

**ANKARA ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**YEMEK TARİFİ PROJESİ**

**BLM4531**

**Batuhan Arda Kibrit**

**20290266**

**Github Linki:**

**<https://github.com/BatuhanArdaKibrit/BLM4531>**

## **ÖZET**

Proje dotnet ve angular kullanılarak güvenli ve stabil çalışan, herkesin kendi tariflerini paylaştığı, düzenlediği ve isterse silebildiği kullanımı basit ama işlevsel bir web sitesidir. Herkesin kendine özgü favori ve tariflerim sekmesi bulunmaktadır. Ana sayfada ise uygulamaya kayıt olmuş kişilerin oluşturduğu tüm tarifler mevcuttur. Bu şekilde herkesin etkileşimde olduğu bir platform oluşturulmaya çalışılmıştır. Uygulamada kayıt olan kişiler sonradan isterlerse Ad-Soyad, Email ve Parola gibi bilgilerini değiştirebilmektedir. Angular tarafında Json Web Token (jwt) sayesinde session ve uygulama güvenliği sağlanmıştır.

# İÇİNDEKİLER

ÖZET.....	1
İÇİNDEKİLER.....	2
1. Projenin Amacı.....	3
2. Projede Kullanılan Teknolojiler.....	3
3. Giriş Ekranı.....	3
4. Sayfalar.....	4
4.1. <u>Tüm tarifler</u> .....	4
4.2. <u>Favoriler sayfası</u> .....	5
4.3. <u>Kullanıcı tarifleri</u> .....	5
4.4. <u>Tarif sekmesi</u> .....	7
5. Hesap Bilgileri.....	9
6. Kaynak kod.....	10
KAYNAKLAR.....	13
EKLER.....	14
Ek-1 - <u>Swagger UI</u> .....	14
Ek-2 - <u>Angular routes</u> .....	14
Ek-3 - <u>Kayıtlı hesap bulunmaktadır</u> .....	15
Ek-4 - <u>Hesabı sil</u> .....	15
Ek-5 - <u>Eski parola uyarı</u> .....	15
Ek-6 - <u>Tarif silme onay</u> .....	16

## 1. Projenin Amacı

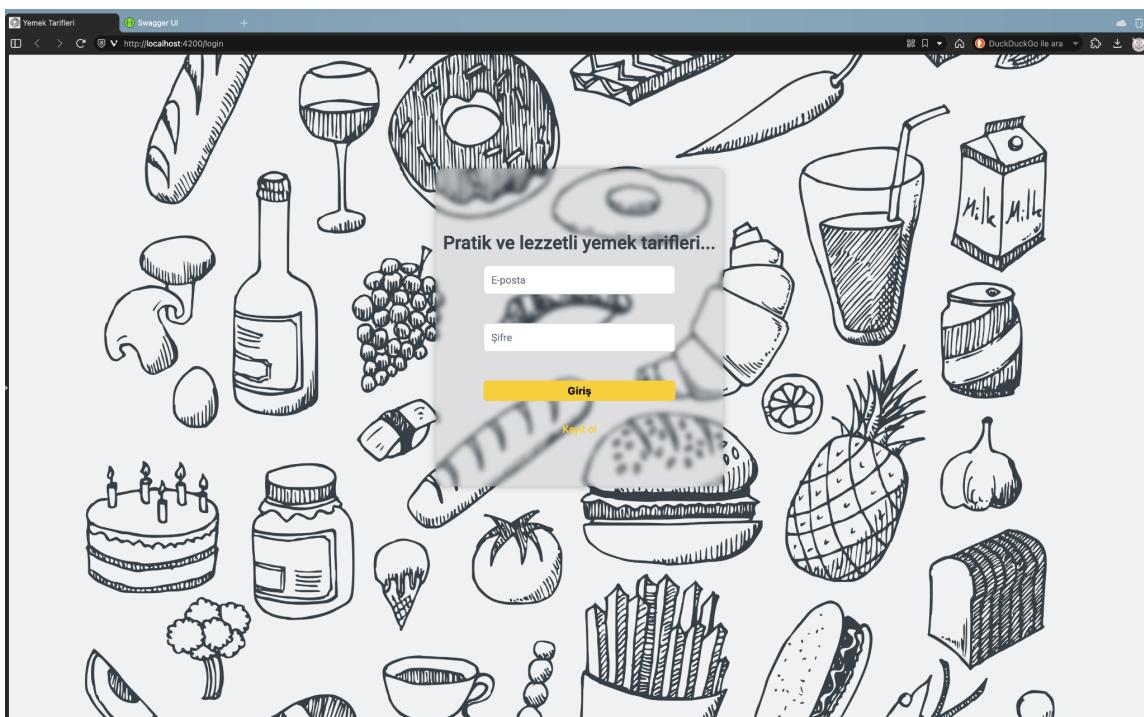
Web sitesini kullanan kullanıcılar hem kendi kaydetmek istedikleri tariflerini kayıt edebilmektedir hem de kullanan diğer kullanıcıların paylaştığı tarifleri görebilmektedir bu sayede herkesin etkileşimde olduğu ve not defteri gibi bir kullanımı olan web sitesi oluşturulmuştur.

## 1. Projede Kullanılan Teknolojiler

Projede frontend kısmında angular version 18 ve backend kısmında ise dotnet version 8 kullanılmıştır. Database olarak da PostgreSQL version 14 kullanılmıştır. Bazı fonksiyonel işlemleri gerçekleştirebilmek için bu teknolojilerin kullanıcılar tarafından yapılmış bazı ek paketleri de kullanılmıştır. Swagger ekranı (Ek-1) ve Frontend de kullanılan route lar (Ek-2) belirtilmiştir.

