PREPARATION

Create a folder in the Apps folder with name of the current exam.  
Inside, create ProjectName.Web cs project with version 2.2 of .netcore  
Create two class library of .netstandard v.2.0 with name ProjectName.Data and ProjectName.Services

Create references between projects like this: Services 🡪 Data / Web 🡪 Services, Data,   
SIS.MvcFramework / Data 🡪 Microsoft.EntityFrameworkCore.SqlServer (NuGet)

1. DATABASE

Inside Data, create ProjectNameDbContext class which inherits DbContext. Next create a new class called DbSettings and make a connection string (see Panda project), just rename the name of the   
Database with the current projectName.

Back in the DbContext class , create the two overrided void methods OnConfiguring and   
OnModelCreating (see Panda project).

Next step is creating the Database models. Create a folder called Models in   
Data and start creating classes with the requirements from the exam document.  
Add the Required attribute where needed. If there is a collection with nav. Prop. ,don’t  
forget to initialize it in the constructor of the class.

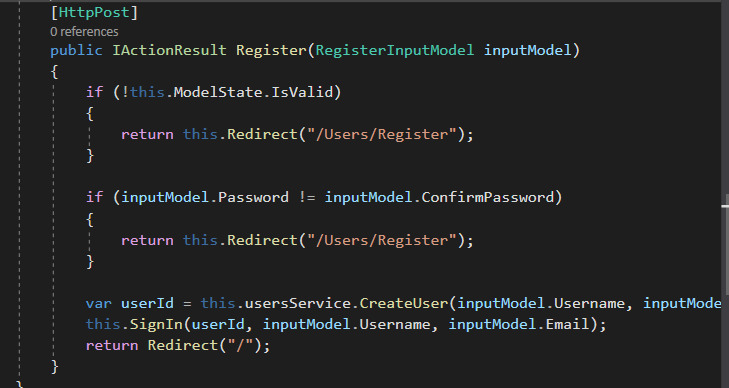
After all of the models are created, make the DbSet properties in the DbContext class.  
In OnModelCreating method, make the needed relations between the tables.

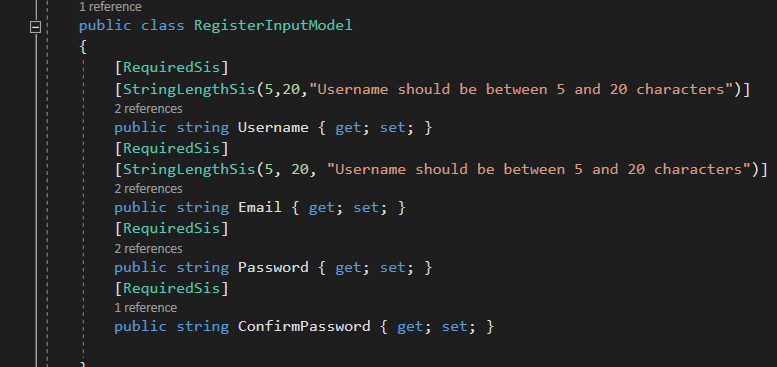
Make a StartUp class in the Web project and inherit the IMvcApplication and   
implement the class methods. This class hold all of the services that need to be register on  
a later stage. In the Configure method write this code:  
using(var db = new ProjectNameDbContext())  
{  
 db.Database.EnsureCreated();  
}  
 Test the server and delete all cookies if there are any.

Next step is to add the WebHost.Start(new Startup()) in the Main method in the Web project.

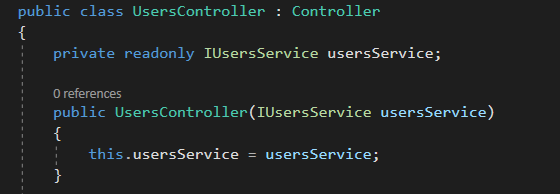
1. VIEWS AND FUNCTIONALITY

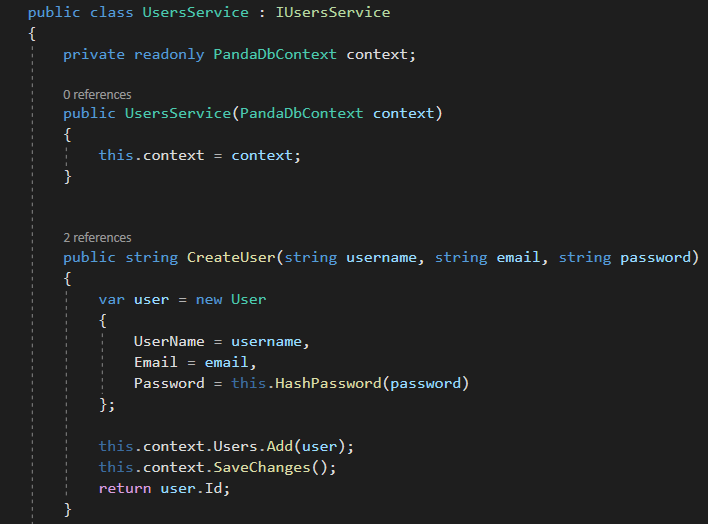
In the Web project make a Controller, Views and ViewModels folders.  
Copy all resources in view folder and make sure to check Copy Always property!

Let’s start with HomeController, it should inherit Controller base class.   
Add a public IActionResult and return the view. The methods should be with the names  
of the html files. The views with Http Get are pretty easy, let’s see a view with a Http Post request.  
  
HTTP POST User Register:  
  


In the method parameter, there is a RegisterInputModel, this is the class that need to hold the info from the view form. The properties of the class should be named exactly how they are in the view. Here is the implementation of the ViewModel class:  
  


This class is placed in the ViewModels folder. As you can see this is the place for   
validation of the properties. Use the custom validation attributes from the SIS framework.

Back to the UserController class:  


In the constructor give as a parameter the IUserService. This is an interface which should be created in the Service project. This services must be created for every controller with post method or more complex functionality. Of course they need the have a proper class that implement the interface   
functionality. Here is an example of a service class:  
  
  
  
 And they should be register in the StartUp class in the ConfigureServices method:

