YAPILANLAR

Uzun zamandır rapor atmadığım zamanlarda yaptıklarım şu şekildeydi:

Kendim ürün ekleyerek filtreleme işlemlerini halletmem gerekiyordu. Ben de bu durum için elimle teker teker eklemek yerine veri tabanı araştırması yaptım. Birçok kanaldan dokümanlardan yaptığım araştırmalar sonucunda SQLite veri tabanını kullanmaya karar verdim. Bu veri tabanını artıları şu şekildeydi:

1- Tabloları direkt olarak manuel oluşturarak projeye ekleme yapabiliyorum.
2- SQL sorgularını çalıştırabiliyordum.
3- CSV'den veri import etme işlemlerini kolaylıkla yapabiliyorum.
4- Veri tabanında eklediğim veya çıkardığım daha doğrusu yapılan bütün işlemlerin anında projeye yansıması gerçekleşiyor.

Başlangıç olarak bir sunucu vs. kullanmayacağım için projenin bu zamana kadar yapılan tüm durumlara en uygun olarak SQLite veri tabanını kullanmak hem daha avantajlı hem de daha kolaylaştırdı bütün işleri.

Peki bu veri tabanını bağlarken neler ile uğraştım?

Öncelikle eğitimleri izlemem doküman taramam çok uzun sürelerimi aldı. Daha sonra kullanmam gereken veri tabanı elemanlarını detaylı olarak oluşturmam biraz zamanımı çaldı. Bu projede filtrelerde kullanılacak her şeyi not ettikten sonra öncelikle veri tabanımı bağladım. Bu noktada NUGET paketini indirdim.

Microsoft.EntityFrameworkCore.Sqlite

Bunu yüklemek için de

dotnet add package Microsoft.EntityFrameworkCore.Design

Komutunu kullandım. Böylelikle paketimizi başarılı şekilde indirdik. Daha sonra *"dotnet -ef"* bilgisayarımda yüklü olmadığı için onu indirdim.

dotnet tool install --global dotnet-ef

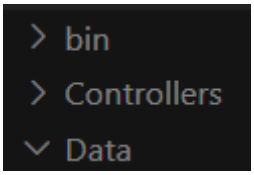
Bunu da indirdikten sonra bazı dosyalar oluşturup bazı dosyaları da düzenlemem gerekiyordu.

dotnet ef migrations add InitialCreate

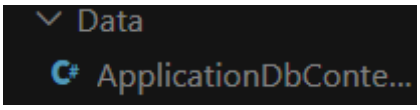
dotnet ef database update

Kodlarını sırasıyla yaptım. Hataları düzeltmek için ise;

Öncelikle yeni bir klasör oluşturdum. İsmini de Data olarak adlandırdım.



Daha sonra bu Data klasörümün içerisine ApplicationDbContext.cs isimli dosya oluşturdum.



Bu dosyanın içeriğini de şu şekilde düzenledim:

```
using Microsoft.EntityFrameworkCore;

namespace Dropshipping.Data
{
    5 references
    public class ApplicationDbContext : DbContext
    {
        0 references
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
            : base(options)
        {
        }

        // Burada veritabanı tablolarınızı tanımlayacaksınız
        // Örnek:
        // public DbSet<Product> Products { get; set; }

    }
}
```

Bu kısımda önemli bir kısım var. **Namespace** ismimizi daha sonra kullanacağız. İsmimi Dropshipping.Data olarak tanımladım.

