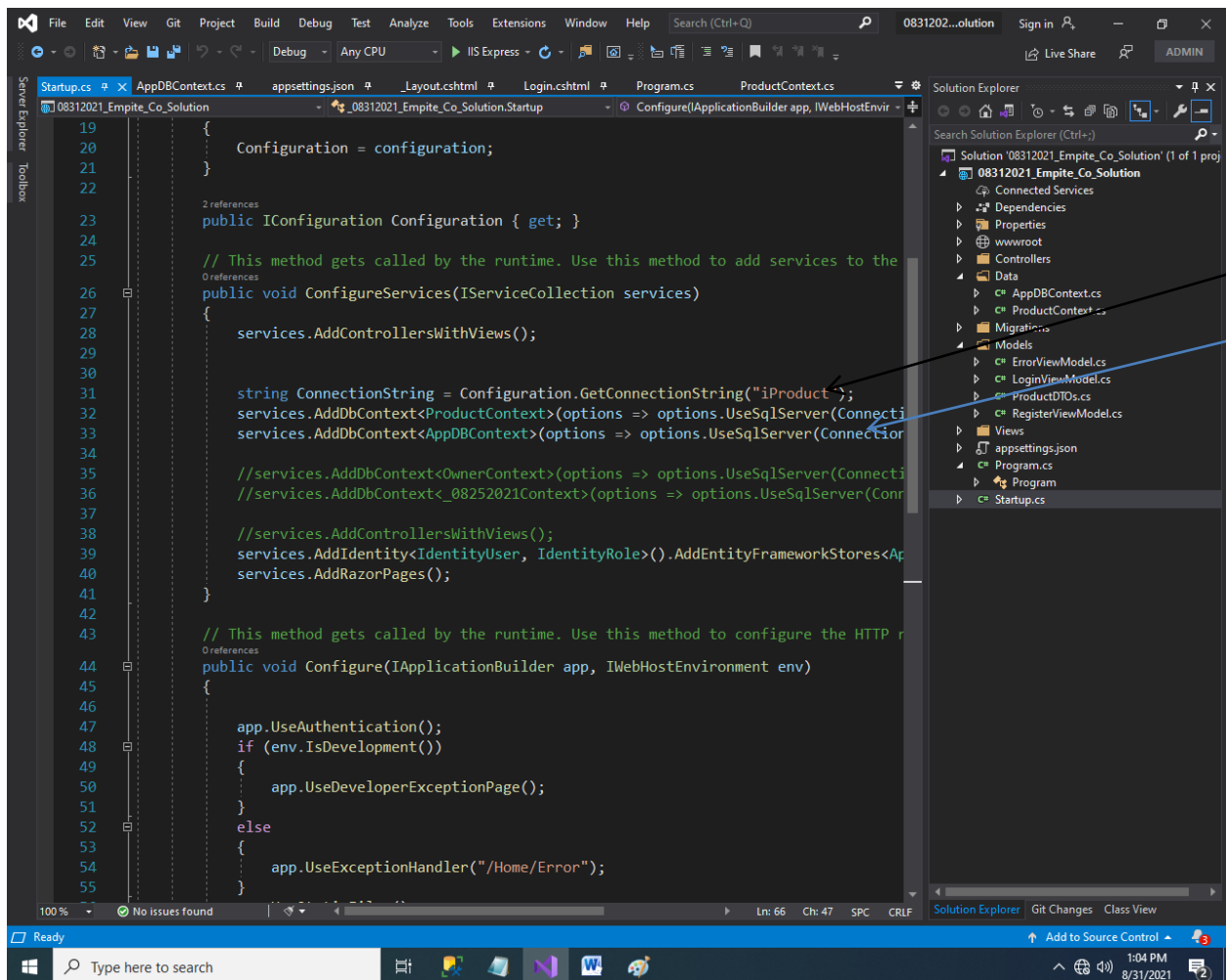
Then I have walked through server explore and connected my SQL server management studio data base to get connection and connection string...



In this case I have already created a data base called “Empite\_Co\_Solution\_DB”,  
After that,  
I had added connection string to “appsettings.json” to set the connectivity  
between my application and data base server.



After that, I have walked through “Startup.cs” to set the connectivity and migration, (Via context)

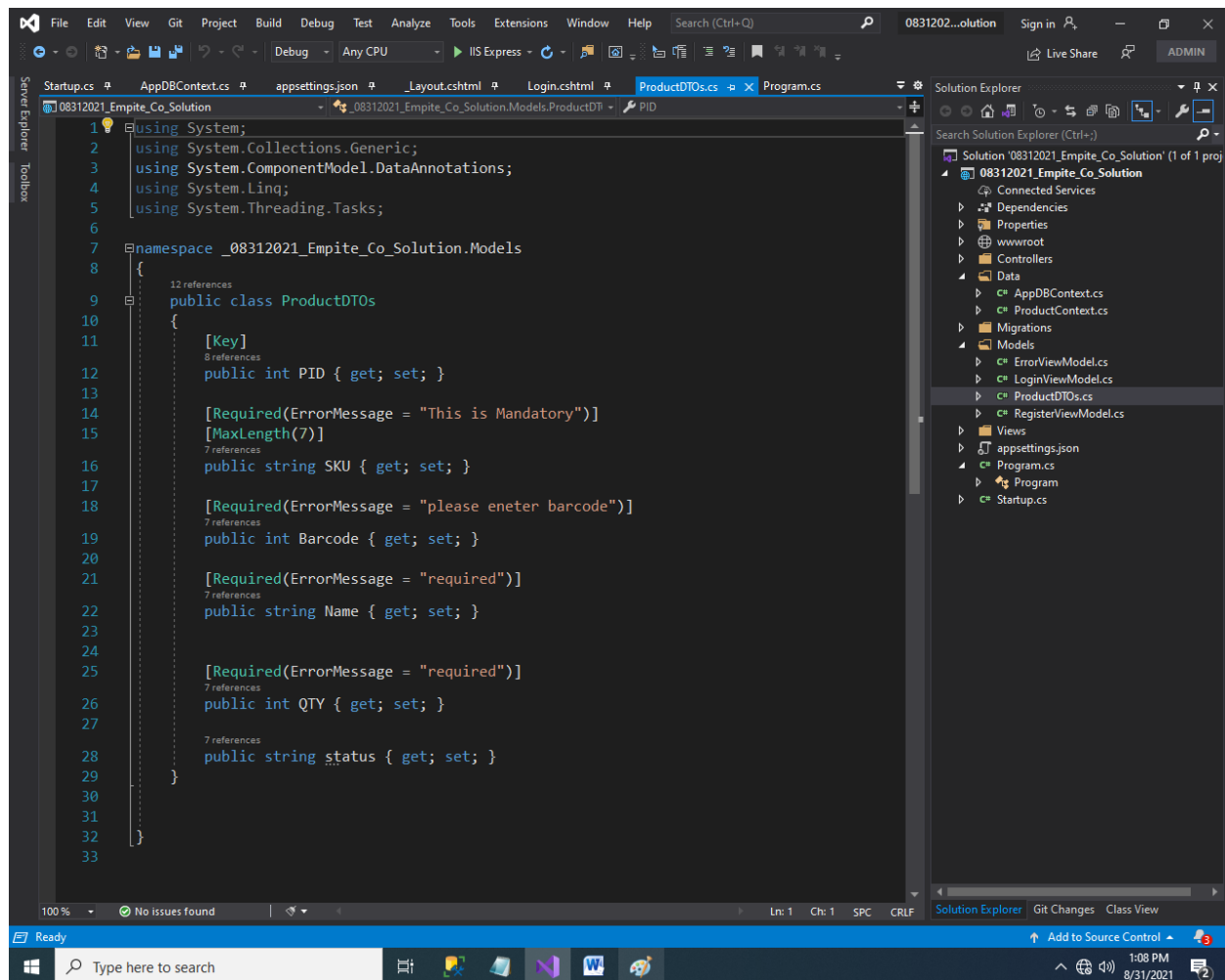


In arrowed area I have initiated connection string to data base and also added my first context (ProductContext) to set the migration.

My next movement was create data table for my Model class in Data base,

As I already mentioned that I have created a folder named “Models” inside that I have created a class called “ProductDTOs.cs”

Here I have added my simple code,



In here, I have added 6 properties and I have used Built in support “Annotations” as well for required fields, I have made some fields as compulsory and also added limitation for string max length for some fields too.

