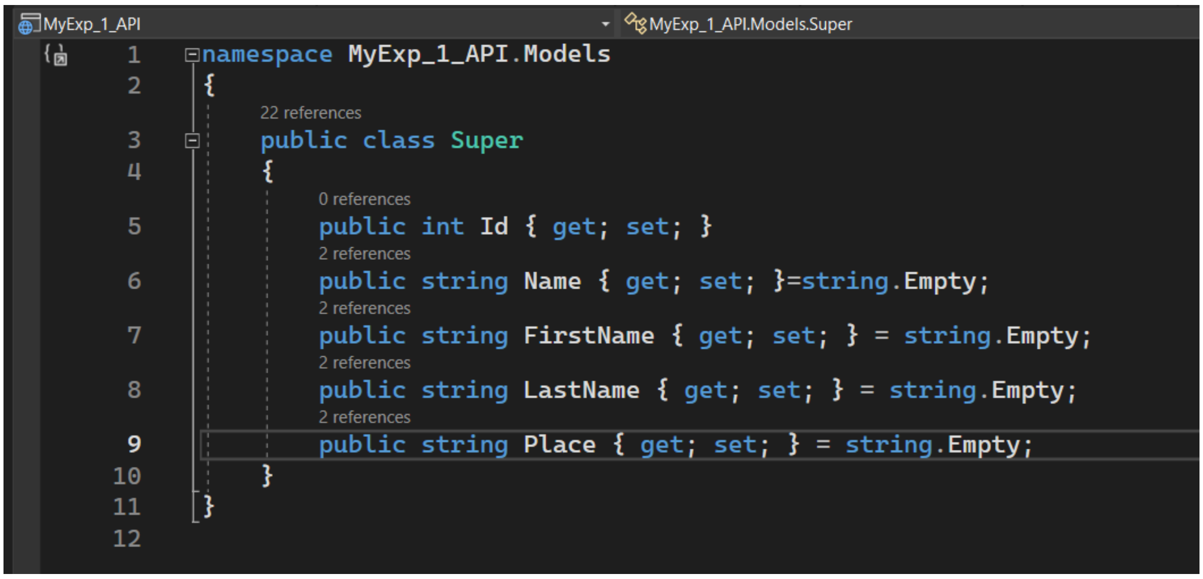
.NET REST API STEPS

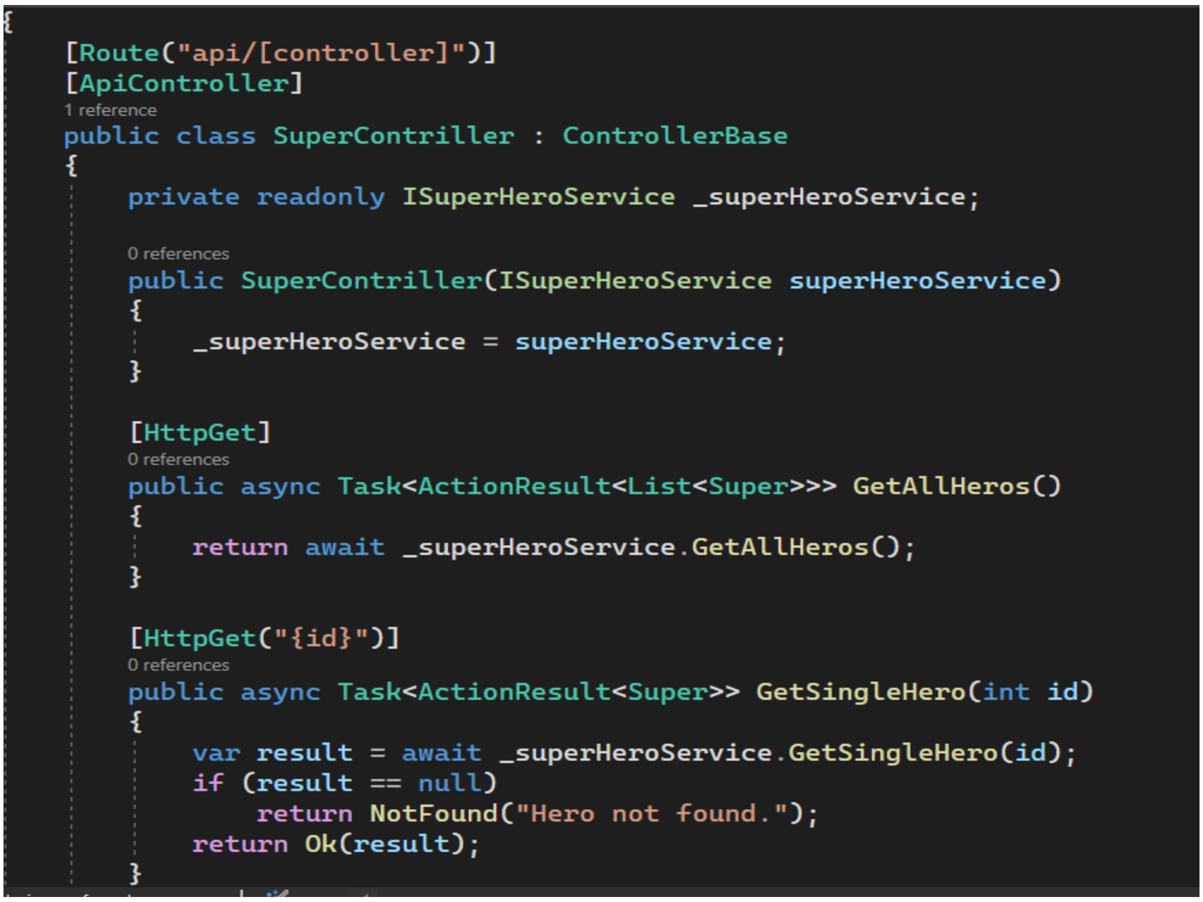
Step1: Open Visual Studio 2022 → Create New Project → ASP.NET Core Web API Click on Next.
Step2: Give The Name for the project and Location to Store then Click on Next
Step3: It Shows Additional information leave as default and Click on Create Button.
Step4: Goto Solution Box Right Click on Project Name and Add new Folder “Models”and Right click on “Models” folder and Add Class “Employees”.Add Properties Inside Employees Class like (public int Id {get,set}; etc).



