



**T.C.
SAKARYA ÜNİVERSİTESİ**

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

WEB PROGRAMLAMA

E-Ticaret Web Sitesi Uygulaması

1A - B191210075 – Nur Sena Aytar

Dr.Öğr.Üyesi HÜSEYİN ESKİ

Github Linki: <https://github.com/nursenaaytar/ECommerceSite>

SAKARYA

2022-2023

GİRİŞ

Bu ödevde Web Programlama dersinde teorik ve pratik olarak öğrenilen bilgilerin proje üzerinde uygulanması amaçlanmıştır. Projede kategorilendirilmiş ürünlerin yayınlandığı, bu kategoriler üzerinde kullanıcıların rollerine göre CRUD işlemlerini sağladığı, dinamik bir web sitesi uygulaması yapılmıştır. Proje ASP.NET CORE MVC mimari deseni üzerine tasarlanmıştır. Projenin Model kısmı için MSSQL Server veritabanı uygulaması kullanılarak Code First yaklaşımıyla veritabanı işlemleri uygulanmıştır. View kısmında ise HTML5, CSS3, Bootstrap, JavaScript, jQuery teknolojileri ve Razor Syntax kullanılarak C# dili kullanımı sağlanmıştır. Controller içerisinde bulunan methodlar yardımıyla Model ve View arasında etkileşim sağlanarak dinamik bir şekilde işlemlerin yapılması sağlanmıştır. Proje üzerinde uygulanan işlemler GitHub uygulaması üzerinden yayınlanmıştır.

NAYTAR Market bir e-ticaret sitesidir. Bu sitenin amacı ürün sipariş edebilmektir.

REGISTER HERE

PROFILE INFORMATION

Full Name...

PhoneNumber...

User Name...

LOGIN INFORMATION

Email

Password

Password Confirmation

Register

[Home](#)

Üye olma ekranıdır. Yukarıdaki alanlar üye olmak isteyen kişiden istenir. Password büyük harf, küçük harf, noktalama işareti ve rakam içermelidir.

```
namespace Microsoft.AspNetCore.Identity
{
    public class UserManager<TUser> : IDisposable where TUser : class
    {
        public const string ResetPasswordTokenPurpose = "ResetPassword";
        public const string ChangePhoneNumberTokenPurpose = "ChangePhoneNumber";
        public const string ConfirmEmailTokenPurpose = "EmailConfirmation";

        protected virtual CancellationTokens CancellationTokens { get; }
        protected internal IUserStore<TUser> Store { get; }
        public virtual ILogger Logger { get; }
        public IPasswordHasher<TUser> PasswordHasher { get; }
        public IList<IUserValidator<TUser>> UserValidators { get; }
        public IList<IPasswordValidator<TUser>> PasswordValidators { get; }
        public ILookupNormalizer KeyNormalizer { get; }
        public IdentityErrorDescriber ErrorDescriber { get; }
        public IdentityOptions Options { get; }
        public virtual bool SupportsUserAuthenticationTokens { get; }
        public virtual bool SupportsUserAuthenticationKey { get; }
        public virtual bool SupportsUserTwoFactorRecoveryCodes { get; }
        public virtual bool SupportsUserTwoFactor { get; }
        public virtual bool SupportsUserPassword { get; }
        public virtual bool SupportsUserSecurityStamp { get; }
        public virtual bool SupportsUserRole { get; }
        public virtual bool SupportsUserLogin { get; }
        public virtual bool SupportsUserEmail { get; }
        public virtual bool SupportsUserPhoneNumber { get; }
        public virtual bool SupportsUserClaim { get; }
        public virtual bool SupportsUserLockout { get; }
        public virtual bool SupportsQueryableUsers { get; }
        public virtual IQueryable<TUser> Users { get; }

        public UserManager(IUserStore<TUser> store, IOptions<IdentityOptions> optionsAccessor, IPasswordHasher<TUser> passwordHasher, IEnumerable<IUserValidator<TUser>> userValidators,
    }
}
```

Kayıt işlemi için Microsoft.AspNetCore.Identity kütüphanesi kullanıldı. Password kontrolü, aynı kullanıcı adının tekrar girilmesi gibi işlemlerin kontrolü bu kütüphane ile sağlandı.