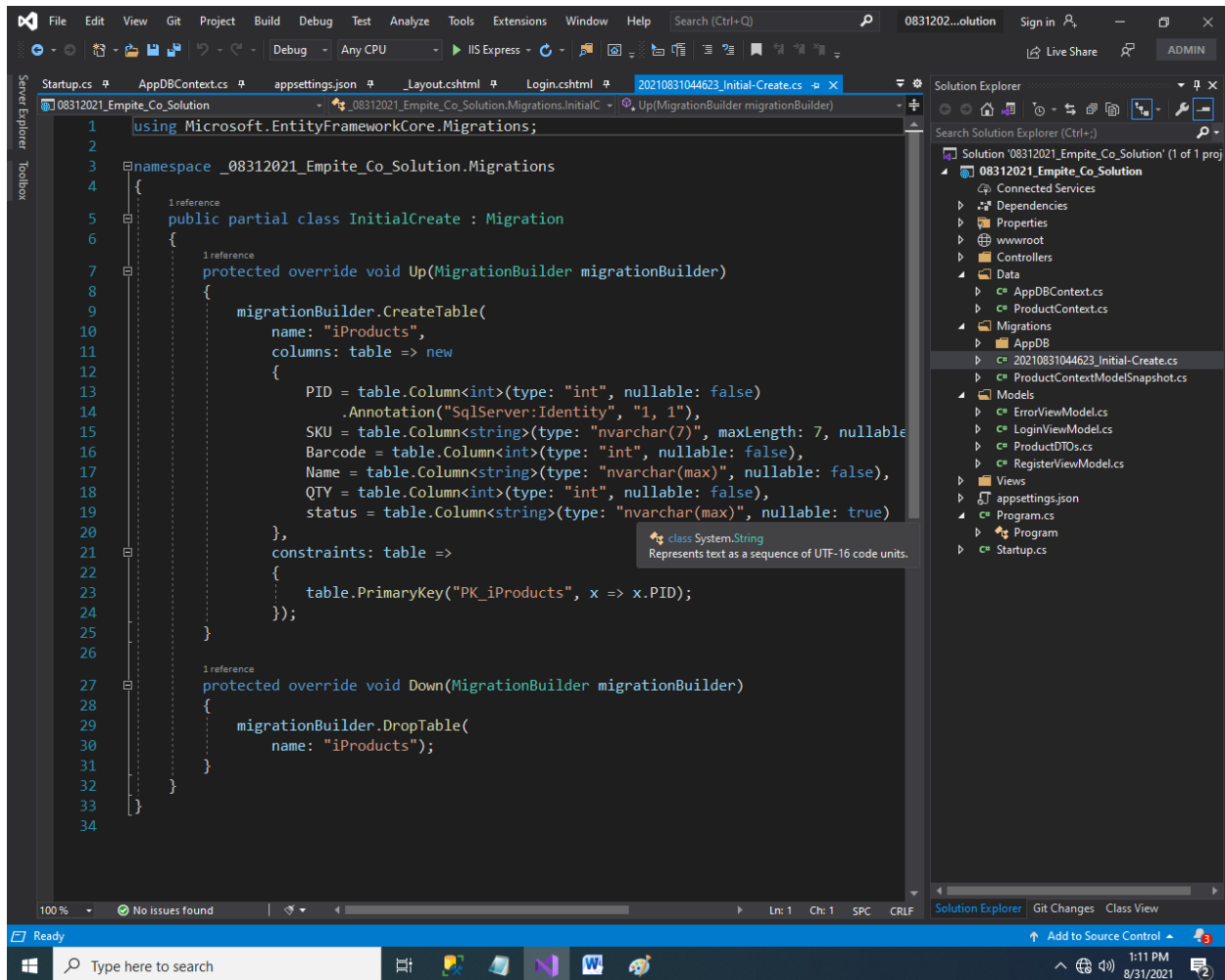
After that I have create data table using “Migration builder”

I walked through Nutget Package manager

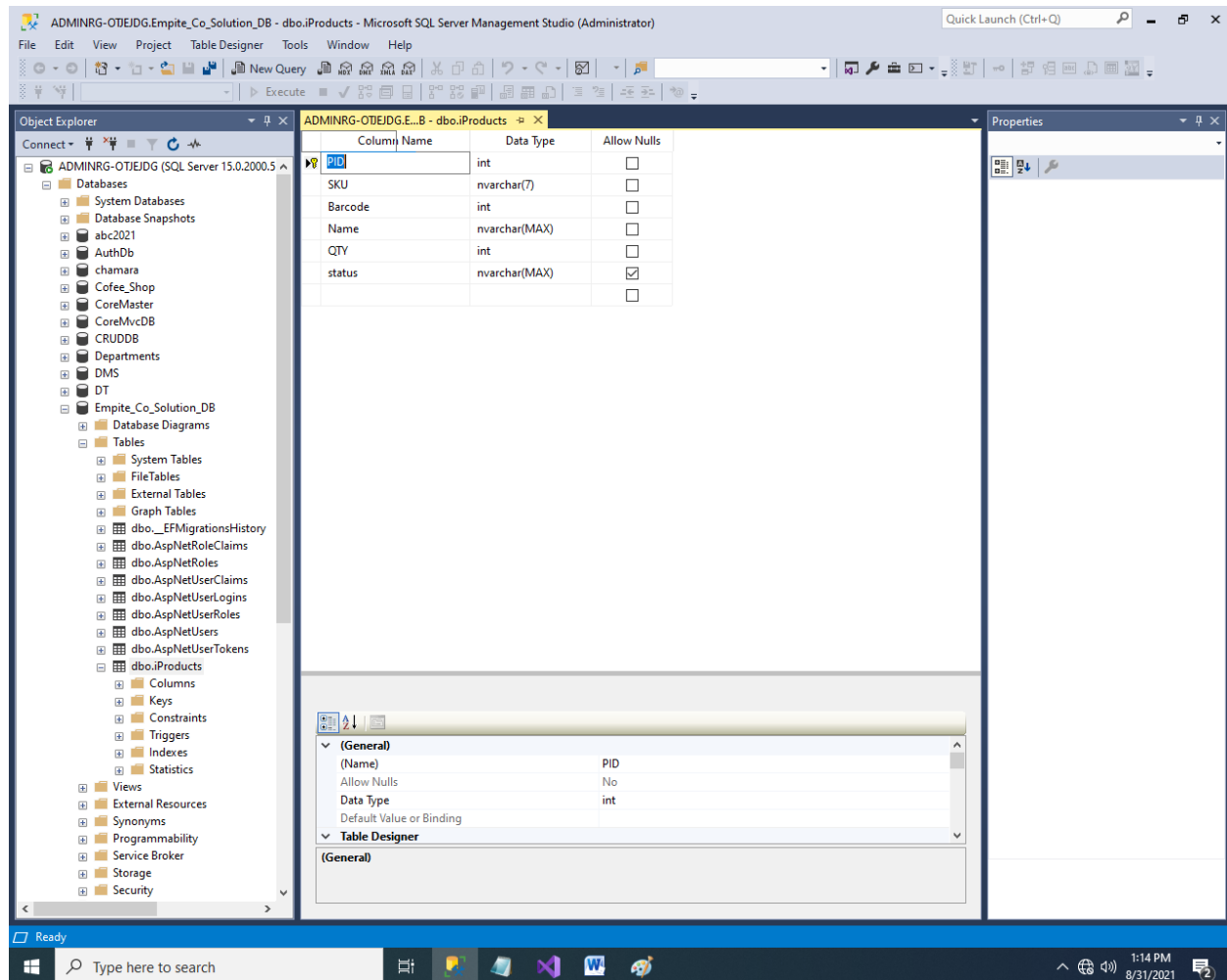
Then

Add-Migration "Initial-Create" //here I added Migration for database.

Update-Database //that make sure that my initial migration was successful or not.



