



**SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
WEB PROGRAMLAMA DERSİ**

PROJE ÖDEVİ RAPORU

1.Öğretim A grubu

Grup Üyeleri

Kübra Yılmaz B181210023

Merve Sarı B181210074

Sedanur Sarıkoç B181210110

Sakarya

Ocak,2021

Projeden Beklenenler

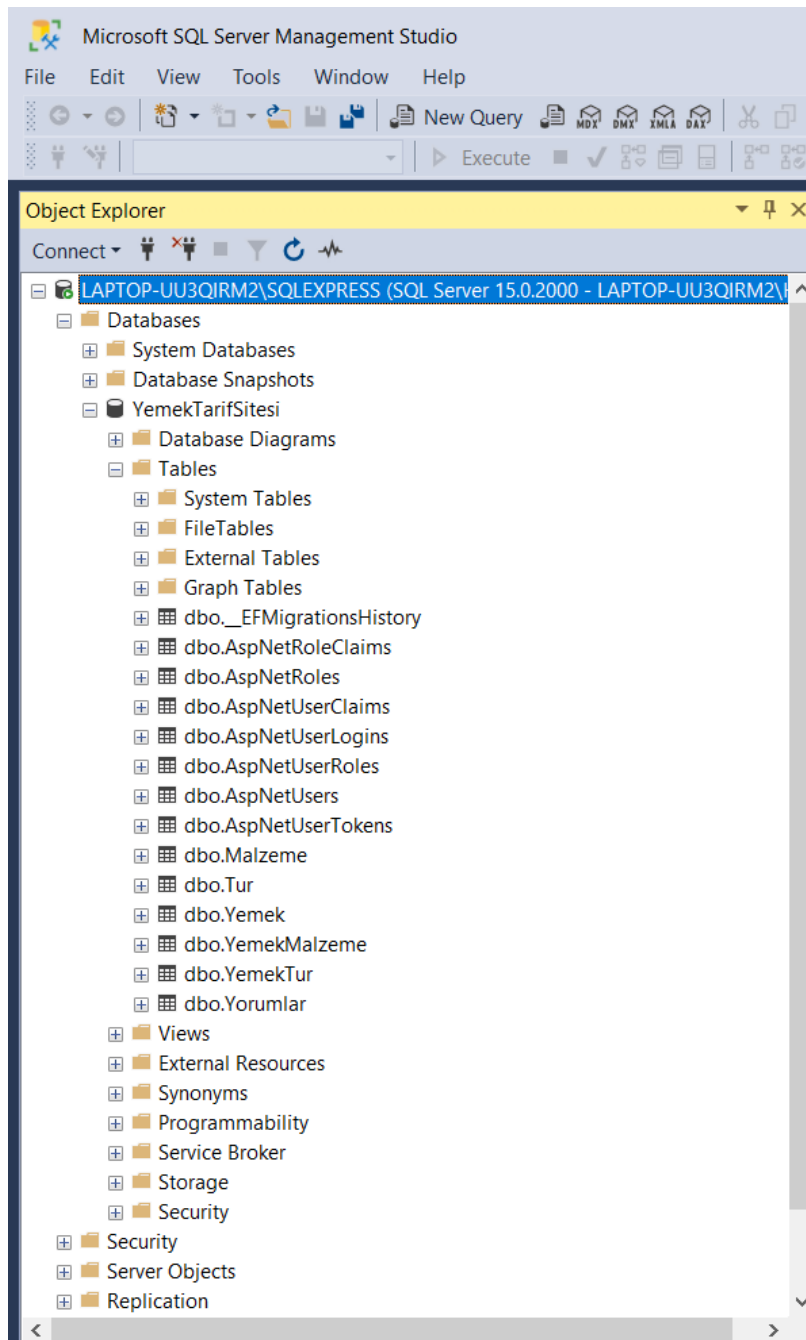
İnsanların, yemek tariflerine daha kolay ve pratik şekilde ulaşmaları beklenmektedir. Ayrıca kategori ayrımı ile istenen türde yemeğe daha kolay ulaşım gerçekleştirilmektedir.

- Projemizi Front-end olarak gerçekleştirdik.
- Kullanıcıların üye olacakları sayfa bulunmaktadır.
- Authorization doğru bir şekilde yapılandırılmıştır. (Üye olmayan kullanıcılar, veri tabanı işlemlerinin yapıldığı admin menüsüne ulaşamamaktadır.)
- Çok dil özelliğini ekledik. (Türkçe/İngilizce dil desteği)

Projede Kullanılan Teknolojiler

- Asp.net Core 3.1 Razor Pages
- C#
- Veritabanı olarak SQL Server
- Entity Framework Core ORM
- Bootstrap Tema
- HTML5, CSS3, Javascript ve JQUERY

Veritabanı Görüntüsü



Yemek Tablosu

Yemek tablosuna ait model ekran görüntümüz bu şekildedir. Yemek oluşturduğumuz zaman ilgili alanlara ait veri alınmaktadır. Id, YemekAdi, Tarif, Fotoğraf alanlarına veri girişi zorunlu iken malzeme, puan alanlarına veri girişi zorunlu değildir.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace YemekTarifSitesi.Models
{
    25 references
    public class Yemek
    {
        16 references
        public int Id { get; set; }
        16 references
        public string YemekAdi { get; set; }
        10 references
        public string? Malzemeler { get; set; }
        13 references
        public string Tarif { get; set; }
        14 references
        public double? Puan { get; set; }
        14 references
        public string Fotoğraf { get; set; }
    }
}
```

Malzeme Tablosu

Malzeme tablosuna ait model ekran görüntümüz bu şekildedir. Malzeme ekleyeceğimiz zaman ilgili alanlara ait veri alınmaktadır. Her alana ait veri girişi zorunludur.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace YemekTarifSitesi.Models
{
    20 references
    public class Malzeme
    {
        13 references
        public int Id { get; set; }
        14 references
        public string Adi { get; set; }
        13 references
        public double Miktar { get; set; }
        13 references
        public string MiktarTur { get; set; }
    }
}
```