LOGIN FORM

Login

FOR NEW PEOPLE

[Register Here](#) (Or) go back to [Home](#) >

Daha önce sisteme kaydı yapılmış kullanıcılar için sisteme giriş alanıdır. Mail ve password doğru girilmesi ile sisteme erişim sağlanır.

Kullanıcı Türleri:

İki farklı kullanıcı tipi mevcuttur: admin ve user.

Kullanıcı oluşturmak için mail doğrulama ile onaylama gerektirmektedir. Üyelik işlemi sırasında girilen mail adresine link gelir, linke tıklanarak yönlendirilen sayfaya erişim sağlandığında üyelik işlemi tamamlanmış olur.


User Kullanıcı:

User kullanıcıları ürün görüntüleyebilir, ürün siparişi verebilir, sipariş takibi yapabilir, isteklerini istek bölümünden admin kullanıcılarına iletebilirler.

User Profil Alanı:

Kullanıcılar profil bilgilerini güncelleyebilirler.

MY PROFILE



Sena Aytaç

[My Wish List](#)

[My Addresses](#)

[My Cards](#)

Name*

Email*

Phone Number*

Birthday

Image

[Choose File](#) No file chosen

[Update](#)

Ayrıca fotoğraflarının alt kısmında bulunan butonlardan çeşitli işlemler gerçekleştirebilirler.

-My Wish List butonu ile;

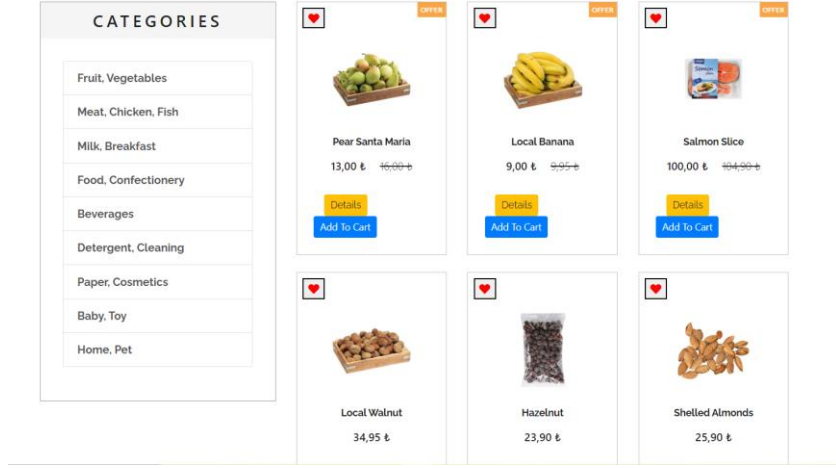
İsteklerini belirtecekleri ekrana yönlendirilirler.

-My Addresses butonu ile;

Sipariş adresi eklemek üzere adres ej-klme ekranına yönlendirilirler.

-My Cards butonu ile;



Siparişlerde kullanmak üzere kart ekleme ekranına yönlendirilirler.



Şekil 1: Ürünler

Ürünler alanından kategorize edilmiş olan sipariş edebilecekleri ürünleri gözlemleyebilir, sepete ekleyebilirler.

Shopping Cart

Image	Name	Price	Quantity	Total	Delete
	Local Banana	9,00 ₺	1	9,00 ₺	

CHECKOUT

Cart Total

₹9,00 ₺

Shipping

Free

Order Total

₹9,00 ₺

Continue Shopping

Şekil 2:Alışveriş Sepeti

Order Details

First Name

Last Name

Address

City

Phone

Email

Payment

Name on Credit Card

Credit Card Number

Month

Year

Cvv

Complete Order

Your Cart

Local Banana x(1)



9,00 ₺

Total (8)

9,00 ₺

Şekil 3:Ürün Sipariş Ekranı

Orders

Order :	Name	Price	Quantity
	Towel 8 Pieces	26,25	1
	Barley Noodle	3,25	1
Customer Name :	Ayşe Nadir	Total:29,50 ₺	
Address:	Istanbul/Kartal		
Phone:	5294883347		
Email:	ayse.nadir@gmail.com		
Payment Status:	CreditCard		
Order Status:	Pending		

