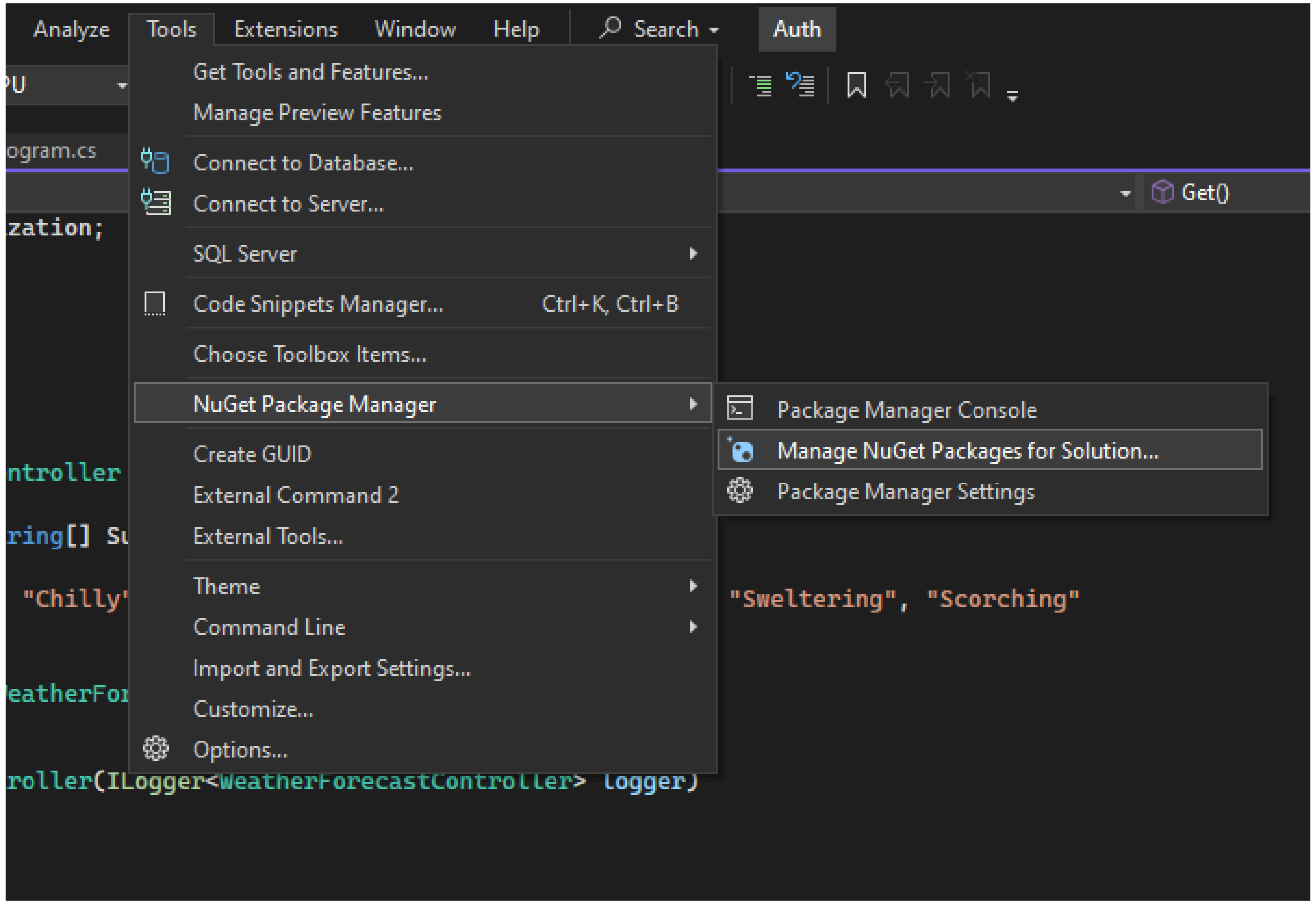
Authentication sử dụng Identity

Nhóm 2

Cài đặt NuGet Package

