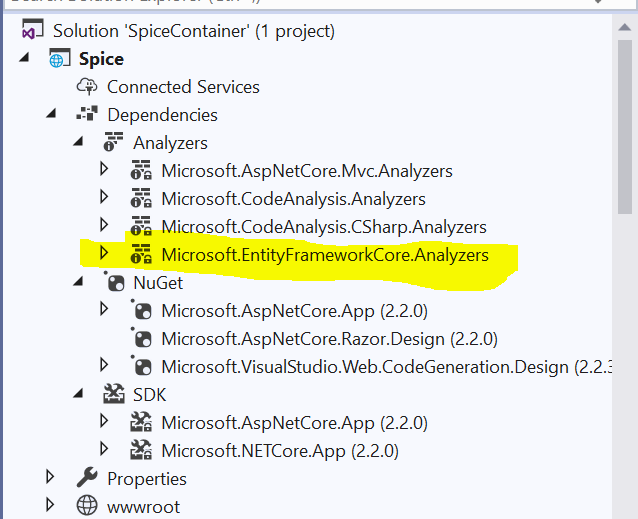
**Spice Projects Note:**

1. When Creating the Project ensure to choose asp.net core and Individual Authentication.
2. Go to startup.cs and edit the connection string with short name such as asp-Spice of data base that you want to create, highlighted below (dark)



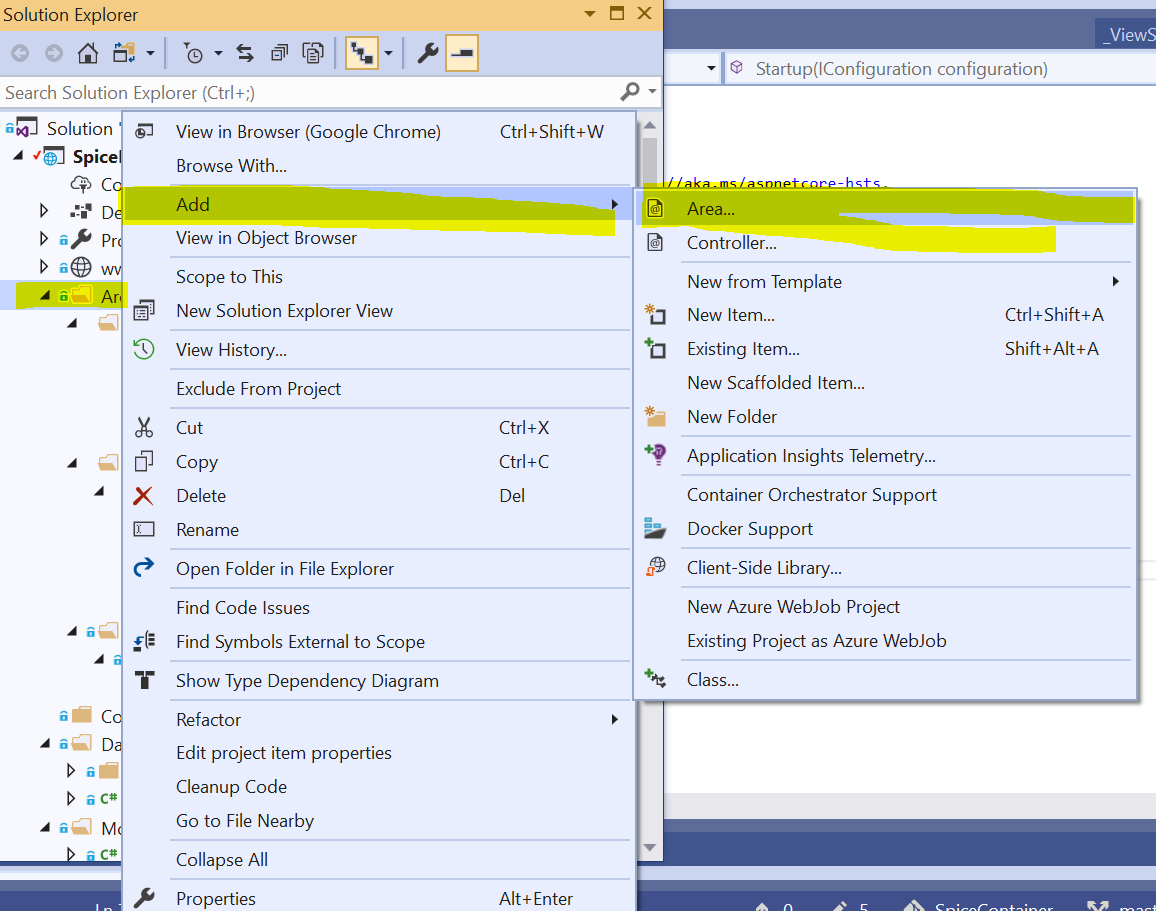
1. Open NuGet Console and type update-database, it will create database in server, If it cannot something is wrong, I have not installed any Entitiy frame work for it to work. Make sure to change the sql server name in connection, highlighted above in yellow.

The above is necessary otherwise you may experience Entitiy framework errors. These packages are installed by default. See below and it works an it works with additional packages.



Creating Separate Areas for both Admin and Customer.

1. Right click and create two areas for each one (Customer and Admin)

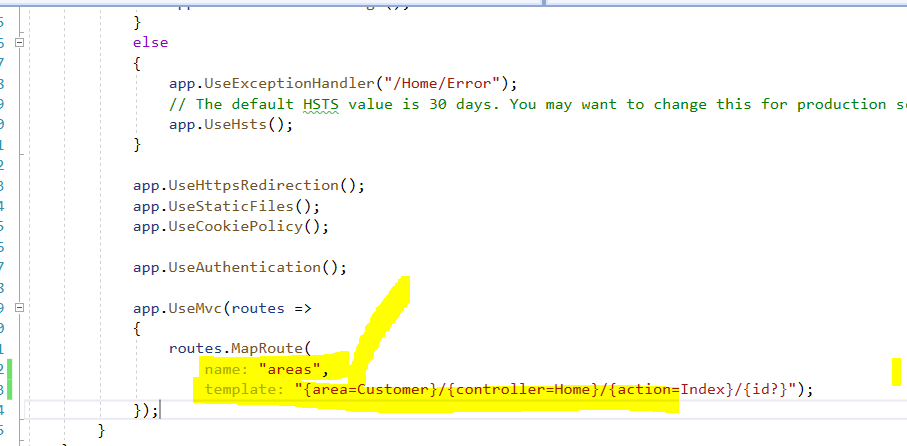


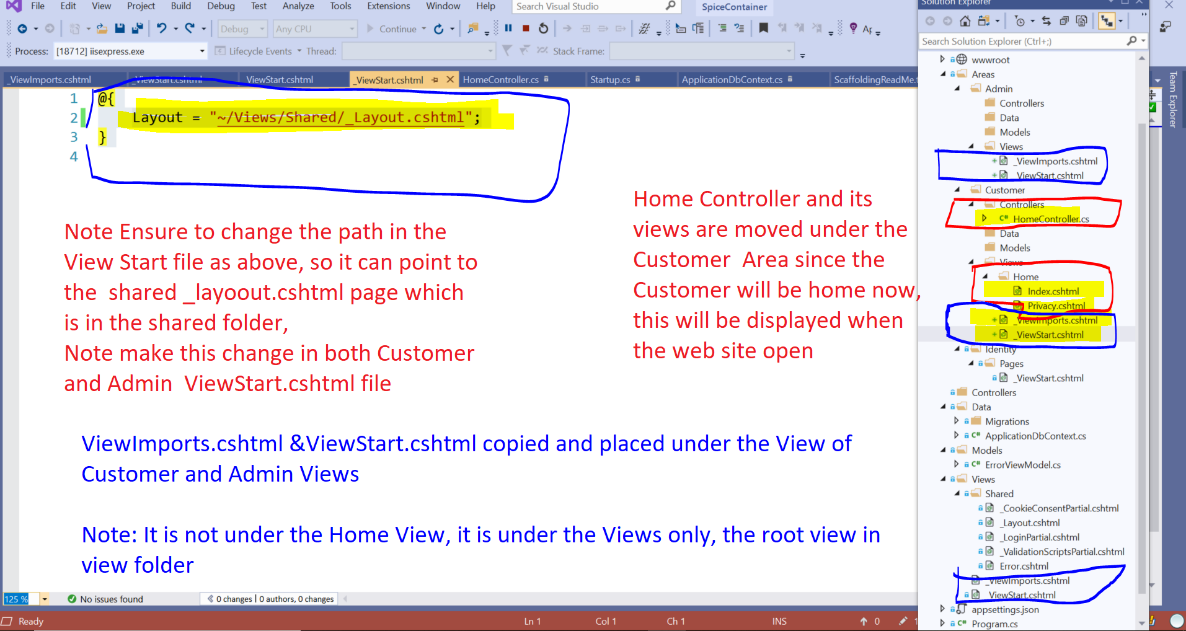
1. You can actually can make one home, make the customer, in the startup file , in the starteup file replace it with original as the following:

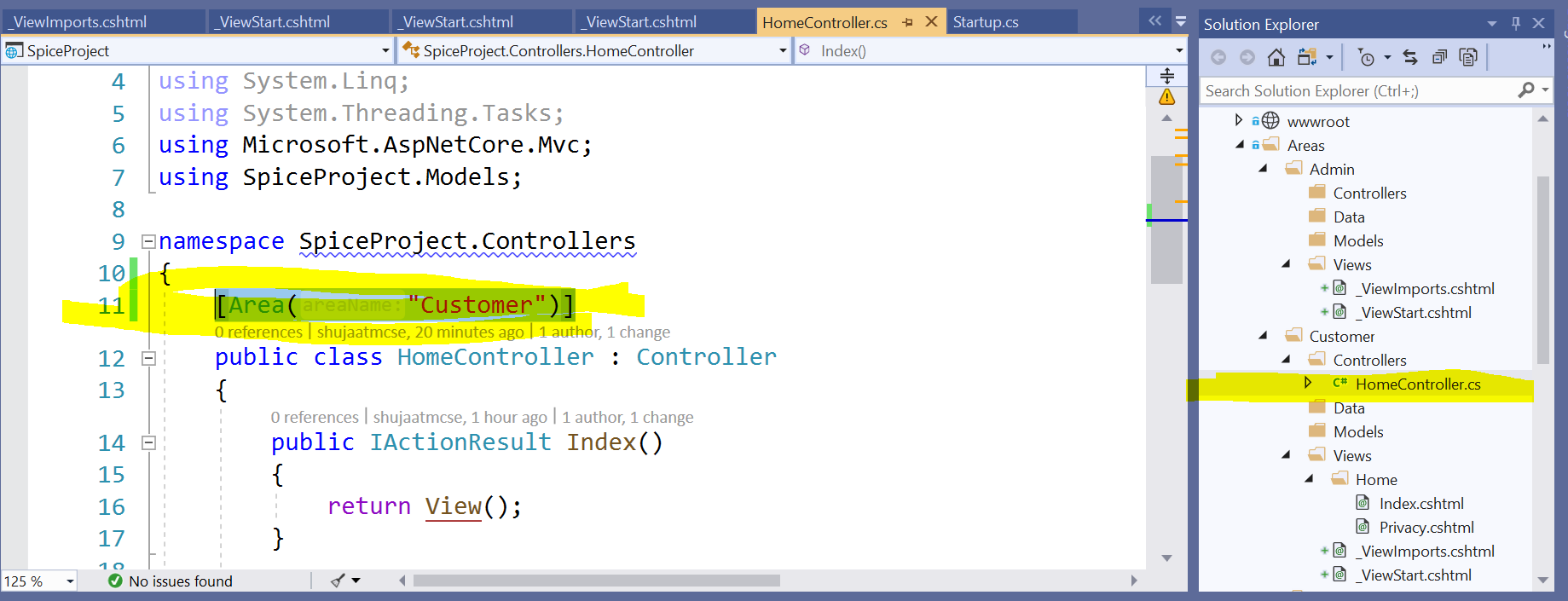
routes.MapRoute(

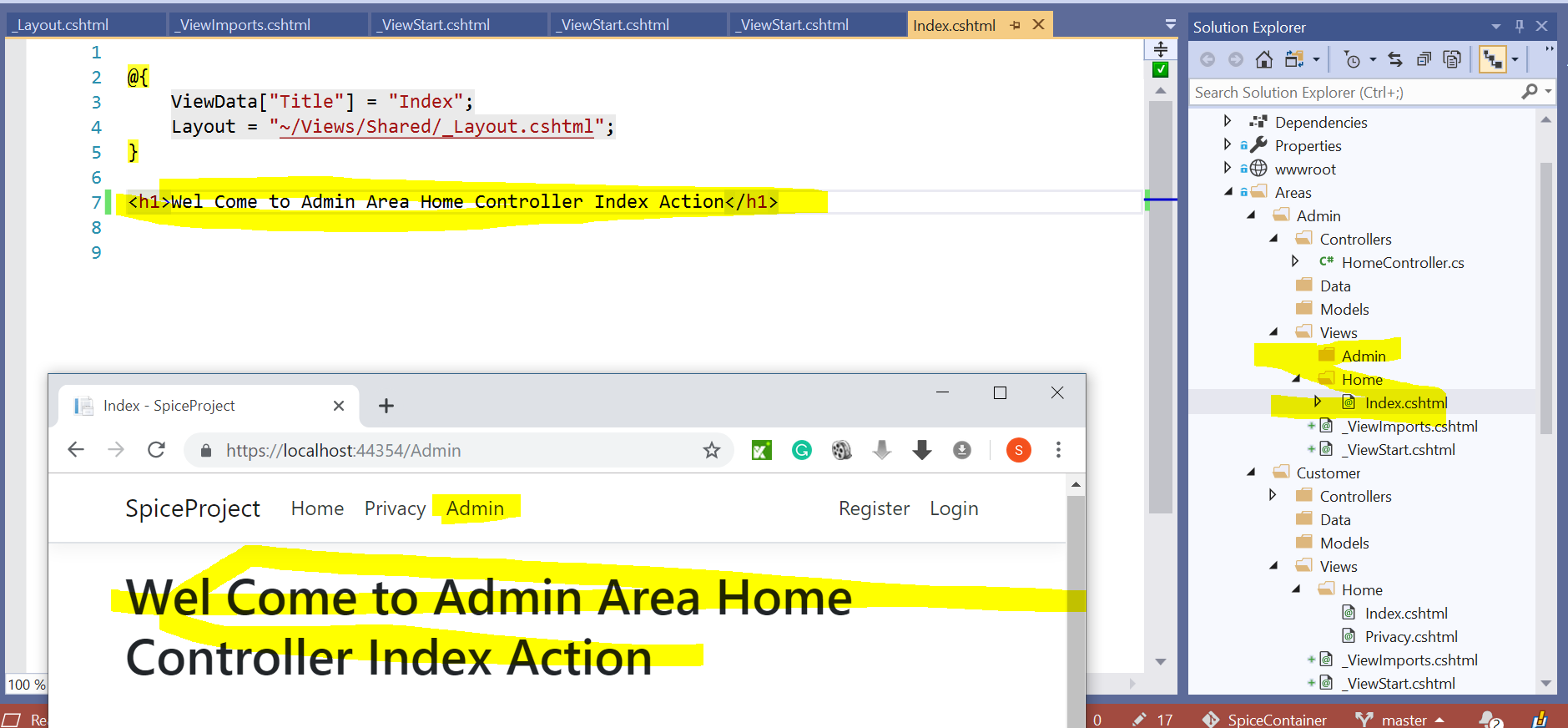
name: "areas",

template: "{area=Customer}/{controller=Home}/{action=Index}/{id?}");

1. 
2. Now you need to have the views and separate controller in both customer and admin. Example for customer is shown below: You may not need to copy the ViewStart.cshtml as it work without this as well.



1. Now we need to tell the Home Controller which we moved to Customer about the Area. Hence do it as below: Add [Area(“Customer”)]
2. 
3. Below is a screen shot that shown the admin controller as well.



The complete code is on github with commits [**Customer&AdminAreaWithReadMeFile**](https://github.com/shujaatmcse/SpiceProject/commit/58e5c0de4b6745a8979f8e822eb71c2a5231481b)

Migration/Model/DBSet.