Order :	Name	Price	Quantity
	Local Banana	9,00	6
Customer Name :		Ayse Nadir	Total:54,00 ₺

Şekil 4:Siparişler Ekranı

Ana SayfaÜrünlerKartlarSiparişlerHakkımızdaİletişim

aytarÇalış

Sakarya Üniversitesi
Kemalpaşa, Esentepe Kampüsü,
Universite Cd., 54050
Sarıyer/Sakarya
4,5 ★★★★★ 1.289 yorum
Daha büyük harıtaayı gösterile

LEAVE A MESSAGE

Your Name

Your Email

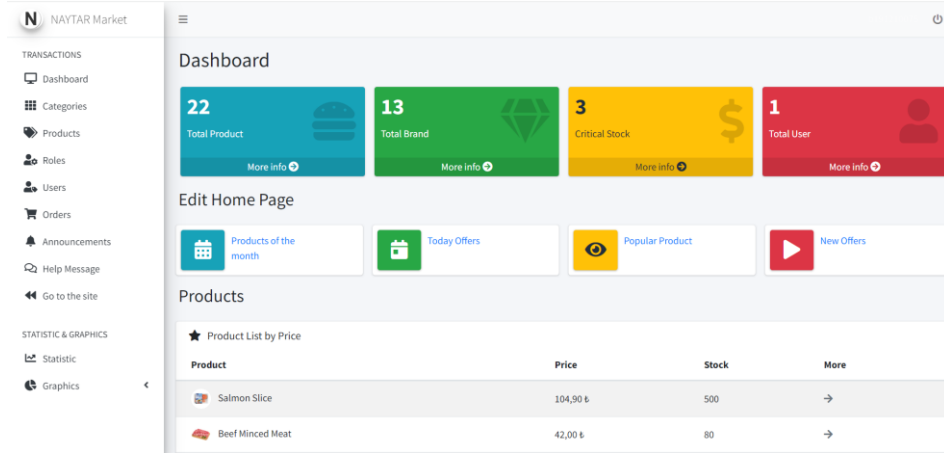
Your message here...

SUBMIT

Şekil 5:İletişim Ekranı

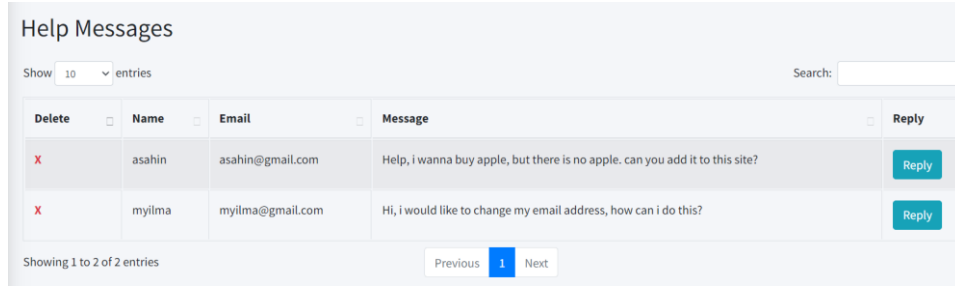
İletişim alanında mesaj bırakılabilir, bırakılan mesajlar admin tarafından görüntülenebilir.

Admin Kullanıcı:



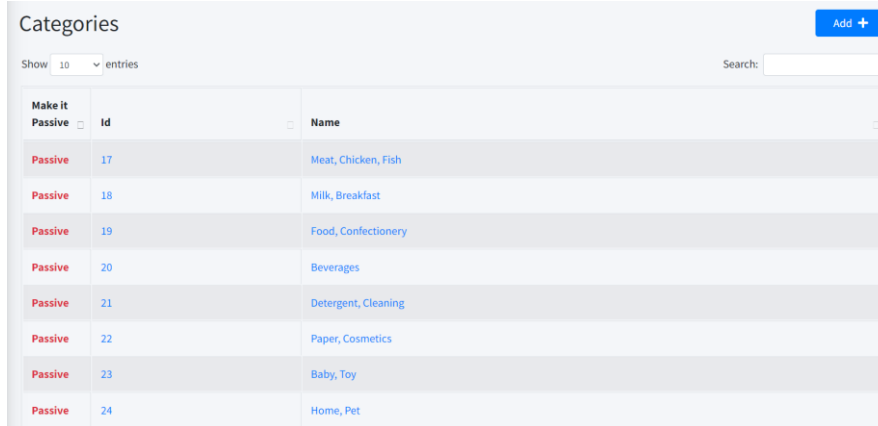
Product	Price	Stock	More
Salmon Slice	104,90 ₺	500	→
Beef Minced Meat	42,00 ₺	80	→