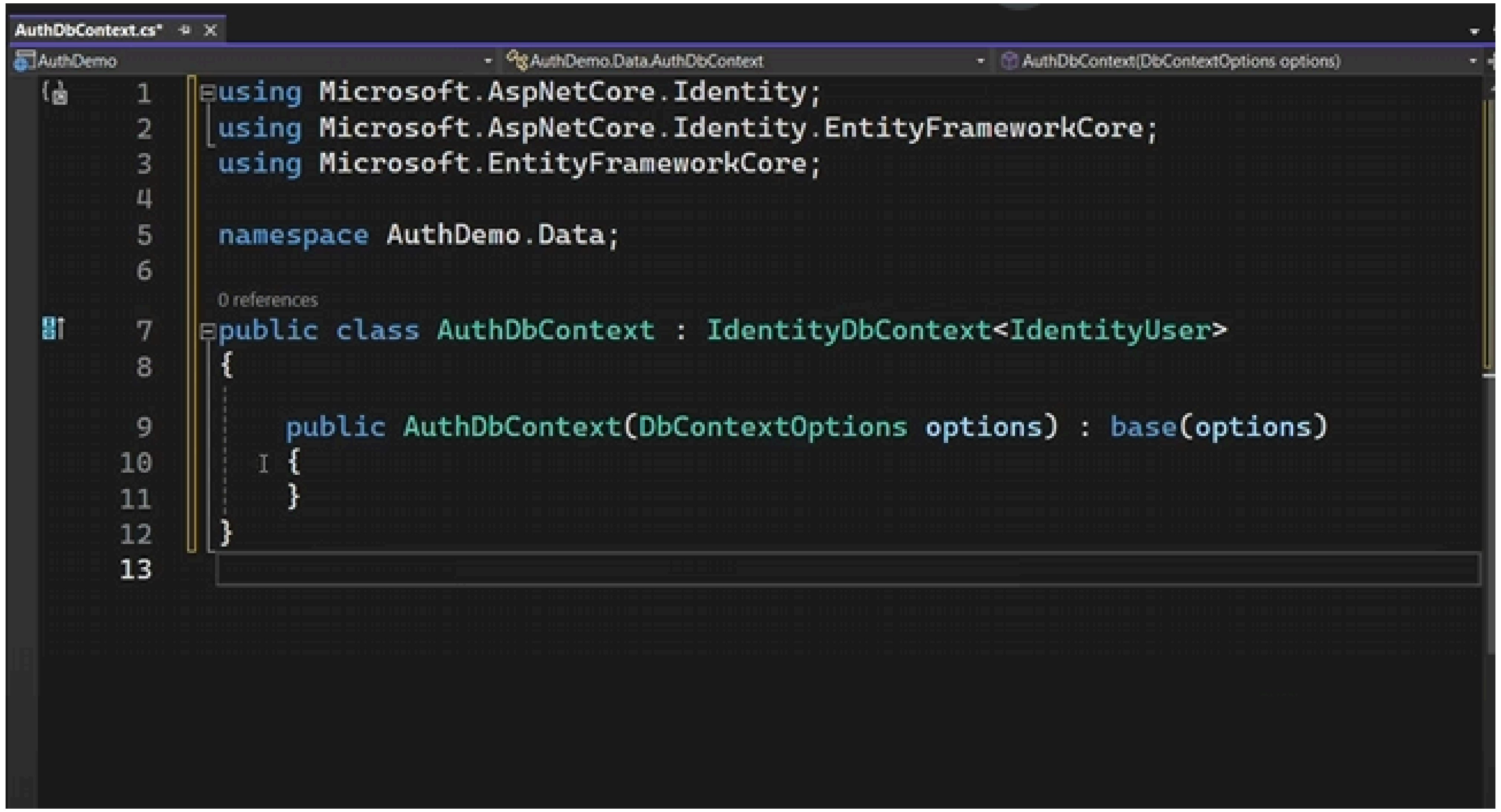
- Chọn Tools -> NuGet Package Manager -> Manage Packages for Solution..
- Tìm và cài đặt Package sau:

  - Microsoft.AspNetCore.Identity.EntityFrameworkCore



Setup DbContext

Cho DbContext thừa kế IdentityDbContext<IdentityUser>

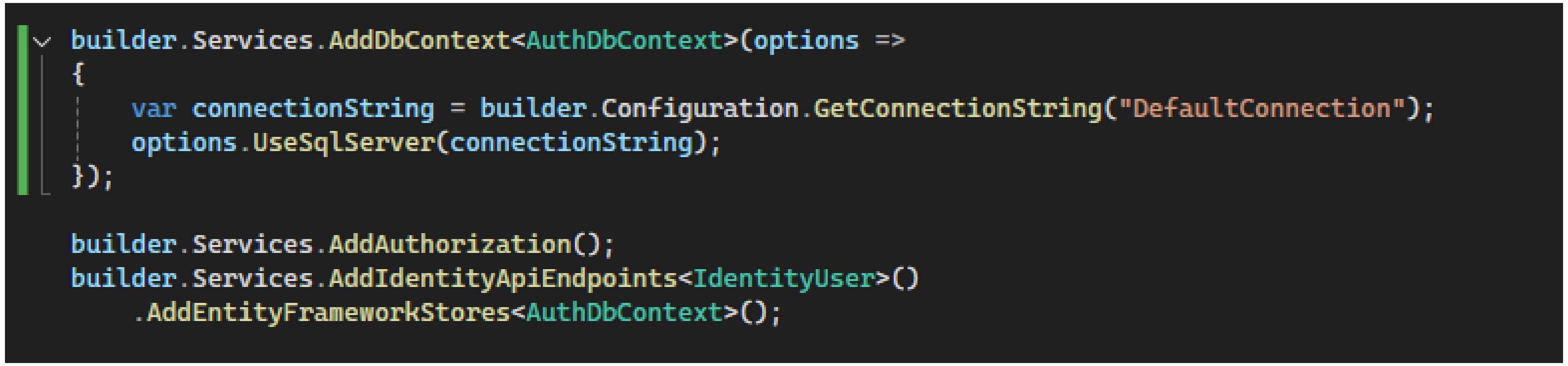


AuthDbContext.cs*
AuthDemo
AuthDemo.Data.AuthDbContext
AuthDbContext(DbContextOptions options)
0 references

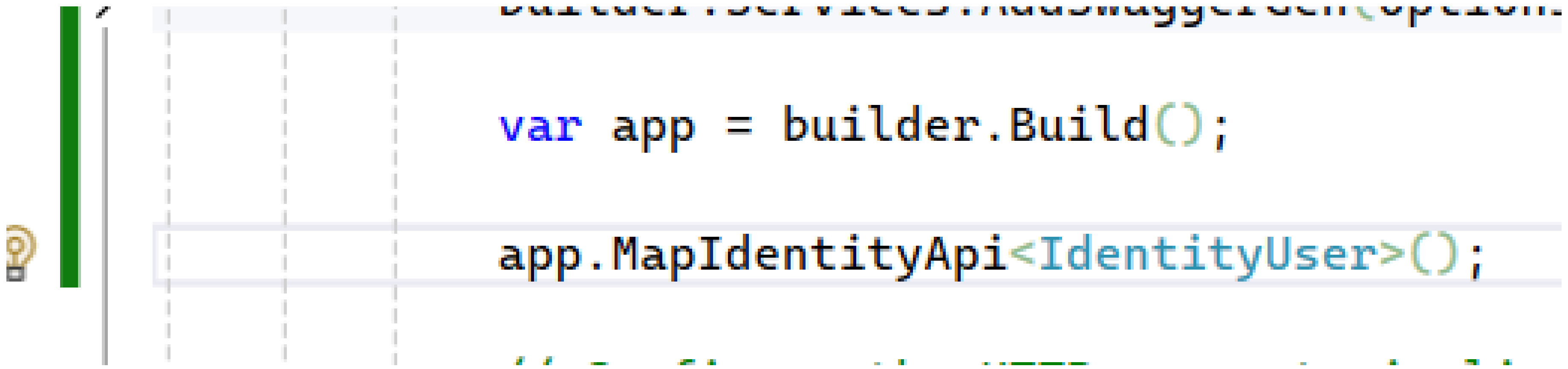
```
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;

namespace AuthDemo.Data;

public class AuthDbContext : IdentityDbContext<IdentityUser>
{
    public AuthDbContext(DbContextOptions options) : base(options)
    {
    }
}
```