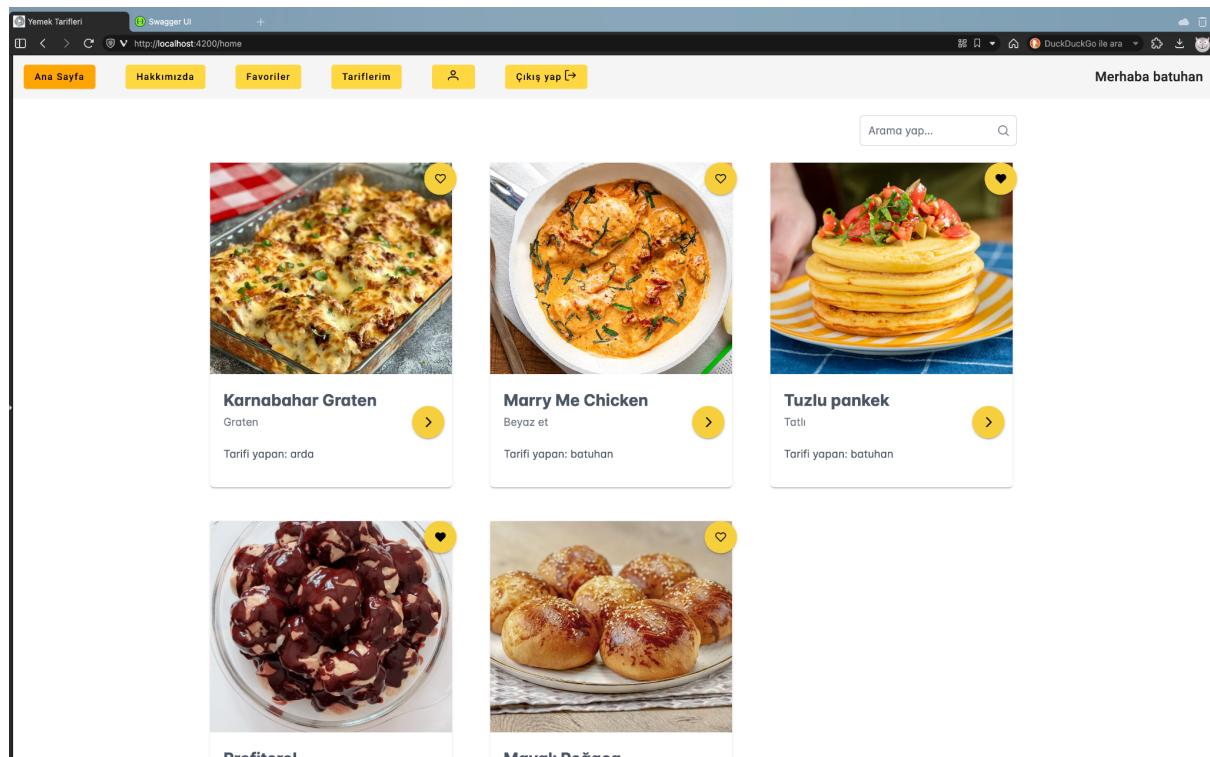
## 2. Giriş Ekranı

Giriş ekranı ve kayıt ol ekranları mevcuttur bu sayfada belli uyarı ekranları mevcuttur böyle bir kullanıcı bulunmamaktadır (Ek-3), şifre yanlış girilmiştir (Ek-4) ve kullanıcı zaten üyedir (Ek-5) uyarıları gibi. Eğer hesaba giriş yapılmış ama çıkış yap tuşuna basılmadıysa tekrar sayfa açılrsa otomatik olarak Ana sayfa açılacaktır. Bu session süresi 1 gündür, 1 günden sonra jwt geçersiz kalacağı için tekrar giriş yapılması gerekecektir. Giriş yapılmaz ise diğer hiçbir sayfa açılmayacaktır.