After that I have verified that it has successful,

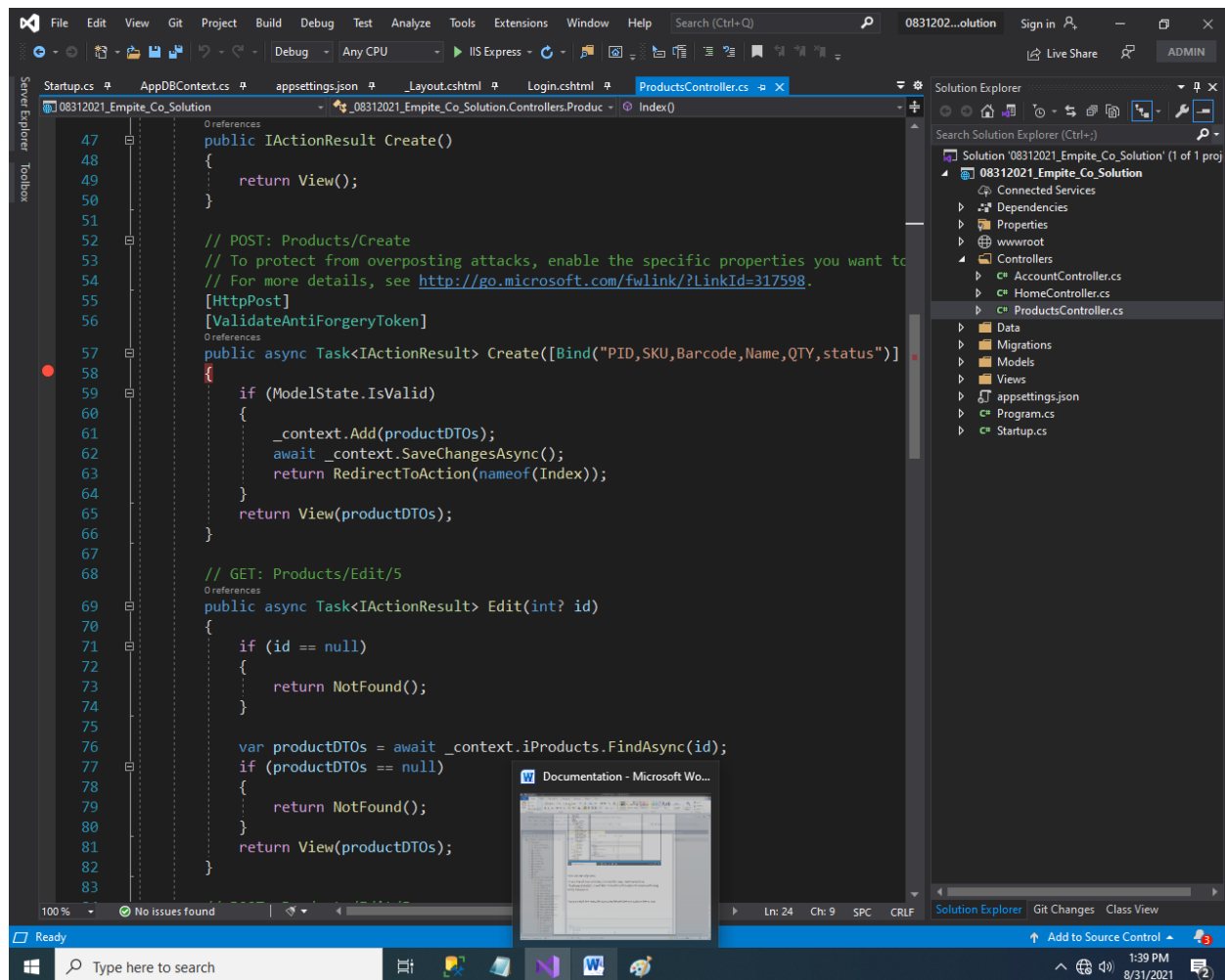


After add this migration,

I have created new controller, in controller class, I have named it as “ProductsController”, I used MVC controller with Read/write actions with using entity framework,

I have created this controller by proving Model class and context class as well.

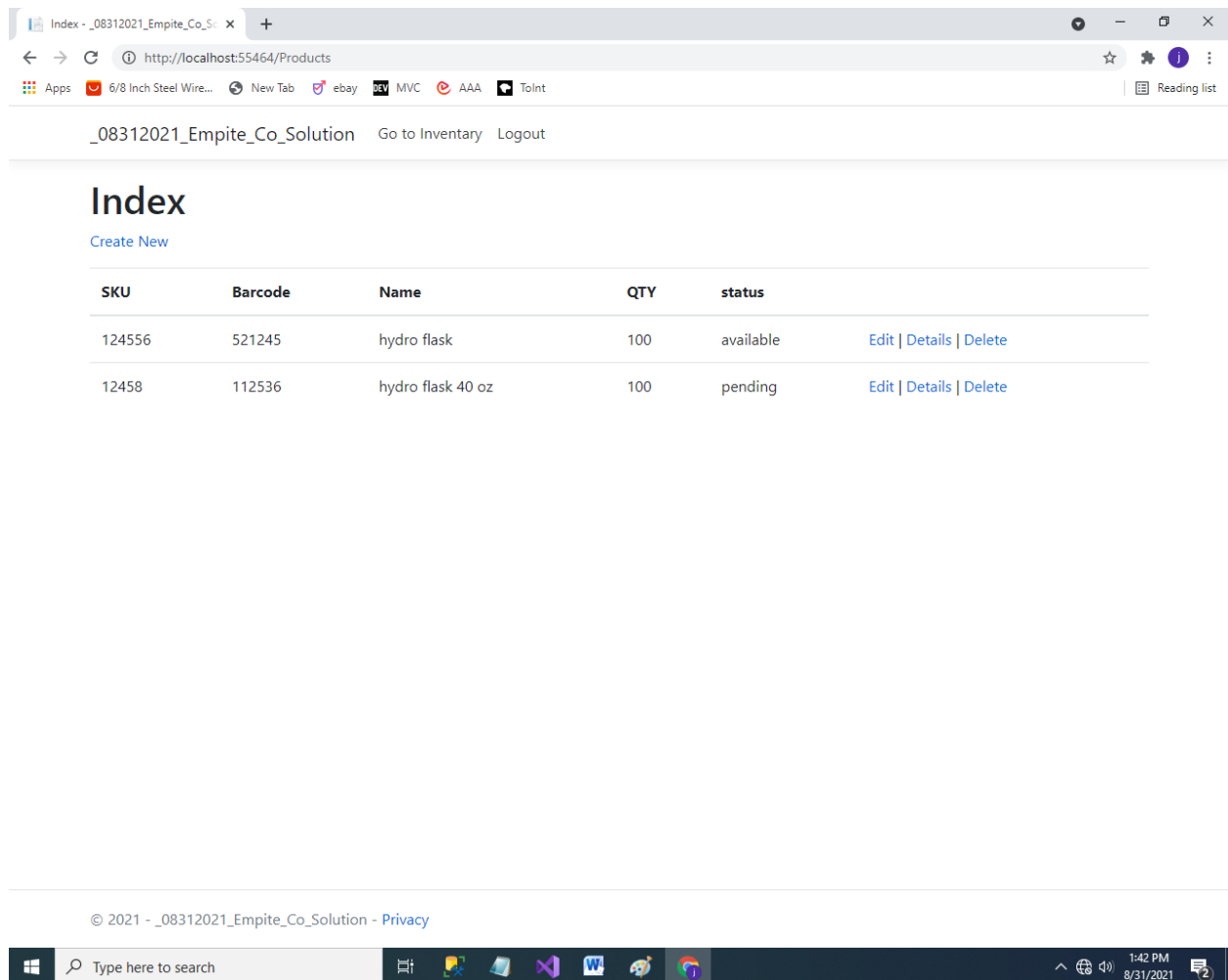




Since I have used scaffolding here, automatically VS has created view for me,

So it has view for

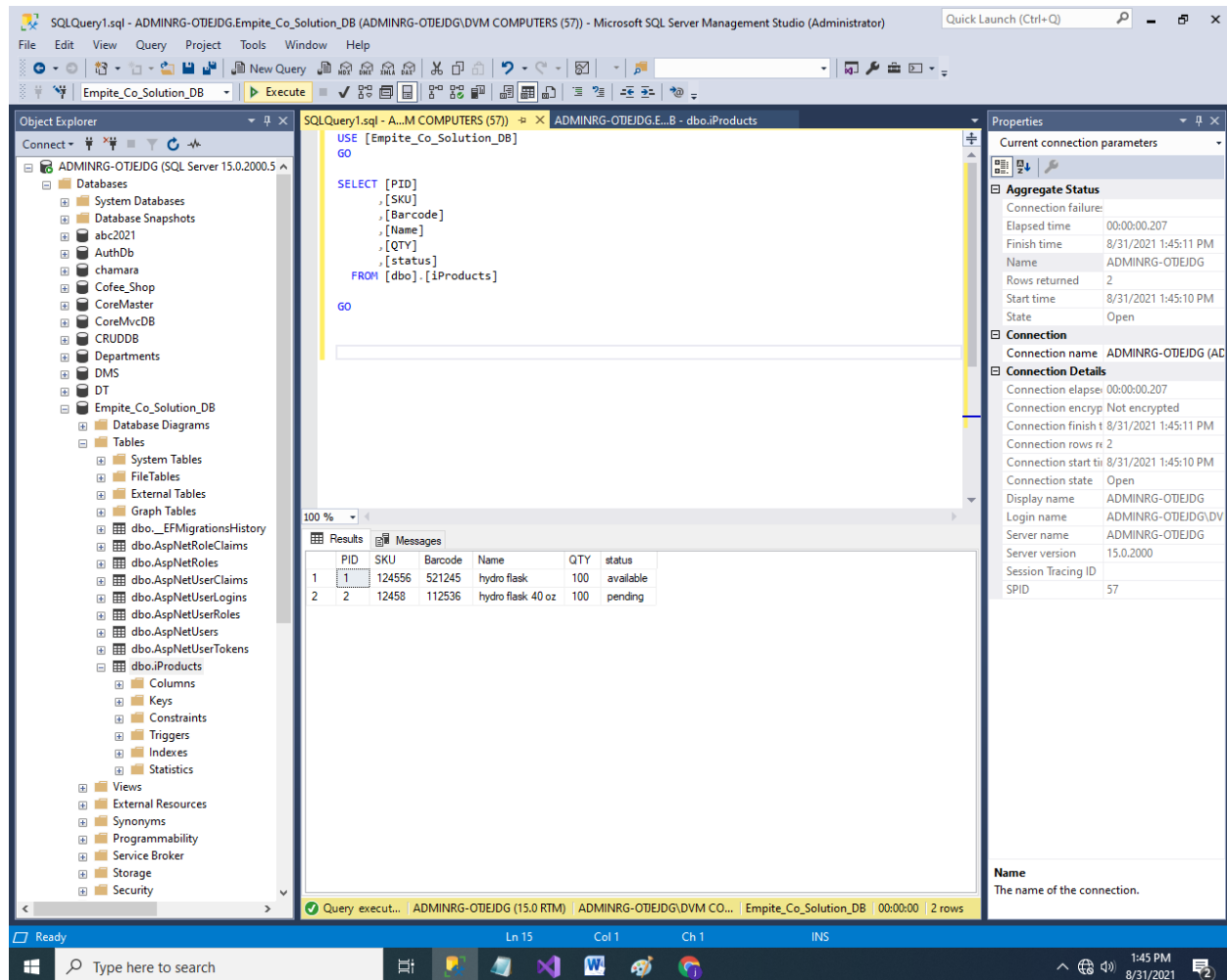
1. CREATE
2. DETAILS
3. INDEX
4. DELETE
5. EDIT