Setup Program.cs

Add các Services: AddDbContext, AddAuthorization, AddIdentityApiEndpoints, .AddEntityFrameworkStores vào Program.cs



```
builder.Services.AddDbContext<AuthDbContext>(options =>
{
    var connectionString = builder.Configuration.GetConnectionString("DefaultConnection");
    options.UseSqlServer(connectionString);
});

builder.Services.AddAuthorization();
builder.Services.AddIdentityApiEndpoints<IdentityUser>()
    .AddEntityFrameworkStores<AuthDbContext>();
```

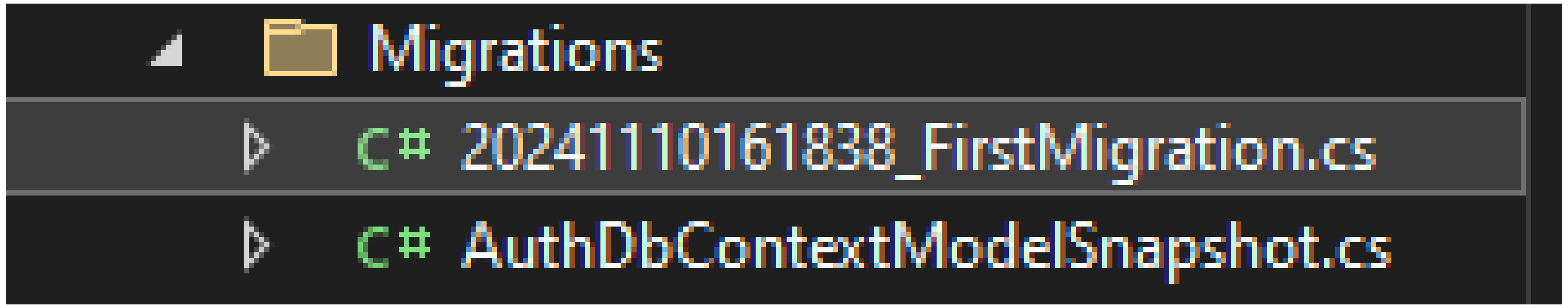


```
var app = builder.Build();

app.MapIdentityApi<IdentityUser>();
```