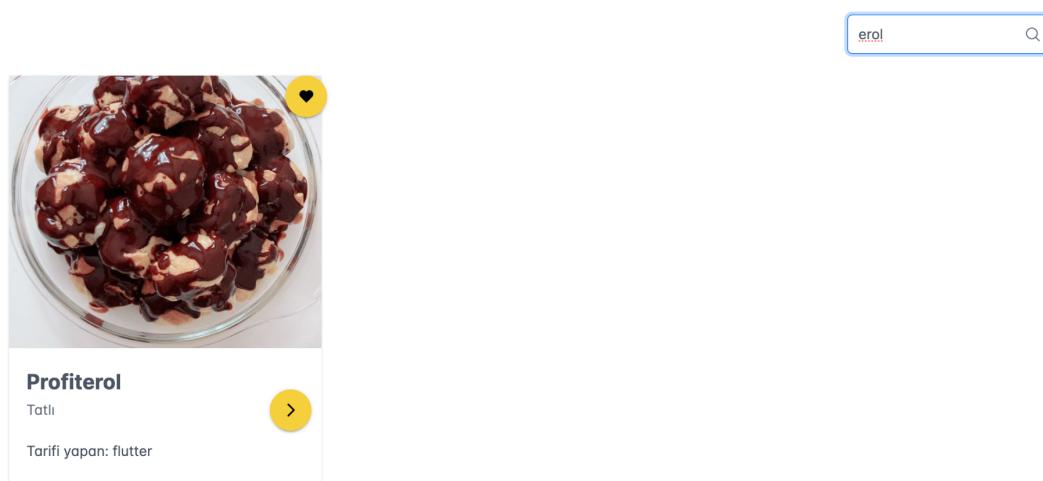


## 3. Sayfalar

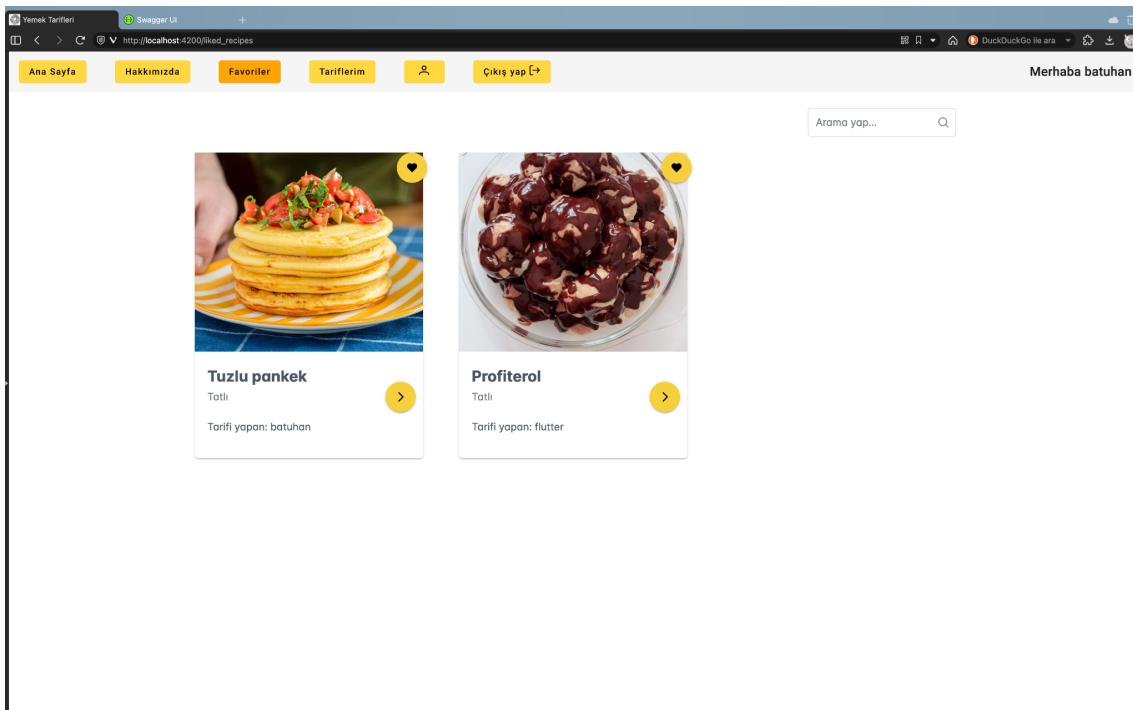
### 4.1. Tüm tarifler



Ana sayfada tüm kullanıcıların oluşturduğu tarifler bulunmaktadır. Arama kutusu da mevcuttur arama kutusunda sonuç bulunamazsa ekrana bastırılmaktadır.



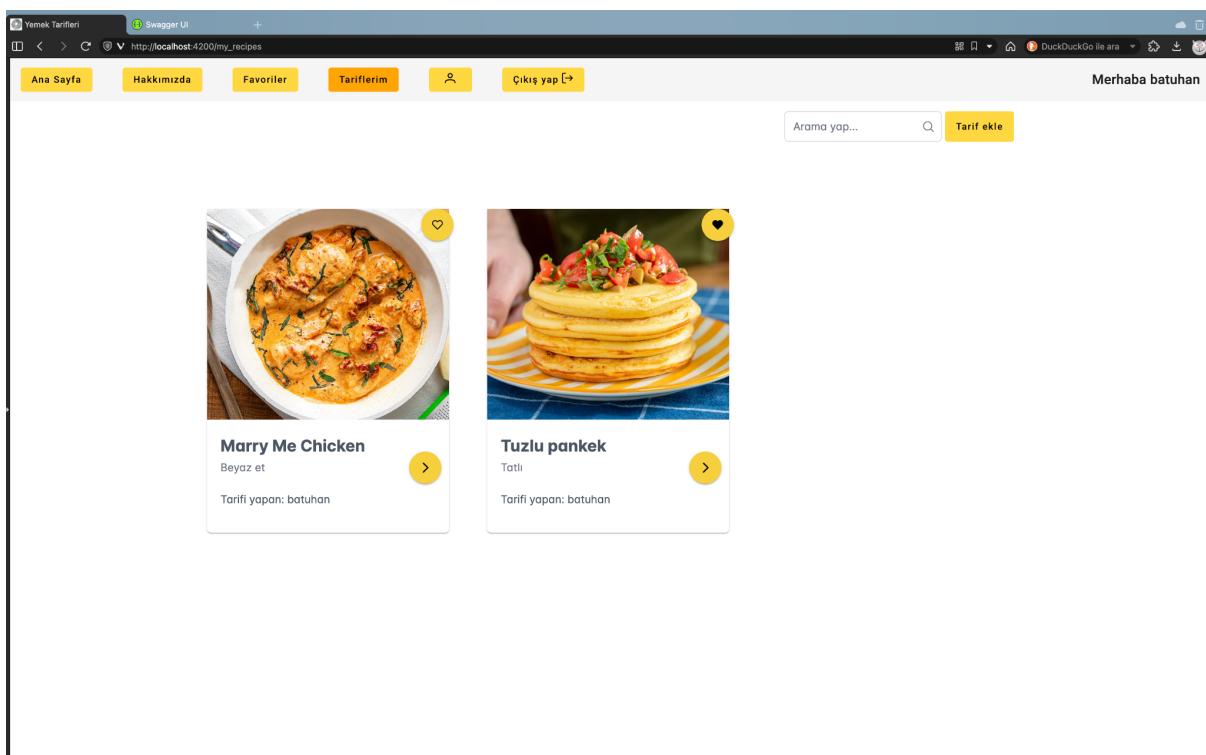
## **4.2. Favoriler sayfası**



Favoriler sayfasında her kullanıcının kendi favorilere eklediği tarifler bulunmaktadır. Her kullanıcının favori ekranı ayrıdır. Favoriler ekranındayken favorilerden kaldırma işlemi gerçekleştirilirse tarif kaybolur ve eğer favoriler kısmında bir tarif bulunmuyorsa tarif bulunmadığını belirten bir yazı çıkmaktadır.

