



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

WEB PROGRAMLAMA PROJE RAPORU

HAYVAN BARINAĞI

B201210077 - Muhammed Emin ÇETİN - 1 A Grubu

https://github.com/muhammedcetinn/net_core-hayvan_barinagi_uygulamasi

SAKARYA

Ağustos, 2023

WEB PROGRAMLAMA Dersi

Hayvan Barınağı

Muhammed Emin Çetin, B201210077

1.Öğretim A grubu

Özet

Proje ilk başlatıldığında giriş yapmayan kullanıcılar sadece veritabanındaki hayvanları görüntüleyebilmektedir. Giriş yapmış normal kullanıcılar hayvanları listeleyebilir, sahiplenmek için talep oluşturabilir, barınağa yeni hayvan vermek için başvurabilir ve taleplerini takip edebilir. Giriş yapmış admin kullanıcıları hayvanları ve hayvan türlerini listeleyebilir ve bunları düzenleyebilir veya silebilir. Admin kullanıcılar, normal kullanıcıların taleplerini görüntüleyip bunları değerlendirebilir.

© 2023Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Herhangi bir kopya işleminde sorumluluk bana aittir.

Anahtar Kelimeler: asp.net 7, SQLServer, Identity, Localization, LINQ

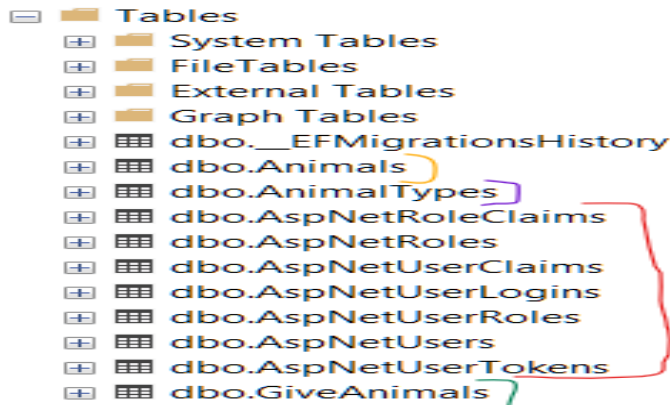
1. GELİŞTİRİLEN YAZILIM

1.1. Amaç

Identity, localization, linq, entity framework vb. kullanarak ayağa kalkabilen hayvan barınağı web projesi geliştirilmesi.

1.2. Algoritma

Kullanıcılar “Register” bölümünden üye olur, “Login” bölümünden giriş yapar. İşlemleri sağlayan Identity sınıfıdır. Admin veya customer olmak üzere iki tip yetki vardır. Yetkilendirme veritabanından yapılır. Adminler veritabanındaki tüm tabloları site içerisinde görebilir. Adminler kullanıcı bilgilerine doğrudan erişemez, sadece kullanıcı talep oluşturmuşsa talebi değerlendirebilmek için kullanıcı hakkındaki gerekli “hakkında” bilgisi görüntülenebilir. Veritabanında Identity tabloları, Animals tablosu, GiveAnimal tablosu ve AnimalTypes tablosu bulunmaktadır.



1.2.1. Animals Tablosu ve Animals / Requests Sekmeleri

Id	Name	Age	Features	AnimalTypeId	ImageUrl	isRequest	Status	Recipient	RecipientAbout
1	Hayv...	1	Çok tatl...	1	\img\logo....	False	True	customer...	Ben bir kullanıcı...
2	Hayv...	2	Çok tatl...	2	\img\logo....	False	False	'NULL'	'NULL'
3	Hayv...	3	Çok tatl...	3	\img\logo....	False	False	'NULL'	'NULL'
4	Hayv...	4	Çok tatl...	4	\img\logo....	False	False	customer...	Ben bir kullanıcı...
5	Hayv...	5	Çok tatl...	1	\img\logo....	False	False	'NULL'	'NULL'
6	Hayv...	6	Çok tatl...	2	\img\logo....	False	False	'NULL'	'NULL'
7	Hayv...	7	Çok tatl...	3	\img\logo....	False	False	'NULL'	'NULL'
8	Hayv...	8	Çok tatl...	4	\img\logo....	True	False	customer...	Ben bir kullanıcı...
9	Kulla...	1	Çok tatl...	1	\img\logo....	False	False	null	null
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Hayvanların genel tanımları bu tabloda bulunmaktadır. “isRequest” ifadesi bir “Customer” ın ilgili hayvana talebi olup olmadığını gösterir. “Status” sütunu ise bu talebin durumunu ifade eder. Kontrol şu şekilde gerçekleşir:

