Proje Raporu

Proje

|  |  |  |  |
| --- | --- | --- | --- |
| Rapor Tarihi | Proje Adı | Hazırlayan: | |
| 22 Haziran 2022 | Depo Ürün Yönetim Sistemi | | Sezer VAROL  Gökhan HASBAL |

PRoJE ÖZETİ

Bir şirketin dışarıdan aldığı ürünleri veya kendi ürünlerini depolayabileceği ve sevkiyat yapabileceği, VarBal Storage isimli bir MVC projesidir. Sezer **Var**ol ve Gökhan Has**bal** tarafından 31 Mayıs – 23 Haziran arasında geliştirildi.

Projenin bölümleri

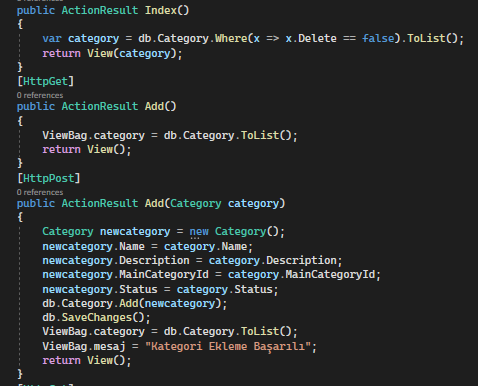
|  |  |  |
| --- | --- | --- |
| Görev | yapılan işlemler | |
| MVC , ENTİTYFRAMEWORK  YAPISININ OIUŞTURULMASI | | Visual Studio programıyla Asp.Net Web Applicaition .net framework projesi oluşturuldu . | |  |
| TEMPLATE ARAŞTIRILMASI | Themeforest’ten kendi projemize uygun taslak bulundu. | | |  |
| TEMPLATE’İNPROJEYE AKTARILMASI | Entity framework’un içinde default halde gelen Content sayfasının tüm içeriği silinip template ait dosyalar buraya aktarıldı ve tüm href ,src linkleri kendi kullanacağımız adreslere göre güncelledik , yine default halde gelen fonts ve js dosyaları silindi projeden, ki mevcut olanla karışmasın. | | |  |
| TEMPLATE’İN ÇALIŞIR HALE GETİRİLMESİ | Templatein çalışır hale gelmesi için öncelikle ana indexi belirleyip index sayfasından bazı elemeler yapılması gerekmektedir. Bu elemeler neticesinde tüm sayfaları içinde barındırıcak olan layout sayfası oluşturulur ve bu sayfada navbar ve footer vardır. | | |  |
| LAYOUT YAPISININ ŞEKİLLENDİRİLMESİ | Önce kullanılacak olan navbar yapısı ve footer kodları belirlenerek sayfaya enjekte edilir. @renderbody’in altı ve üstü kullanılır renderbodynin üstü navbar altı footer yapısını barındırır. | | |  |
| MODEL NESNELERİNİN SİSTEMATİK ŞEKİLDE ADLANDIRILMASI | Kullanacağım model’ler oluşturup foreign key bağlantıları oluşturuldu içermesi gereken bilgiler aktarıldı. Bu modeller neticesinde controllerlar oluşturuldu. | | |  |
| CONTROLLERLARIN OLUŞTURULMASI | Kullandığımız modeller neticesinde controllerlar oluşturulup isimlendirilmesi yapıldı ve bu controllerların barındırdığı sayfalar tek tek oluşturuldu. Sayfalara modeller çağırılıp gerekli ilişkilendirilmeler yapıldı. | | |  |
| VERİTABANI İŞLEMLERİNİ ADIMLARI | Veri tabanı işlemleri için webconfig sayfasında yapılması gereken birkaç adım mevcuttur <connectionstring> html tadıyla database adı vs. tanımlanır. Datacontext yapısı gene burada tanımlanır datacontext i tüm sayfalardaki veri işlemlerinde kullanırız. O yüzden sayfalara çağırdığımız kütüphanelere falan dikkat etmemiz gerekmektedir. | | |  |
| VERİ TABANININ OLUŞTURULMASI | Projede kullanmam gereken veritabanı için modeller sayfama bir datacontext modeli oluştururuz bu datacontext modelininin içinde dbset kodu ile modelleri database’e set ederiz. | | |  |
| VERİ TABANININ OLUŞTURULMASI | Package Manager Console’dan migration yapısı çağırılır.  1-)Enable-Migrations  2-)add-migration Varbal  3-)add-migration Varbal  4-)update-database yapılarak veritabanı oluşturulur. | | |  |
| VERİ TABANINDAKİ TABLOLAR | About,About Feature,Address,Booking,Brand,Category,Contact Us,Invoice,Logs,NewsGet,Order,Our Worker,Permission,Seo,Shelf,Shipment,User,Vehicle,Vision And Mission ,Product,Slider, ,Contact tabloları mevcuttur bunlar her biri modellerin içinde hangi columnları olacaksa yazılıp kodlandırılması gerçekleşir properity yapılarıyla. | | |  |
| SAYFA YAPILARININ OLUŞTURULMASI | Controller’daki action result’ta bulunan sayfalar kullanmamız gerektiği gibi (varsa adminlayout, userlayout, layout veya layoutsuz olarak oluşturulur). | | |  |
| SAYFALARIN DÜZENLENMESİ | Hangi sayfada ne yapılması isteniyorsa ona clean kod mantığıyla sayfalarımız kodlanır ve sayfamızda kullanacak veritabanından gelecek veriler varsa modeller yardımıyla çağırılır. | | |  |
| VERİ EKLEME,GUNCELLEME İŞLEMLERİNİN OLUŞTURULMASI | Gerekli controller’larla ekleme silme düzenleme işlemleri gerçekleştirme işlemleri yapılmaktadır. | | |  |
|  |  | | |  |