So far it is working perfectly,

By using this it can POST a product, update a product (HTTPPUT), view and delete as well,

I have checked the application from data base terminal and data base has also updated successfully...

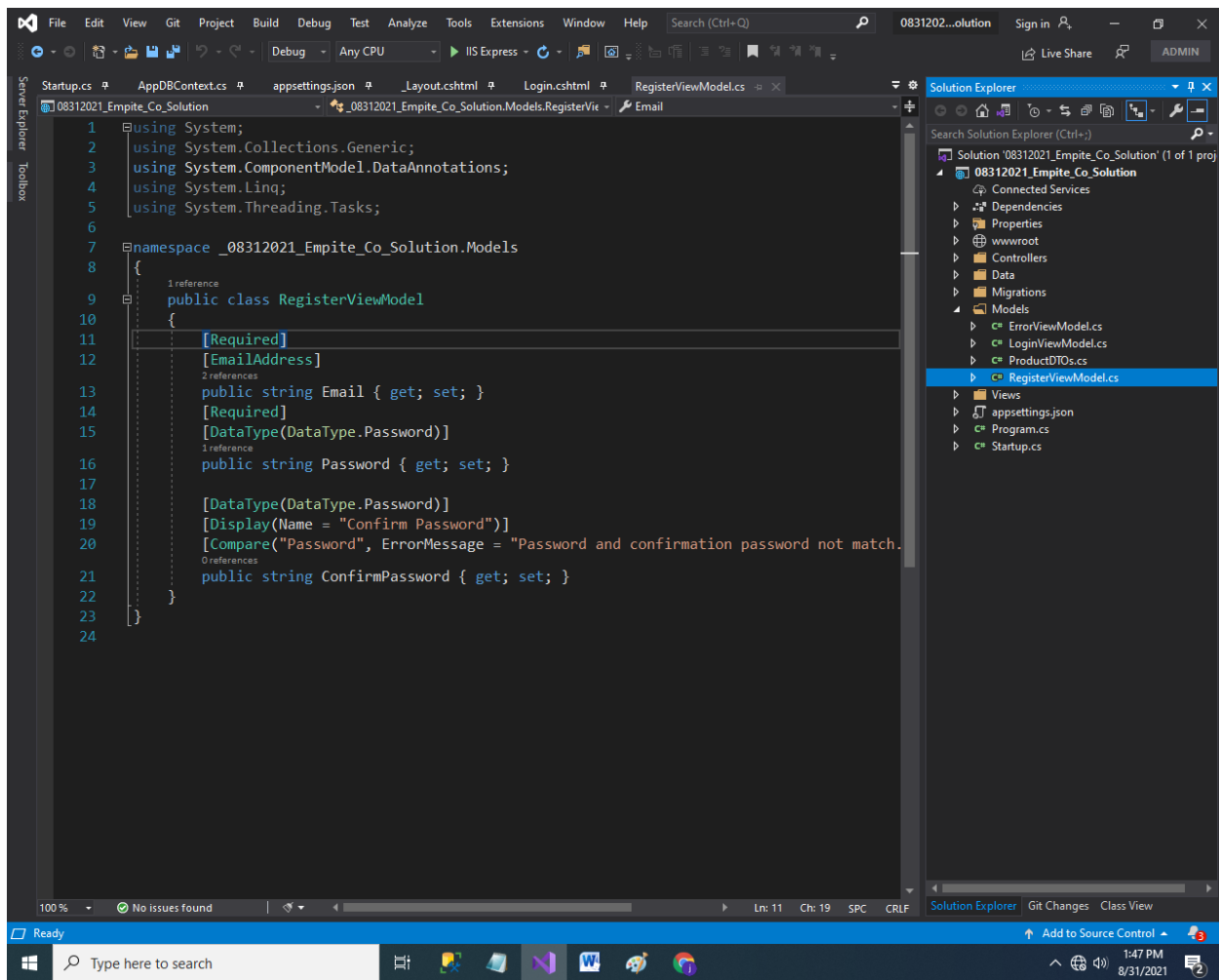


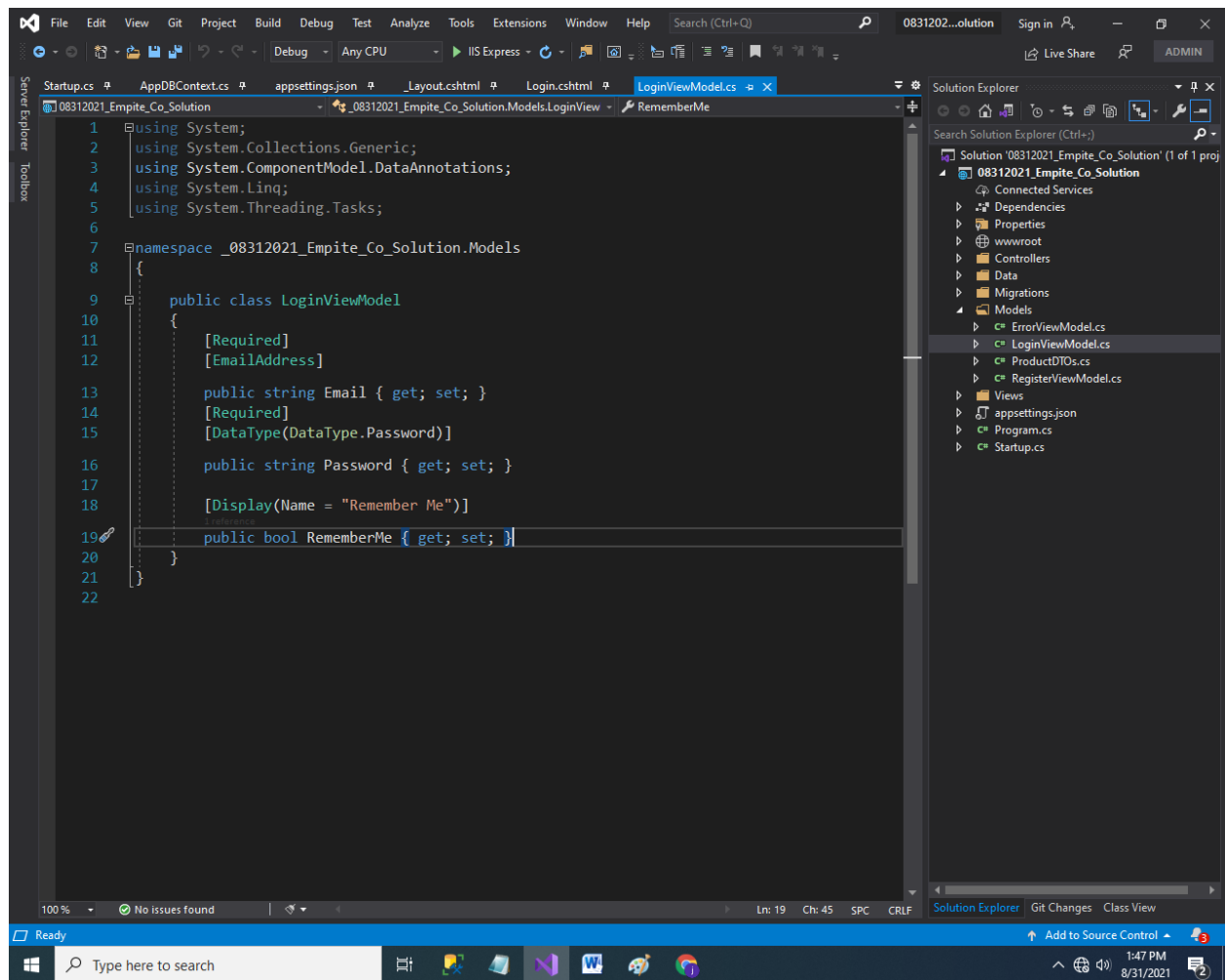
So I have created very basic application using ASP.NET core MVC.