Tür Tablosu

Tür tablosuna ait model ekran görüntümüz bu şekildedir. Tür ekleyeceğimiz zaman ilgili alanlara ait veri alınmaktadır. Her alana ait veri girişi zorunludur.

```
Yemek.cs    Kullanici.cs    Views.Home.Index.en-US.resx
TarifSitesi
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace YemekTarifSitesi.Models
{
    19 references
    public class Tur
    {
        13 references
        public int Id { get; set; }
        13 references
        public string Adi { get; set; }
    }
}
```

Kullanıcı Tablosu

Kullanıcı tablosuna ait model ekran görüntümüz bu şekildedir. Kullanıcı ekleyeceğimiz zaman ilgili alanlara ait veri alınmaktadır. Her alana ait veri girişi zorunludur.

```
using Microsoft.AspNetCore.Identity;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace YemekTarifSitesi.Models
{
    31 references
    public class Kullanici : IdentityUser
    {
        0 references
        public string Ad { get; set; }
        0 references
        public string Soyad { get; set; }
    }
}
```

YemekMalzeme Tablosu

YemekMalzeme tablosuna ait model ekran görüntümüz bu şekildedir. Yemek ve Malzeme arasında çoka çok bağlantı olduğu için bu ara tablo oluşturulmuştur.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace YemekTarifSitesi.Models
{
    19 references
    public class YemekMalzeme
    {
        11 references
        public int Id { get; set; }

        9 references
        public int YemekId { get; set; }

        8 references
        public int MalzemeId { get; set; }

        9 references
        public Yemek Yemek { get; set; }

        12 references
        public Malzeme Malzeme { get; set; }
    }
}
```

YemekTur Tablosu

YemekTur tablosuna ait model ekran görüntümüz bu şekildedir. Yemek ve Tur arasında çoka çok bağlantı olduğu için bu ara tablo oluşturulmuştur.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace YemekTarifSitesi.Models
{
    19 references
    public class YemekTur
    {
        11 references
        public int Id { get; set; }

        8 references
        public int YemekId { get; set; }

        9 references
        public int TurId { get; set; }

        12 references
        public Yemek Yemek { get; set; }

        9 references
        public Tur Tur { get; set; }
    }
}
```

```

<div class="tab-content" id="v-pills-tabContent">
  <div class="tab-pane fade show active" id="v-pills-1" role="tabpanel" aria-labelledby="v-pills-day-1-tab">
    @foreach (var item in Model.Yemek)
    {
      <div class="ftco-animate" style="width: 100%;">
        <div class="col-md-4 blog-entry" style="width: 50%; height: 500px; float: left; ">
          
          <div class="text pt-3 pb-4 px-4">
            <div class="meta">
            </div>
            <h3 class="heading"><a href="#">@item.YemekAdi</a></h3>
            <p style="color:white"></p>
            <p class="clearfix">
              <a href="#" class="float-left meta-chat">Puan: @item.Puan</a>
            </p>
          </div>
        </div>
        <div class="col-md-8 blog-entry" style="margin-left: 35%; height: 500px; width: 100%; ">
          <b>Malzemeler:</b>
          @foreach (var item2 in Model.YemekMalzeme)
          {
            if (item2.YemekId == item.Id)
            {
              @item2.Malzeme.Miktar @item2.Malzeme.MiktarTur &nbsp;&nbsp;&nbsp; @item2.Malzeme.Adi, &nbsp;&nbsp;&nbsp;
            }
          }
          <br /><b>Tarif:</b> @item.Tarif
        </div>
      </div>
    }
  </div>