Tüm yetkilere sahiptir. Ürün ekleyebilir, güncelleme yapabilir, silebilir.



Delete	Name	Email	Message	Reply
X	asahin	asahin@gmail.com	Help, i wanna buy apple, but there is no apple. can you add it to this site?	Reply
X	myilma	myilma@gmail.com	Hi, i would like to change my email address, how can i do this?	Reply

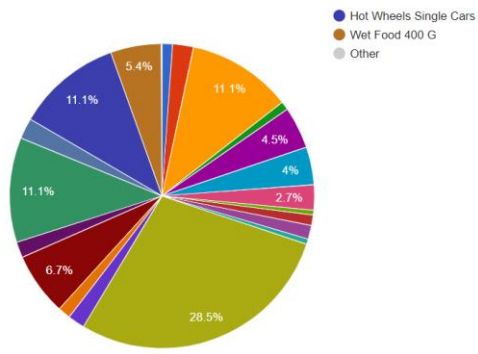
Kullanıcı isteklerini görüntüleyebilir, isteklere yanıt verebilir.



Make it Passive	Id	Name
Passive	17	Meat, Chicken, Fish
Passive	18	Milk, Breakfast
Passive	19	Food, Confectionery
Passive	20	Beverages
Passive	21	Detergent, Cleaning
Passive	22	Paper, Cosmetics
Passive	23	Baby, Toy
Passive	24	Home, Pet

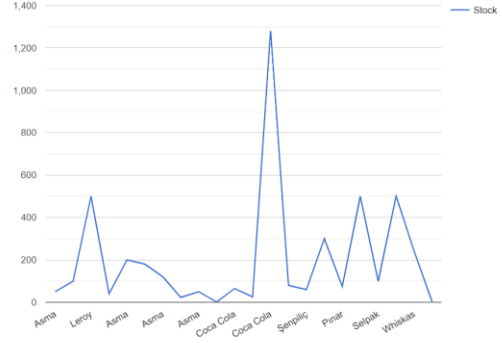
Ürün kategorilerinin aktif, pasiflik durumlarını değiştirebilir.

Product - Stock Graphic

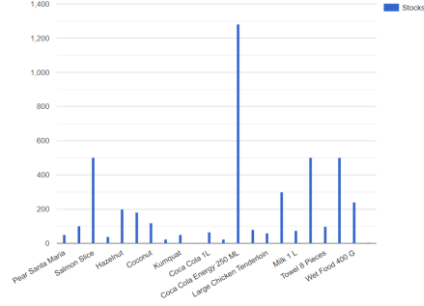


2/2

Brand - Stock Graphic



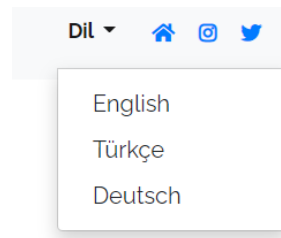
Product - Stock Graphic



Ürün stoklarını pie-chart, line chart ve column chart grafiği ile gözlemleyebilir.