Projemizde Kod Yapılarımıza Örnekler Aşağıda Gösterildiği Gibidir.

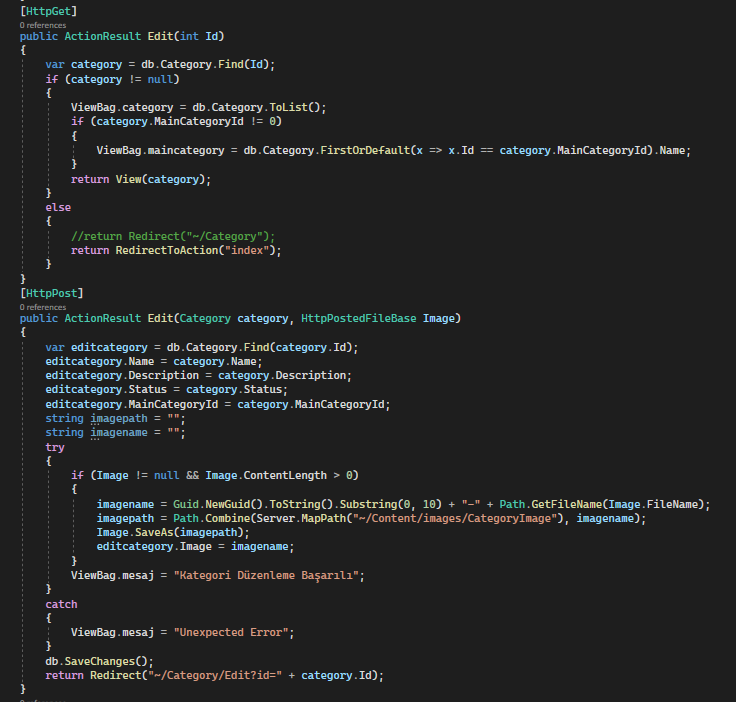
CRUD (create-read-update-delete) işlemlerimize örnek