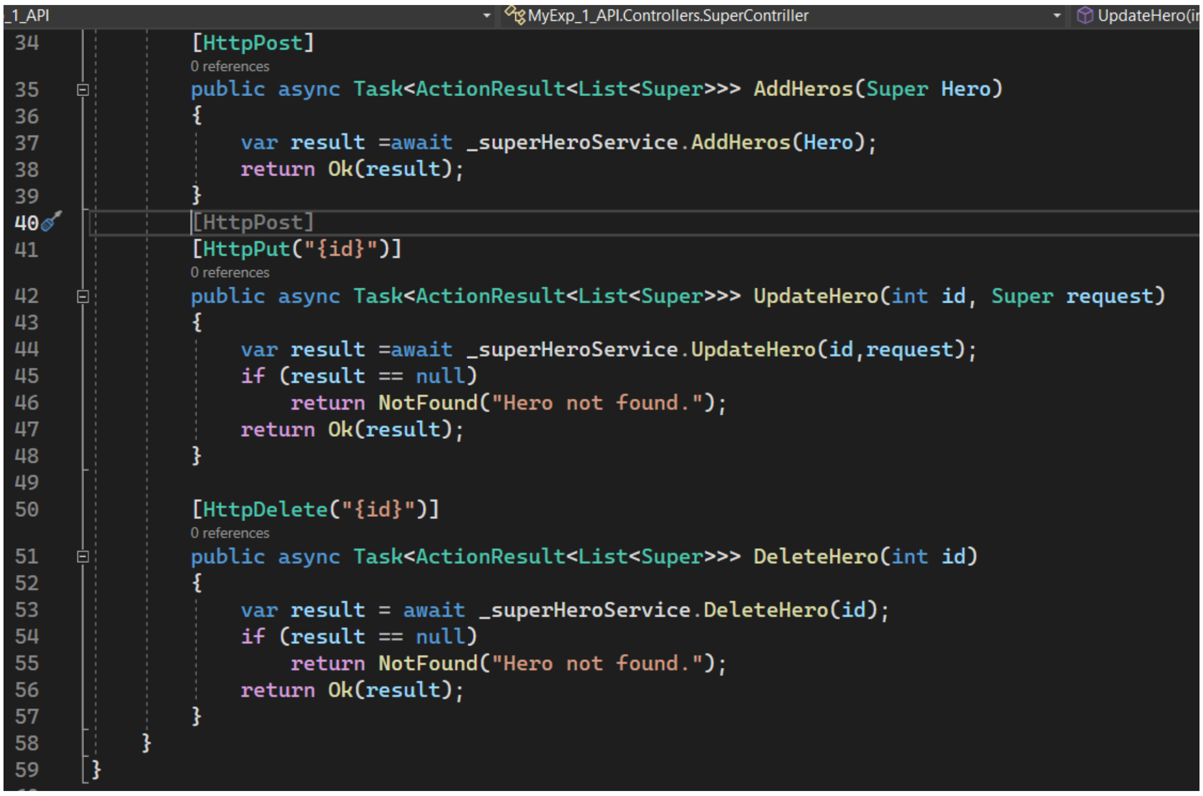
```
namespace MyExp_1_API.Models
{
    22 references
    public class Super
    {
        0 references
        public int Id { get; set; }
        2 references
        public string Name { get; set; }=string.Empty;
        2 references
        public string FirstName { get; set; } = string.Empty;
        2 references
        public string LastName { get; set; } = string.Empty;
        2 references
        public string Place { get; set; } = string.Empty;
    }
}
```

Step5:Right Click on Controller folder → Add→Controller.It shows window in left side **Select API→MVC Controller-Empty** Click on Add.It shows the new window, give it name (EmployeesController) and Click on Add.

Step6:



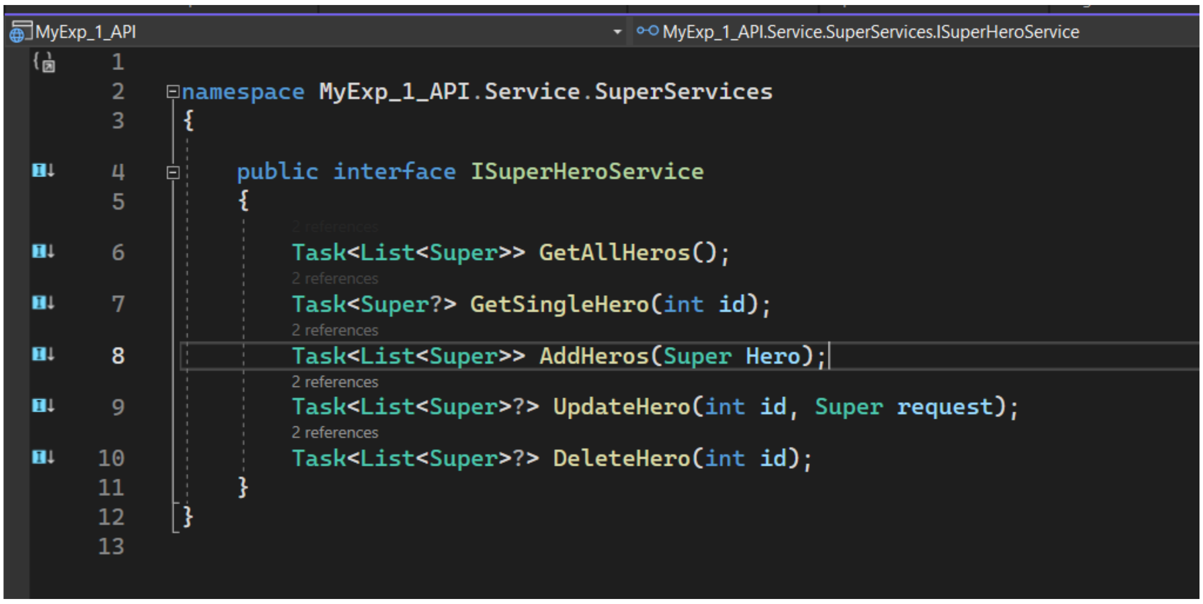
```
{
    [Route("api/[controller]")]
    [ApiController]
    1 reference
    public class SuperContriller : ControllerBase
    {
        private readonly ISuperHeroService _superHeroService;

        0 references
        public SuperContriller(ISuperHeroService superHeroService)
        {
            _superHeroService = superHeroService;
        }

        [HttpGet]
        0 references
        public async Task<ActionResult<List<Super>>> GetAllHeros()
        {
            return await _superHeroService.GetAllHeros();
        }

        [HttpGet("{id}")]
        0 references
        public async Task<ActionResult<Super>> GetSingleHero(int id)
        {
            var result = await _superHeroService.GetSingleHero(id);
            if (result == null)
                return NotFound("Hero not found.");
            return Ok(result);
        }
```



```
_1_API                                MyExp_1_API.Controllers.SuperContriller                    UpdateHero(i
            [HttpPost]
            0 references
            public async Task<ActionResult<List<Super>>> AddHeros(Super Hero)
            {
                var result =await _superHeroService.AddHeros(Hero);
                return Ok(result);
            }
            [HttpPost]
            [HttpPut("{id}")]
            0 references
            public async Task<ActionResult<List<Super>>> UpdateHero(int id, Super request)
            {
                var result =await _superHeroService.UpdateHero(id,request);
                if (result == null)
                    return NotFound("Hero not found.");
                return Ok(result);
            }

            [HttpDelete("{id}")]
            0 references
            public async Task<ActionResult<List<Super>>> DeleteHero(int id)
            {
                var result = await _superHeroService.DeleteHero(id);
                if (result == null)
                    return NotFound("Hero not found.");
                return Ok(result);
            }
        }
    }
```

