

# Requirement Analysis

Thursday, September 7, 2023 9:37 AM

[kdivyaswarnkar@gmail.com](mailto:kdivyaswarnkar@gmail.com)

Divya@1234

- Create a **HMS application** with **ASP.NET Web Application (.NET Framework) (.NET Framework 4.5.2Version)** Authentication should be Individual User Accounts.
- Create a new project **HMS.Entities (Class Library (.NET Framework))** in Same solution where we can add all the requirements. for separation of concerns.
- Create the classes basic Entities
- **our requirements**  
**two accommodation types**

hotel rooms	residency apartments
3 types of hotel rooms	2 types of apartments
standard	Two bedrooms
deluxe	Three bedrooms
Suits	

- Create a class AccommodationType
  - ID
  - Name //hotel Rooms , residency Apartments Apartment
  - Description

## AccommodationTypes

ID	Name
1	Hotel Rooms
2	Apartment

## AccommodationPackages

ID	AccommodationTypeID	Name	NoOfRoom	FeePerNight
1	1	Standard	1	50
2	1	Deluxe	1	70
3	1	Suites	1	100
4	2	2 Bedrooms	2	120
5	2	3 Bedrooms	3	150

### Accommodation:-

Ex:- **select \* from Accommodations where AccommodationPackageID =2**

ID	AccommodationPackageID	Name

1	1	190
2	1	191
3	2	192
4	1	193
5	3	194
6	2	195
7	4	Apartment No 232
8	4	Apartment No 237
9	5	Apartment No 200
10	4	Apartment No 231

### Bookings :-

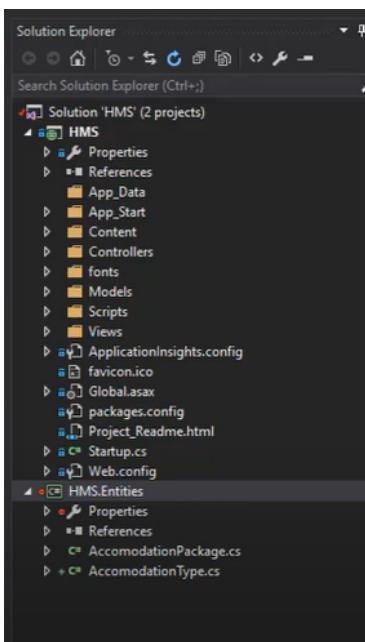
ID	AccommodationID	FromDate	Duration(No of Stay Nights)
1	3	22June2023	4 days

```
Untitled - Notepad
File Edit Format View Help
AccommodationTypes
ID      Name
1       Hotel Room
2       Apartment

AccommodationPackages
ID      AccommodationTypeID      Name          NoOfRoom      FeePerNight
1       1                         Standard       1             50
2       1                         Deluxe        1             70
3       1                         Suites        1             100
4       2                         2 Bedroom    2             120
5       2                         3 Bedroom    3             150

Accommodations -- select * from Accommodations where AccommodationPackageID = 2
ID      AccommodationPackageID      Name
1       1                           198
2       1                           191
3       2                           192
4       1                           193
5       3                           194
6       2                           195
7       4                           Apartment No 232
8       4                           Apartment No 237
9       5                           Apartment No 200
10      4                           Apartment No 231

Bookings
ID      AccommodationID      FromDate      Duration(No Of Stay Nights)
1       3                      22June2019   4
```



```
AccommodationType.cs  # X
HMS.Entities
1  using System;
```

```

2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace HMS.Entities
8  {
9      14 references
10     public class AccomodationType
11     {
12         2 references
13         public int ID { get; set; }
14
15         4 references
16         public string Name { get; set; }
17         3 references
18         public string Description { get; set; }
19     }
20 }

```

If we want to reference anything then we have to create 2 properties in class

```

AccommodationPackage.cs  ✘ AccomodationType.cs
HMS.Entities -> HMS.Entities.AccommodationPackage -> FeePerNight
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace HMS.Entities
8  {
9      public class AccommodationPackage
10     {
11         public int ID { get; set; }
12
13         public int AccommodationTypeID { get; set; }
14         public AccomodationType AccommodationType { get; set; }
15
16         public string Name { get; set; }
17         public int NoOfRoom { get; set; }
18         public decimal FeePerNight { get; set; }
19     }
20 }