## **4.3. Kullanıcı tarifleri**

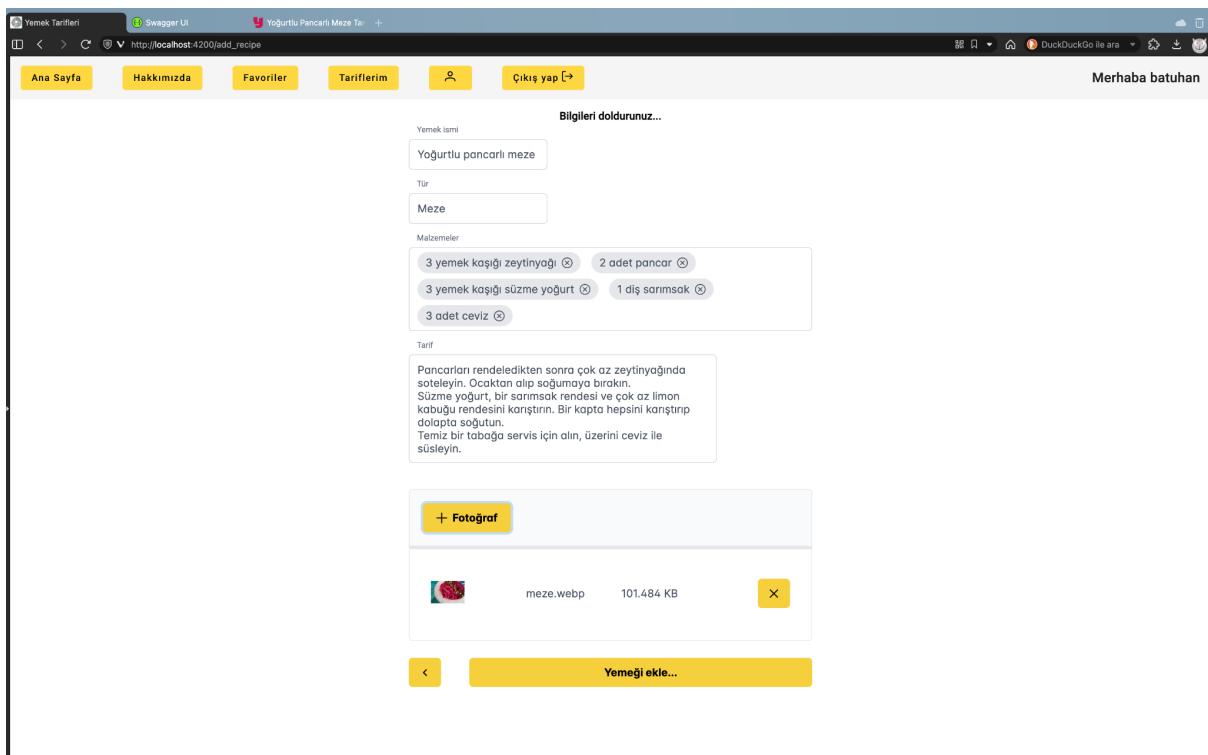
Tariflerim sayfasında Ana sayfa ve Favorilerim sayfasındaki arama kutusuna ek olarak Tarif Ekle tuşu bulunmaktadır bu tuş sayesinde kullanıcı Ana Sayfaya ve kendi Tariflerim sayfasına tarif eklemiş olacaktır.



The screenshot shows the 'add\_recipe' form page of the Yemek Tarifleri application. The top navigation bar is identical to the homepage. The main area contains fields for entering recipe information:

- Bilgileri doldurunuz...**
- 
- 
- 
- 
- 
-

Herhangi bir tarif ekledikten sonra ekran aşağıdaki ekran görüntüsünde olduğu gibi gözükmektedir fotoğraf sürüklenecek de yüklenemilmektedir. Sadece resim dosyalı kabul edilmektedir. Database de resim dosyası byte array ve malzemeler de string array olarak tutulmaktadır.



#### 4.4. Tarif sekmesi

Tarif sekmesine herhangi bir sayfada kutu içinde bulunan tariflerin sağ alt kısmında “>” sembollü tuşa basılarak tarifin detaylarına ulaşılabilirmektedir. Tarif sekmesinde bulunan Tarifi Sil ve Tarifi Düzenle butonları sadece tarifi yapan kişi tıkladı ise görünür olmaktadır. Eğer tarifi yazmadığı halde başka yollardan tarifi düzenlemeye çalışan birisi olursa bunun için önlem alınmıştır.

Tarifi sil butonuna tıklandığında (Ek-6) onaylama kutusu çıkmaktadır eğer işlem onaylanırsa tarif bir daha ulaşılamayacak şekilde silinmektedir.

Tarifi düzenle butonunda ise istenilen herhangi bir değişken değiştirilebilmektedir bunlar sadece fotoğraf veya sadece isim ya da tüm her şeyi baştan sona değiştirilebilir bunda herhangi bir sınır yoktur.