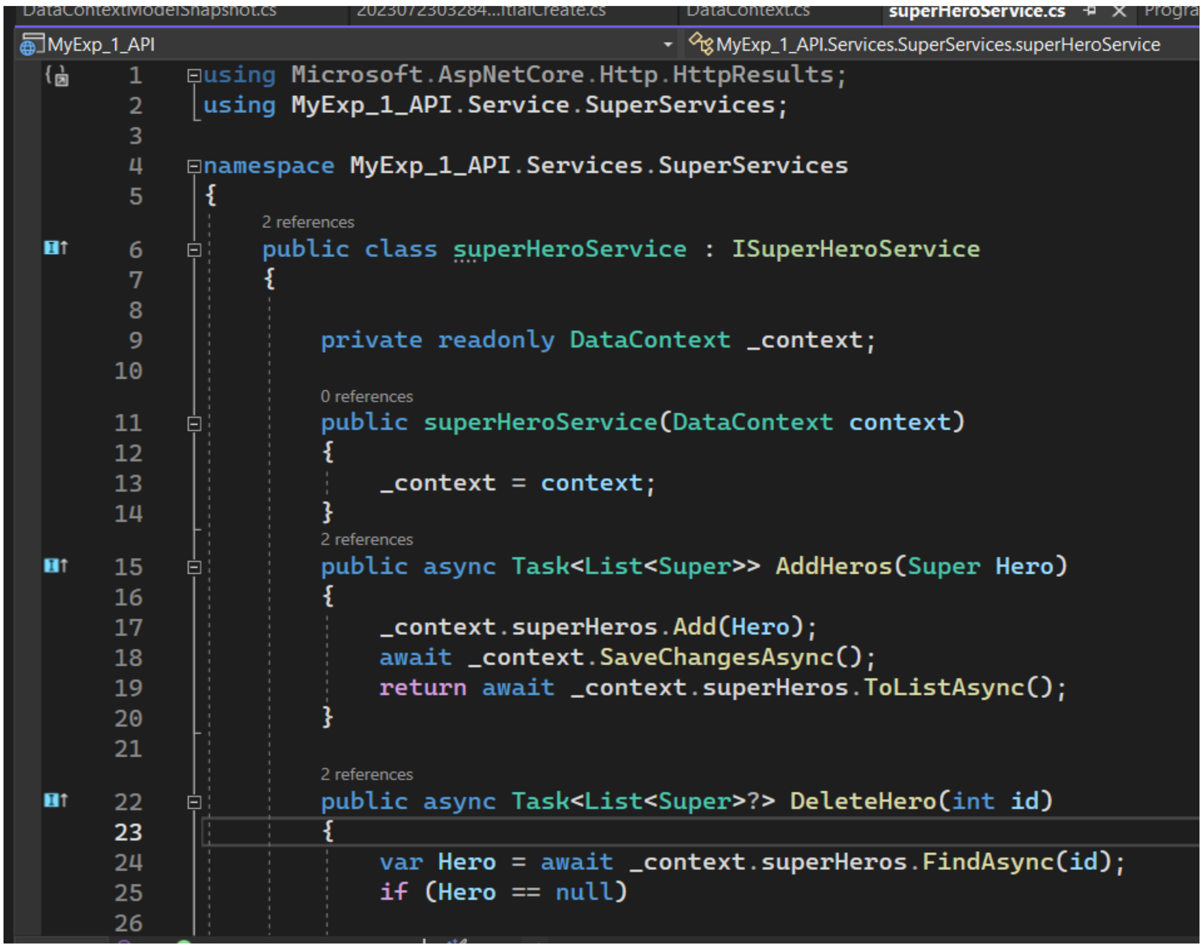
Step7:Add new Folder “**Services”.Right click and add another Folder called EmployeesServices.Right click on EmployeesServices and Add→ New Item→ Interface**. Give it name **“IEmployeeService.cs”** and also Add new Class Called **“EmployeesService.cs”.**

