



KÜTAHYA DÜMLUPINAR ÜNİVERSİTESİ

Mühendislik Fakültesi

Bilgisayar Mühendisliği

Görsel Programlama Çözümleri

Final Sınav Raporu

Öğretim Üyesi

Doç. Dr. HASAN TEMURTAŞ

Öğrenci

Kubilay Dicle 202013172019
Ömer Can Bolat 202113172072

Rapor İeriđi

- 1. Genel Bilgi*
- 2. Web Sitesi Hakkında Genel Bakış*
- 3. Web Sitesi İçin Kullanılan Teknolojiler Ve Kütüphaneler*
- 4. Görsellerle Adım Adım Proje İncelemesi*
- 5. Kod Görselleri*
- 6. Kaynaklar*

1. Genel Bilgi

- Proje, HTML & CSS & JS, C#, .Net kullanılarak hazırlandı.
- Web sitenin adı “Yemek Lazım”
- Web sitenin türü E-Ticaret sitesi

2. Web Sitesi Hakkında Genel Bakış

YemekLazım, kullanıcıların yemek tariflerine hızlı ve kolay bir şekilde erişmesini sağlayan, aynı zamanda alışveriş deneyimini entegre ederek süreci daha pratik hale getiren bir web tabanlı platformdur. Bu uygulama, modern mutfak kullanıcılarının karşılaştığı iki temel ihtiyacı karşılamak için geliştirilmiştir: **yemek tariflerine erişim** ve **malzeme tabanlı yemek önerileri**.

Projenin Amacı

Günümüzde yemek yapmak, sadece bir zorunluluk değil, aynı zamanda bir hobidir. Ancak, bu süreç bazen zaman alıcı ve karmaşık olabilir. Özellikle tariflere ulaşmak, malzemeleri organize etmek ve eksik malzemeleri belirlemek kullanıcılar için zahmetli bir süreçtir. YemekLazım, bu sorunları çözmeyi amaçlayarak aşağıdaki hedeflere odaklanır:

- 1. Kullanıcıların tariflere kolayca ulaşmasını sağlamak**
- 2. Mevcut malzemelere göre tarif önerileri sunmak**
- 3. Eksik malzemeleri belirleyip alışveriş sürecini kolaylaştırmak**
- 4. Kullanıcıların zaman yönetimini optimize etmek ve mutfak deneyimini geliştirmek**

Uygulamanın Temel Özellikleri

1. Yemek Tarifi Arama

Bu özellik, kullanıcıların istedikleri yemeklerin tariflerine ve malzeme listelerine erişimini sağlar.

Detaylar:

- **Yemek Adı Girişi:**

Kullanıcı, yapmak istediği yemeğin adını arama çubuğuna yazar. Örneğin, “Makarna” yazıldığında, sistem veri tabanındaki tüm makarna tariflerini getirir.

- **Tarif ve Malzeme Listesi:**

Seçilen tarif, yapım aşamalarıyla birlikte kullanıcıya sunulur. Tarifte birlikte gerekli olan tüm malzemeler detaylı bir şekilde listelenir.

- **Alışveriş Sepeti:**

Kullanıcı, listelenen malzemeleri kolayca alışveriş sepetine ekleyebilir. Bu özellik, kullanıcının ihtiyaçlarını organize etmesini ve eksik malzemeleri satın almasını kolaylaştırır. Sepet, kullanıcının alışveriş deneyimini optimize eder.

Örnek Kullanım:

Bir kullanıcı, “Lahmacun” tarifine erişmek istediğinde sistem, tarifin yanı sıra gerekli tüm malzemeleri (un, kıyma, maydanoz vb.) listeler ve bu malzemeleri alışveriş sepetine ekleme seçeneği sunar.

2. Malzeme Tabanlı Tarif Önerisi

Bu özellik, kullanıcıların mevcut malzemelerine göre yemek tarifleri bulmasını sağlar.

Detaylar:

- **Malzeme Girişi:**

Kullanıcı, evde bulunan malzemeleri sisteme manuel olarak girebilir ya da sistemdeki hazır malzeme listesinden seçim yapabilir. Örneğin, “domates”, “soğan” ve “yumurta” malzemelerini girdiğinde sistem, bu malzemelerle yapılabilecek yemekleri analiz eder.

- **Tarif Önerileri:**

Sistemde bulunan algoritma, kullanıcı tarafından girilen malzemeleri analiz eder ve bu malzemelerle uyumlu tarifleri listeler. Bu tarifler, mevcut malzemelerle neler yapılabileceğini gösterir.

- **Eksik Malzemelerin Belirtilmesi:**

Kullanıcı, mevcut malzemelerle sadece belirli tarifleri yapabiliyorsa, sistem eksik malzemeleri belirler ve bu eksik malzemeler tamamlandığında başka hangi tariflerin yapılabileceği konusunda kullanıcıyı bilgilendirir. Böylece, kullanıcı eksik malzemeleri sepete ekleyerek alışverişini planlayabilir.

Örnek Kullanım:

Bir kullanıcı, “patates”, “soğan” ve “sıvı yağ” gibi malzemeleri seçtiğinde sistem, bu malzemelerle yapılabilecek tarifleri (örneğin, “Patates Kızartması”) önerir. Aynı zamanda, “kıyma” eklenirse “Kıymalı Patates” yemeğini yapabileceğini belirtir.

3. Admin Paneli Hakkında

Admin Panelinin Özellikleri

Admin paneli, yemek tarifi web sitesinin temel bileşenleri olan Tarifler ve Malzemeler için kapsamlı bir yönetim aracı sunmaktadır. Aşağıda panelin sunduğu CRUD (Create, Read, Update, Delete) işlemleri detaylı bir şekilde açıklanmıştır:

1. Tarif Yönetimi

Yemek tarifleri, sistemdeki en önemli veri setlerinden biridir. Admin paneli üzerinden aşağıdaki işlemler gerçekleştirilebilir:

Yeni Tarif Oluşturma (Create): Yöneticiler, yeni tarifler ekleyebilir ve tarif adı, açıklaması, hazırlık süresi gibi detayları girebilir.

Tarif Listeleme (Read): Sistemde mevcut olan tüm tarifler bir tablo üzerinde listelenir. Bu liste, tarif adı, açıklama ve ilişkili malzemeleri gibi bilgileri içerir.

Tarif Güncelleme (Update): Mevcut tariflerin bilgileri düzenlenebilir. Örneğin, tarifin içeriği güncellenebilir veya eksik malzemeler eklenebilir.

Tarif Silme (Delete): Yöneticiler, artık gerekli olmayan tarifleri sistemden kolayca kaldırabilir.

2. Malzeme Yönetimi

Malzemeler, tariflerle ilişkili olduğu için sistemin önemli bir parçasını oluşturur. Admin paneli, malzemelerle ilgili aşağıdaki işlemleri destekler:

Yeni Malzeme Ekleme (Create): Yeni bir malzeme eklenerek sistemde kullanılabilir hale getirilir. Malzeme adı ve açıklama gibi bilgiler girilebilir.

Malzeme Listeleme (Read): Sistem içerisindeki tüm malzemeler, bir tablo görünümünde listelenir. Malzemelerin hangi tariflerle ilişkili olduğu da görülebilir.

Malzeme Güncelleme (Update): Mevcut malzemeler üzerinde deęişiklik yapılabilir. Örneęin, malzemenin adı veya açıklaması güncellenebilir.

Malzeme Silme (Delete): Artık sistemde yer almaması gereken malzemeler kolayca kaldırılabilir.

3. Tarif ve Malzeme İlişkilendirme

Admin paneli, tarifler ve malzemeler arasında bir ilişki kurma yeteneęi de sunar:

Yöneticiler, bir tarifi hangi malzemeleri içerdığını kolayca belirtebilir.

Yeni bir tarif oluştururken veya mevcut bir tarifi düzenlerken, kullanılacak malzemeler seçilebilir ya da çıkarılabilir.

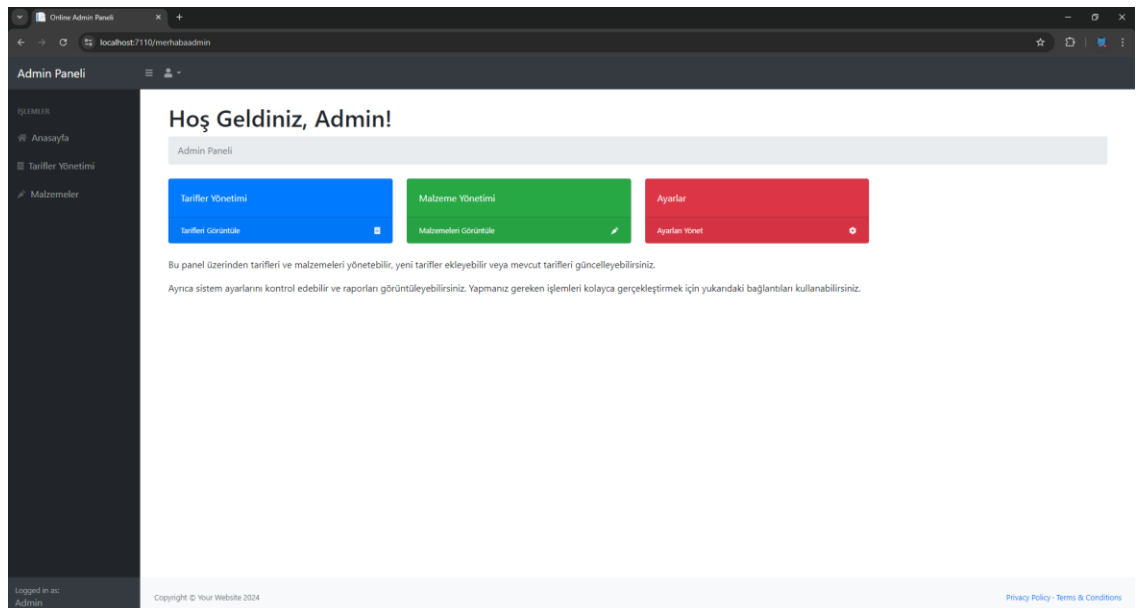
Panelin Avantajları

Kullanıcı Dostu Arayüz: Panelin tasarımı, yöneticilerin karmaşık işlemleri kolayca gerçekleştirmesine olanak tanır.

Veri Tutarlılığı: Tarifler ve malzemeler arasındaki ilişkiler düzenli bir şekilde yönetilir.

Hızlı İşlemler: CRUD işlemleri hızlı ve verimli bir şekilde gerçekleştirilir.