Listeleme ve değer ekleme

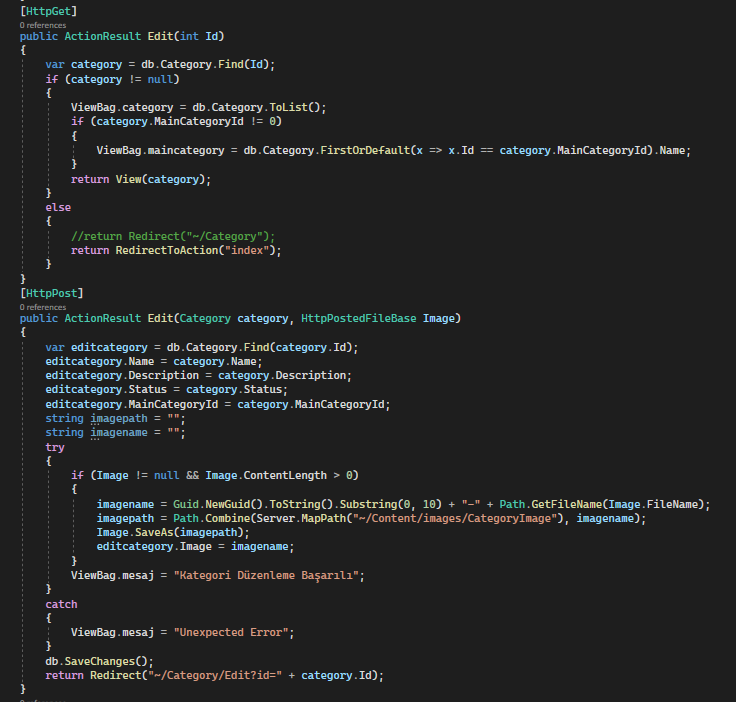


Basit, birden fazla tablo bilgileri gerektirmeyen actionlarımız için, uygun modeller ve LINQ sorgularıyla bilgiler listelendi.

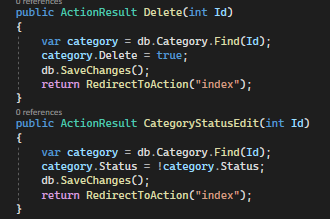
ekleme ksıımlarında bu tablolar, formlar aracılığıyla alınıp, database'imize kaydedildi.



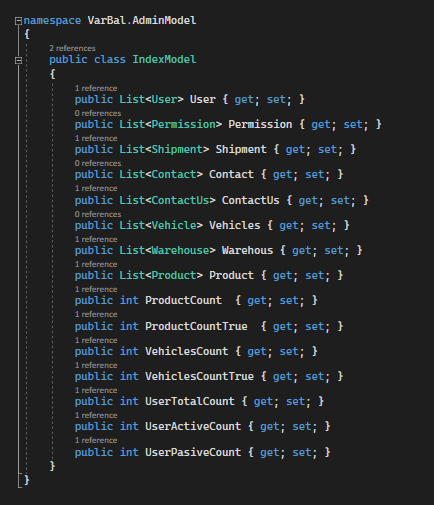
Düzenleme kısmında ise, düzenlenecek tablo girdisinin id'si tutuldu, Find methodu ile bulunup cshtml kısmında yine form vasıtasıyla yeni girdiler alınarak database güncellendi. bazı controllerlara gösterildiği gibi fotoğraf eklenme özelliği konuldu.

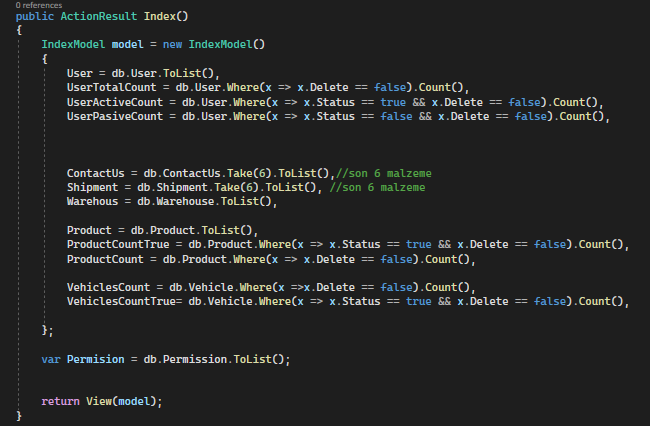


Tek buton tıklanmasıyla verilerin status değişimi ve delete işleminin yapılması için uygun actionlar açılıp, index.cshtml içerisine yerleştirildi.