IEmployeeService.cs



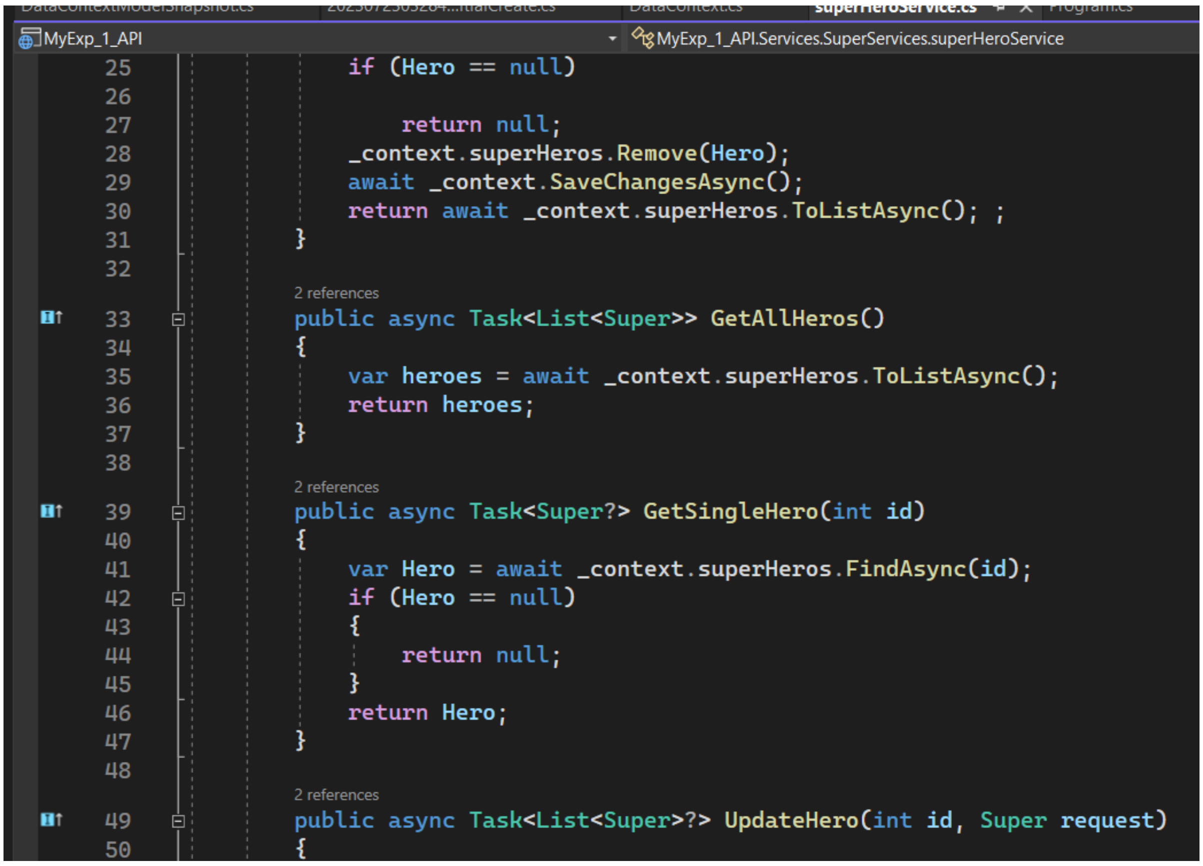
```csharp

namespace MyExp_1_API.Service.SuperServices
{
    public interface ISuperHeroService
    {
        2 references
        Task<List<Super>> GetAllHeros();
        2 references
        Task<Super?> GetSingleHero(int id);
        2 references
        Task<List<Super>> AddHeros(Super Hero);
        2 references
        Task<List<Super>?> UpdateHero(int id, Super request);
        2 references
        Task<List<Super>?> DeleteHero(int id);
    }
}
```

For **EmployeesService.cs**


DataContextModelSnapshot.cs   20230723032846...itialCreate.cs   DataContext.cs   superHeroService.cs   Progra