```

```

HMS - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Accommodation.cs  ✘ AccomodationPackage.cs  AccomodationType.cs
HMS.Entities -> HMS.Entities.Accommodation -> Name
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace HMS.Entities
8  {
9      public class Accommodation
10     {
11         public int ID { get; set; }
12
13         public int AccommodationPackageID { get; set; }
14         public AccommodationPackage AccommodationPackage { get; set; }
15
16         public string Name { get; set; }
17         public string Description { get; set; }
18     }
19 }
20

```

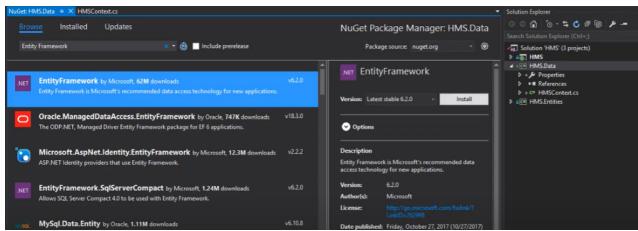
```

Booking.cs  ✘ AccomodationPackage.cs  AccomodationType.cs
HMS.Entities -> HMS.Entities.Booking -> HMS.Entities.Accommodation
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace HMS.Entities
8  {
9      1 reference
10     public class Booking
11     {
12         0 references
13         public int ID { get; set; }
14
15         0 references
16         public int AccommodationID { get; set; }
17         0 references
18         public Accommodation Accommodation { get; set; }
19
20         0 references
21         public DateTime FromDate { get; set; }
22
23         /// <summary>
24         /// No Of Stay Nights
25         /// </summary>
26         0 references
27         public int Duration { get; set; }
28     }
29 }

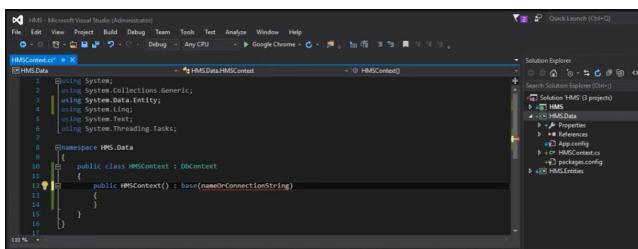
```



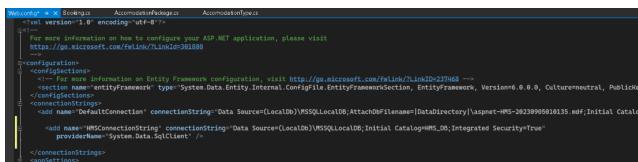
- Create a new project **HMS.Data (Class Library (.NET Framework))** in Same solution where we have to create a new class name **DbContext**.
- Before inherited the class with DbContext class first we need to install EntityFramework in our class library project usir



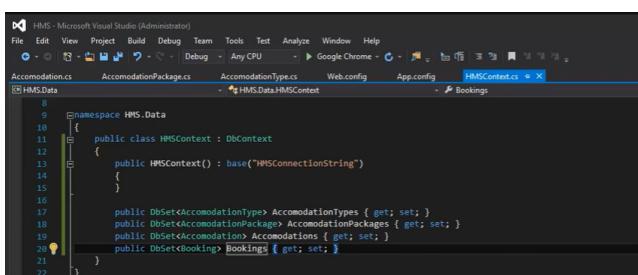
Now Inherit the HMSContext class from DbContext Class and add the required namespace **System.Data.Entity**



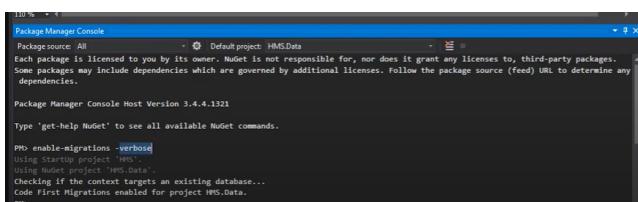
For add the value of connectionstring we need to do some changes in our (appconfig file of same project file or our main wherever suitable for us )



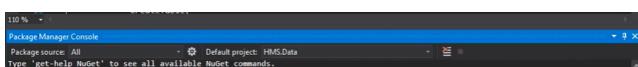
Now we need to add Dbsets in to the Context class



We need to do migrations in HMS.Data class library add the commands (code first approach)  
enable-migrations -verbose