```

Burada tüm yemeklere ait fotoğraf, puan, yemek ismi, malzemeler ve tarif bilgileri yazdırılıyor.

```

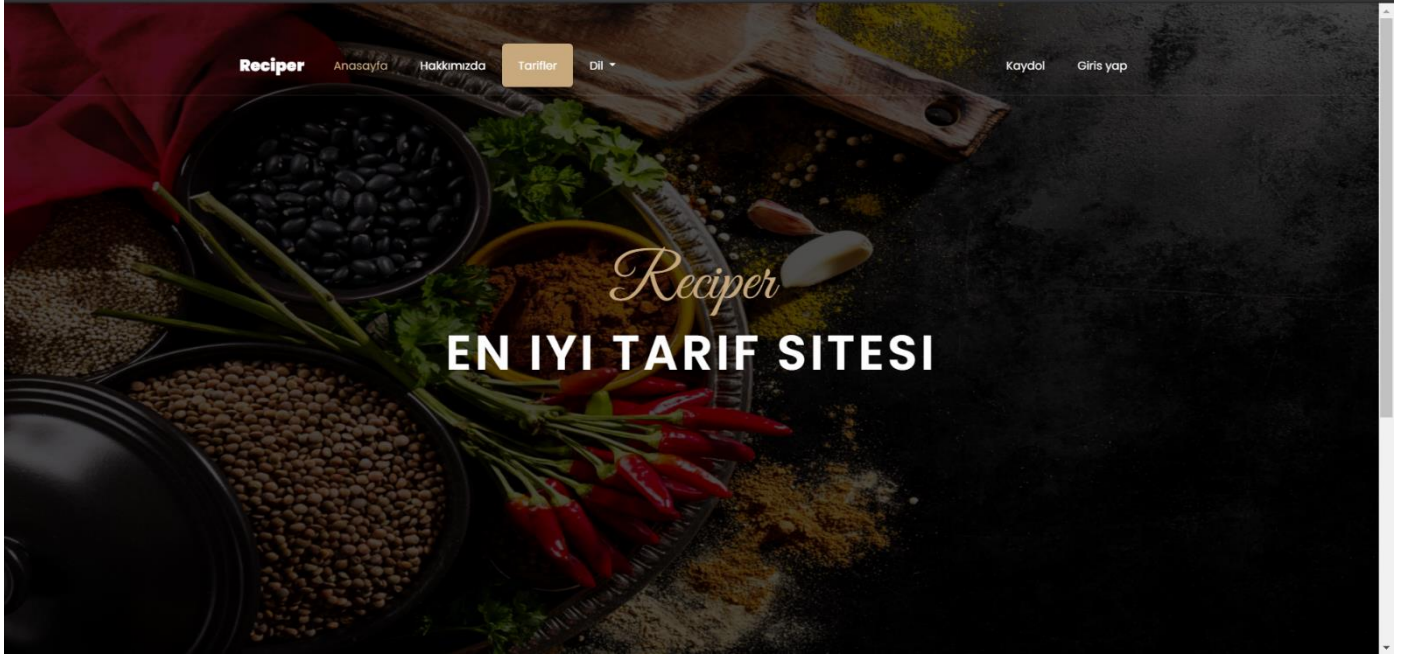
@model YemekTarifSitesi.Models.YemekTarifModel
@{
  ViewData["Title"] = "Tarifler";
  Layout = "~/Views/Shared/_Layout.cshtml";
}

@functions{
  public string YemekGetir(int id)
  {
    <div class="d-flex align-self-stretch">
      @foreach (var item in Model.YemekTur)
      {
        if (item.TurId == id)
        {
          <div class="col-md-4 ftco-animate">
            <div class="blog-entry">
              
              <div class="text pt-3 pb-4 px-4">
                <div class="meta">
                </div>
                <h3 class="heading"><a href="#">@item.Yemek.YemekAdi</a></h3>
                <p class="clearfix">
                  <a href="#" class="float-left meta-chat">Puan: @item.Yemek.Puan</a>
                </p>
              </div>
            </div>
          </div>
        }
      }
    </div>
    return string.Empty;
  }
}

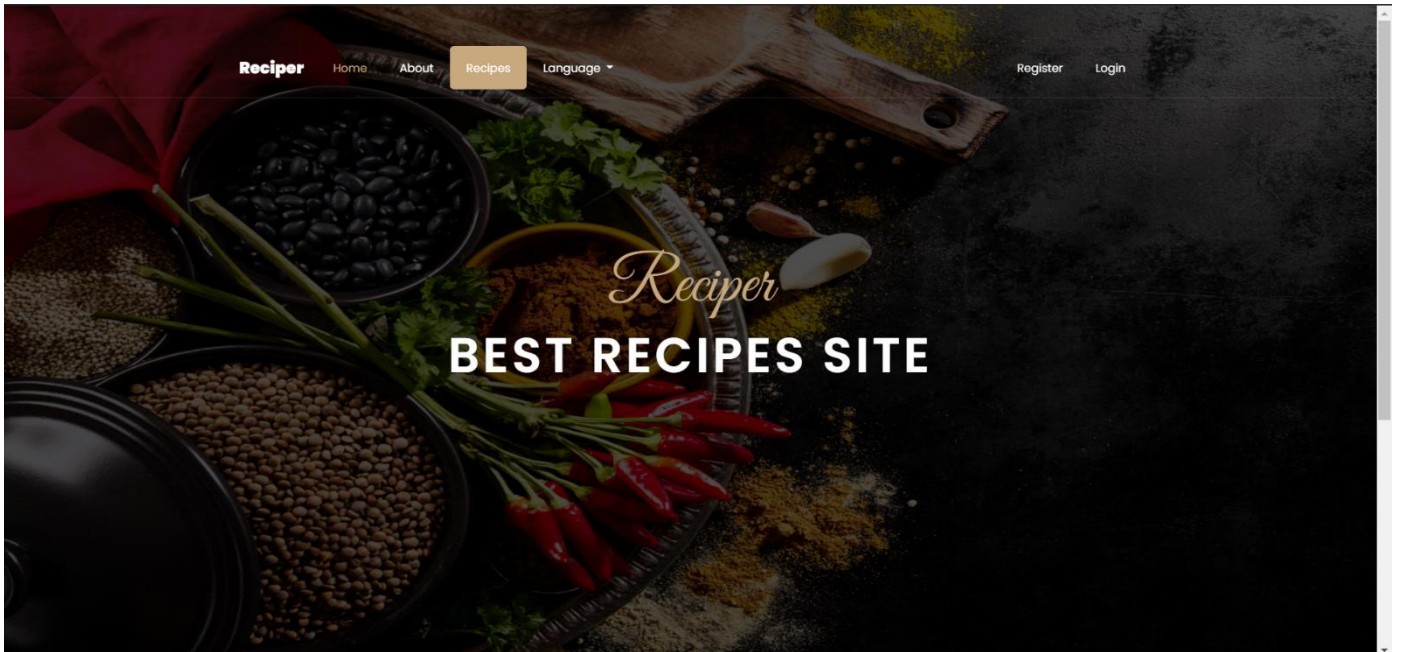
```

Burada kategoriye göre yemek tarifi yazdırıyoruz. Tüm kategorilerin id'leri dönüyor ve kullanıcının istediği kategorinin id'sine eşit olduğunda o kategorideki tüm yemekleri listeliyor.