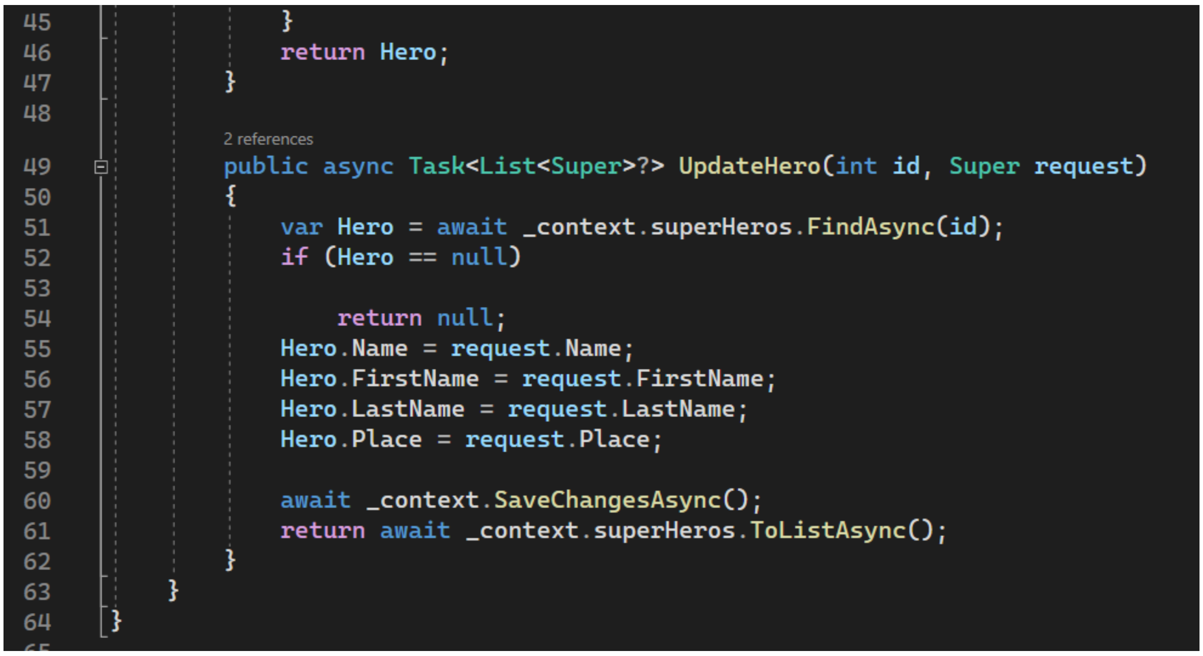
MyExp_1_API   MyExp_1_API.Services.SuperServices.superHeroService

```csharp
using Microsoft.AspNetCore.Http.HttpResults;
using MyExp_1_API.Service.SuperServices;

namespace MyExp_1_API.Services.SuperServices
{
    2 references
    public class superHeroService : ISuperHeroService
    {

        private readonly DataContext _context;

        0 references
        public superHeroService(DataContext context)
        {
            _context = context;
        }
        2 references
        public async Task<List<Super>> AddHeros(Super Hero)
        {
            _context.superHeros.Add(Hero);
            await _context.SaveChangesAsync();
            return await _context.superHeros.ToListAsync();
        }

        2 references
        public async Task<List<Super>?> DeleteHero(int id)
        {
            var Hero = await _context.superHeros.FindAsync(id);
            if (Hero == null)

```


DataContextModelSnapshot.cs   20230723032846...itialCreate.cs   DataContext.cs   superHeroService.cs   Program.cs

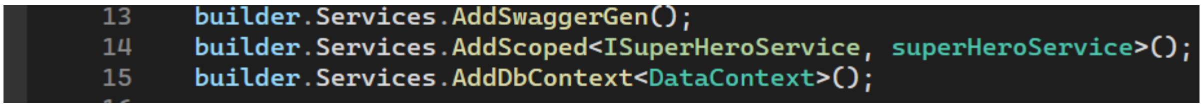
MyExp_1_API   MyExp_1_API.Services.SuperServices.superHeroService

```csharp
            if (Hero == null)

                return null;
            _context.superHeros.Remove(Hero);
            await _context.SaveChangesAsync();
            return await _context.superHeros.ToListAsync(); ;
        }

        2 references
        public async Task<List<Super>> GetAllHeros()
        {
            var heroes = await _context.superHeros.ToListAsync();
            return heroes;
        }

        2 references
        public async Task<Super?> GetSingleHero(int id)
        {
            var Hero = await _context.superHeros.FindAsync(id);
            if (Hero == null)
            {
                return null;
            }
            return Hero;
        }

        2 references
        public async Task<List<Super>?> UpdateHero(int id, Super request)
        {
```



```
            }
            return Hero;
        }

        2 references
        public async Task<List<Super>?> UpdateHero(int id, Super request)
        {
            var Hero = await _context.superHeros.FindAsync(id);
            if (Hero == null)

                return null;
            Hero.Name = request.Name;
            Hero.FirstName = request.FirstName;
            Hero.LastName = request.LastName;
            Hero.Place = request.Place;

            await _context.SaveChangesAsync();
            return await _context.superHeros.ToListAsync();
        }
    }
}
```

Step8:
Add this in Program.cs file for Data Context and Scoped



```
builder.Services.AddSwaggerGen();
builder.Services.AddScoped<ISuperHeroService, superHeroService>();
builder.Services.AddDbContext<DataContext>();
```

Step8: **Goto View Menu → Other Windows → Package Manager Console.**

And type a commands

For checking all the packages is install in our project or not

>> dotnet ef

If it shows an error message then install it.