Then my next task is implement login technique and integrates this will current project,

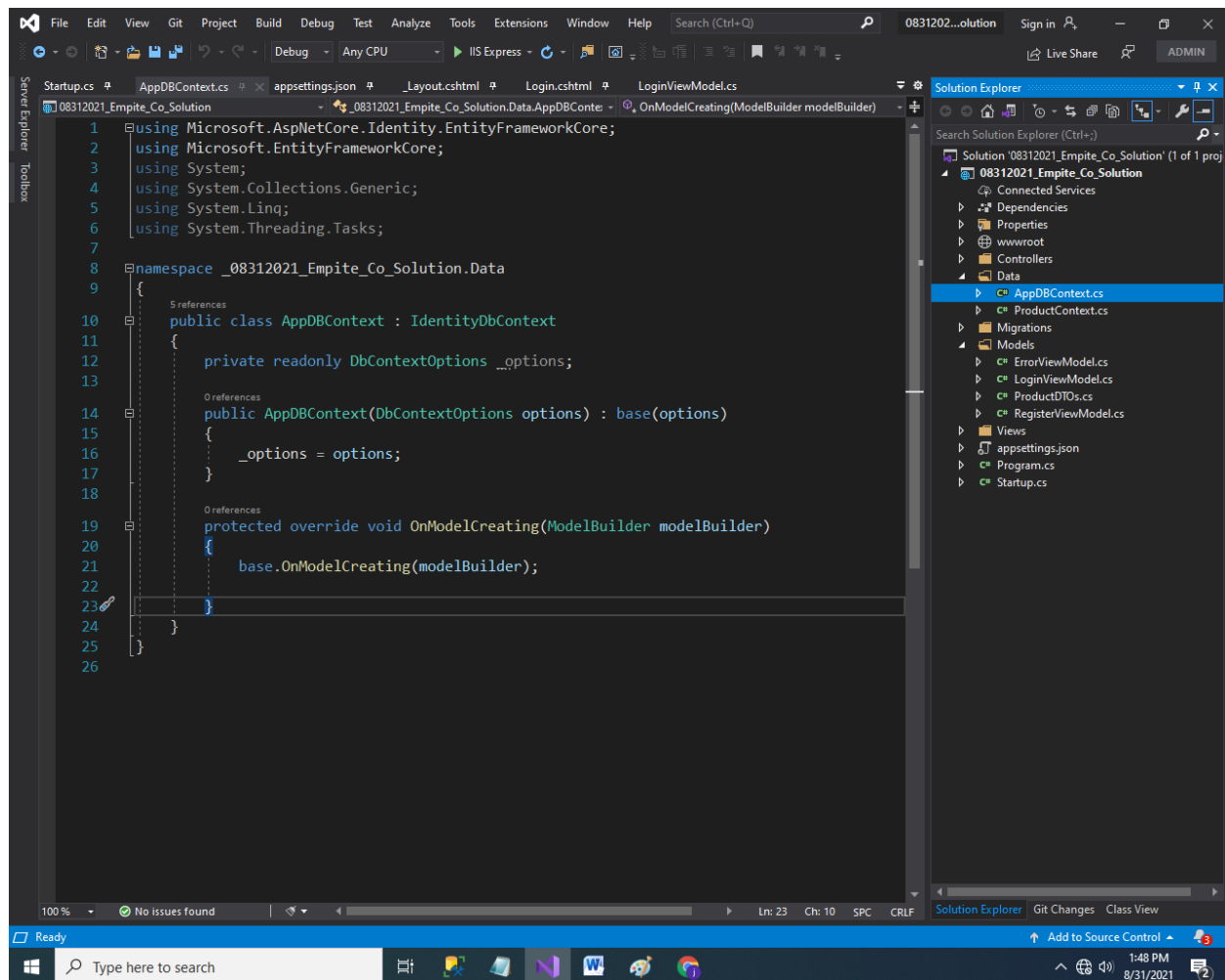
Here below I have showed the method/way I used.

I have created two view models for register and login,





Then I created another context class for this integration.



After that I have referenced this context also “Startup.cs” to avoid conflicts between namespaces in project.

After that I did another migration using migration builder,

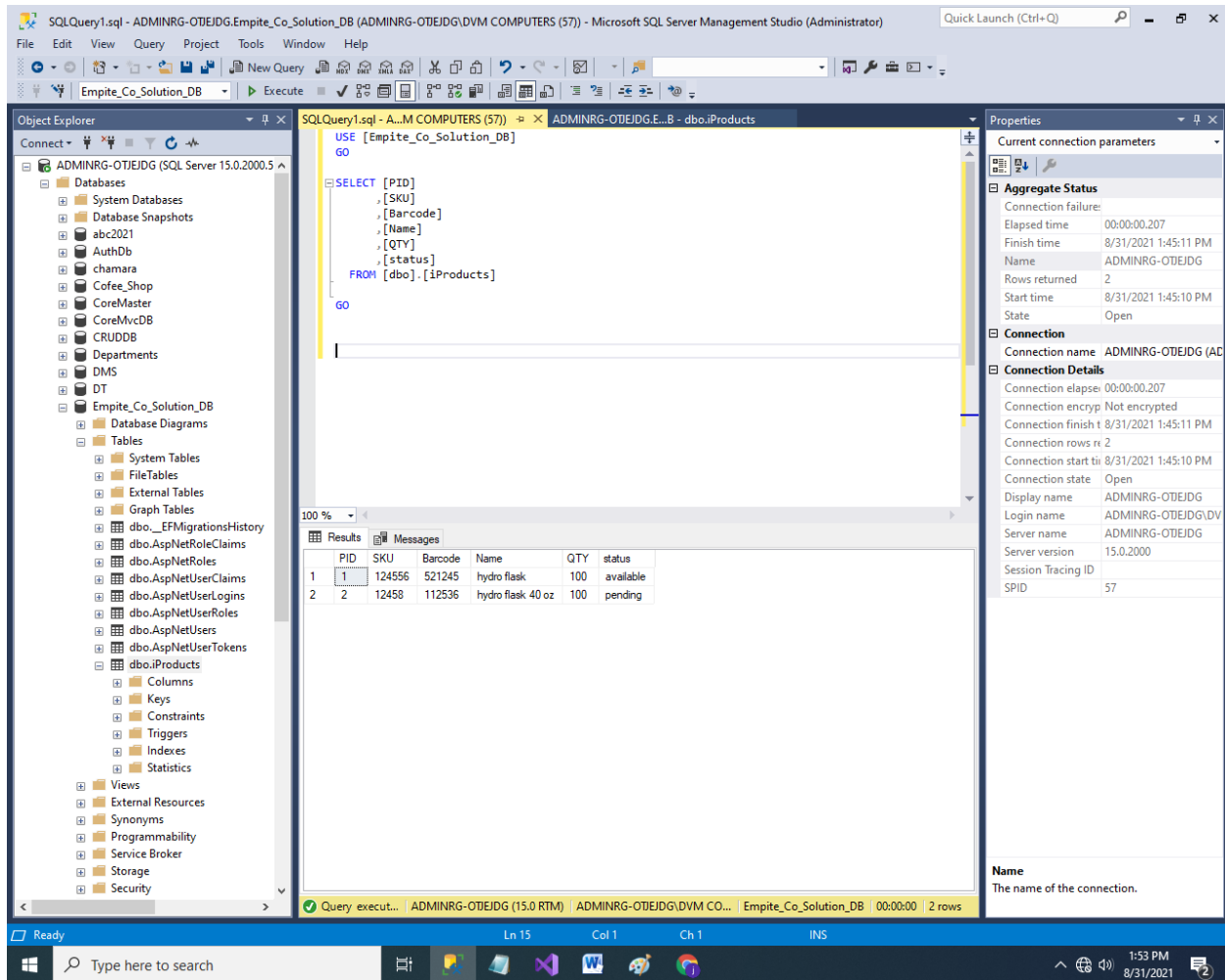
Since I have already used one migration this time I Used,

Add-Migration “Second-create” -context AppDbContext // here I have specified which context need to use at this time,

Then after I have created table also using,

Update-database” -context AppDbContext

That would worked properly,



This time I used empty controller scheme to implement my project.

So I have added an empty controller named “AccountController”.

I have used below methods to generate my code.

I have used built in support “`private readonly UserManager<IdentityUser> _userManager;`  
`private readonly SignInManager<IdentityUser> _signInManager;`”

to implement my code.

