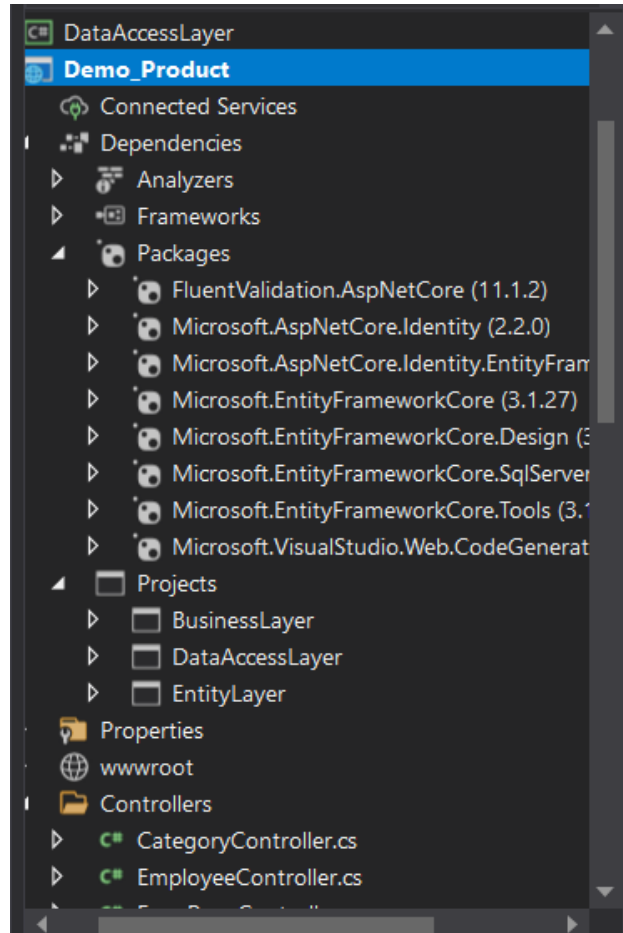


Projemizin Kurulum Aşaması ve Çalıştırılması

Projemizde Microsoft Visual Studio Community 2019 IDE'sini kullandık. Sebebimiz ise oluşturduğumuz ASP.NET Core WEB Application projelerine uyumlu ve stabil çalışması, kullanım kolaylığı, intellisense gibi özelliklerini sıralandırabiliriz. IDE kurulumumuzdan sonra ise Visual Studio Installer ile ASP.NET ve web geliştirme, .NET masaüstü geliştirme paketlerinin kurulu olmasına da dikkat ediyoruz.

Projemiz boyunca tam destekli olduğu için ASP.Net Core 3.1.27 kullandık. Bu sebeple kullandığımız IdentityServer paketleri, EntityFramework paketlerinde de aynı şekilde 3.1 sürümlerini tercih ettik.

Projemiz N katmanlı mimari yapısında oluşturulduğu için dosyalarımızı indirdikten sonra katmanlarımızın kurulumu, EntityFramework kütüphanelerinin kurulumu gerekmektedir. Bu işlemi ise Nuget Packages kısmından yapıyoruz.



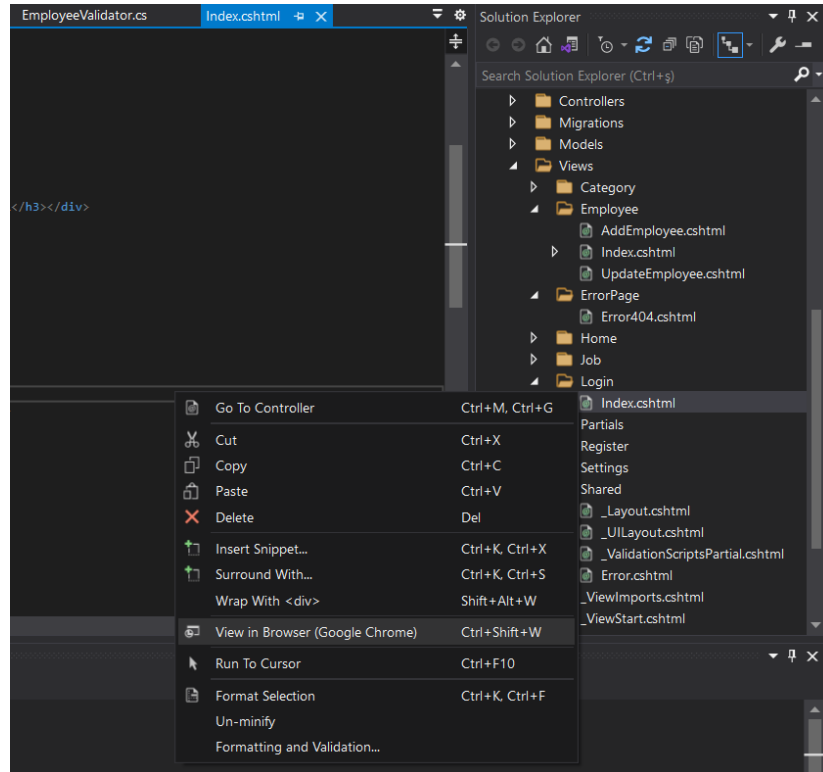
Şekil 1.1: Packages kurulumunu yaptık.

Kurulum yapmamızın yanısıra diğer katmanlardaki verilerimizi kullanabilmemiz için projelerimizi yani katmanlarımızı birbirine referans verecek şekilde ayarlamalıyız.

Veritabanı işlemlerimiz içinse Microsoft SQL Server Management Studio 2018 kullandık. Server başlatırken seçtiğimiz portu ise Veri Erişim Katmanında oluşturduğumu Concrete sınıfında yol olarak verdik ve bu şekilde bağlantıyı kurduk.

```
Context.cs Index.cshtml Error404.cshtml LoginController.cs CategoryValidator.cs ProductValidator.cs EmployeeValidator.cs
DataAccessLayer.DataAccessLayer.Concrete.Context
1 using EntityLayer.Concrete;
2 using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
3 using Microsoft.EntityFrameworkCore;
4 using System;
5 using System.Collections.Generic;
6 using System.Text;
7
8 namespace DataAccessLayer.Concrete
9 {
10     public class Context: IdentityDbContext<AppUser, AppRole, int>
11     {
12         /*Veritabanı bağlantımızı burada yaptığımız için Veritabanımızı oluştururken ya da herhangi bir
13         veri göçü durumunda Package Manager Console üzerinden Default Project kısmında
14         DataAccessLayer seçtik.*/
15
16         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
17         {
18             optionsBuilder.UseSqlServer("server=DESKTOP-PUSH3QC\\MSSQLSERVER01;database=TelefonRehberi;integrated security=true;");
19             //database configürasyonu yaptık
20             //Veritabanımıza Telefon Rehberi ismini verdik ve SQL Server Management'ı açarken bu server'a bağlandık.
21
22             public DbSet<Product> Products { get; set; }
23
24             public DbSet<Employee> Employees { get; set; }
25
26             public DbSet<Category> Categories { get; set; }
27
28             public DbSet<Job> Jobs { get; set; }
29
30             /* Veri göçünde add-migration mig_ornek dedik ve
31             update-database komutu ile verilerimizi tablolarda güncelledik*/
32         }
33     }
34 }
```

Şekil 1.2: Veri Tabanı Bağlantısı



1.3: Projemizi başlatmak içinse Login Index dosyamıza girip sağ tık-> view in browser seçeneğini seçiyoruz