Yetkilendirme ve Güvenlik: Sadece yetkili kullanıcıların admin paneline erişebilmesi sağlanmıştır.



Fotoęraf: Admin Paneli

YemekLazım'ın Faydaları

1. Zaman Yönetimi:

Kullanıcılar, tarif arama ve alışveriş süreçlerinde zamandan tasarruf sağlar. Malzeme tabanlı öneriler, hızlı karar vermeye olanak tanır.

2. Kaynak Verimliliği:

Kullanıcılar, mevcut malzemelerini en iyi şekilde değerlendirerek israfı azaltabilir. Eksik malzeme bilgisi, gereksiz alışverişini önler.

3. Kullanıcı Dostu Arayüz:

Basit ve sezgisel tasarımı sayesinde YemekLazım, kullanıcıların kolayca tarif aramasına, malzeme eklemesine ve alışveriş sepetlerini yönetmesine olanak tanır.

4. Alışveriş Entegrasyonu:

Eksik malzemeler hızlıca sepete eklenerek alışverişe hazır hale getirilir. Bu, mutfakta pratik bir süreç oluşturur.

Teknik Yapı ve Çalışma Şekli

- **Frontend:**

Kullanıcı arayüzü HTML, CSS ve JavaScript kullanılarak tasarlanmıştır. Dinamik özellikler için JavaScript aktif rol oynamaktadır. Kullanıcılar yemek tariflerine ve malzeme girişine yönelik işlemleri burada gerçekleştirir.

- **Backend:**

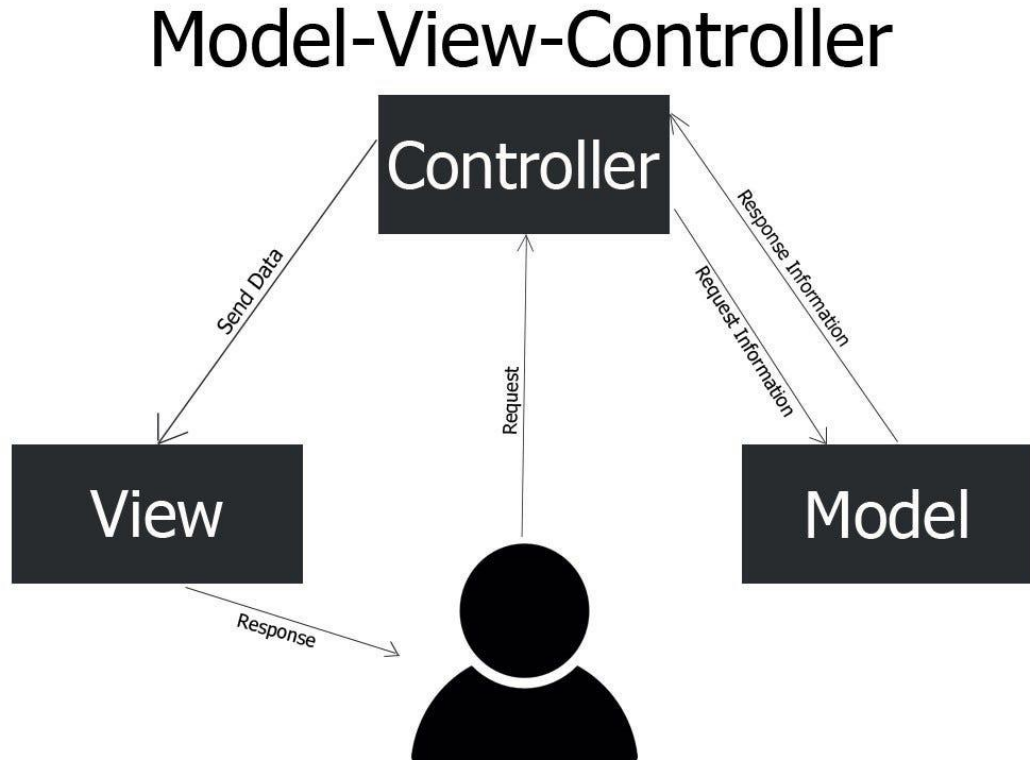
C# ve .NET Framework ile geliştirilmiştir. Kullanıcı taleplerine göre tarif verilerini işler ve malzeme analizleri yapar.

- **Veritabanı:**

MsSQL veritabanı sistemi, yemek tarifleri, malzemeler ve kullanıcı verilerini saklamak için kullanılır. Tariflerin malzemelerle eşleştirilmesi, öneri algoritmalarıyla desteklenir

3. Web-Sitesi için Kullanılan Teknolojiler Ve Araçlar

- ASP.NET MVC (Model-View-Controller)



Fotoğraf: 1.1

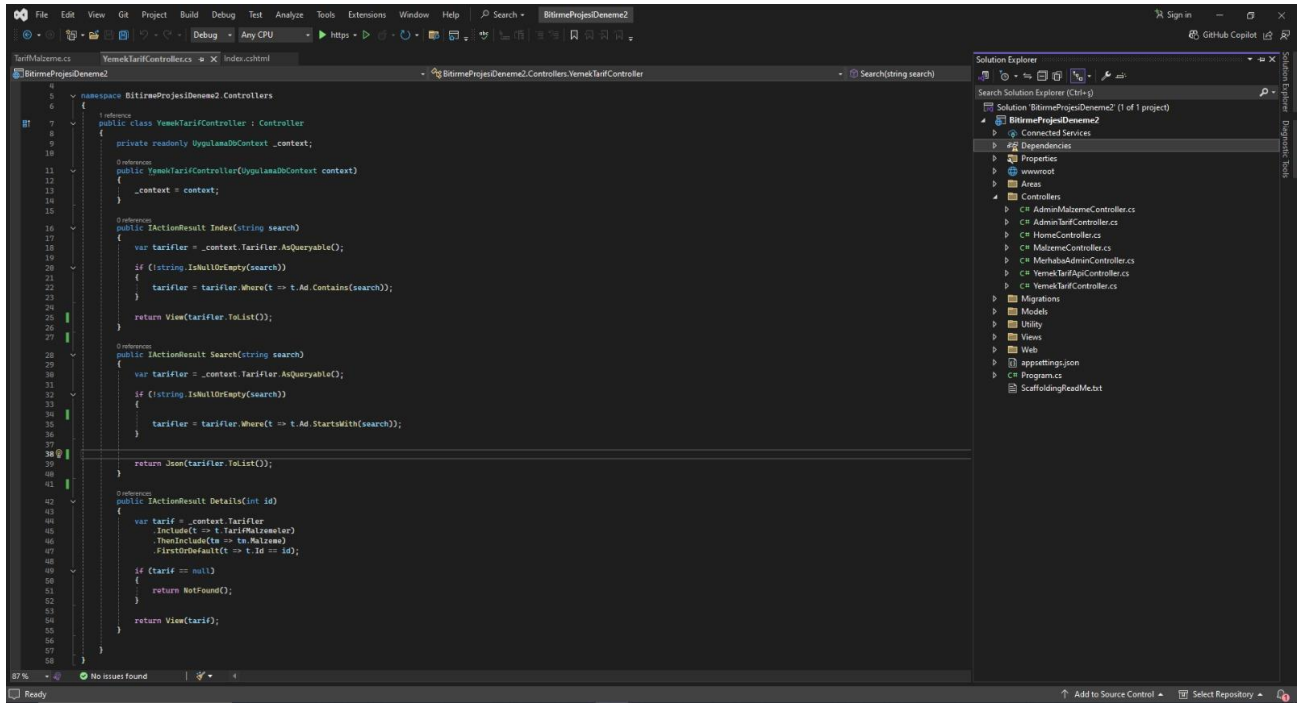
ASP.NET MVC, web uygulamaları geliştirmek için kullanılan bir framework olup Model-View-Controller (MVC) mimari desenine dayanır. Bu yapı, uygulamayı üç temel katmana ayırarak kodun daha düzenli, bakımı kolay ve test edilebilir olmasını sağlar:

Model: Uygulamanın veri katmanıdır. Tarifler, malzemeler gibi veriler burada yönetilir.

View: Kullanıcı arayüzünü oluşturur. Kullanıcılar için tariflerin listelendiği sayfalar, malzeme seçim formları ve üyelik giriş/çıkış ekranları burada geliştirilmiştir.

Controller: İş mantığı ve kullanıcı taleplerini yönetir. Kullanıcının tarif araması, malzemeleri seçmesi gibi işlemleri kontrol eder ve uygun View ile Model'i birbirine bağlar.

Bu yapı sayesinde projede kullanıcıların yemek tariflerine ulaşması, tariflere malzemelerle filtreleme yapması ve üyelik sistemiyle giriş yapması gibi işlemler etkin bir şekilde sağlanmıştır.



Fotoğraf: MVC 1.2

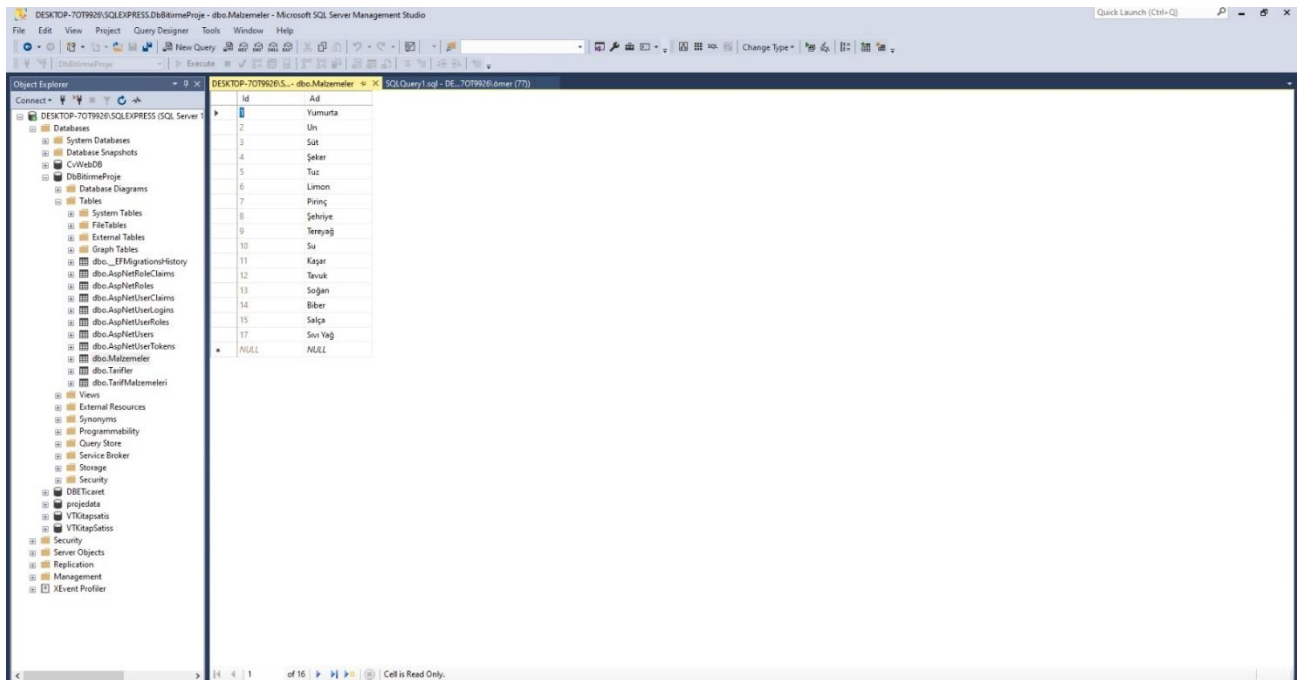
➤ MSSQL (Microsoft SQL Server)

Microsoft SQL Server, projenin veritabanı yönetim sistemi olarak kullanılmıştır.

