.NET Course

Understand file Structure

<PROGRAM.CS>

```
namespace MyApp
{
   public class Program
   {
      public static void Main(string[] args)
      {
         var builder = WebApplication.CreateBuilder(args); //Creating web Application Builder

         builder.Services.AddControllersWithViews();        // Add services to the container.
```

←-- builder.Services.AddDbContext<MyAppContext>

```
    (options =>
options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnectionString")));
```

→//Add this code after create connection string and create AddDbContext file after
using Microsoft.EntityFrameworkCore; check auto import it

```
         var app = builder.Build();   //Building the application

         // Configure the HTTP request pipeline.
```

←-------- handling errors

```
         if (!app.Environment.IsDevelopment())
         {
```

```
            app.UseExceptionHandler("/Home/Error");
            // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
            app.UseHsts();
        }  ---------->

        app.UseHttpsRedirection();
        app.UseRouting();

        app.UseAuthorization();
        app.MapStaticAssets(); //Enabel static files (img,css,js)

←========Configure default route
        app.MapControllerRoute(
            name: "default",
            pattern: "{controller=Home}/{action=Index}/{id?}")
            .WithStaticAssets();  ---------->

        app.Run();   //Run App
    }
  }
}
```

Then Crete Module

```
namespace MyApp.Models
{
  public class Item
  {
```

```
    public int id { get; set; }
    public string name { get; set; }

  }
}
```

First install Nuget packages Entity framework(Core,Sql,Tools and Design) Then using the Server Explora create the connection and get connection String

Create one (MyAppContext) class for database (Module or anywhere) and inherit DbContext (I Create This Create Data Folder and inside that)

```
using Microsoft.EntityFrameworkCore;
using MyApp.Models;

namespace MyApp.Data
{
  public class MyAppContext : DbContext
  {
    public MyAppContext(DbContextOptions<MyAppContext> options) : base(options) //create this
    {
    }

    public DbSet<Item> Items { get; set; } //Add Existind modules
  }
}
```

2)After Add this code in Program.cs

```
builder.Services.AddDbContext<MyAppContext>
  (options =>
options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnectionString")));
```

1)And Connection string on Aappsetting.json file set connection string

```
"AllowedHosts": "*",

"ConnectionStrings": {

  "DefaultConnectionString": "Data Source=LAPTOP-S3G4E30I\\SQLEXPRESS01;Initial Catalog=myapp_db;Integrated Security=True;Connect Timeout=30;Encrypt=False;Trust Server Certificate=True;Application Intent=ReadWrite;Multi Subnet Failover=False"

}
```

After that go to package manager console and type Add-Migration “Initial Migration” after successfully creating Migration folder and inside Initial Migration file correctly generate sql commands for our Modules and Then Type again Update-Database After this our table create on Database

---

CRUD

Ug

One to one

```
using System.ComponentModel.DataAnnotations;


namespace example.Models

{

   public class Student

   {

      [Key]

      public int Id { get; set; }

      public string Name { get; set; }


      public ClassRoom? ClassRoom { get; set; }

      public List<Teacher> Teachers { get; set; } = new List<Teacher>();

   }

}
```

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace example.Models
{
   public class ClassRoom
   {
      [Key]
      public int Id { get; set; }
      public string Name { get; set; }
      [ForeignKey("Student")]
      public int StudentId { get; set; }
      public Student? Student { get; set; }
   }
}
using example.Models;
using Microsoft.EntityFrameworkCore;

namespace example.Data
{
   public class DataContext :DbContext
   {
      public DataContext(DbContextOptions<DataContext> options) : base(options)
      {
      }
      public DbSet<Student> Students { get; set; }
      public DbSet<ClassRoom> ClassRooms { get; set; }
      public DbSet<Teacher> Teachers { get; set; }
```

```
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<Student>()
        .HasOne(c => c.ClassRoom)
        .WithOne(cd => cd.Student)
        .HasForeignKey<ClassRoom>(cd => cd.StudentId)
        .OnDelete(DeleteBehavior.Cascade);



        modelBuilder.Entity<Student>()
          .HasMany(w => w.Teachers)
          .WithOne(c => c.Student)
          .HasForeignKey(c => c.StudentId)
          .OnDelete(DeleteBehavior.Cascade);




    }
  }
}
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace example.Models
{
  public class Teacher
  {
    [Key]
    public int Id { get; set; }
    public string Name { get; set; }
```

```
    [ForeignKey("Student")]
    public int StudentId { get; set; }
    public Student? Student { get; set; }
  }
}
```

```
dotnet tool install --global dotnet-ef
dotnet ef –version
dotnet ef migrations add InitialCreate
dotnet ef database update
```