To create the Category model do the following:

1. create the Category Model Class in Model folder.
2. In your ApplicationDBContext.cs add : public DbSet<Category> Categories { get; set; }
3. At Nuget Console type: add-migration MainCategoryAddedFirstMigration
4. Type: Update-database

To add a forgien Key in the Code first approach, see example below how the parent class is referenced in the sub class.

public class SubCategory

{

public int Id { get; set; }

public string Name { get; set; }

[Required]

[Display(Name = "Category")]

public int CategoryId { get; set; }

[ForeignKey("CategoryId")] // Category is refenced to the Id of Category in //Category class, each sub category can belong to a category, the Category class

// is referenced after the virtual word.

public virtual Category Category { get; set; }

}

//The Category Class: Note the Id in Category class is just Id

public class Category

{

[Key]

public int Id { get; set; }

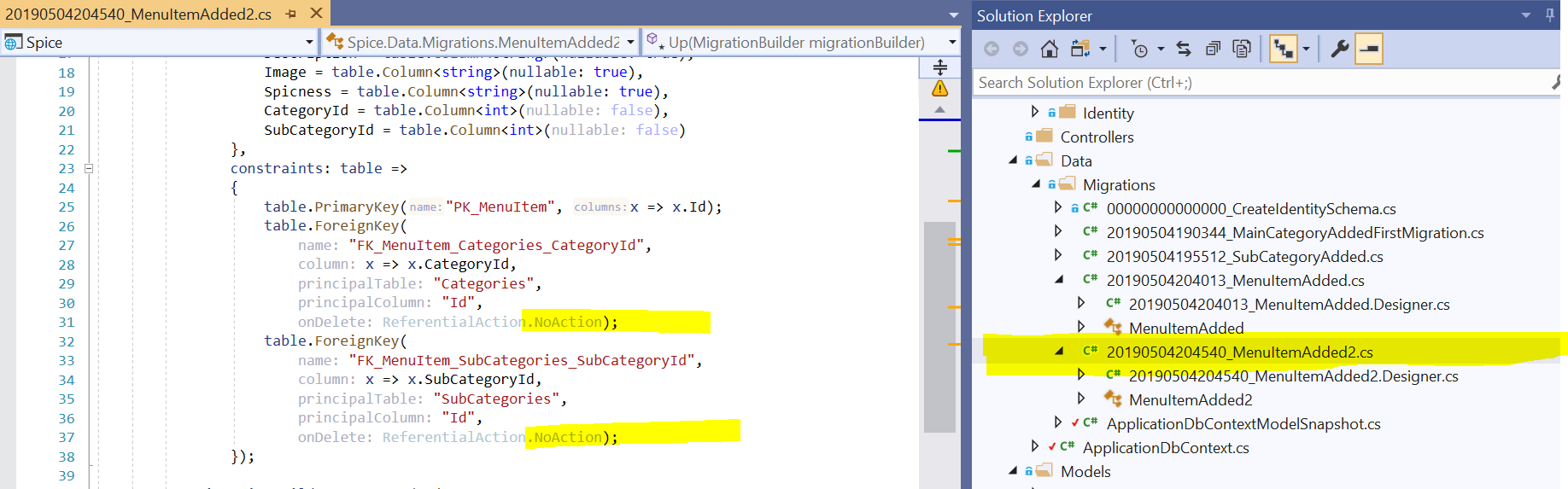
[Required]

public string Name { get; set; }

}

**Adding the Enum for MenuItem and its detail members:**

you can get an error when updating the database at Nuget Console , so go to the context detail file and change the casecad to NoAction, then rerun the update-database command. See example below



The error is because of Forgien Key constraint, if parent deleted what will the child do.

For Adding Enum: The Html.getEnumList must be used in the view.

<div class="form-group"><**label** **asp-for**="Spycness" class="control-label"></**label**>

<**select** **asp-for**="Spycness" class="form-control"**aspitems**="Html.GetEnumSelectList<MenuItem.Espicy>()"></**select**></div>

Below is a complete class of MenuItem.

public class MenuItem

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

public string Image { get; set; }

//Enum Start

public string Spycness { get; set; } // We will use this as Input in View

public enum Espicy { Na=0, NotSpicy=1, Spicy=2, HotSpicy=3} //This will be Html.get Elments

//End,

public decimal Price { get; set; }

//referencing Category

public int CategoryId { get; set; }

[ForeignKey("CategoryId")]

public Category Category { get; set; }

//referencing Sub Category

public int SubCategoryId { get; set; }

[ForeignKey("SubCategoryId")]

public SubCategory SubCategory { get; set; }

}