Bir kullanıcı sahiplenme oluşturduğunda “isRequest” “true” olur ve “Recipient” sütununa sahiplenmek isteyen kullanıcının username’i, “RecipientAbout” sütununa ise kullanıcının hakkında bilgisi işlenir. “isRequest” sütunu “True” olan hayvanlar ana sayfada listelenmez. Bu talepler Admin tarafına iletilir. Admin talebi onaylarsa “Status” sütunu “True” olur ve kullanıcı talep sayfasında “Onaylandı” mesajını görüntüler. Admin talebi reddederse “isRequest” sütunu tekrar “false” yapılır ve “Recipient” kısmında username’i yazan kullanıcı bunu “Reddedildi” olarak görüntüler. Bu işlemler “AnimalController” içerisindeki “Create/Delete/Edit-Request” aksiyonlarında yapılmaktadır.

Giriş yapmış kullanıcının ekran görüntüsü:

 **Animal Shelter**

Animals | My Requests | Give Animal to Shelter Requests

Hello customer1@gmail.com! | Logout


Animals

[Give Animal to Shelter](#)


Image	Name	Age	Features	Type	
	Hayvan2	2	► Click For Animal Features	Dog	Create request

1 numaralı butondan kullanıcı barınağa hayvan verme talebinde, 2 numaralı butondan ilgili hayvanı sahiplenme talebinde bulunabilir.

Kullanıcı 1 Numaralı sekineden sahiplenme taleplerine, 2 numaralı sekmeden barınağa hayvan verme taleplerine ulaşabilir. Kullanıcılar taleplerini şöyle görmektedir:

Requests					
Image	Name	Age	Features	Type	Status
	KullanıcıHayvanı1	1	► Click For Animal Features	Cat	Accepted

Admin için talepler şöyle görüntülenir:

Requests							
Recipient	Image	Name	Age	Features	Type		
<div>▼ customer1@gmail.com</div> <div>Ben bir kullanıcıyım - Customer 1</div>		KullanıcıHayvanı3	3	▶ Click For Animal Features	Hamster	Accept	Decline

Admin için hayvanlar sayfası şöyle görülmektedir:

Animals						
Image	Name	Age	Features	Type		
	Hayvan2	2	▶ Click For Animal Features	Dog	Edit	Delete
	Hayvan3	3	▶ Click For Animal Features	Rabbit	Edit	Delete
	Hayvan4	4	▶ Click For Animal Features	Hamster	Edit	Delete
	Hayvan5	5	▶ Click For Animal Features	Cat	Edit	Delete
	Hayvan6	6	▶ Click For Animal Features	Dog	Edit	Delete
	Hayvan7	7	▶ Click For Animal Features	Rabbit	Edit	Delete
	KullanıcıHayvanı1	1	▶ Click For Animal Features	Cat	Edit	Delete
Add						

Talepler ve hayvanlar tablosu listelenirken “isRequest” ve “Status” lerin kontrolü view içerisinde if/else yapısıyla sağlanmıştır.

```
@foreach (var animal in Model)
{
    if (animal.isRequest == false && animal.Status == false)
    {
```

Kullanıcı taleplerini kendisi listelerken Animals tablosunda kullanıcıya ait verilerin sorgusu linq ile yapılmıştır.

```
string username = User.Identity.Name;
List<GiveAnimal> animallist = _giveAnimalRepository.GetReq(u => u.Recipient == username, includeProps: "AnimalType").ToList();
return View(animallist);
```

1.2.2. AnimalTypes Tablosu ve Sekmesi

	Id	NameEN	NameTR
►	1	Cat	Kedi
	2	Dog	Köpek
	3	Rabbit	Tavşan
	4	Hamster	Fare
	5	Other	Diğer
*	NULL	NULL	NULL

Bu sekmeyi sadece adminler görüntüleyebilir. Tabloya veri girişi, güncellemesi ve çıkışı sadece adminler tarafından gerçekleştirilir. Bu tabloda veriler iki sütunda tutulur. Sütunlardan birisi verinin türkçe ifadesiyken diğeri ingilizce ifadesidir.

```
Thread.CurrentThread.CurrentCulture.Name.ToString() == "en-US"
```