Yemek Tarifleri    Swagger UI    +

http://localhost:4200/food\_detail/30

Ana Sayfa Hakkımızda Favoriler Tarifierim Çıkış yap [→]

Merhaba batuhan

# Marry Me Chicken

Beyaz et

Tarifi yapan: batuhan

**Malzemeler**

- 2 adet tavuk göğüsü
- 1/3 su bardağı un
- 1 çay kaşığı tuz
- 1/2 çay kaşığı karabiber
- 1 tatlı kaşığı toz kırmızı biber

**Tarif**

Marry me chicken için tavuk göğüsleri yatay olarak ortadan ikiye kesin ve dört adet ince göğüs parçası elde edin. Un, tuz, karabiber ve kırmızı toz biberi bir tabaka karıştırın. Tavuk göğüsleri arkası önlü olacak şekilde bir karışma bulanın. Geniş bir tavada zeytinyağı ve tereyağına orta ateşte istatin. Tavuk göğüsleri tavaya alın ve her iki tarafı da kuzarıp renk elana dek pişirin. Piştiğinden sonra bir tabağa alıp kenara koyn. Tavukları pişirdiğiniz tencerede fazla yağ varsa birazını dökün ve tekrar ocağa alın. Sarımsaklıları ince ince doğrayın veya eziciden geçirin ve aynı tava da kavurmayı başlayın. Üzerine tavuk suyunu ekleyin ve bir spatula yardımıyla karıştırın. Ocağı kısık ateşte alın ve kremayı da ilave edin. Ara ara yavaşça karıştırarak pişirin. Kremalı haflaşmış kıvam elincice baharatları, rendelenmiş parmesan peynirini ve kurutulmuş domatesleri de ekleyip karıştırın. Piştiğiniz tavuk göğüsleri tavadaki sosun içerişine ekleyin ve sosla beraber birkaç dakika pişirin. Ocağın altını kapatın ve tava kapağına sarın. Sade makarna, haşlanmış sebzeler veya pilav ile servis edebilirsiniz. Afiyet olsun!

Favoriler ekle Tarifi sil Tarifi düzenle

Yemek Tarifleri    Swagger UI    Yoğurtlu Pancarlı Meze Tarifi    +

http://localhost:4200/edit\_recipe/30

Malzemeler

2 adet tavuk göğüsü 1/3 su bardağı un 1 çay kaşığı tuz

1/2 çay kaşığı karabiber 1 tatlı kaşığı toz kırmızı biber

**Tarif**

Marry me chicken için tavuk göğüsleri yatay olarak ortadan ikiye kesin ve dört adet ince göğüs parçası elde edin. Un, tuz, karabiber ve kırmızı toz biberi bir tabakta karıştırın. Tavuk göğüsleri arkası önlü olacak şekilde bir karışma bulanın. Geniş bir tavada zeytinyağı ve tereyağına orta ateşte istatin. Tavuk göğüsleri tavaya alın ve her iki tarafı da kuzarıp renk elana dek pişirin. Piştiğinden sonra bir tabağa alıp kenara koyn. Tavukları pişirdiğiniz tencerede fazla yağ varsa birazını dökün ve tekrar ocağa alın. Sarımsaklıları ince ince doğrayın veya eziciden geçirin ve aynı tava da kavurmayı başlayın. Üzerine tavuk suyunu ekleyin ve bir spatula yardımıyla karıştırın. Ocağı kısık ateşte alın ve kremayı da ilave edin. Ara ara yavaşça karıştırarak pişirin. Kremalı haflaşmış kıvam elincice baharatları, rendelenmiş parmesan peynirini ve kurutulmuş domatesleri de ekleyip karıştırın. Piştiğiniz tavuk göğüsleri tavadaki sosun içerişine ekleyin ve sosla beraber birkaç dakika pişirin. Ocağın altını kapatın ve tava kapağına sarın. Sade makarna, haşlanmış sebzeler veya pilav ile servis edebilirsiniz. Afiyet olsun!