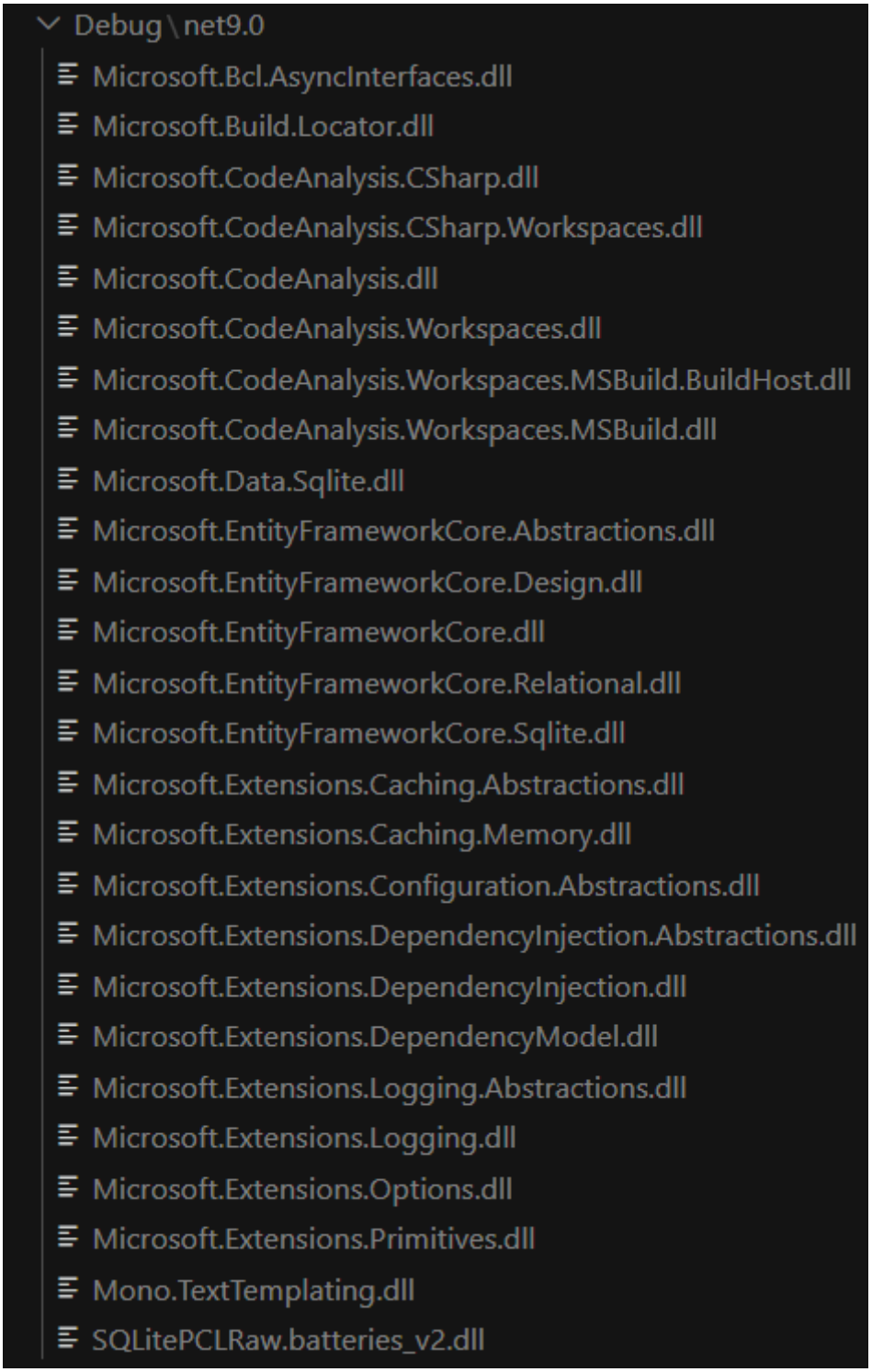
Daha sonra appsettings.json dosyamızı düzenlememiz gerekiyordu.

```
{} appsettings.json > ...
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=Dropshipping.db"
  }
}
```

Burada yaptığımız şey eğer yoksa Dropshipping.db klasörü oluşturmak varsa da her seferinde bağlantı kurmak ile uğraşmak istediğimiz için burada tanımladık. Böylelikle uygulamayı başlattığımız zamanda ilk olarak appsettings.json başlatıldığı için veri tabanında yapılan tüm değişiklikler güncellenerek işlemlere öyle devam edilecek.