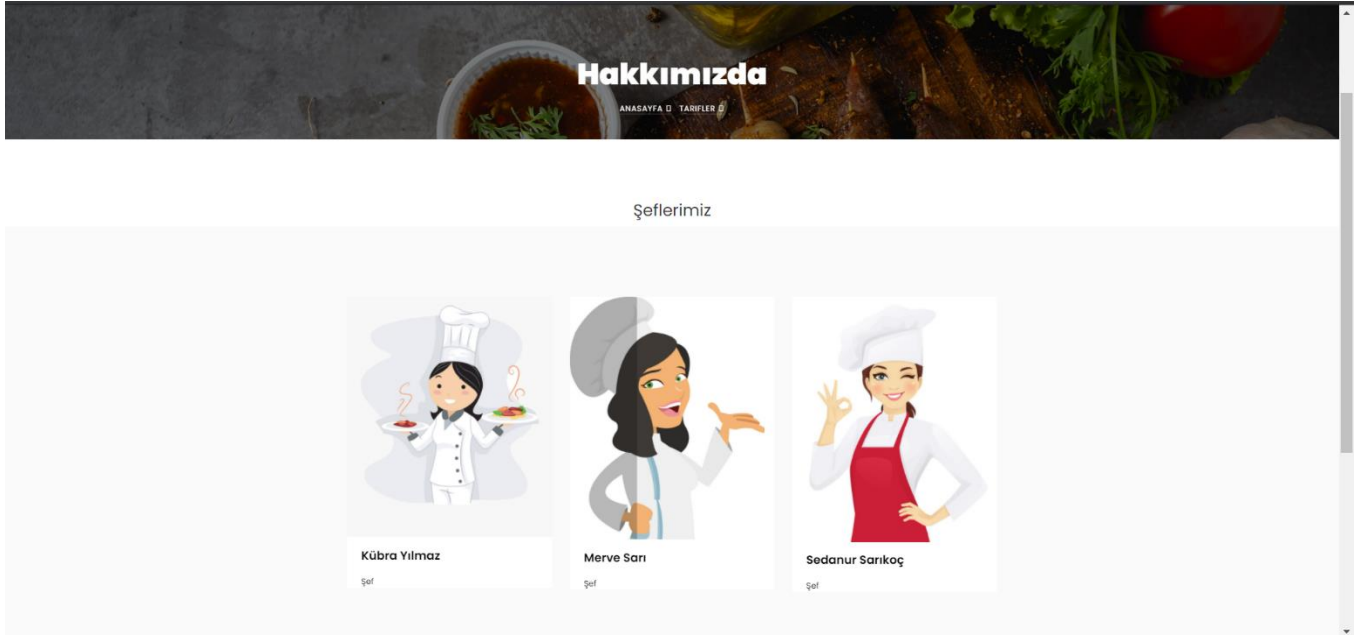
Anasayfa-Türkçe



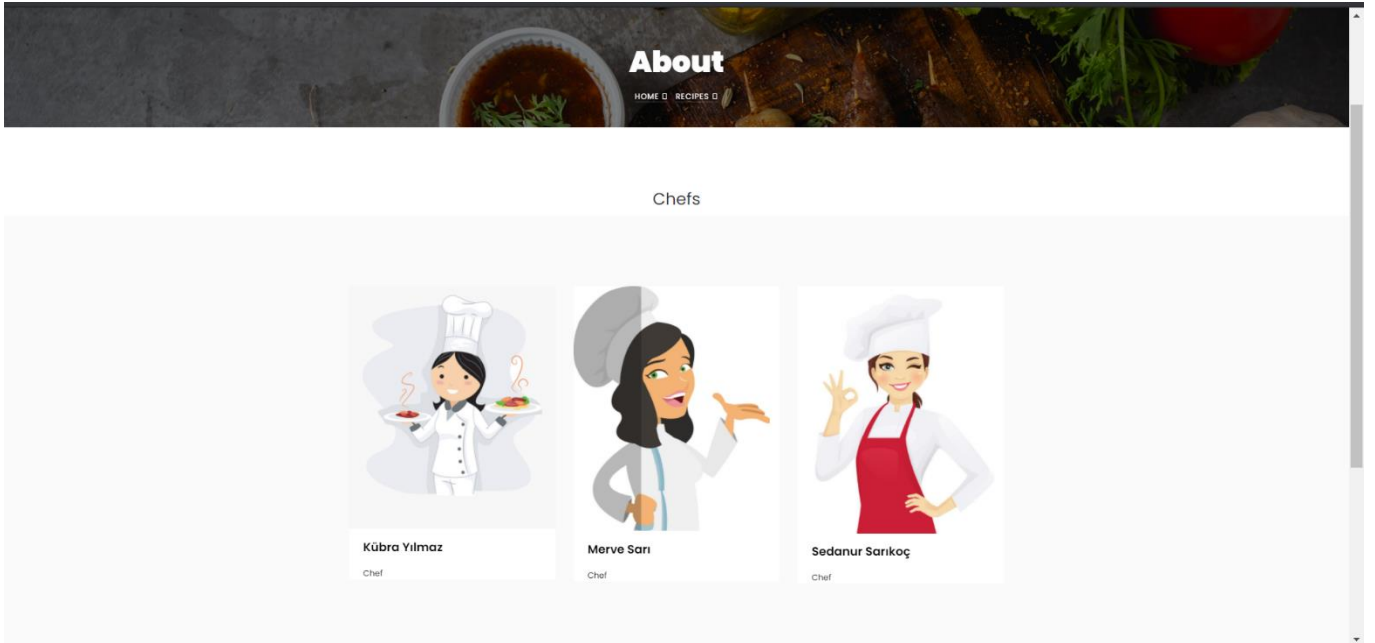
Anasayfa-İngilizce



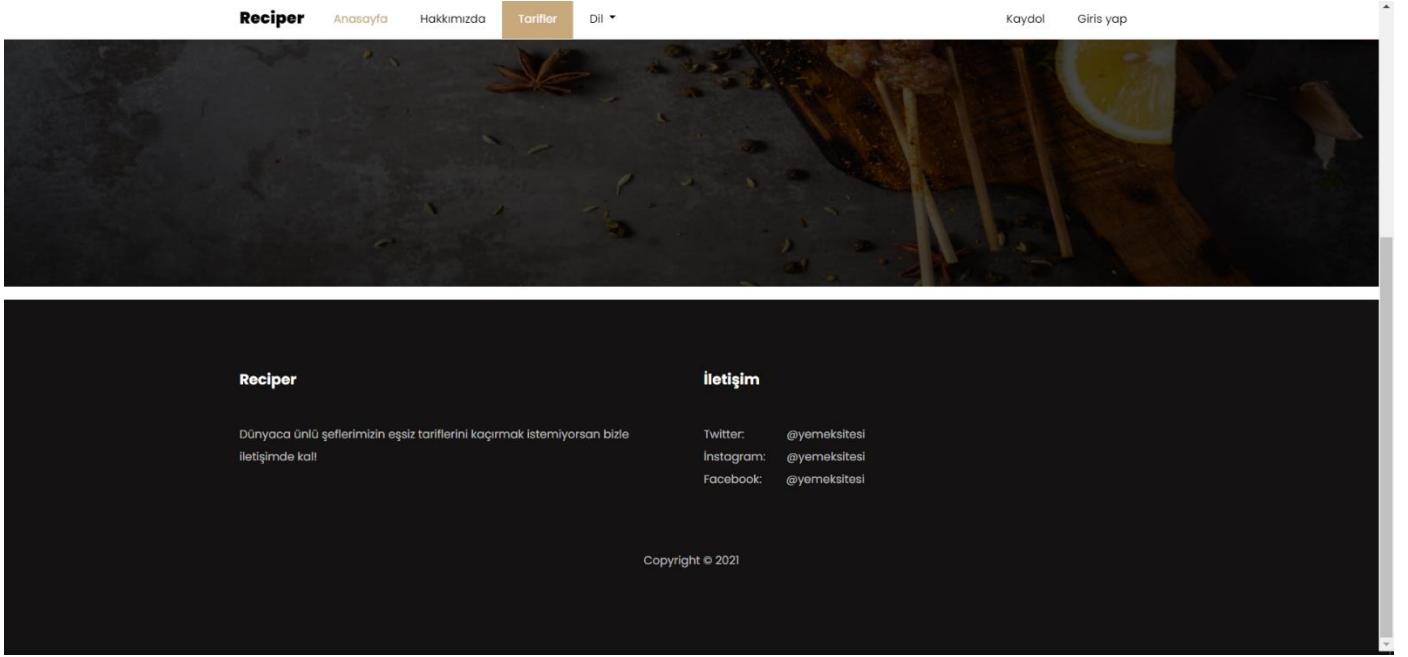
Hakkımızda-Türkçe



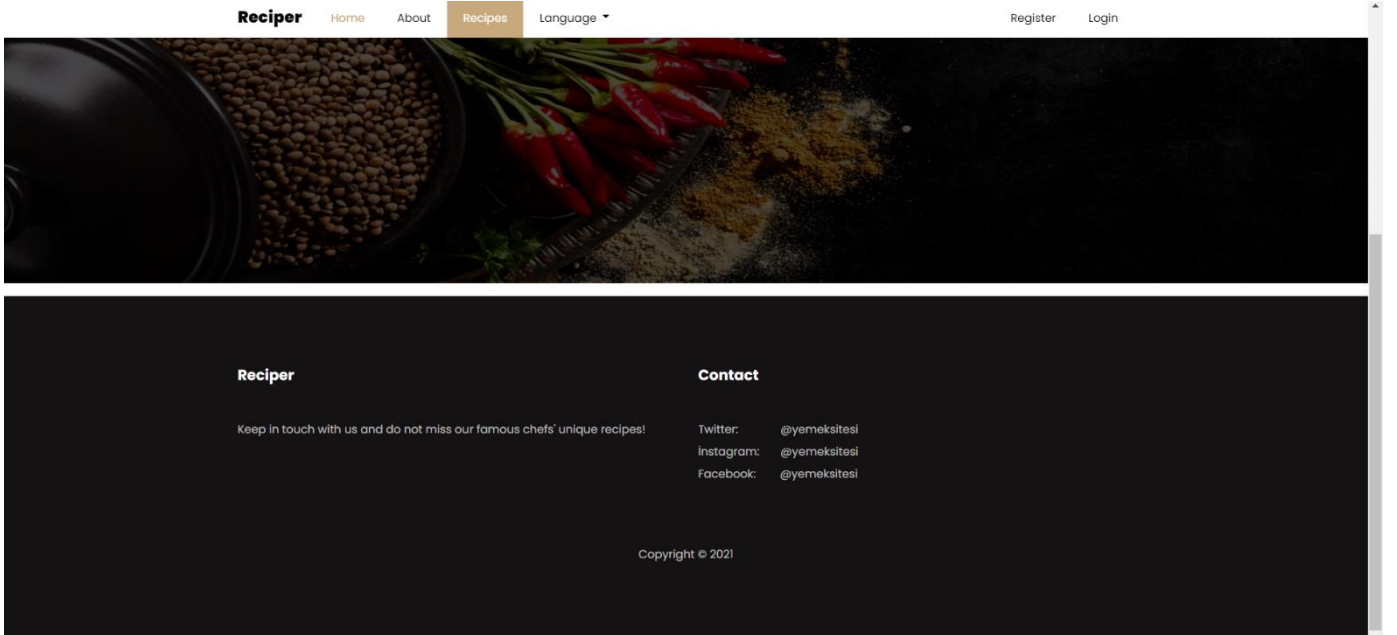
Hakkımızda-İngilizce