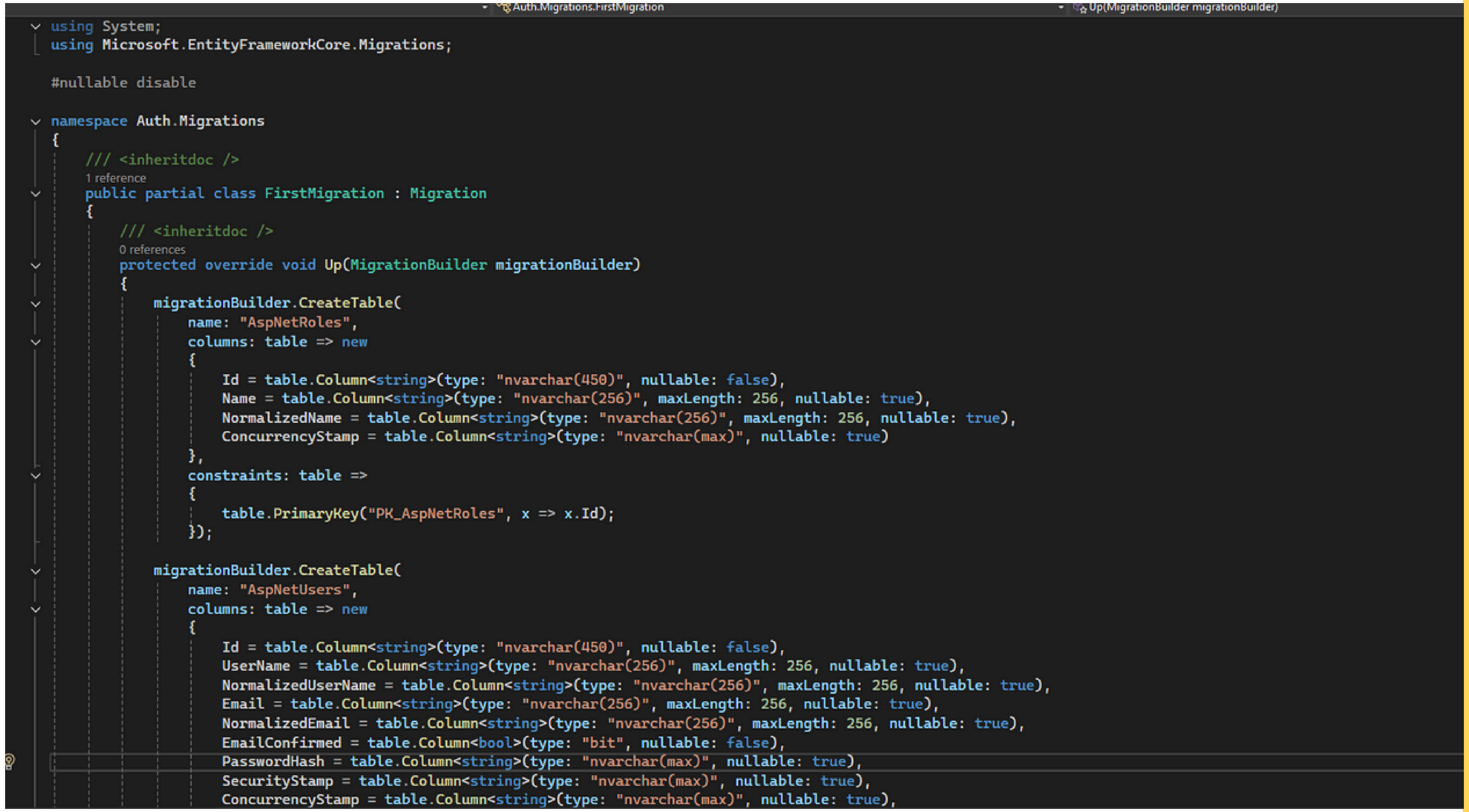
Tạo Migration và update database để tạo Table

Mở Package Manager Console và tạo Migration:

Add-Migration TênMigration



Migrations
C# 20241110161838_FirstMigration.cs
C# AuthDbContextModelSnapshot.cs



Auth.Migrations.FirstMigration
Up(MigrationBuilder migrationBuilder)