Veritabanında yemek tarifleri, malzemeler, tarif-malzeme ilişkileri ve kullanıcı bilgileri gibi tablolar tasarlanmıştır.

Tablolar arasında birbiriyle ilişkili veriler Entity Framework kullanılarak yönetilmiştir.

Verilerin güvenli ve performanslı bir şekilde saklanması için MSSQL tercih edilmiştir.



➤ Entity Framework (ORM)

Entity Framework, veritabanı işlemlerini kolaylaştıran bir ORM (Object-Relational Mapping) aracıdır.

Bu proje için Code-First yaklaşımı kullanılmıştır. Bu yaklaşımda önce veri modelleri tanımlanmış, ardından veritabanı Entity Framework tarafından otomatik olarak oluşturulmuştur.

Tarifler ile malzemeler arasındaki çoktan çoğa ilişkiler TarifMalzeme adlı ara tablo modeli ile oluşturulmuştur.

LINQ sorguları kullanılarak veritabanına erişim ve veriler üzerinde işlemler gerçekleştirilmiştir.

➤ Identity Core (Üyelik Sistemi)

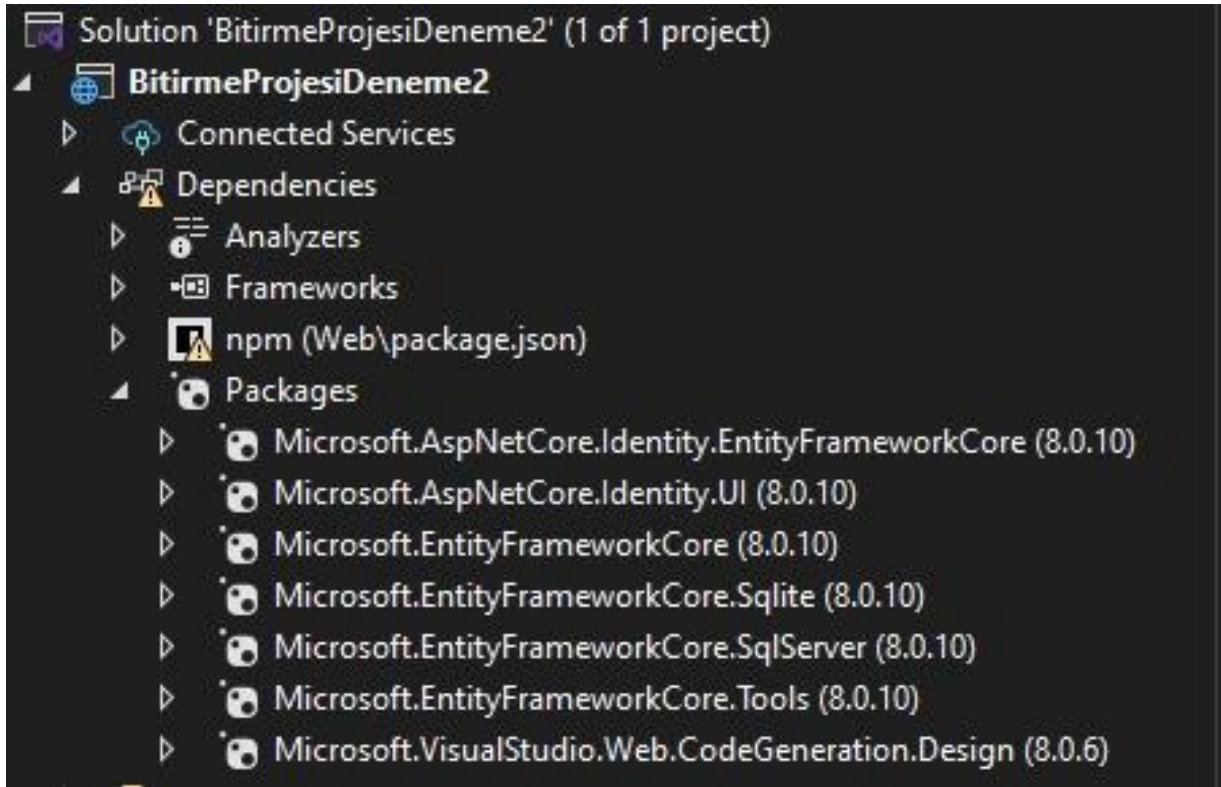
ASP.NET Identity Core, projenin üyelik ve kimlik doğrulama sistemi için kullanılmıştır.

Kullanıcı Yönetimi: Kullanıcılar kayıt olabilir, giriş yapabilir ve profillerini yönetebilir.

Kimlik Doğrulama: Kullanıcıların güvenli bir şekilde sisteme giriş yapması sağlanmıştır.

Yetkilendirme: Admin yetkisine sahip kullanıcıların tarifi düzenleyebilmesi ve malzeme ekleyebilmesi gibi işlemler için rol tabanlı erişim kontrolü uygulanmıştır.

ApplicationUser Sınıfı: Identity Core'un temel IdentityUser sınıfı genişletilerek projeye özgü ek alanlar (örneğin telefon numarası) dahil edilmiştir.



Fotoğraf: Entity Framework & Identity Core

➤ HTML (HyperText Markup Language)

HTML, web sayfalarının iskelet yapısını oluşturur. Projede:

Tarif listesi ve malzeme seçim formları gibi sayfalar HTML ile hazırlanmıştır.

Dinamik içerikler, Razor syntax kullanılarak ASP.NET ile entegre bir şekilde oluşturulmuştur.

➤ CSS (Cascading Style Sheets)

CSS, web sayfalarının görsel tasarımını oluşturmak için kullanılmıştır.

Modern ve kullanıcı dostu bir arayüz tasarımı sağlamak amacıyla renk paletleri, yazı tipleri ve düzenleme işlemleri yapılmıştır.

Daha profesyonel bir görünüm için hover efektleri, animasyonlar ve responsive tasarım özellikleri uygulanmıştır.

➤ JavaScript (JS)

JavaScript, dinamik içerikler ve kullanıcı etkileşimleri için kullanılmıştır.

Form doğrulamaları: Örneğin, malzeme seçim formunda kullanıcıların geçerli giriş yapmasını sağlamak.

Dinamik işlemler: Seçilen malzemelere göre tariflerin anında güncellenmesi gibi işlemler.

AJAX ile sunucuya yapılan çağrılar sayesinde sayfanın yeniden yüklenmesine gerek kalmadan veri alışverişi sağlanmıştır.

➤ Visual Studio



Fotoğraf: Visual Studio

Visual Studio, projenin geliştirilmesi için kullanılan entegre bir geliştirme ortamıdır (IDE).

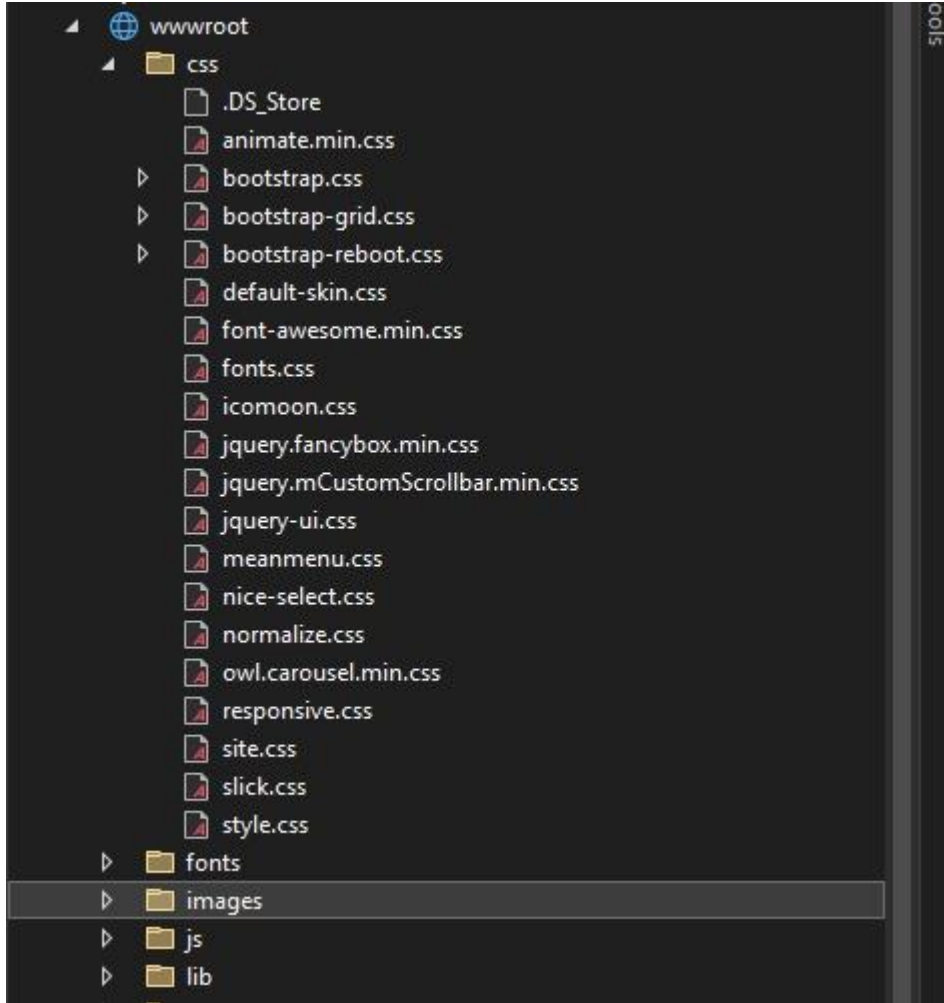
Kodlama, hata ayıklama ve test işlemleri burada yapılmıştır.