Then I written HTTPPost method to create register for new user,

```
[HttpPost]
public async Task<IActionResult> Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = new IdentityUser
        {
            UserName = model.Email,
            Email = model.Email,
        };

        var result = await _userManager.CreateAsync(user, model.Password);

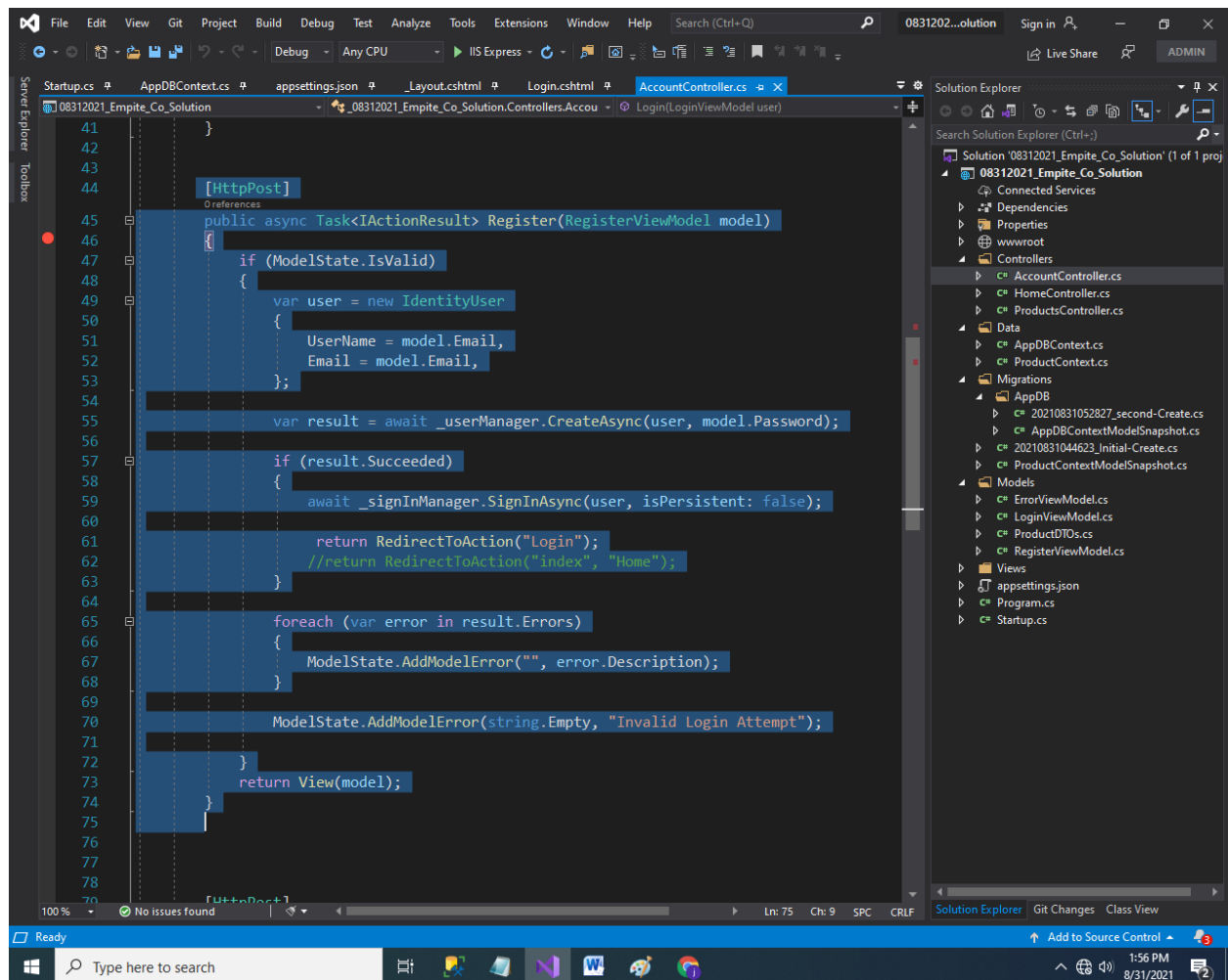
        if (result.Succeeded)
        {
            await _signInManager.SignInAsync(user, isPersistent: false);

            return RedirectToAction("Login");
            //return RedirectToAction("index", "Home");
        }

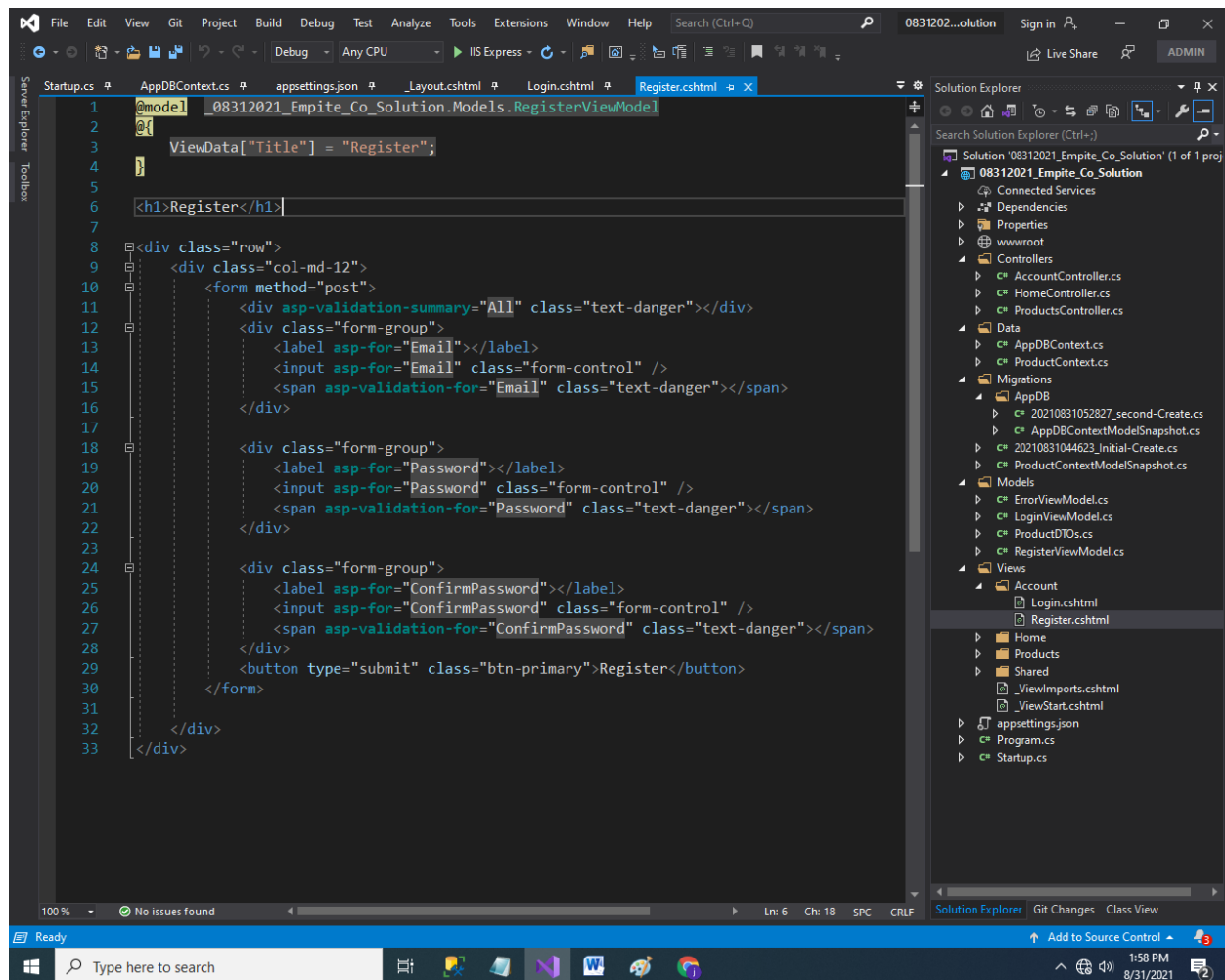
        foreach (var error in result.Errors)
        {
            ModelState.AddModelError("", error.Description);
        }

        ModelState.AddModelError(string.Empty, "Invalid Login Attempt");
    }
    return View(model);
}
```





After that I have created a new RAZOR view for Register method, by giving model class and context class as well.