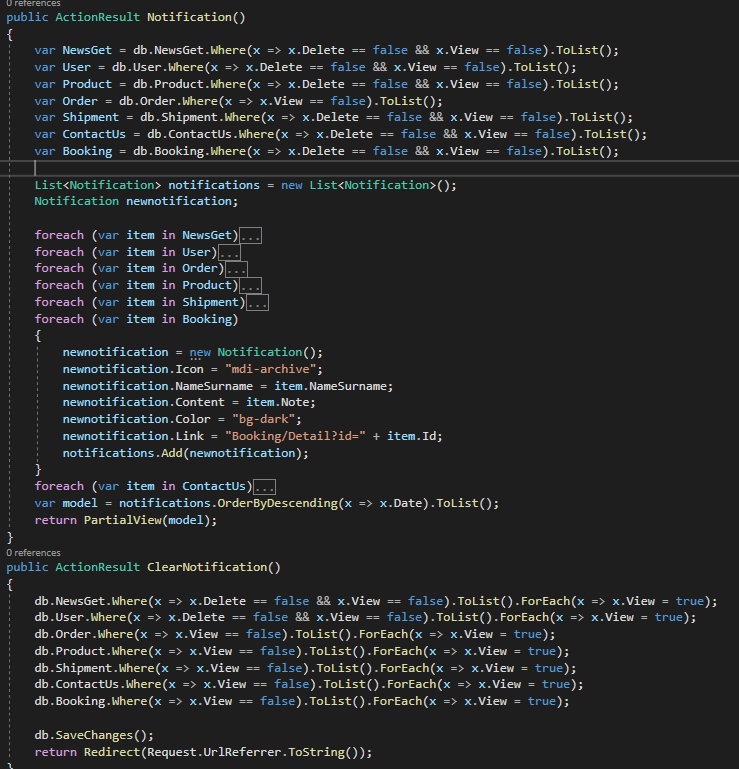


Daha detaylı sayfalar için, şekilde görüldüğü gibi çok amaçlı modeller oluşturuldu yine uygun sorgularla bu modeller doldurulup, ilgili yerlerde kullanılmak üzere view ile cshtml kısmına aktarıldı

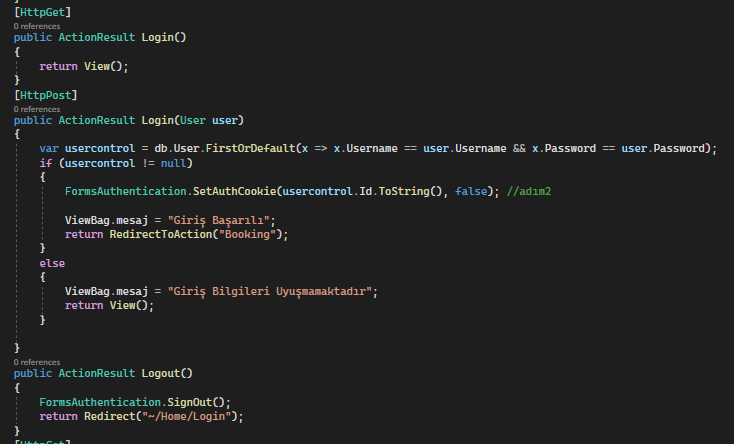




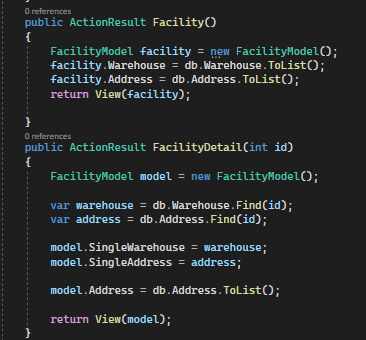
Admin panelde bulunan notification kısmı işlevsel hale getirildi, ve görülmesi istenen verilerin sorgular yardımıyla çağrılması sağlandı. bu bildirimlerin, tıklandığında ilgili sayfanın detayına gitmesi, ve view değerlerinin otomatik değiştirilip bu bildirim listesinden düşmesi sağlandı. Notification kısmına, bütün bildirimleri tek tıklama ile kaldıracak bir buton tasarlandı.