Gelişmiş IntelliSense desteği sayesinde kod yazımı hızlandırılmıştır.

➤ GIT (Kaynak Kod Yönetimi)

GIT, proje geliştirme sürecinde kodların versiyon kontrolü için kullanılmıştır. Projenin farklı sürümleri yönetilmiş, gerektiğinde eski sürümlere dönülebilmektedir.

➤ Bootstrap (CSS Framework)



Fotoğraf: Bootstrap

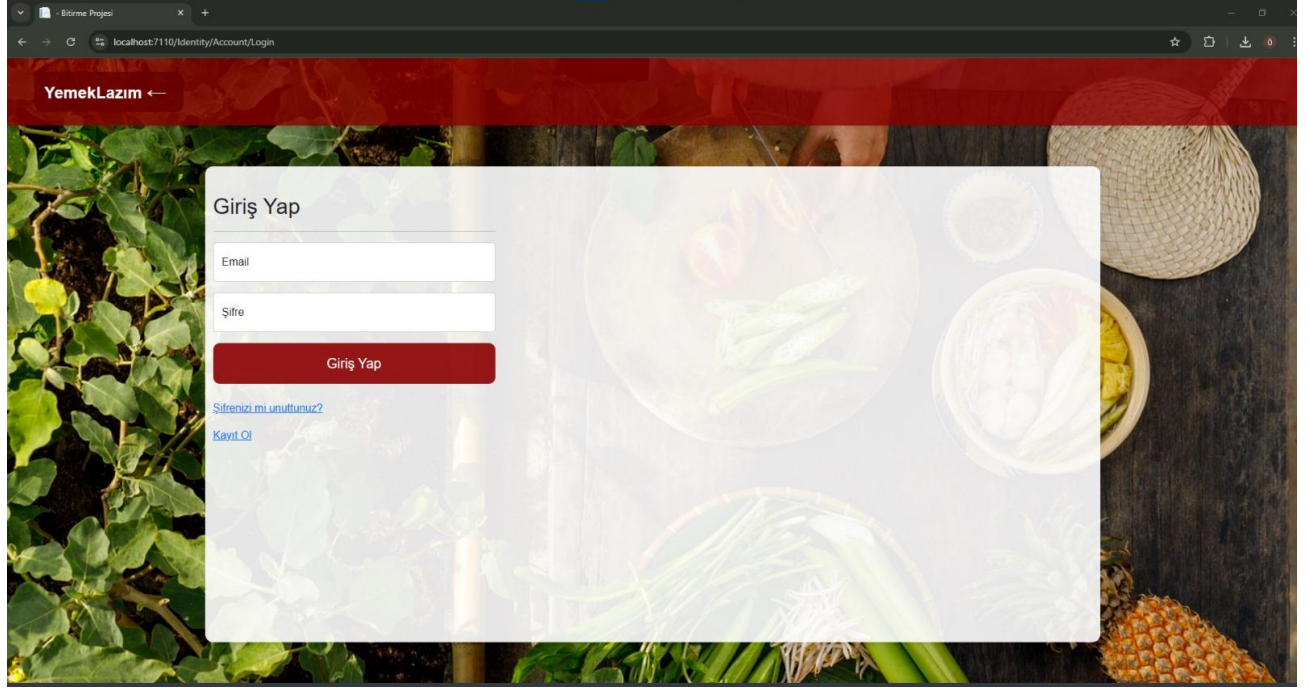
Kullanıcı arayüzü tasarımını daha hızlı ve kolay bir şekilde oluşturmak için Bootstrap kütüphanesi kullanılmıştır.

Responsive tasarım: Farklı ekran boyutlarında sayfaların uyumlu görünmesi sağlanmıştır.

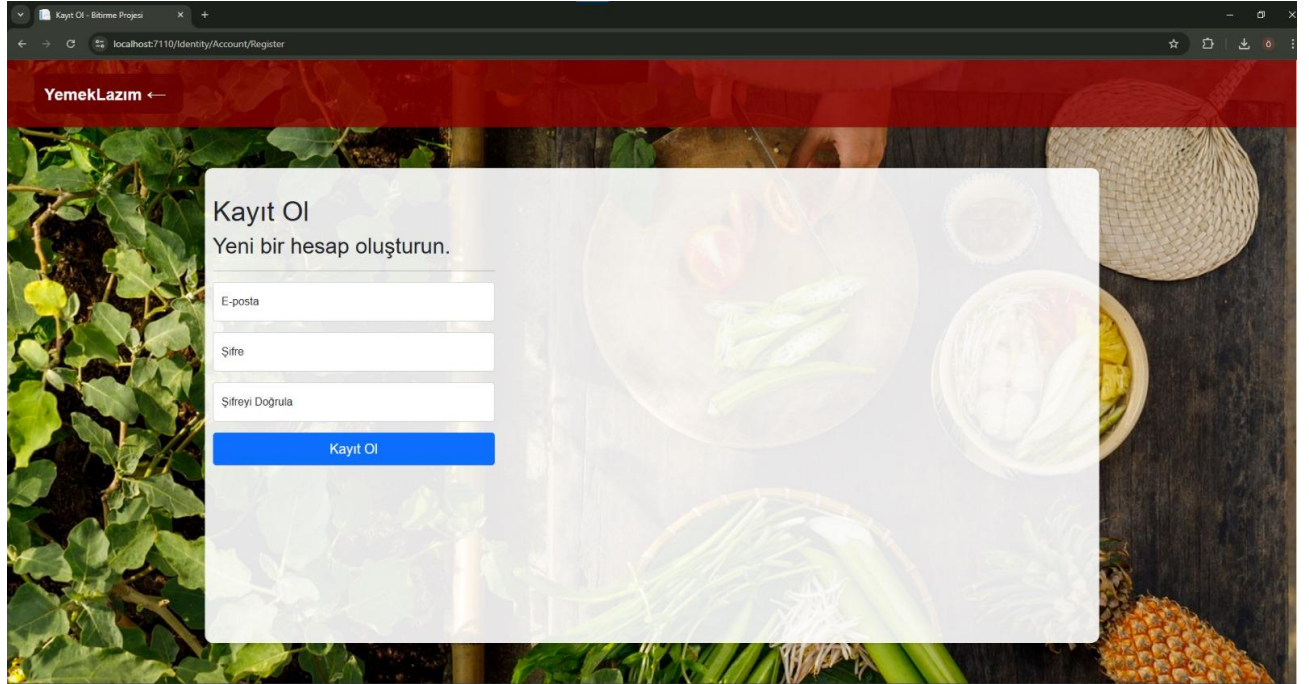
UI bileşenleri: Button, card, navbar gibi Bootstrap bileşenleriyle profesyonel bir tasarım oluşturulmuştur.

Bu teknolojilerin bir araya gelmesiyle, modern ve kullanıcı dostu bir yemek tarifleri web sitesi geliştirilmiştir. Kullanıcılar hem tarayıcı üzerinden güvenli bir şekilde giriş yapabilmekte hem de ihtiyaç duydukları yemek tariflerine kolaylıkla ulaşabilmektedir.