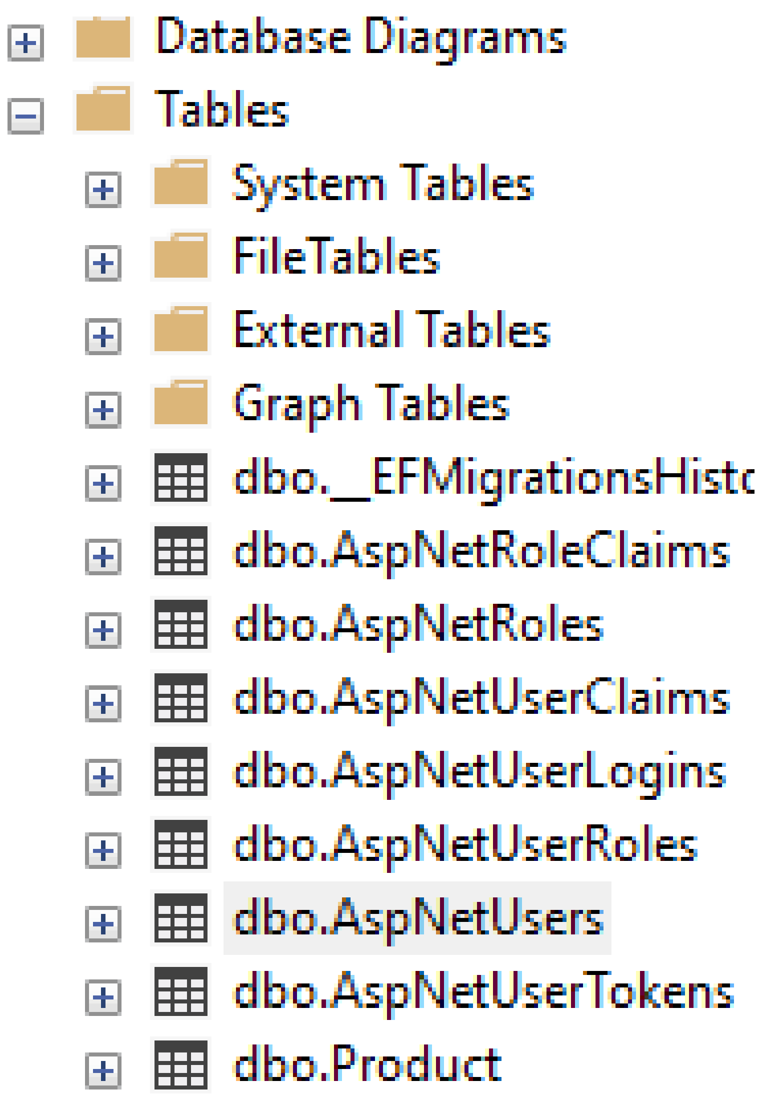
```
using System;
using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace Auth.Migrations
{
    /// <inheritdoc />
    1 reference
    public partial class FirstMigration : Migration
    {
        /// <inheritdoc />
        0 references
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "AspNetRoles",
                columns: table => new
                {
                    Id = table.Column<string>(type: "nvarchar(450)", nullable: false),
                    Name = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),
                    NormalizedName = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),
                    ConcurrencyStamp = table.Column<string>(type: "nvarchar(max)", nullable: true)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_AspNetRoles", x => x.Id);
                });

            migrationBuilder.CreateTable(
                name: "AspNetUsers",
                columns: table => new
                {
                    Id = table.Column<string>(type: "nvarchar(450)", nullable: false),
                    UserName = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),
                    NormalizedUserName = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),
                    Email = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),
                    NormalizedEmail = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),
                    EmailConfirmed = table.Column<bool>(type: "bit", nullable: false),
                    PasswordHash = table.Column<string>(type: "nvarchar(max)", nullable: true),
                    SecurityStamp = table.Column<string>(type: "nvarchar(max)", nullable: true),
                    ConcurrencyStamp = table.Column<string>(type: "nvarchar(max)", nullable: true),
```

Tạo Migration và update database để tạo Table

Sau khi add Migration thành công, tiếp theo ta update Database :