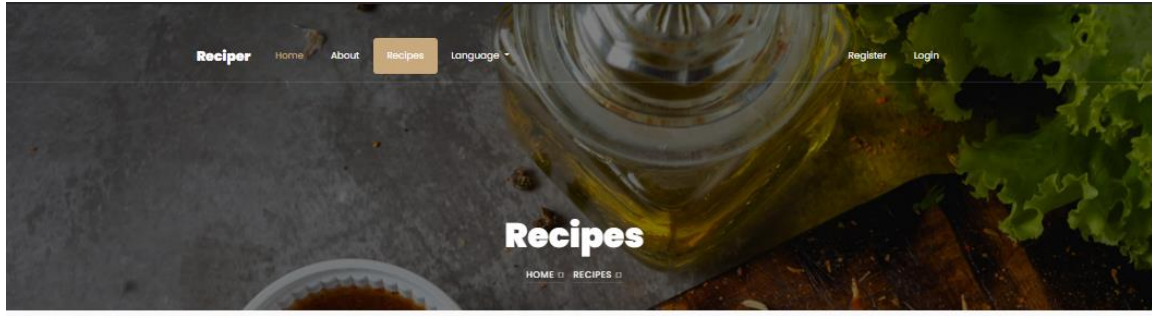


Layout-Türkçe




Layout-İngilizce





AllBreakfastLunchDinnerDessertDrink



Pancake

Puan: 4


Malzemeler: 1 su bardağı un, 1 su bardağı süt, 2 adet yumurta, 2 yemek kaşığı şeker, 1 paket kabartma tozu, 1 paket vanilya.