Görsellerle Adım Adım Proje İncelemesi



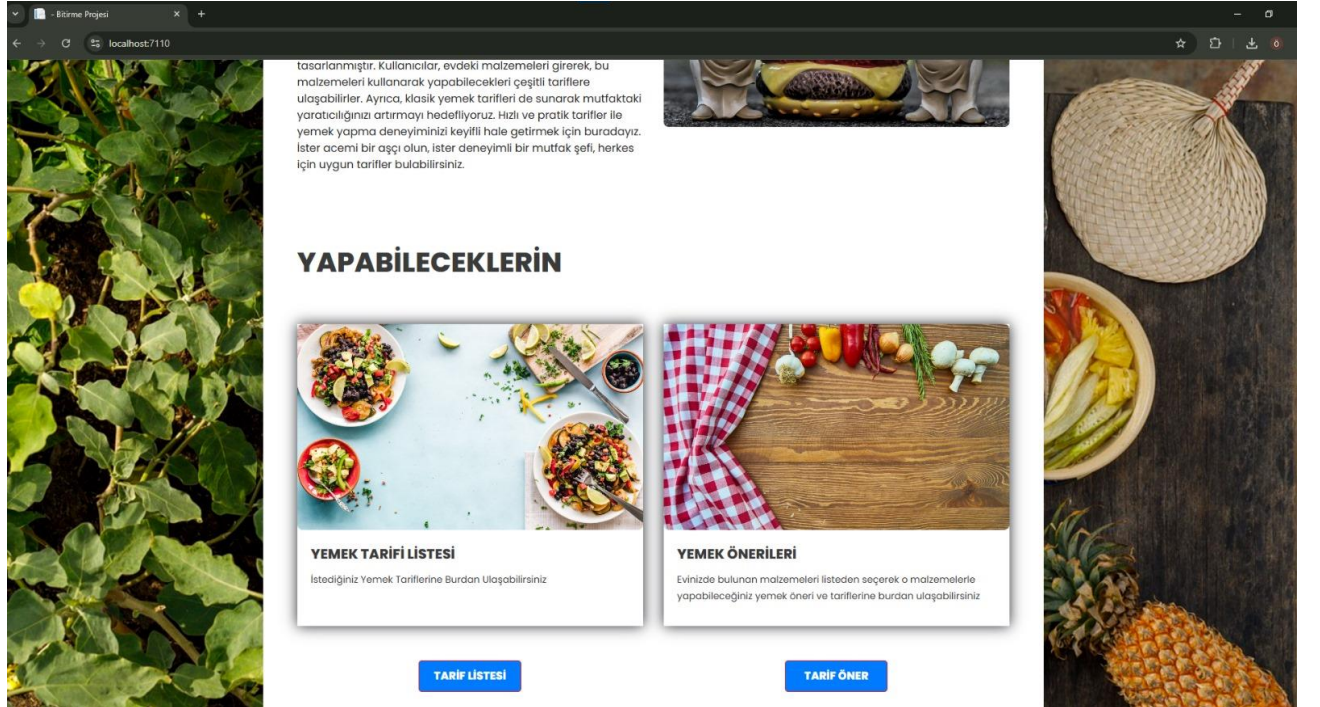
Fotoğraf : Giriş Ekranı



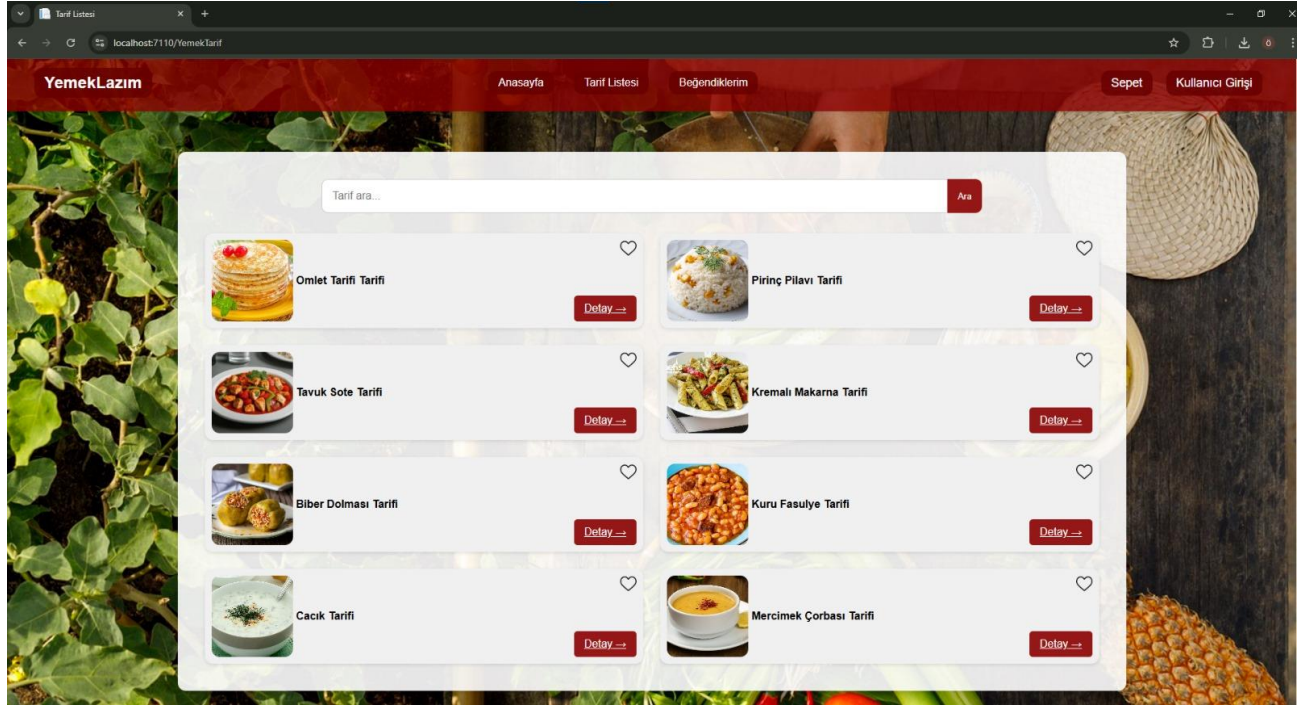
Fotoğraf : Kayıt Ol Ekranı



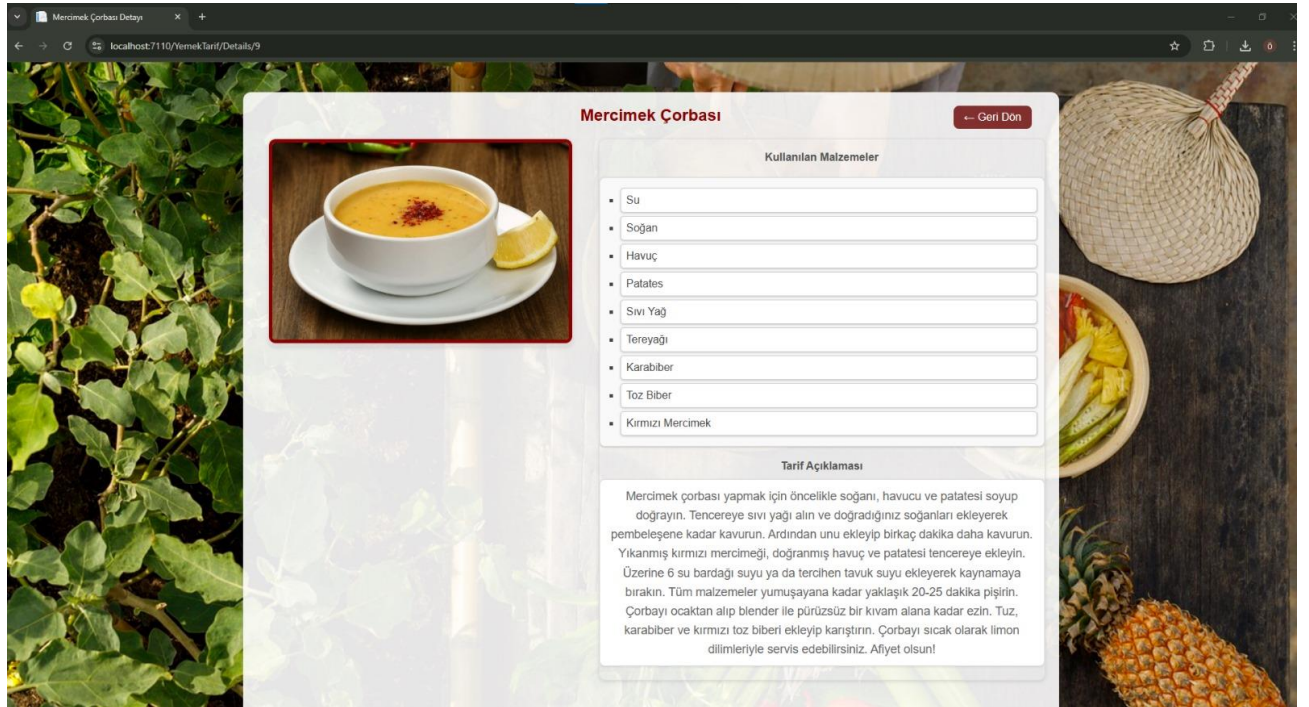
Fotoğraf : Anasayfa



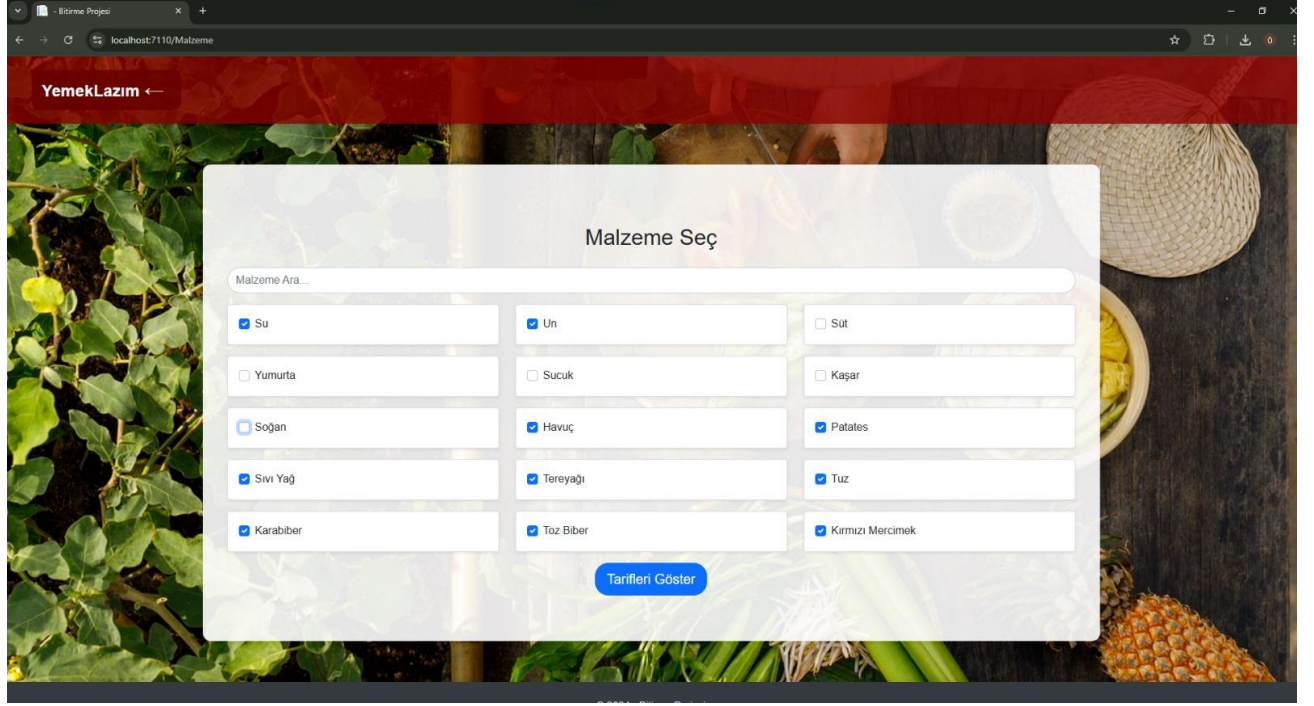
Fotoğraf : Anasayfa.2



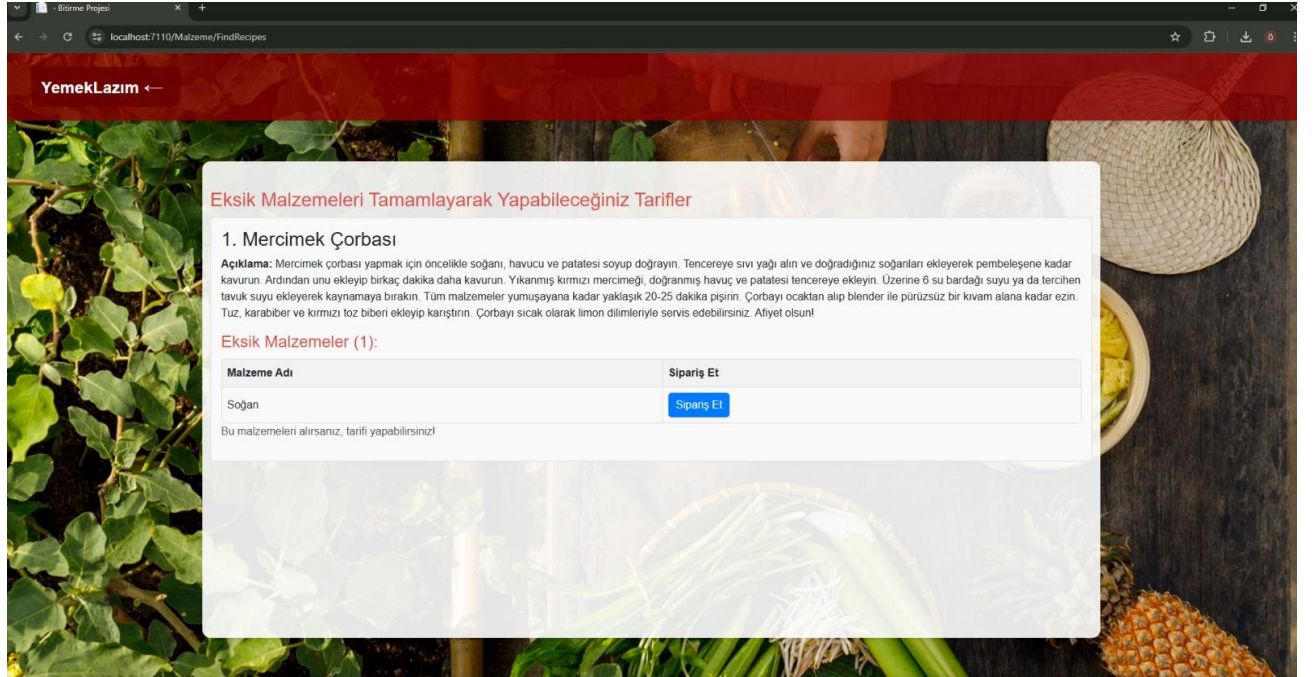
Fotoğraf : Tarif Ekranı



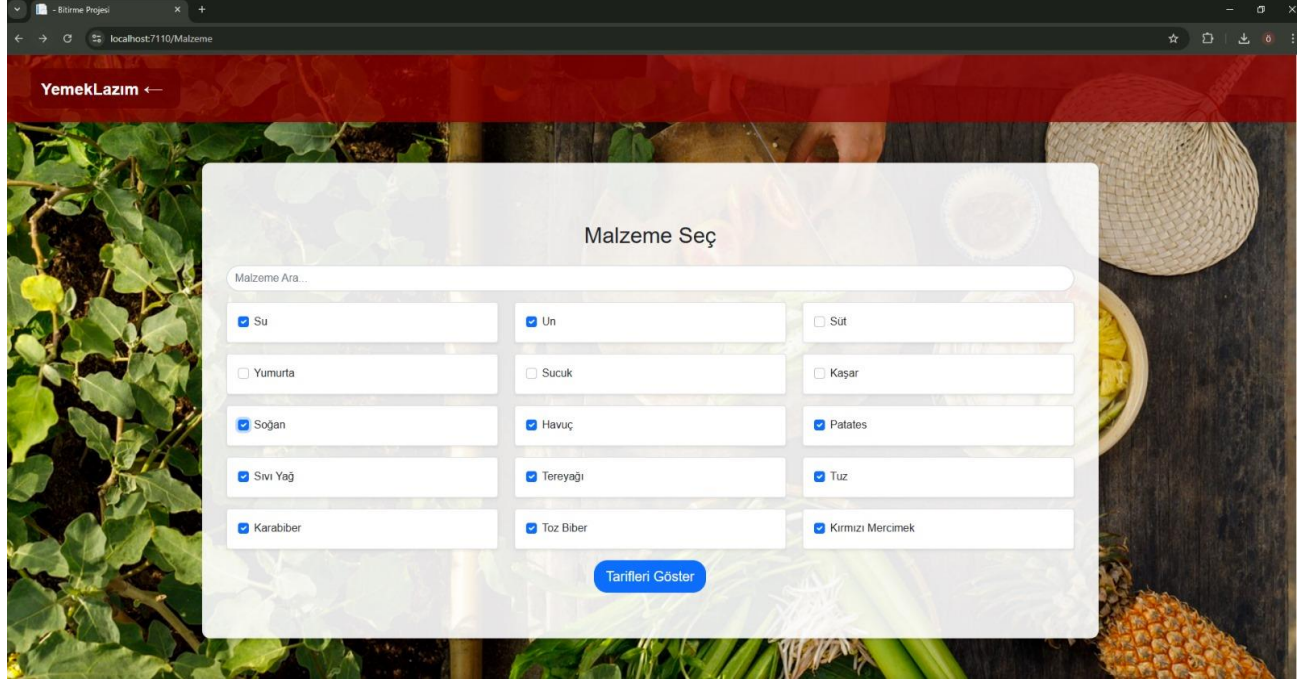
Fotoğraf : Yemek Tarif Ekranı



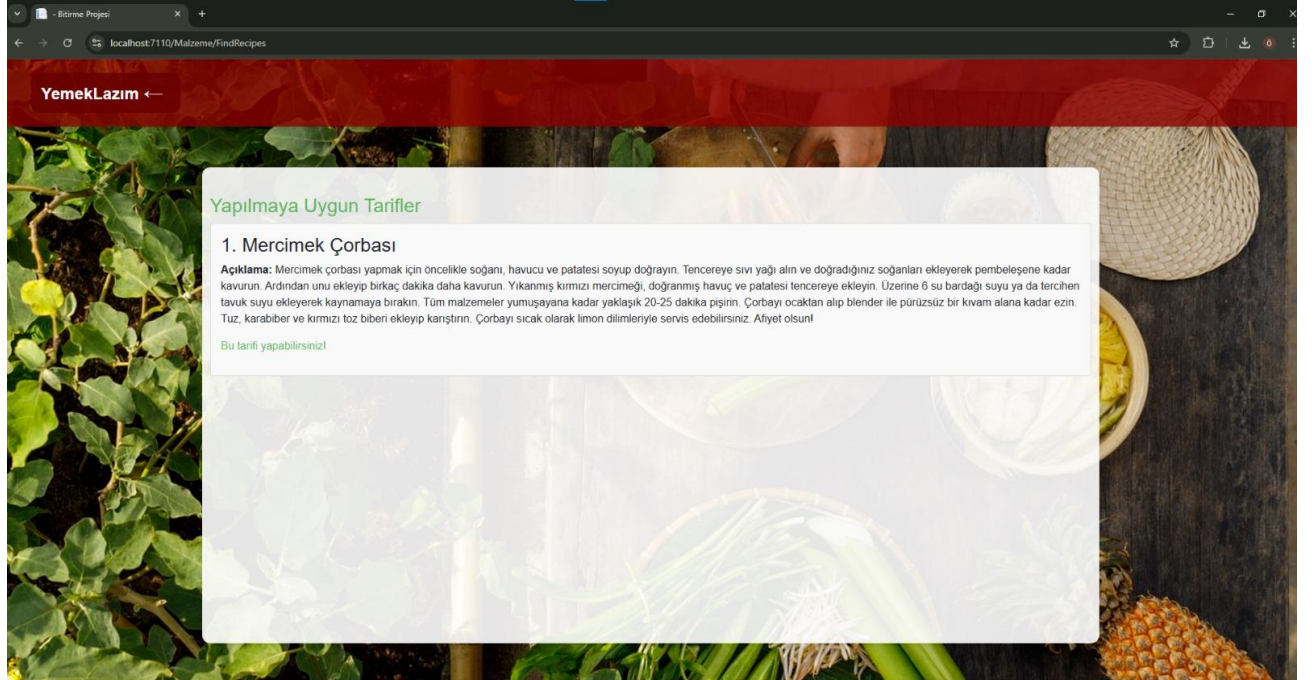
Fotoğraf : Malzeme Ekranı



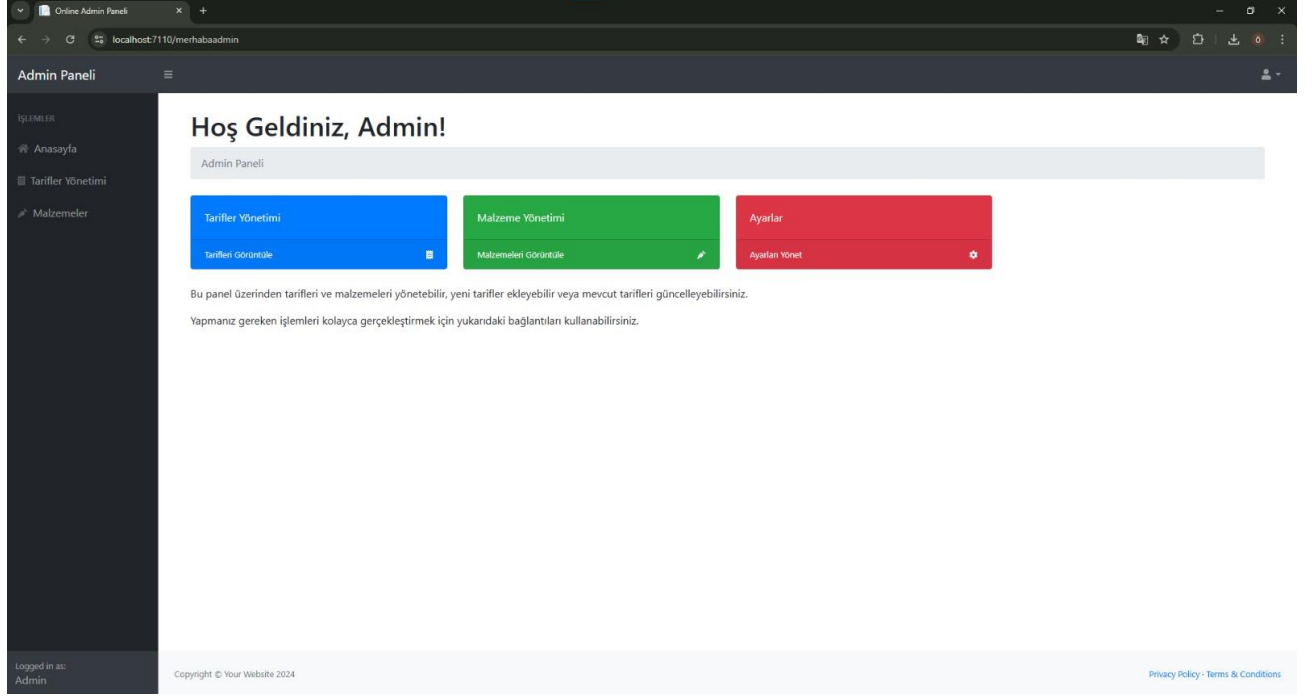
Fotoğraf : Örnek Malzeme Ekranı



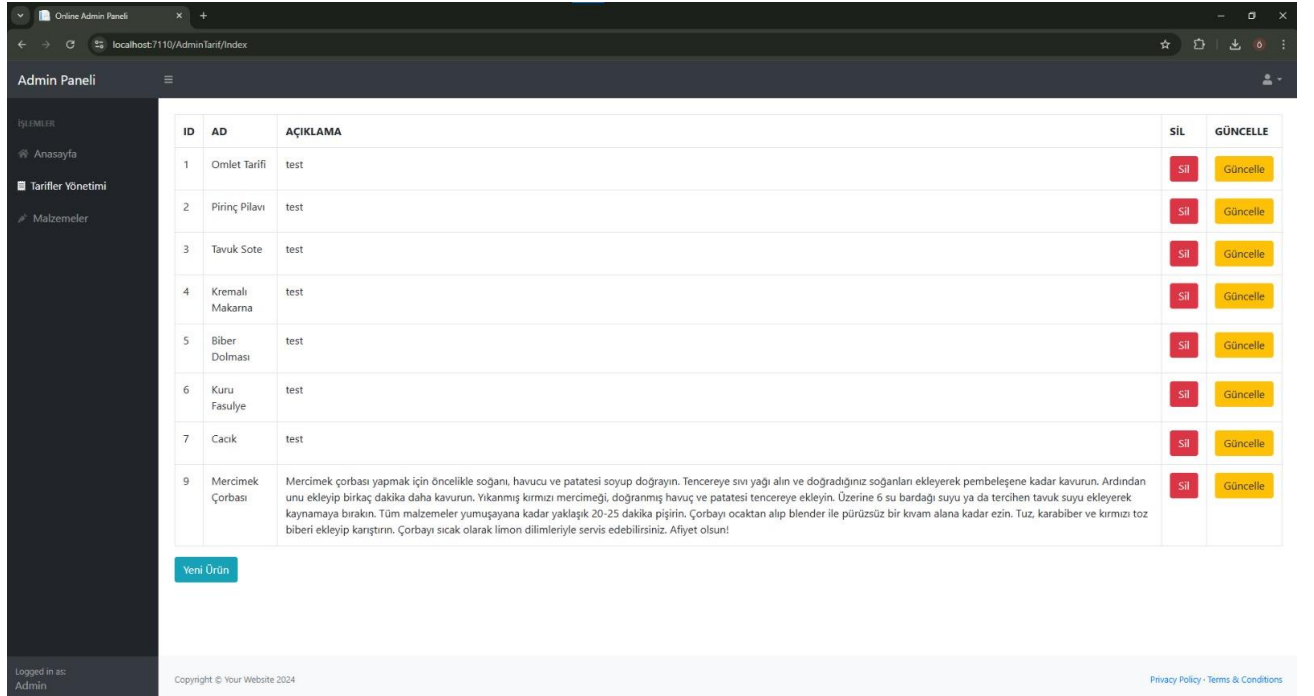