Veri tabanı bağlama işleminde en zor ve önemli kısmı ise Program.cs dosyamızda yazacağımız eklememiz gerekenlerdi.

Bu noktada NUGETpaketini indiripo SQL dosyalarını da tam anlamı ile kurduktan sonra:



Bu işlemlerin ardından ilk olarak

```
using System.Globalization;
using Microsoft.EntityFrameworkCore;
using Dropshipping.Data;
```

Komutlarını ekledik. Burada Dropshipping.Data namespace'mizi de ekleyerek veri tabanı bağlamak için kullanılacak tüm dosyaların bağlantısı kurduk.

VE SON OLARAK DA:

```
builder.Services.AddControllersWithViews();

builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseSqlite(builder.Configuration.GetConnectionString("DefaultConnection")));

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

Kodumuz ile başarılı bir şekilde veri tabanı bağlantımızı kurmuş olduk. Şimdi projemizi çalıştıracağız eğer her şey doğru bir şekilde tamamlanmışsa Dropshipping.db adında dosyamız kendiliğinden oluşturacaktır.

Ad	Durum	Değiştirme tarihi	Tür	Boyut
Models		11.03.2025 16:06	Dosya klasörü	
obj		17.05.2025 13:47	Dosya klasörü	
Properties		11.03.2025 16:06	Dosya klasörü	
Views		11.03.2025 16:06	Dosya klasörü	
wwwroot		11.03.2025 16:06	Dosya klasörü	
appsettings.Development.json		6.03.2025 14:00	JSON	1 KB
appsettings.json		17.05.2025 13:36	JSON	1 KB
BitirmeProjem.csproj		17.05.2025 13:40	VisualStudio.Launc...	1 KB
BitirmeProjem.sln		6.03.2025 15:20	Visual Studio Solut...	2 KB
Dropshipping		17.05.2025 13:48	Data Base File	16 KB
Dropshipping.db-shm		17.05.2025 13:49	DB-SHM Dosyası	32 KB
Dropshipping.db-wal		17.05.2025 13:49	DB-WAL Dosyası	0 KB
Program.cs		17.05.2025 13:46	C#	2 KB

Ve dosyamız burada oluşturuldu. Her şey tamamlandı.

Peki bundan sonra neler yapılacak?

• Web scraping ile denemeler yaparak veri çekme işlemleri denedim. Lakin burada karşılaştığım sorun AMAZON'un veri çekme işlemlerinde Proxyler ile engellemeler yaptığını gördüm.

Bu durumdan dolayı bu zamana kadar biraz yavaş kalmış olduk. Şimdi tüm verileri veri tabanında manuel olarak kullanılacak şeyler eklenecek.

• Daha sonra burada dosyaların içerisinde gerekli filtrelemeler nereye eklenecekse o kısma yönlendirmeler yapılacak ve ürün listelemeleri aktarılacak.

•Ürün listelemeleri uygun filtrelemelerden geçerken sorunlar olursa bu sorunların çözümü ile ilgili araştırmalar, eklemeler, çıkarmalar yapılacak.

• Ürünler başarılı bir şekilde aktarımları yapılırsa gerçek ürün verileri ile denemeler yapılacak.

• Satış grafikleri manuel olarak eklenip görselleştirmeler eklenecek.

• Bir yandan da API bağlantısı kurabilmek için araştırmalar ve görüşmeler devam edecek.

• Yazılımın demosu oluşturulacak ve kullanım için testlere bakılacak.

• Çıkan sorunlar ile uğraşılacak.

• Tüm sorunların çözümü bulunacak.

• Her şey istenildiği gibi olduğu vakitte projemiz son aşamaya aktarılacak.

• Son aşama olarak da testler yapılacak ve proje artık hazır hâle gelmiş olacak.