Tarif: Öncelikle yumurtalar ve şekeri karıştırma kabına alarak iyice çırpalım. Süt, un, kabartma tozu, vanilya ilave edip çırpalım. Kek kıvamından biraz daha koyu olacak şekilde hamur hazırlayalım. Teflon ya da yapışmaz bir tavaya az sıvı yağı dökelim, fırça yardımı ile her tarafına dağıtalım. Sıvı yağı az kullanmak pankek yapmanın püf noktalarındandır. 1 büyük kaşık hamur dökülerek hamurun kendi kendine yayılmasını bekleyelim. Üzeri göz göz olmaya başlayan pankeklerimizi spatula yardımı ile ters çevirelim. Her iki tarafı da pişen pankeklerimizi servis tabağına alalım. Afiyet Olsun!

Burada tüm tarifler sıralanıyor.


HOME RECIPES

AllBreakfastLunchDinnerDessertDrink



Pancake

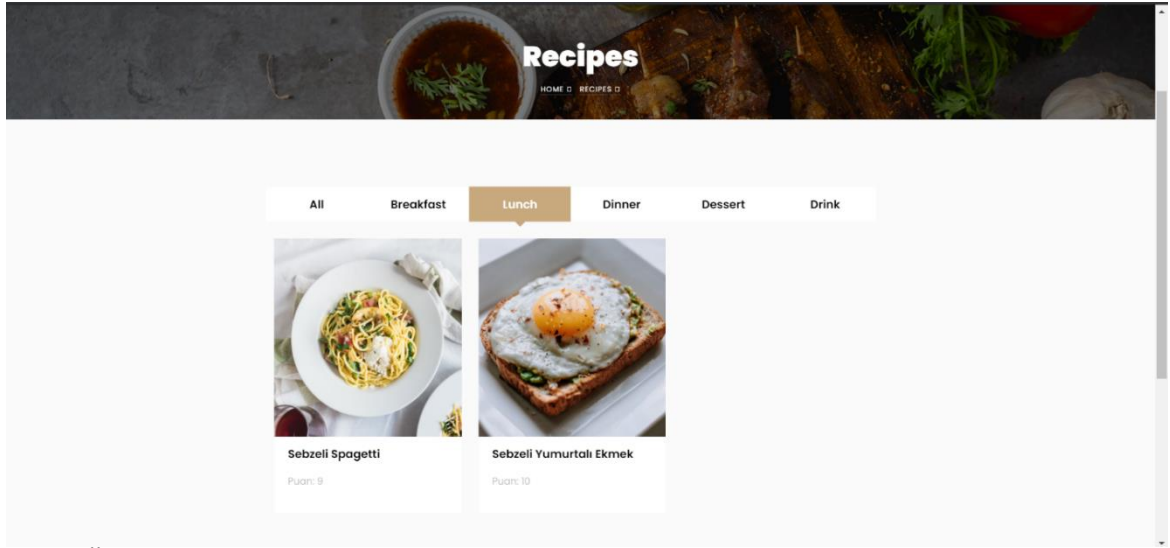
Puan: 4



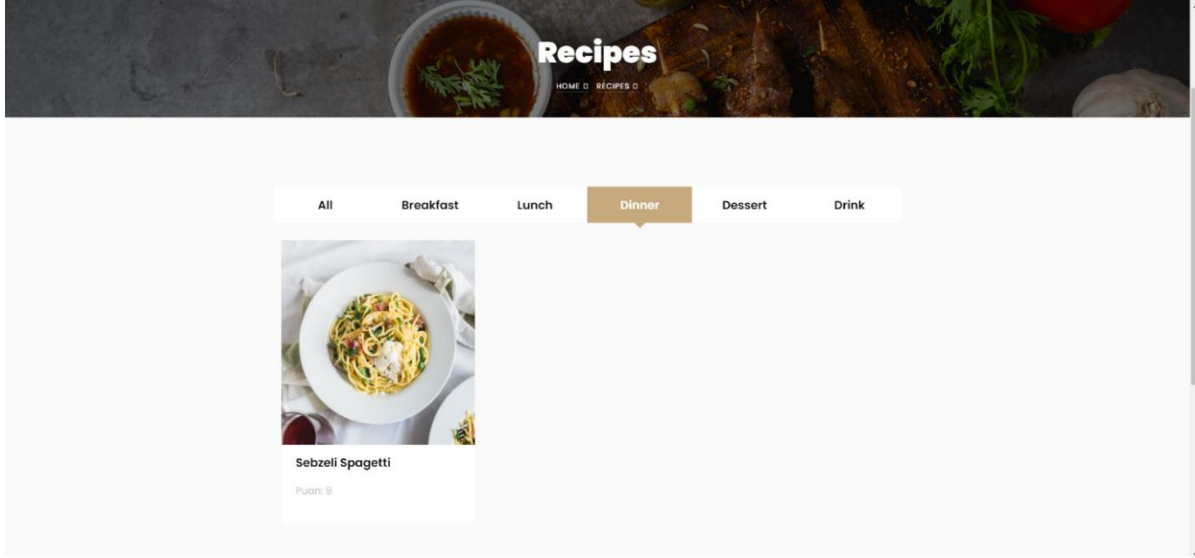
Sebzeli Yumurtalı Ekmek

Puan: 10

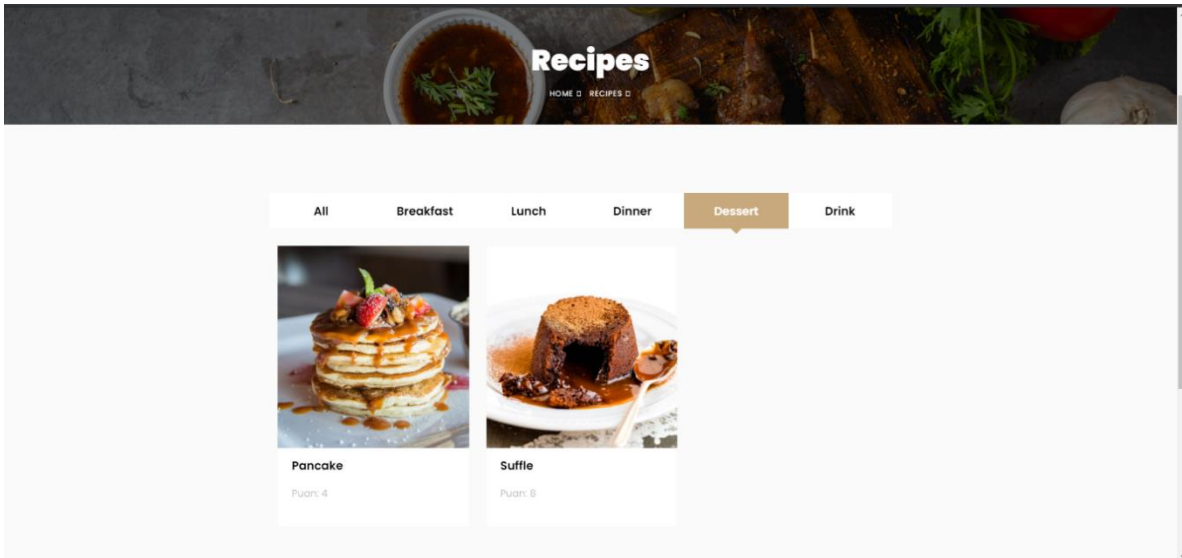
Burada Kahvaltı kategorisine ait yemekler sıralanıyor.



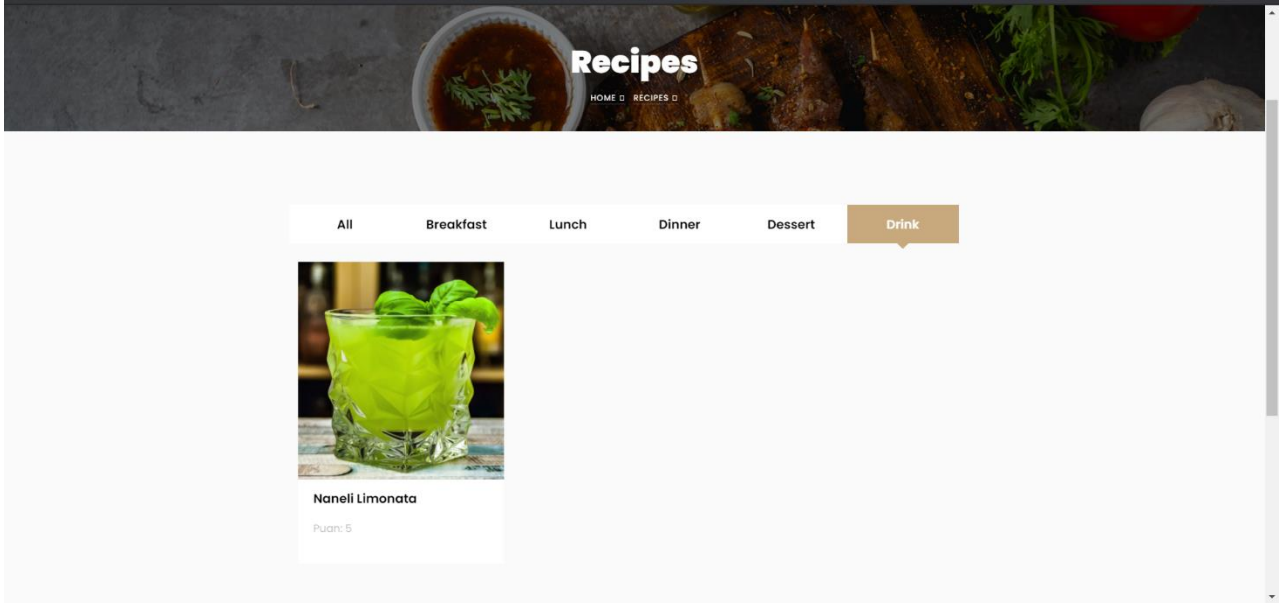
Burada Öğle kategorisine ait yemekler sıralanıyor.