Hayvan türlerinin gösterildiği sütunlarda kültür kontrolü yapılarak tablodan ilgili dilin verisi çekilir. İlgiliyi sekmeyi admin şöyle görüntüler:

Animal Types

Type		
Cat	Edit Type	Delete Type
Dog	Edit Type	Delete Type
Rabbit	Edit Type	Delete Type
Hamster	Edit Type	Delete Type
Other	Edit Type	Delete Type

Add New Type

Add New Type

Name - TR

Name - ENG

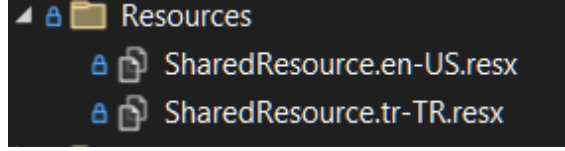
Add New Type

Back to table

Yeni hayvan türü girişi yaparken adminden ilgili tür için hem ingilizce hem türkçe veri beklenir.

1.2.3. Localization

Proje hem türkçe hem ingilizceyi desteklemektedir. Barınaktaki hayvanların özellikleri hariç tüm verilere localization uygulanmıştır. Localization Animaltypes tablosu için veri tabanında çift dilli veri ile sağlanırken sekmeler gibi yerel ifadeler için için proje içerisinde Kaynak(.resx) oluşturulmuştur.

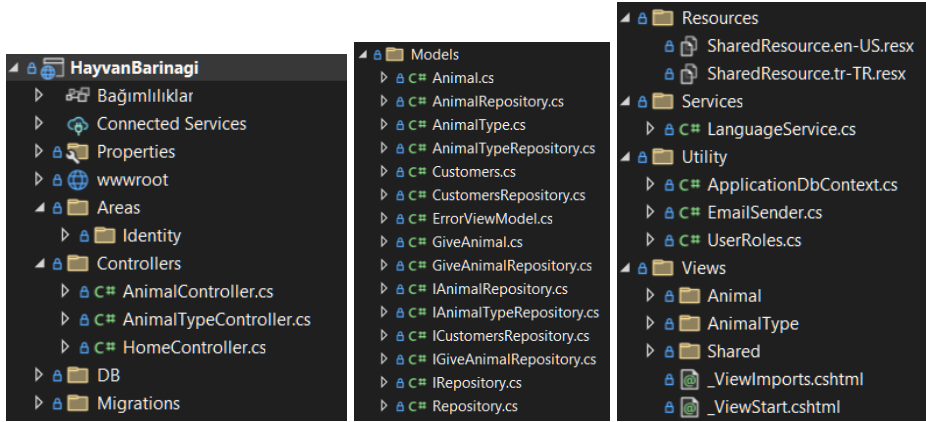


View’de kullanımı şöyledir: `@_localization.GetKey("str_ClickForFeatures")`

.Resx ten örnek görüntü :

	Ad	Değer
str_About	About	Hakkında
str_Accept	Accept	Kabul et
str_Accepted	Accepted	Kabul edildi

1.2.4 Dosya yapısı



1.2.5 Identity

```
//Identity İşlemleri
builder.Services.AddIdentity<IdentityUser, IdentityRole>(options => options.SignIn.RequireConfirmedAccount = true)
    .AddEntityFrameworkStores<ApplicationDbContext>().AddDefaultTokenProviders();
//Identity ekledikten sonra razor kullanıcaz dedik
builder.Services.AddRazorPages();
//email
builder.Services.AddScoped<IEmailSender, EmailSender>();
```

Login, Register, Authorization, Authentication işlemleri için Asp.Mvc.Identity kullanılmıştır. Burada

kütüphanenin bize sağladığı tablodaki verilere ek üyelerin isim-soyisim ve hakkında bilgilerine de ihtiyaç duyulduğu için bu bilgiler Customers modelinde verilmiş Login.cshtml.cs ve Register.cshtml.cs dosyalarında entegre edilmiştir.

2. SONUÇ

Bu projede Identity, Localization gibi temel anahtar kelimeler kullanılmış istenen özellikleri kapsayan bir ASP Net Core Mvc Hayvan Barınağı Projesi yapılmıştır.