Now I need to show return view for newly created Register method, So I have created view in controller,

```
public IActionResult Register()
{
    return View();
}
```

After running this,

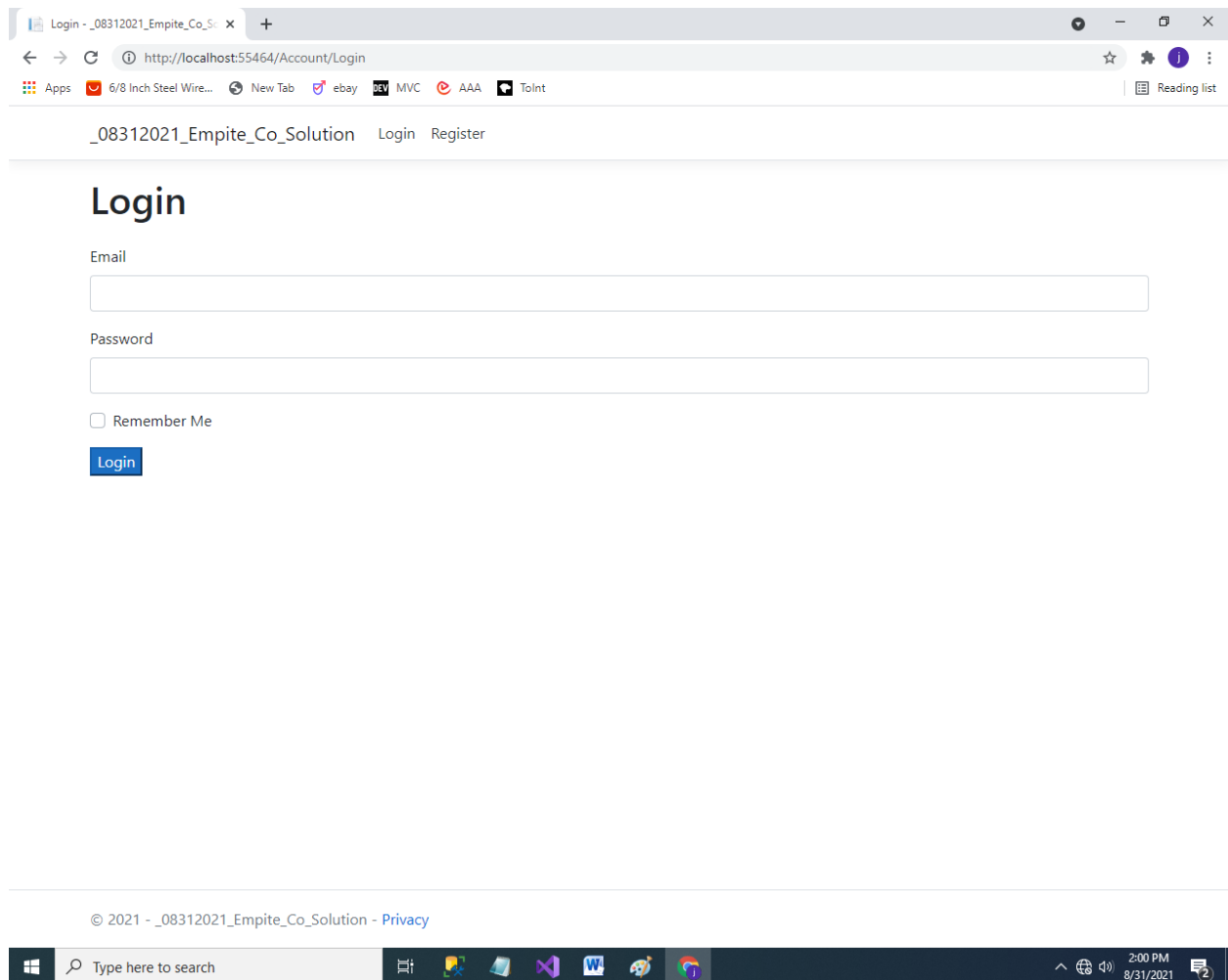
## Register

Email

Password

Confirm Password

Register



So now application is working properly,

Then I have added logout function too,

```
public async Task<IActionResult> Logout()
{
    await _signInManager.SignOutAsync();

    return RedirectToAction("Login");
}
```

After that I have modified \_Layout.cshtml page to authorized the user

```
@if (User.Identity.IsAuthenticated)
{
    <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="Home" asp-action="Privacy">Privacy</a>
        </li>*@

        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-
controller="Products" asp-action="Index">Go to Inventory</a>
        </li>

        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-
controller="Account" asp-action="Logout">Logout</a>
        </li>

    </ul>
}
else
{

```

(I took this code from stack overflow)

Then finally I added `[Authorize]` notation hence this will facilitate my application, proper login sequence.

Finally I have bound the product page controller via asp controller view.

Code is

```

li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-
controller="Products" asp-action="Index">Go to Inventory</a>
        </li>
management page.
//this will walk user to inventory

        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-
controller="Account" asp-action="Logout">Logout</a>
        </li>

```

So finally this application work like follow,

When the beginning,

1. User need to log into system
2. If the user does not have login credential he/she need to create user name and password, then only he/she can access system,
3. Once after successful registration, a user will redirected to login page, then after it asking to provide login credential that he/she used to register,
4. If they are matched with stored he/she can access the system,
5. Otherwise nope.

Real screen shots

The screenshot shows a web browser window with the address bar displaying 'http://localhost:55464/Account/Login'. The browser's taskbar at the bottom shows various open applications including '6/8 Inch Steel Wire...', 'New Tab', 'ebay', 'MVC', 'AAA', and 'Toint'. The login page itself has a header with the text '\_08312021\_Empite\_Co\_Solution' and links for 'Login' and 'Register'. The main heading is 'Login'. Below this, there are two input fields: 'Email' with the value 'udanapeiris1995@gmail.com' and 'Password' with masked characters '.....'. A checkbox labeled 'Remember Me' is present below the password field. A blue 'Login' button is at the bottom of the form. At the very bottom of the browser window, a footer contains the text '© 2021 - \_08312021\_Empite\_Co\_Solution - Privacy'. The Windows taskbar at the bottom of the screen shows the search bar, taskbar icons, and the system clock indicating 2:11 PM on 8/31/2021.

Login - \_08312021\_Empite\_Co\_S...

http://localhost:55464/Account/Login

Apps 6/8 Inch Steel Wire... New Tab ebay MVC AAA Toint Reading list

\_08312021\_Empite\_Co\_Solution Login Register

## Login

Email

udanapeiris1995@gmail.com

Password

.....