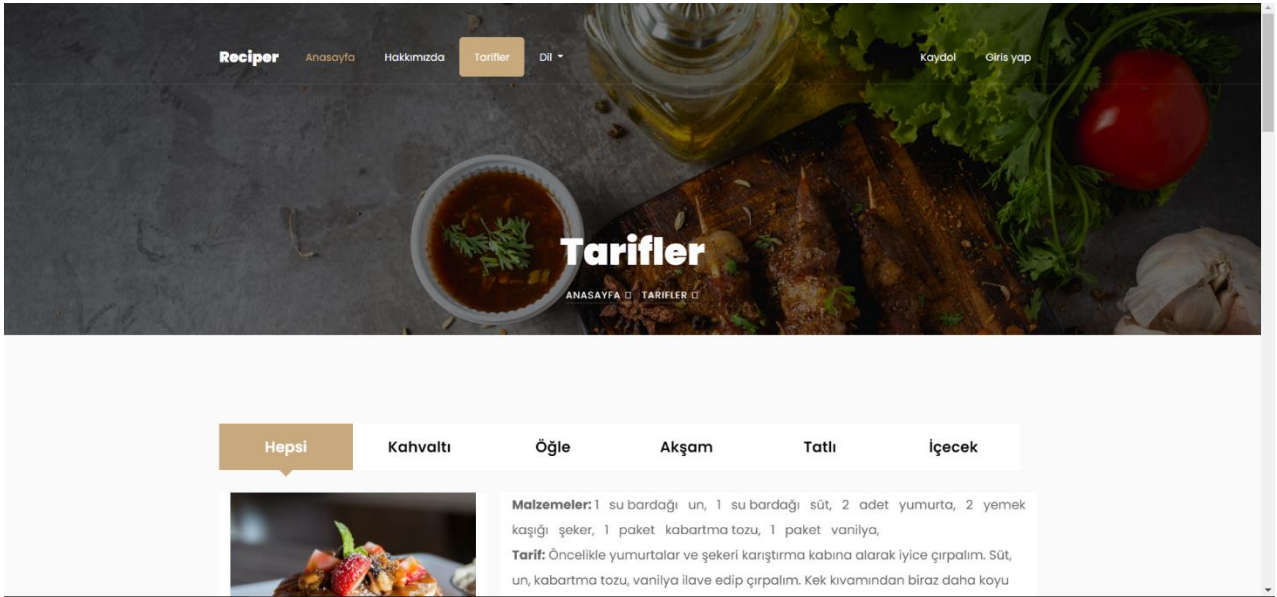
Burada Akşam kategorisine ait yemekler sıralanıyor.



Burada Tatlı kategorisine ait yemekler sıralanıyor.



Burada İçecek kategorisine ait yemekler sıralanıyor.



Tarifler sayfası için önceden koymuş olduğumuz ekran görüntüleri, sayfanın İngilizce haline aittir. Sayfanın Türkçe halinde ise menü kısmı ve Tarifler sayfası böyle görünmektedir. Tarifler sayfasındaki kategoriler ise böyle gözükmektedir.

Burada sitemizin Kaydol kısmı
gözükmektedir. Normalde burada sadece
mail ve şifre isteniyordu. Bunlara ek
olarak biz tabloya ad ve soyad alanı
ekledik.

Burada sitemizin Giriş kısmı
gözükmektedir.

Register

Create a new account.

Use another service to register.

Ad

Soyad

Email

Password

Confirm password

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.

Log in

Use a local account to log in.

Use another service to log in.

Email

Password

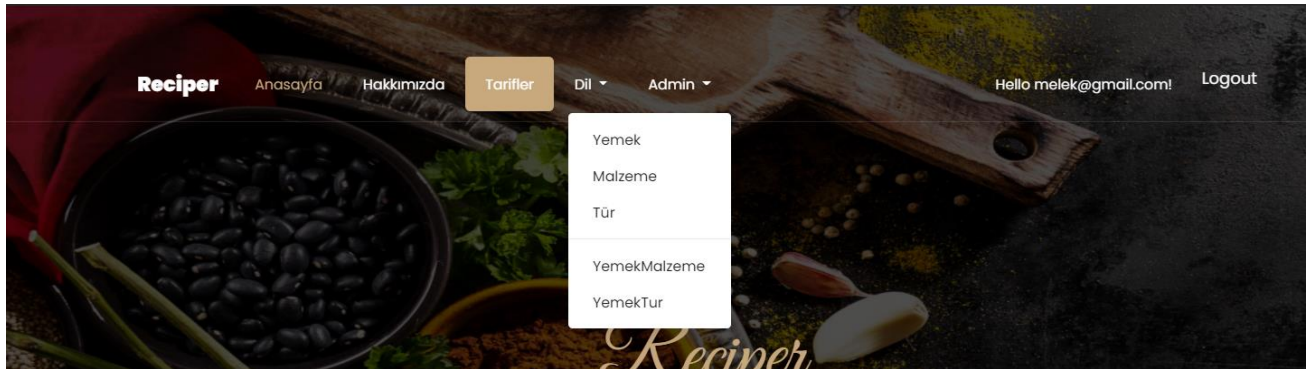
☐ Remember me?

Log in

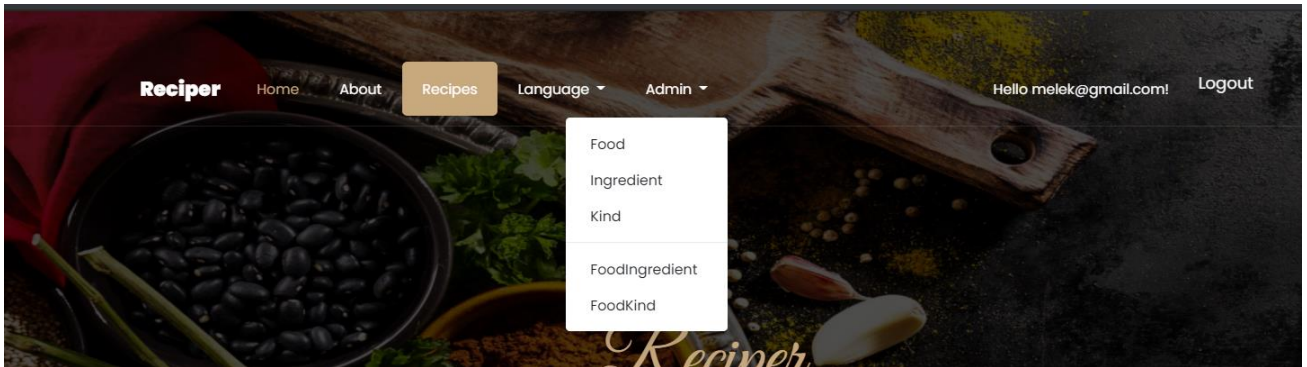
[Forgot your password?](#)

[Register as a new user](#)

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.



Kullanıcı giriş yaptığında normalde gözükmeyen admin menüsü gösteriliyor.



Admin menüsü-İngilizce