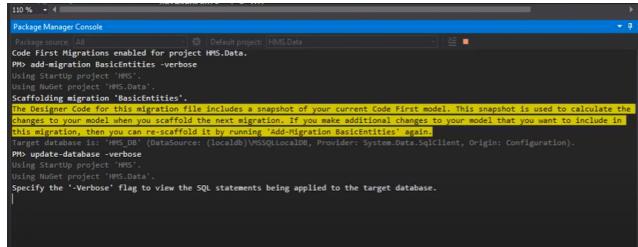


add-migration BasicEntities -verbose



```
PM> enable-migrations -verbose
PM> add-migration InitialCreate -verbose
PM> using Microsoft.EntityFrameworkCore;
PM> using Microsoft.EntityFrameworkCore.Design;
PM> Checking if the context targets an existing database...
PM> Code First Migrations enabled for project HMS.Data.
PM> add-migration BasicEntities -verbose
PM> using System;
PM> using Microsoft.EntityFrameworkCore;
PM> using Microsoft.EntityFrameworkCore.Design;
PM> Scaffold migration "BasicEntities".
The Designer Code for this migration file includes a snapshot of your current Code First model. This snapshot is used to calculate the changes to your model when you scaffold the next migration. If you make additional changes to your model that you want to include in this migration, then you can re-scaffold it by running 'Add-Migration BasicEntities' again.
PM> Scaffold database for 'HMS DB' at 'C:\Users\localh\VisualStudio\HMS\HMS.Data\bin\Debug\netcoreapp3.1\HMS.Data.scaffolding'.
PM> [REDACTED]
```

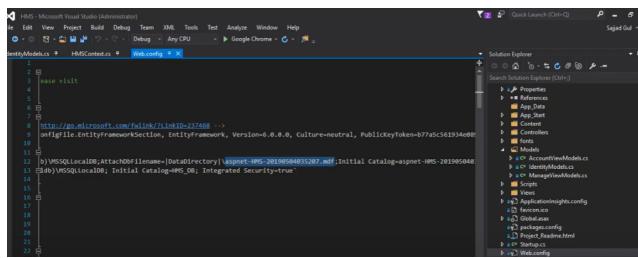
Update-database -verbose



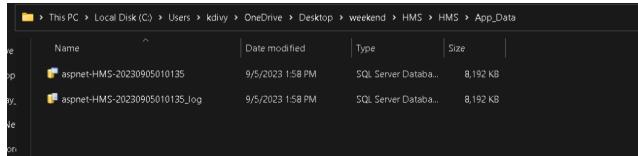
We can check now in our sql server where we can see all the tables which is created using the migrations

**REMEMBER :-**

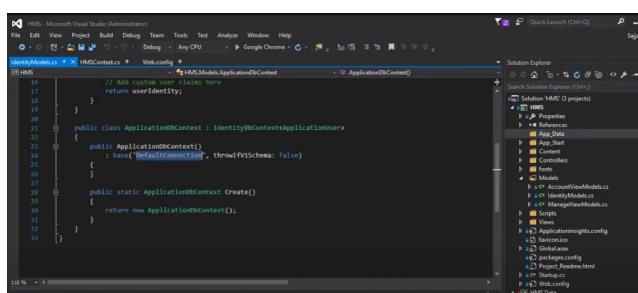
When we created the application then we selected the individual user accounts so now when we register and login then it will give the result and registration and log we can do login and logout and registration. We can see this mdf file in appdata>open in file explorer



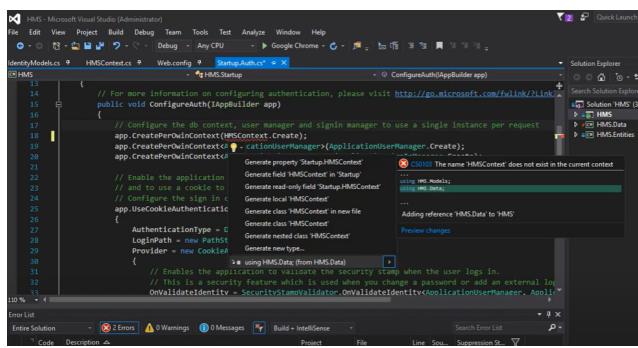
It is default database which is added automatically we want the data should be add in the users table in our HMS database



So we can do 2 things we can replace the Defaultconnection from the base or we can remove the ApplicationDbContext class.



Here I removed ApplicationDbContext Class, replace the ApplicationDbContext in to HMSDbContext wherever we referenced.