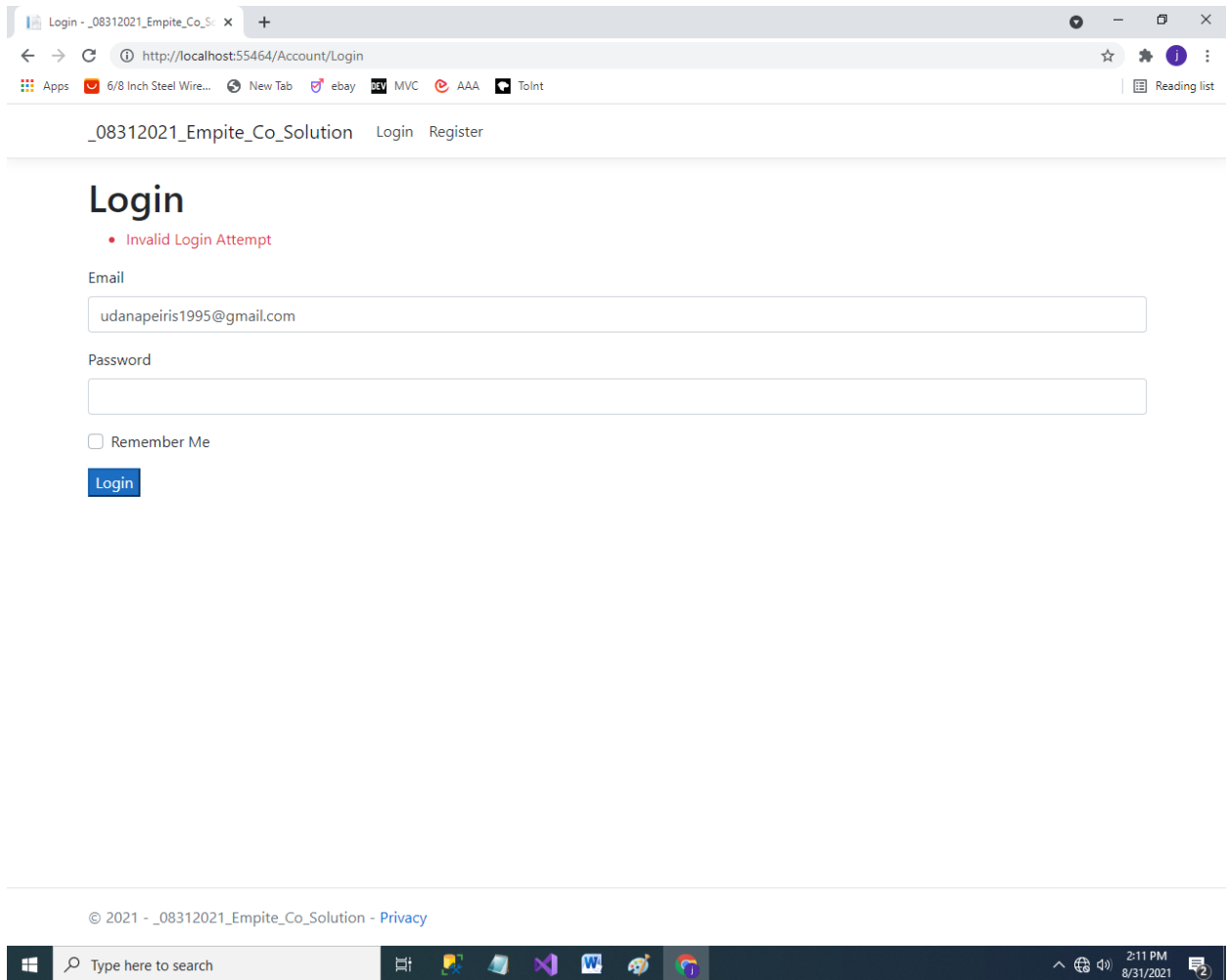
☐ Remember Me

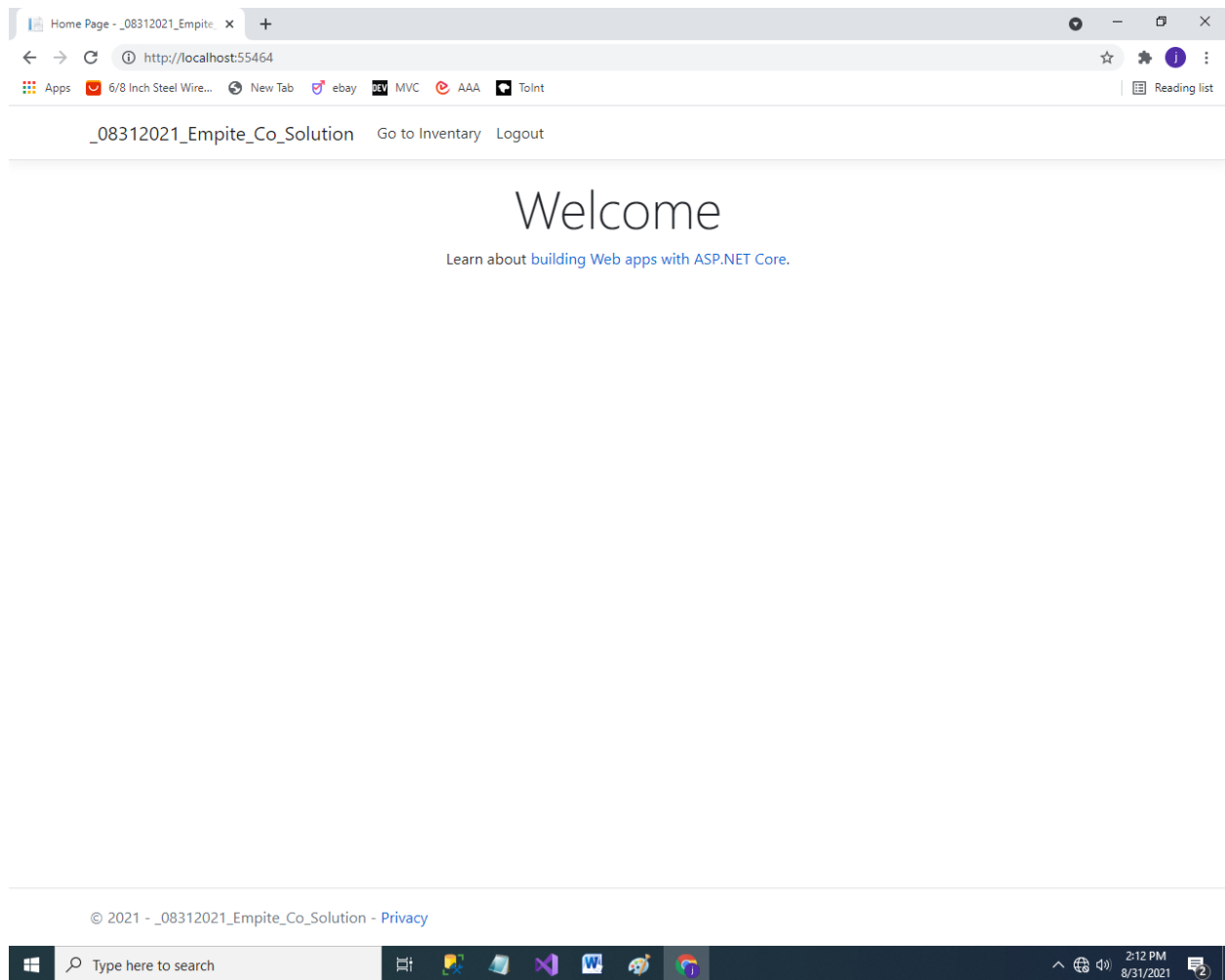
Login

© 2021 - \_08312021\_Empite\_Co\_Solution - Privacy

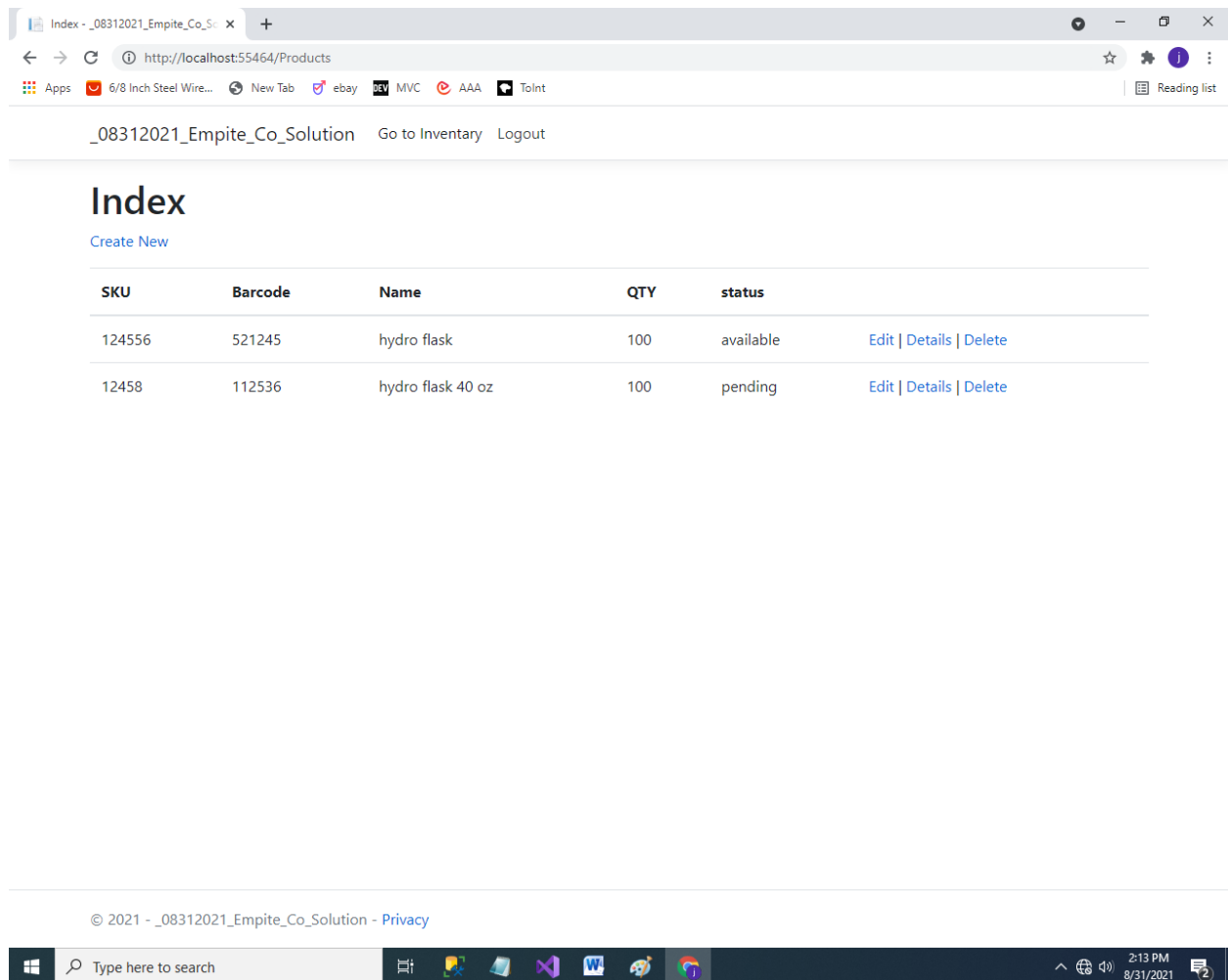
Type here to search

2:11 PM 8/31/2021









## Limitations!

I have made this project with single user with all administrative privileges,