Fotoğraf : Örnek Malzeme Ekranı.2



Fotoğraf : Örnek Malzeme Ekranı.2



Fotoğraf : Admin



Fotoğraf : Admin

Online Admin Paneli

localhost:7110/AdminTarif/TarifGuncelle/9

Admin Paneli

İŞLEMLER

Anasayfa

Tarifler Yönetimi

Malzemeler

Tarif Güncelleme Sayfası

Ad

Mercimek Çorbası

Açıklama

Mercimek çorbası yapmak için öncelikle soğanı, havucu ve patatesi soyup doğrayın. Tencereye sıvı yağı alın ve doğradığınız soğanları ekleyerek pembeleşene kadar kavurun. Ardından unu ekleyip birkaç dakika daha kavurun. Yıkay

Tarif İçin Gerekli Malzemeler:

Malzeme ara...

- ☒ Su
- ☐ Un
- ☐ Süt
- ☐ Yumurta
- ☐ Sucuk
- ☐ Kaşar
- ☒ Soğan
- ☒ Havuç
- ☒ Patates
- ☒ Sıvı Yağ
- ☒ Tereyağı
- ☐ Tuz
- ☒ Karabiber
- ☒ Toz Biber
- ☒ Kırmızı Mercimek

Güncelle

Fotoğraf : Admin

Online Admin Paneli

localhost:7110/AdminMalzeme/Index

Admin Paneli

İŞLEMLER

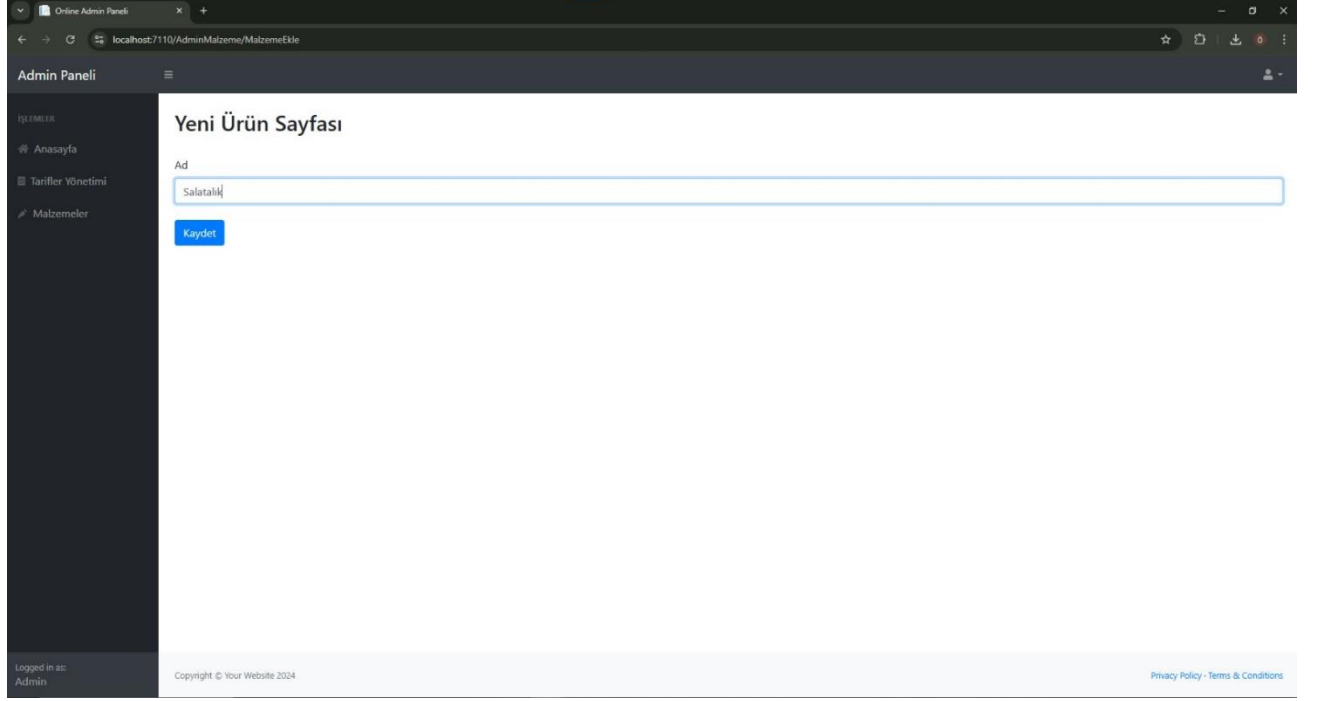
Anasayfa

Tarifler Yönetimi

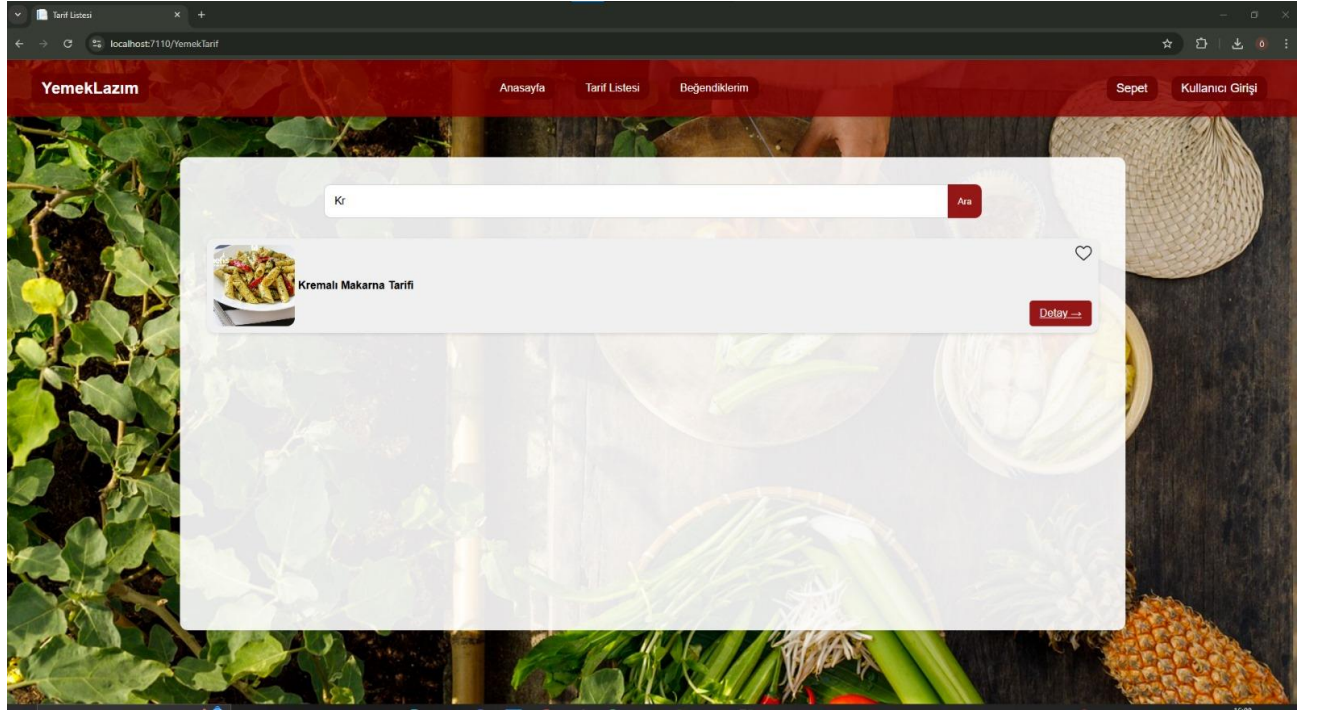
Malzemeler

ID	MALZEME	SİL	GÜNCELLE
1	Su	SİL	Güncelle
2	Un	SİL	Güncelle
3	Süt	SİL	Güncelle
4	Yumurta	SİL	Güncelle
5	Sucuk	SİL	Güncelle
6	Kaşar	SİL	Güncelle
7	Soğan	SİL	Güncelle
8	Havuç	SİL	Güncelle
9	Patates	SİL	Güncelle
10	Sıvı Yağ	SİL	Güncelle
11	Tereyağı	SİL	Güncelle
12	Tuz	SİL	Güncelle
13	Karabiber	SİL	Güncelle