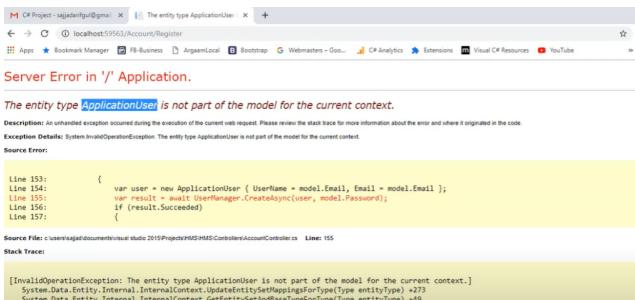


```

10 public class HMSContext : DbContext
11 {
12     public HMSContext() : base("HMSConnectionString")
13     {
14     }
15 
16     public static HMSContext Create()
17     {
18         return new HMSContext();
19     }
20 
21     public DbSet<AccommodationType> AccommodationTypes { get; set; }
22     public DbSet<AccommodationPackage> AccommodationPackages { get; set; }
23     public DbSet<Accommodation> Accommodations { get; set; }
24     public DbSet<Booking> Bookings { get; set; }
25 
26 }
27 
28

```

We will get this error so we need to do some changes in our code and add HMSUser class in entities and copy paste some code



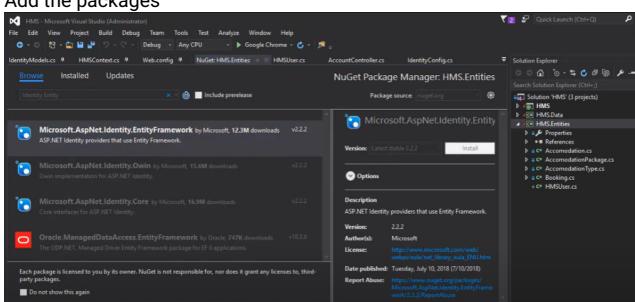
```

1 Existing System.Data.Entity;
2 using System.Security.Claims;
3 using System.Threading.Tasks;
4 using Microsoft.AspNetCore.Identity;
5 using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
6 
7 namespace HMS.Models
8 {
9     // You can add profile data for the user by adding more properties to your ApplicationUser class. Please visit
10    // http://go.microsoft.com/fwlink/?LinkID=317594 to learn more.
11    public class ApplicationUser : IdentityUser
12    {
13        public async Task<ClaimsIdentity> GenerateUserIdentityAsync(UserManager< ApplicationUser > manager)
14        {
15            // Note the authenticationType must match the one defined in CookieAuthenticationOptions.AuthenticationType
16            var userIdentity = await manager.CreateIdentityAsync(this, DefaultAuthenticationType.ApplicationCookie);
17            // Add custom claims here
18            return userIdentity;
19        }
20    }
21

```

The above code into the user class

Add the packages

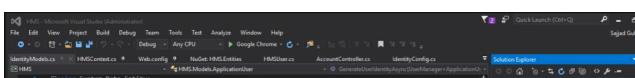


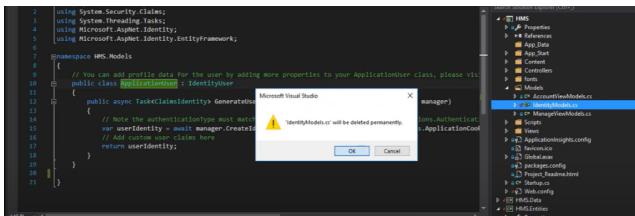
```

1 using Microsoft.AspNetCore.Identity;
2 using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
3 using System;
4 using System.Collections.Generic;
5 using System.Linq;
6 using System.Security.Claims;
7 using System.Threading.Tasks;
8 
9 namespace HMS.Entities
10 {
11     public class HMSUser : IdentityUser
12     {
13         public async Task<ClaimsIdentity> GenerateUserIdentityAsync(UserManager<HMSUser> manager)
14         {
15             // Note the authenticationType must match the one defined in CookieAuthenticationOptions.AuthenticationType
16             var userIdentity = await manager.CreateIdentityAsync(this, DefaultAuthenticationType.ApplicationCookie);
17             // Add custom claims here
18             return userIdentity;
19         }
20     }
21
22 }
23

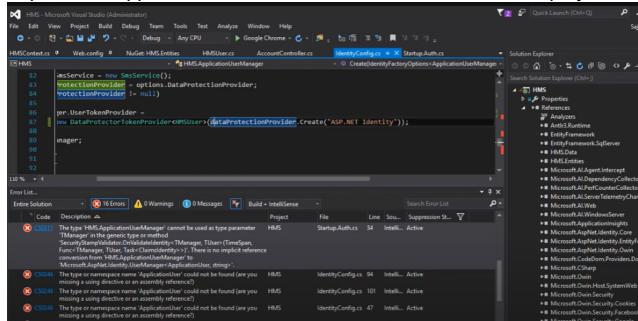
```