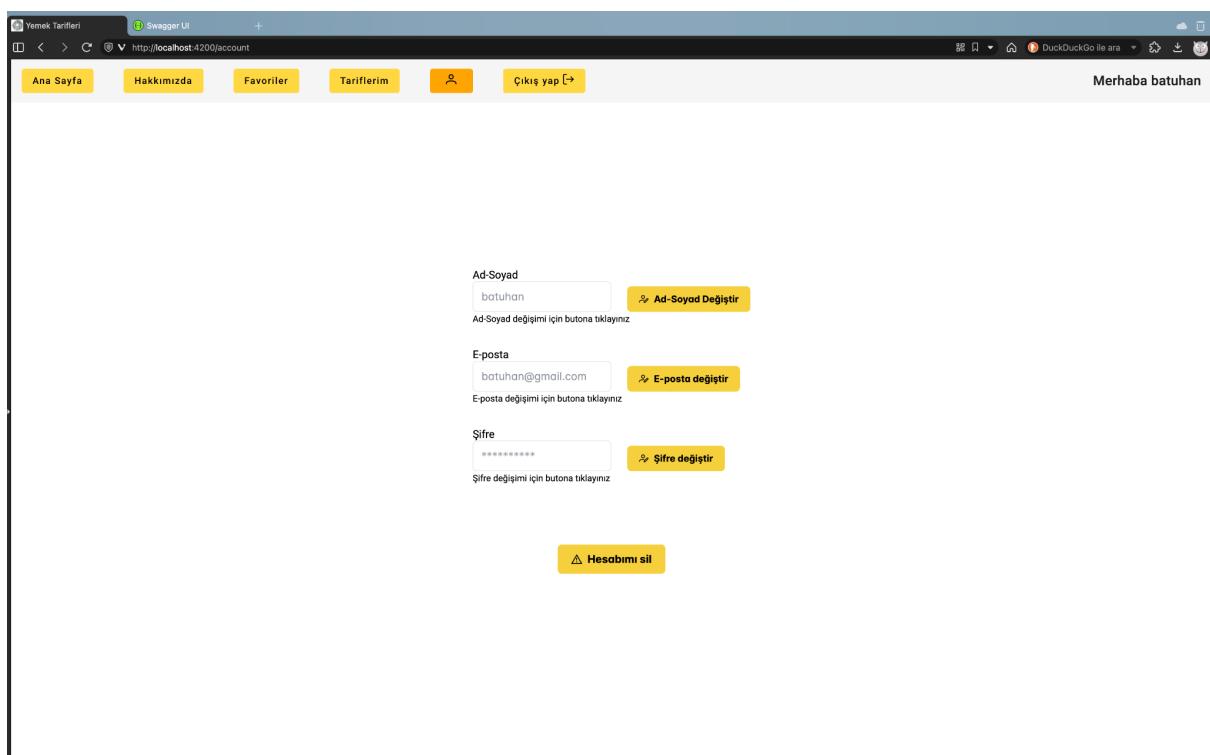
Fotoğraf

marry\_me\_chicken.webp 203.168 KB

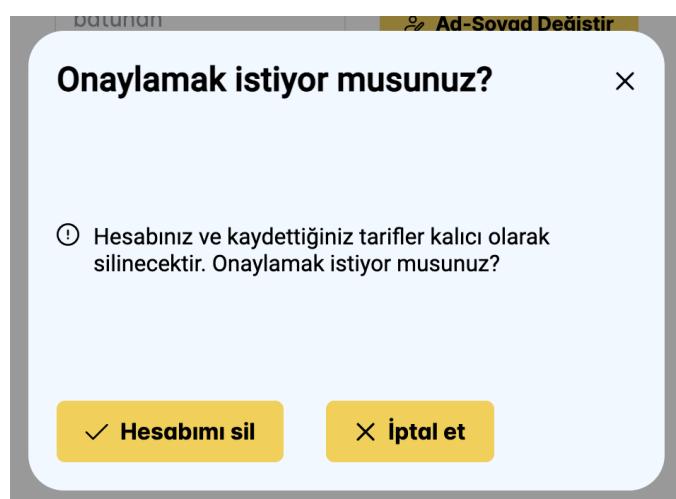
Yemeği düzenle...

## 4. Hesap Bilgileri

Hesap bilgileri sayfasında Ad-Soyad, Email ve Şifre değiştirme gibi seçenekler bulunmaktadır. Ad-Soyad değişiminde sayfa yenilenmesine gerek kalmadan yukarıda bulunan Merhaba yazısı güncellenir ve Tariflerdeki tarifi yapan kısmı değişir. Eğer değişiklik başarılı olduysa başarılı olduğunu belirten bir pop-up çıkmaktadır eğer başarısız olduysa hata olduğunu belirten bir pop-up çıkmaktadır.



Hesabımı sil tuşuna tıklandığında bir onay kutusu çıkmaktadır eğer onay kutusu reddedilirse işlem iptal edilecektir, devam edilirse de hesap kalıcı olarak silinecektir



Eğer şifre değiştirmek istenirse eski şifre kontrolü gerçekleştirmektedir eğer eski şifre hatalıysa hata olduğunu belirten pop-up çıkacaktır.

## Eski Şifre

## Yeni Şifre

<

Kaydet

## 5. Kaynak Kod

```
[HttpPost]
0 references
public async Task<ActionResult<Response>> SaveFood([FromForm] FoodDtoReq request){
    var response = new Response();
    Food food = new Food();
    int userId = await context.Users.Where(u => u.Email == request.CreatedBy).Select(u => u.Id).FirstOrDefaultAsync();
    var Ingredients = request.Ingredients.Split(',').ToList();
    IFormFile image = request.Image;
    using (MemoryStream ms = new MemoryStream())
    {
        image.CopyTo(ms);
        byte[] imageBytes = ms.ToArray();
        food.Image = imageBytes;
    }
    food.Name = request.Name;
    food.Type = request.Type;
    food.Recipe = request.Recipe;
    food.CreatedBy = userId;
    food.Ingredients = Ingredients;
    food.ImageName = request.ImageName;

    await context.Foods.AddAsync(food);
    await context.SaveChangesAsync();
    response.Message = "Yemek başarılı bir şekilde kaydedilmiştir.";
    return Ok(response);
}
```

Yemek ekleme post isteği yukarıda görüldüğü gibidir resim dosyası Database'e byte array olarak kayıt edilmektedir. Malzemeler ise „ , “ ile ayrılmış tek bir string olarak gelmektedir bunu String array e çevirdikten sonra Database'e kayıt edilir.

```
[HttpPost("register")]
0 references
public async Task<ActionResult<Response>> Register(UserDto request){
    bool isUserExists = await context.Users.AnyAsync(u => u.Email == request.Email);
    var response = new Response();
    if(isUserExists){
        response.Message="Bu kullanıcı zaten var.";
        return BadRequest(response);
    }
    userService.CreatePasswordHash(request.Password, out byte[] passwordHash, out byte[] passwordSalt);
    User user = new User(){
        Name = request.Name,
        Email = request.Email,
        PasswordHash = passwordHash,
        PasswordSalt = passwordSalt
    };
    await context.Users.AddAsync(user);
    await context.SaveChangesAsync();
    response.Message = "Kullanıcı başarı ile oluşturuldu.";
    return Ok(response);
}
```