update-database

Khi Update database thành công, MySQL Server sẽ tự động tạo ra những bảng bên ->

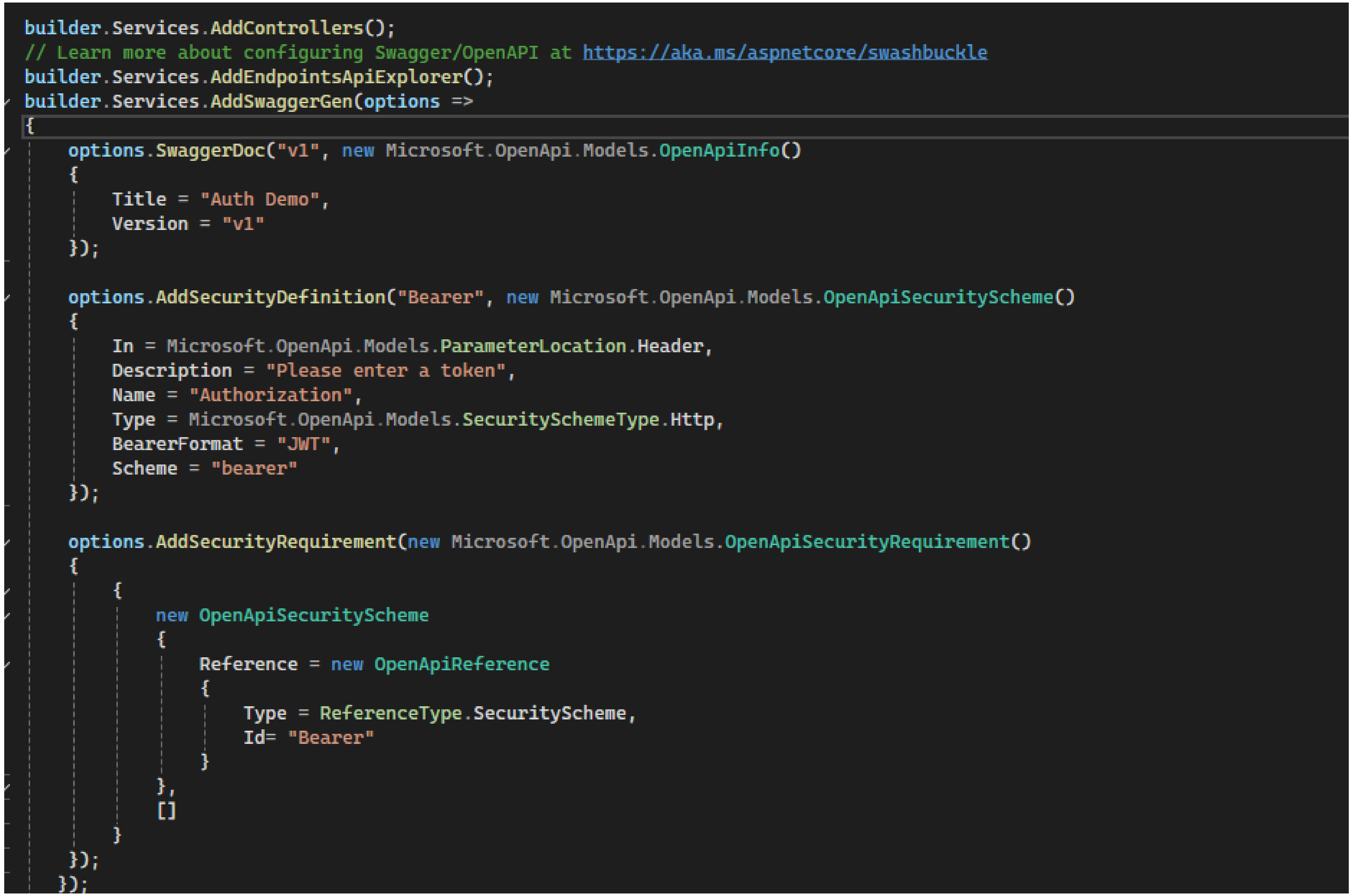


```
⊞ Database Diagrams
⊟ Tables
    ⊞ System Tables
    ⊞ FileTables
    ⊞ External Tables
    ⊞ Graph Tables
    ⊞ dbo.__EFMigrationsHistc
    ⊞ dbo.AspNetRoleClaims
    ⊞ dbo.AspNetRoles
    ⊞ dbo.AspNetUserClaims
    ⊞ dbo.AspNetUserLogins
    ⊞ dbo.AspNetUserRoles
    ⊞ dbo.AspNetUsers
    ⊞ dbo.AspNetUserTokens
    ⊞ dbo.Product
```

Setup Program.cs

Thêm Services.AddSwaggerGen bao gồm những options:

- SwaggerDoc
- AddSecurityDefinition
- AddSecurityRequirement



```
builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen(options =>
{
    options.SwaggerDoc("v1", new Microsoft.OpenApi.Models.OpenApiInfo()
    {
        Title = "Auth Demo",
        Version = "v1"
    });

    options.AddSecurityDefinition("Bearer", new Microsoft.OpenApi.Models.OpenApiSecurityScheme()
    {
        In = Microsoft.OpenApi.Models.ParameterLocation.Header,
        Description = "Please enter a token",
        Name = "Authorization",
        Type = Microsoft.OpenApi.Models.SecuritySchemeType.Http,
        BearerFormat = "JWT",
        Scheme = "bearer"
    });

    options.AddSecurityRequirement(new Microsoft.OpenApi.Models.OpenApiSecurityRequirement()
    {
        {
            new OpenApiSecurityScheme
            {
                Reference = new OpenApiReference
                {
                    Type = ReferenceType.SecurityScheme,
                    Id= "Bearer"
                }
            },
            []
        }
    });
   });
```