>>dotnet tool install --global dotnet -ef

Then check it.

Step9: Right Click on project file in solution explorer Click on Manage **Nuget Package Manager**

Goto Browse

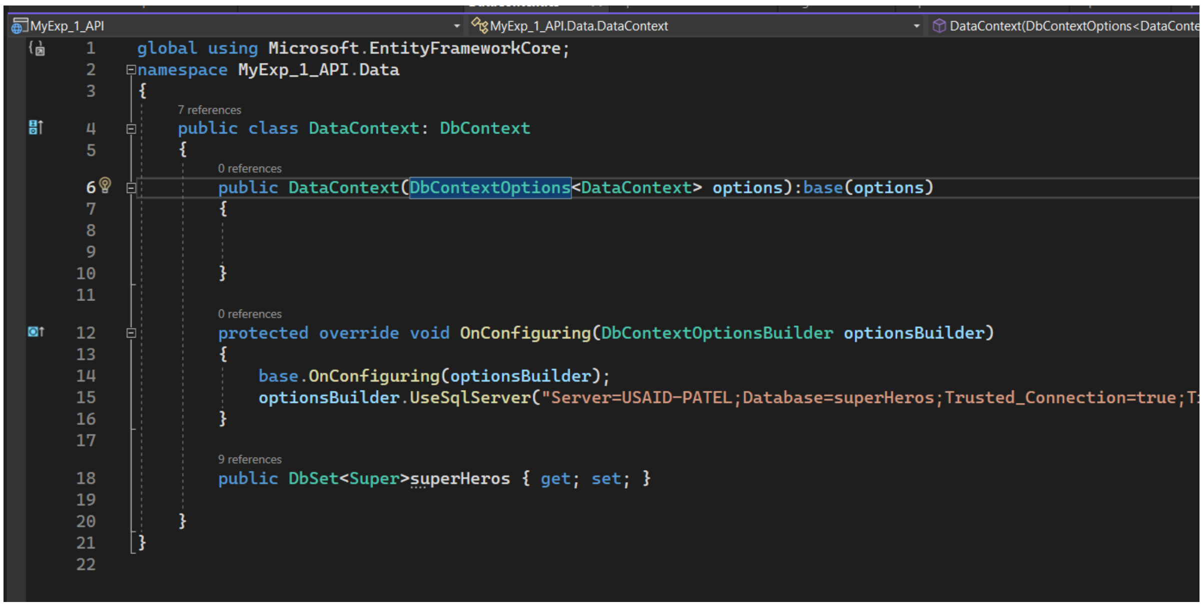
Search for:

Microsoft.EntityFrameworkCore
Microsoft.EntityFrameworkCore.Design
Microsoft.EntityFrameworkCoreSqlServer

Install all.

Step10: Add Folder **“Data”**

Right Click on Data folder and add the new Class Name “DataContext”.



MyExp_1_API | MyExp_1_API.Data.DataContext | DataContext(DbContextOptions<DataConte

```
global using Microsoft.EntityFrameworkCore;
namespace MyExp_1_API.Data
{
    7 references
    public class DataContext: DbContext
    {
        0 references
        public DataContext(DbContextOptions<DataContext> options):base(options)
        {


        }

        0 references
        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            base.OnConfiguring(optionsBuilder);
            optionsBuilder.UseSqlServer("Server=USAID-PATEL;Database=superHeros;Trusted_Connection=true;T
        }

        9 references
        public DbSet<Super>superHeros { get; set; }

    }
}
```

Step11: Goto Package Console Manager again type commands

>>ls //it show list of folder and files

>>cd project folder // in this case Emplyee

>>cd Employee

>>dotnet ef migrations add InitialCreate //It will create schema for Database.

Step12 : dotnet ef database update

Step13: Run the project. Check all the operations