Kayıt olurken çalışan post isteği yukarıda oluşturulduğu gibidir şifre Database de direkt bir string olarak değil Hashlenmiş bir şekilde tutulmaktadır. Giriş işlemi gerçekleşirken de decrypt işlemi gerçekleşmektedir.

```
ngOnInit(): void {
    this.isOwner = this.route.snapshot.data['isOwner']
    if(!this.isOwner){
        this.router.navigate(['/home'])
    }
}
```

Bu kod parçası Anguların tarifi düzenleme kısmında bulunmaktadır eğer Tarif kullanıcının değilse hiçbir türlü düzenleme imkanı bulunmamaktadır.

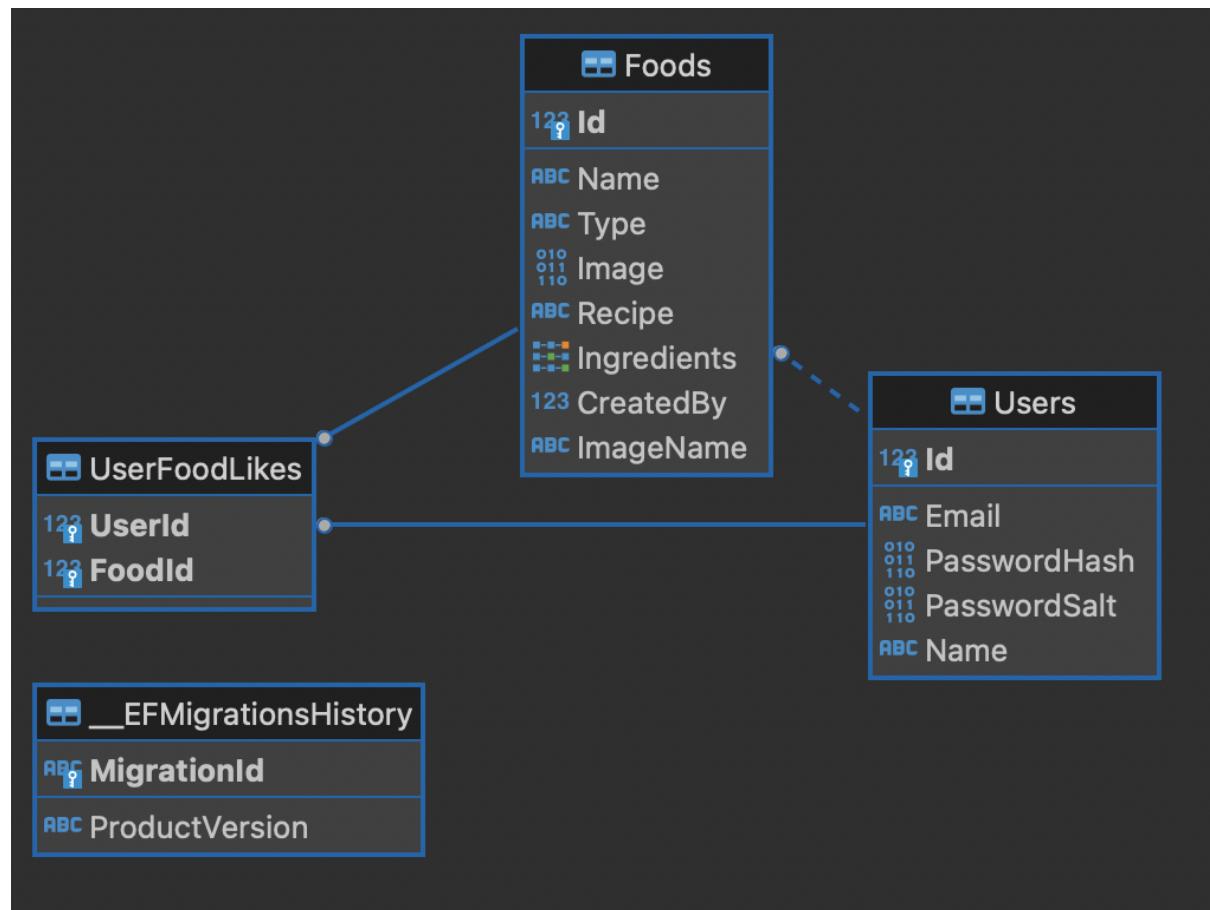
```

<main>
  <div id="search-box">
    <div id="search">
      <span class="p-input-icon-right">
        <i class="pi pi-search"></i>
        <input type="text" pInputText [(ngModel)]="search" (input)="searchCheck()" placeholder="Arama yap...">
      </span>
    </div>
  </div>
  <article>
    @if(result.length){
      @for (food of result; track $index;)
      {
        <app-food-card [food]="food"></app-food-card>
      }
    }@else{
      <p>{{emptyString}}</p>
    }
  </article>
</main>

```

Bu html dosyası Sayfalar içindeki tüm tariflerin bulunduğu Ana sayfayı temsil etmektedir eğer herhangi bir tarif bulunmuyorsa ona göre yazı ya da aranan tarif yoksa Aradığınız tarif bulunmamaktadır yazısı çıkmaktadır.

Database ER diyagramı aşağıda görüldüğü şekilde migration lar da oluşturulmuştur. Hangi kullanıcı hangi yemeği favorilere eklediğini kaydetmek için “UserFoodLikes” tablosu kullanılmaktadır.