Kullanıcı işlemleri için yaratılan giriş ve çıkış action içerikleri. Kullanıcı adı ve şifre ile giriş yapılması uygun görüldü.



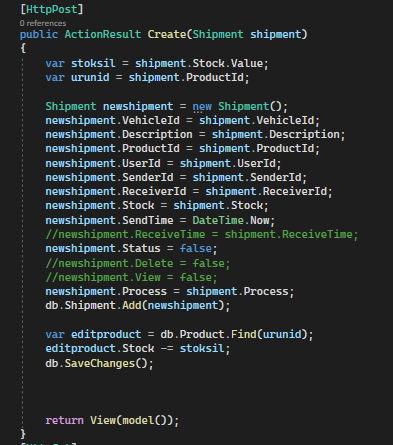
Bütün depolarımızı listeyen bir facility action'ı ve bu sayfa içinde, tıklanıldığında ilgili depo'nun detayına giden facilitydetail action ve sayfası düzenlendi.



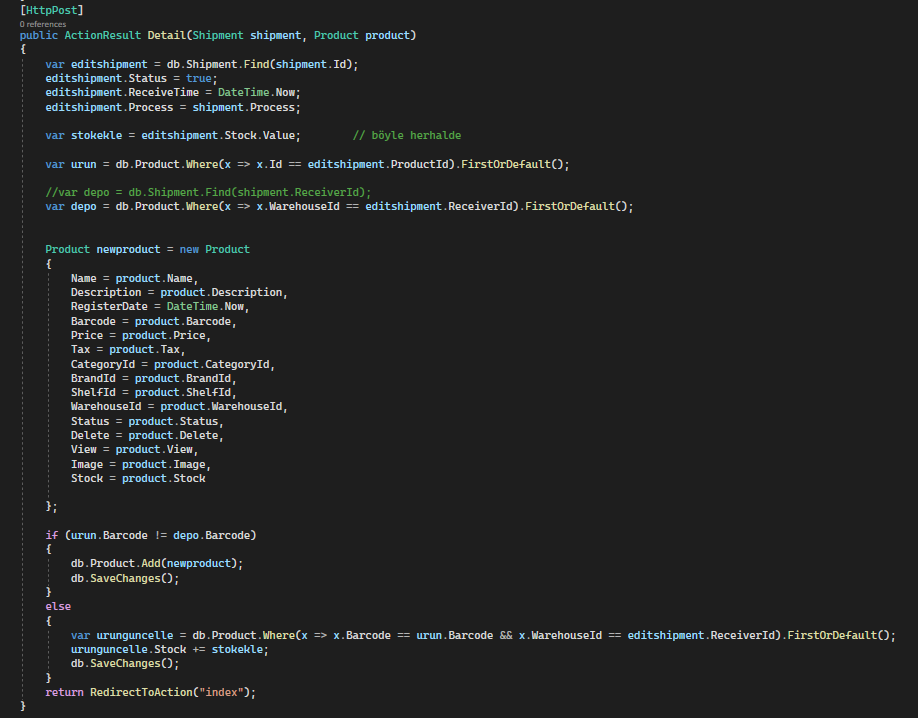
Üye kaydı ile birlikte, üyenin mail adresine bu üye için VarBal tarafından tasarlanmış bir mail yollamak için Smtp yapısı kullanıldı.



Sevkiyat yaratıldığında, ürün sevkiyatın başladığı deponun stokundan azaltıldı, ilgili bilgiler sevkiyat tablosuna aktarıldı.



Alıcı kişi, bu sevkiyatın detay sayfasına girip ürünü kabul ettiğinde. Bu ürün, bu depoda yoksa, ürün eski depo bilgileriyle bu depoya eklendi, eğer bu depoda var ise ürün stoku güncellendi. Kontrol id'ler ve her ürünün kendine özgü VarBal barkod numarası ile yapıldı.



Proje Ortakları