Fotoğraf : Admin



Fotoğraf : Admin



Fotoğraf : Admin

Kod GÖRSELLERİ

```
4
5 namespace BitirmeProjesiDeneme2.Controllers
6 {
7     1 reference
8     public class YemekTarifController : Controller
9     {
10         private readonly UygulamaDbContext _context;
11
12         0 references
13         public YemekTarifController(UygulamaDbContext context)
14         {
15             _context = context;
16
17         0 references
18         public IActionResult Index(string search)
19         {
20             var tarifler = _context.Tarifler.AsQueryable();
21
22             if (!string.IsNullOrEmpty(search))
23             {
24                 tarifler = tarifler.Where(t => t.Ad.Contains(search));
25             }
26
27             return View(tarifler.ToList());
28         }
29
30         0 references
31         public IActionResult Search(string search)
32         {
33             var tarifler = _context.Tarifler.AsQueryable();
34
35             if (!string.IsNullOrEmpty(search))
36             {
37                 tarifler = tarifler.Where(t => t.Ad.StartsWith(search));
38             }
39
40             return Json(tarifler.ToList());
41         }
42
43         0 references
44         public IActionResult Details(int id)
45         {
46             var tarif = _context.Tarifler
47                 .Include(t => t.TarifMalzemeler)
48                 .ThenInclude(tm => tm.Malzeme)
49                 .FirstOrDefault(t => t.Id == id);
50
51             if (tarif == null)
52             {
53                 return NotFound();
54             }
55
56             return View(tarif);
57         }
58     }
59 }
```

Fotoğraf : Tarif Listeleme

```

0 references
public MalzemeController(UygulamaDbContext context)
{
    _context = context;
}

0 references
public IActionResult Index()
{
    var malzemeler = _context.Malzemeler.ToList();
    return View(malzemeler);
}

[HttpPost]
0 references
public IActionResult FindRecipes(int[] selectedIngredients)
{
    if (selectedIngredients == null || selectedIngredients.Length == 0)
    {
        ViewBag.EksikMalzemeBilgisi = "Lütfen en az bir malzeme seçin.";
        return View("FindRecipes", new List<Tarif tarif, List<string> eksikler>());
    }

    var tarifler = _context.Tarifler.Include(t => t.TarifMalzemeler).ToList();
    var uygunTarifler = new List<Tarif tarif, List<string> eksikler>();

    foreach (var tarif in tarifler)
    {
        var tarifMalzemeleri = tarif.TarifMalzemeler.Select(tm => tm.MalzemeId).ToList();

        var eksikler = tarifMalzemeleri.Except(selectedIngredients).ToList();
        var fazlaMalzemeler = selectedIngredients.Except(tarifMalzemeleri).ToList();

        if (selectedIngredients.All(si => !tarifMalzemeleri.Contains(si)))
        {
            continue;
        }

        if (eksikler.Any())
        {
            var eksikMalzemeAdları = eksikler.Select(em => _context.Malzemeler.Find(em)?.Ad).ToList();
            uygunTarifler.Add((tarif, eksikMalzemeAdları));
        }
        else
        {
            uygunTarifler.Add((tarif, null));
        }
    }

    if (!uygunTarifler.Any())
    {
        ViewBag.EksikMalzemeBilgisi = "Seçilen malzemelerle hiçbir tarif bulunamadı.";
    }
}

```

Fotoğraf : malzemeye göre tarif bulma


```

private readonly UygulamaDbContext _context;

public AdminMalzemeController(UygulamaDbContext context)
{
    _context = context;
}

public ActionResult Index()
{
    var malzemeler = _context.Malzemeler.ToList();
    return View(malzemeler);
}

[HttpGet]

public ActionResult MalzemeEkle()
{
    return View();
}

[HttpPost]

public ActionResult MalzemeEkle(Malzeme m)
{
    _context.Malzemeler.Add(m);
    _context.SaveChanges();
    return RedirectToAction("Index");
}

public ActionResult MalzemeSil(int id)
{
    var malzeme = _context.Malzemeler.Find(id);
    _context.Malzemeler.Remove(malzeme);
    _context.SaveChanges();
    return RedirectToAction("Index");
}

[HttpGet]

public ActionResult MalzemeGuncelle(int id)
{
    var malzeme = _context.Malzemeler.Find(id);
    return View(malzeme);
}

[HttpPost]

public ActionResult MalzemeGuncelle(Malzeme m)
{
    var malzeme = _context.Malzemeler.Find(m.Id);
    malzeme.Ad = m.Ad;
    _context.SaveChanges();
    return RedirectToAction("Index");
}
}

```

Fotoğraf : Admin Malzeme Yönetimi

```

0 references
public ActionResult TarifSil(int id)
{
    var tarif = _context.Tarifler.Find(id);
    _context.Tarifler.Remove(tarif);
    _context.SaveChanges();
    return RedirectToAction("Index");
}

[HttpGet]
0 references
public ActionResult TarifGuncelle(int id)
{
    var tarif = _context.Tarifler.Find(id);
    ViewBag.Malzemeler = _context.Malzemeler.ToList();
    ViewBag.SelectedMalzemeler = _context.TarifMalzemeleri
        .Where(tm => tm.TarifId == id)
        .Select(tm => tm.MalzemeId)
        .ToList();
    return View(tarif);
}

[HttpPost]
0 references
public ActionResult TarifGuncelle(Tarif t, List<int> selectedMalzemeIds)
{
    var tarif = _context.Tarifler.Find(t.Id);
    tarif.Ad = t.Ad;
    tarif.Açıklama = t.Açıklama;
    _context.SaveChanges();

    var eskiMalzemeler = _context.TarifMalzemeleri.Where(tm => tm.TarifId == t.Id).ToList();
    _context.TarifMalzemeleri.RemoveRange(eskiMalzemeler);
    _context.SaveChanges();

    if (selectedMalzemeIds != null)
    {
        foreach (var malzemeId in selectedMalzemeIds)
        {
            var tarifMalzeme = new TarifMalzeme
            {
                TarifId = t.Id,
                MalzemeId = malzemeId
            };
            _context.TarifMalzemeleri.Add(tarifMalzeme);
        }
    }
}

```

Fotoğraf : Admin Tarif Yönetimi

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
  <meta name="description" content="" />
  <meta name="author" content="" />
  <title> Online Admin Paneli</title>
  <link href="/Web/dist/css/styles.css" rel="stylesheet" />
  <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.2/js/all.min.js" crossorigin="anonymous"></script>
  <style>
    .navbar-nav {
      margin-left: auto;
    }
  </style>
</head>
<body>
  <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">
    <a class="navbar-brand" href="#">Admin Paneli</a>
    <button class="btn btn-link btn-sm order-1 order-lg-0" id="sidebarToggle" href="#">
      <i class="fas fa-bars"></i>
    </button>
    <ul class="navbar-nav">
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" id="userDropdown" href="#" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false"><i class="fas fa-user fa-fw"></i></a>
        <div class="dropdown-menu dropdown-menu-right" aria-labelledby="userDropdown">
          <a class="dropdown-item" href="#">Ayarlar</a>
          <div class="dropdown-divider"></div>
          <a class="dropdown-item" href="Login.html">Çıkış Yap</a>
        </div>
      </li>
    </ul>
  </nav>
  <div id="layoutSidenav">
    <div id="layoutSidenav_nav">
      <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
        <div class="sb-sidenav-menu">
          <div class="nav">
            <div class="sb-sidenav-menu-heading">İşlemler</div>
            <a class="nav-link" href="/MerhabaAdmin/Index">
              <div class="sb-nav-link-icon"><i class="fas fa-home"></i></div>
              Anasayfa
            </a>
            <a class="nav-link" href="/AdminTarif/Index">
              <div class="sb-nav-link-icon"><i class="fas fa-receipt"></i></div>
              Tarifler Yönetimi
            </a>
            <a class="nav-link" href="/AdminMalzeme/Index">
              <div class="sb-nav-link-icon"><i class="fas fa-carrot"></i></div>
              Malzemeler
            </a>
          </div>
        </div>
        <div class="sb-sidenav-footer">
          <div class="small">Logged in as:</div>
          Admin
        </div>
      </nav>
    </div>
  </div>

```

Fotoğraf : Admin Layout

```

namespace BitirmeProjesiDeneme2.Utility
{
    21 references
    public class UygulamaDbContext : IdentityDbContext
    {
        0 references
        public UygulamaDbContext(DbContextOptions<UygulamaDbContext> options) : base(options) { }
        10 references
        public DbSet<Tarif> Tarifler { get; set; }
        10 references
        public DbSet<Malzeme> Malzemeler { get; set; }
        5 references
        public DbSet<TarifMalzeme> TarifMalzemeleri { get; set; }
        0 references
        public DbSet<ApplicationUser> ApplicationUsers { get; set; }
    }
}

```

Fotoğraf : Veritabanı context nesnesi

```

using BitirmeProjesiDeneme2.Utility;
using Microsoft.EntityFrameworkCore;
using Microsoft.AspNetCore.Identity;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();

builder.Services.AddDbContext<UygulamaDbContext>(options =>
options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

builder.Services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true).AddEntityFrameworkStores<UygulamaDbContext>();

builder.Services.AddRazorPages();

builder.Services.AddHttpClient();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapRazorPages();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

```

Fotoğraf : program.cs

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Server=DESKTOP-70T9926\\SQLEXPRESS;Database=YemekLazimDb;Trusted_Connection=True;TrustServerCertificate=True"
  }
}
```

Fotoğraf : program.cs

Kaynaklar

1. MSSQL Documentation - Veritabanı yönetimi ve tasarımı.
<https://learn.microsoft.com/en-us/sql/>
2. Yemek Tarifleri Siteleri Örnek tarifler ve içerik.
<https://www.nefisyemektarifleri.com>
3. .NET Framework Documentation - Backend geliştirme rehberi.
<https://learn.microsoft.com/en-us/dotnet/>