## **KAYNAKLAR**

<https://www.postgresql.org/docs/current/>

<https://angular.io/docs>

<https://primeng.org>

<https://material.angular.io>

<https://learn.microsoft.com/en-us/dotnet/>

<https://www.nuget.org>

<https://www.npmjs.com>

<https://fonts.google.com/icons>

# EKLER

## Ek-1 - Swagger UI

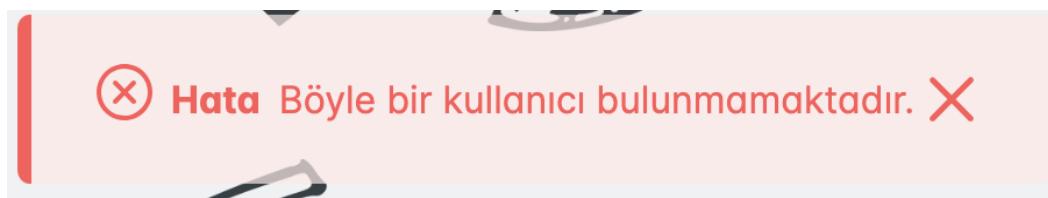
The screenshot shows the Swagger UI interface for a RESTful API. The API is organized into three main sections: Account, Auth, and Food.

- Account:** Contains four methods:
  - PUT /api/Account/name
  - PUT /api/Account/email
  - PUT /api/Account/passwd
  - DELETE /api/Account/{email}
- Auth:** Contains two methods:
  - POST /api/Auth/register
  - POST /api/Auth/login
- Food:** Contains 14 methods:
  - POST /api/Food
  - GET /api/Food/{id}
  - DELETE /api/Food/{id}
  - PUT /api/Food/{id}
  - GET /api/Food/all
  - GET /api/Food/user
  - POST /api/Food/like
  - GET /api/Food/like
  - POST /api/Food/dislike
  - GET /api/Food/{id}/check\_ownership

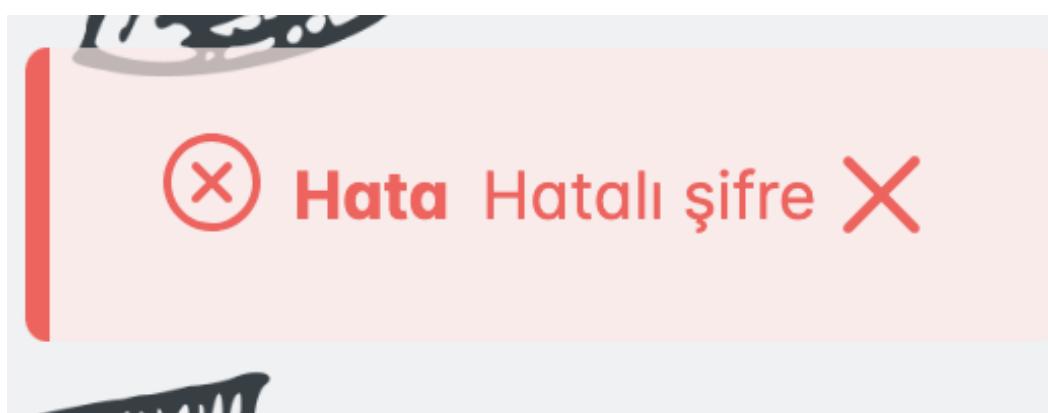
## Ek-2 - Angular routes

```
const routes: Routes = [
  {path:"", redirectTo:"/home", pathMatch:"full"},
  {path:"login", component: LoginComponent},
  {path:"home", component: HomeComponent, canActivate: [AuthGuard]},
  {path:"account", component: AccountComponent, canActivate: [AuthGuard]},
  {path:"change_account_settings/:id", component: AccountSettingsComponent, canActivate: [AuthGuard]},
  {path:"my_recipes", component: UserRecipesComponent, canActivate: [AuthGuard]},
  {path:"add_recipe", component: AddRecipeComponent, canActivate: [AuthGuard]},
  {path:"edit_recipe/:id", component: EditRecipeComponent, canActivate: [AuthGuard], resolve:{isOwner : FoodResolver}},
  {path:"liked_recipes", component: LikedRecipesComponent, canActivate: [AuthGuard]},
  {path:"food_detail/:id", component: FoodDetailComponent, canActivate: [AuthGuard], resolve:{isOwner : FoodResolver}},
  {path:"**", redirectTo:"/home", pathMatch:"full"}]
```

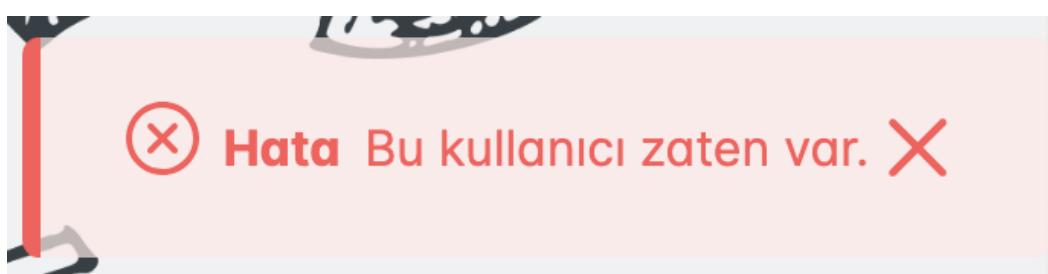
**Ek-3 - Kullanıcı bulunmamaktadır**



**Ek-4 - Hatalı parola**



**Ek-5 - Kayıtlı hesap bulunmaktadır**



**Ek-6 - Tarif silme onay**