Now delete the IdentityModel class from the project

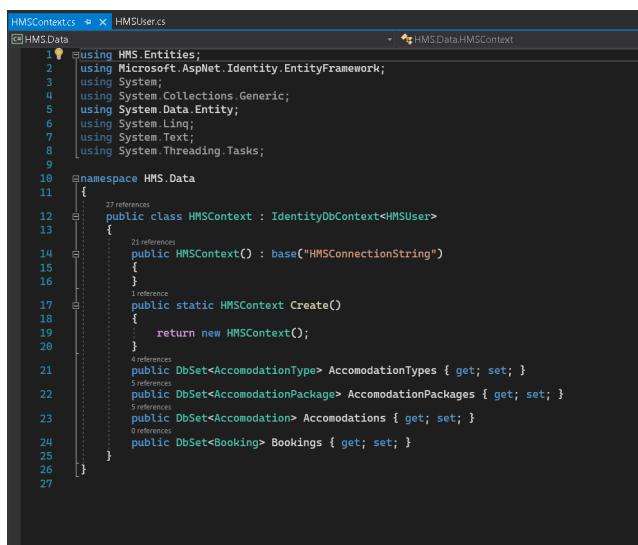
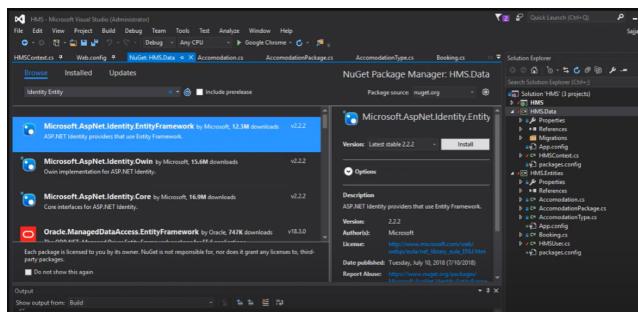




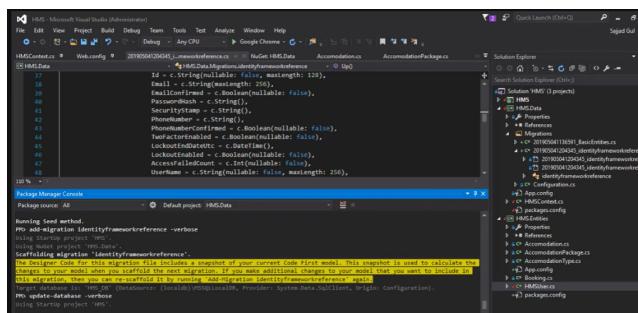
Replace the ApplicationUser into the HMSUser from the whole project



Now update the HMSDbContext class



Note :- whenever we will do any changes in context then we need to add migration and update database



The screenshot shows the Visual Studio IDE interface. On the left, the Solution Explorer displays a project named "IdentityConfig.cs". In the center, the SQL Server Object Explorer shows a connection to "SQL Server (localhost)\MSSQLLocalDB (SQL Server 13.0.0)". Under the "Tables" node, there is a single table named "AspNetUsers" with two rows of data. On the right, the "T-SQL" window displays the SQL query used to select all users from the "AspNetUsers" table.

ID	UserName	Email	EmailConfirmed	NormalizedEmail	NormalizedUserName	PasswordHash	SecurityStamp
1	aspid1@gmail.com	aspid1@gmail.com	0	ASPID1@GMAIL.COM	ASPID1@GMAIL.COM	A1405e10f1a5d45b172a2a14f1a08898	e4f76937-34c2-416d-9162-15e405612345
2	aspid2@gmail.com	aspid2@gmail.com	0	ASPID2@GMAIL.COM	ASPID2@GMAIL.COM	42010ffcc0a049e9a0474e4a474e4a47	c49a3e30-3420-4109-8556-5e5e012345

```
select * from AspNetUsers
```