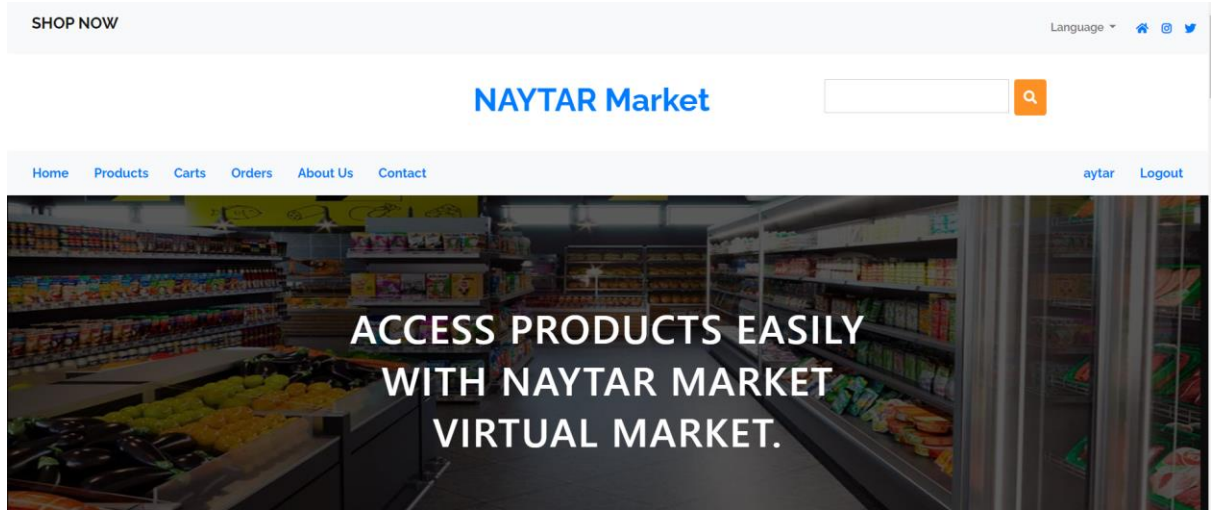
Çoklu Dil Desteği

Uygulamaya giriş yapıldıktan sonra sağ üst kısımda bulunan dil seçme işlemiyle menüde dil değişikliği yapılabilir. Türkçe, İngilizce ve Almanca dil seçeneği bulunmaktadır.



Şekil 6: Dil Seçim Alanı

İngilizce Dil Desteği İle:



Almanca Dil Desteęi İle:



Türkçe Dil Desteęi İle:



Dil Desteği İçin Kullanılan Bazı Kodlar:

```
services.Configure<RequestLocalizationOptions>(  
    options =>  
    {  
        var supportedCultures = new List<CultureInfo>  
        {  
            new CultureInfo("en-US"),  
            new CultureInfo("nl-NL"),  
            new CultureInfo("tr-TR"),  
        };  
  
        options.DefaultRequestCulture = new RequestCulture(culture: "en-US", uiCulture: "en-US");  
  
        options.SupportedCultures = supportedCultures;  
        options.SupportedUICultures = supportedCultures;  
        options.RequestCultureProviders.Insert(0, new QueryStringRequestCultureProvider());  
    });
```

```
@inject LanguageService language
```

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">  
    <ul class="navbar-nav mr-auto"></ul>  
    @if (User.Identity.IsAuthenticated)  
    {  
        <a class="nav-link" asp-area="" asp-controller="Customer" asp-action="Profile">@User.Identity.Name</a>  
        <a class="nav-link" asp-area="" asp-controller="Account" asp-action="Logout">@language.Getkey("Logout")</a>  
    }  
    else  
    {  
        <a class="nav-link" asp-area="" asp-controller="Account" asp-action="Login">@language.Getkey("Login")</a>  
        <a class="nav-link" asp-area="" asp-controller="Account" asp-action="Register">@language.Getkey("Register")</a>  
    }  
</div>
```

Şekil 7:Authorization ve Dil Desteği Çoklu Kod Örneği

Api Hizmeti:

ApiController oluşturuldu.

```
[Route("api/[controller]/[action]")]  
[ApiController]  
1 reference  
public class ApiController : Controller  
{  
    BorsaContext _context = new BorsaContext();  
    private IProductService _productService;  
    private ICategoryService _categoryService;  
    0 references  
    public ApiController(IProductService productService, ICategoryService categoryService)  
    {  
        _productService = productService;  
        _categoryService = categoryService;  
    }  
}
```

Api route işlemi [Route("api/[controller]/[action]")] ile sağlandı.

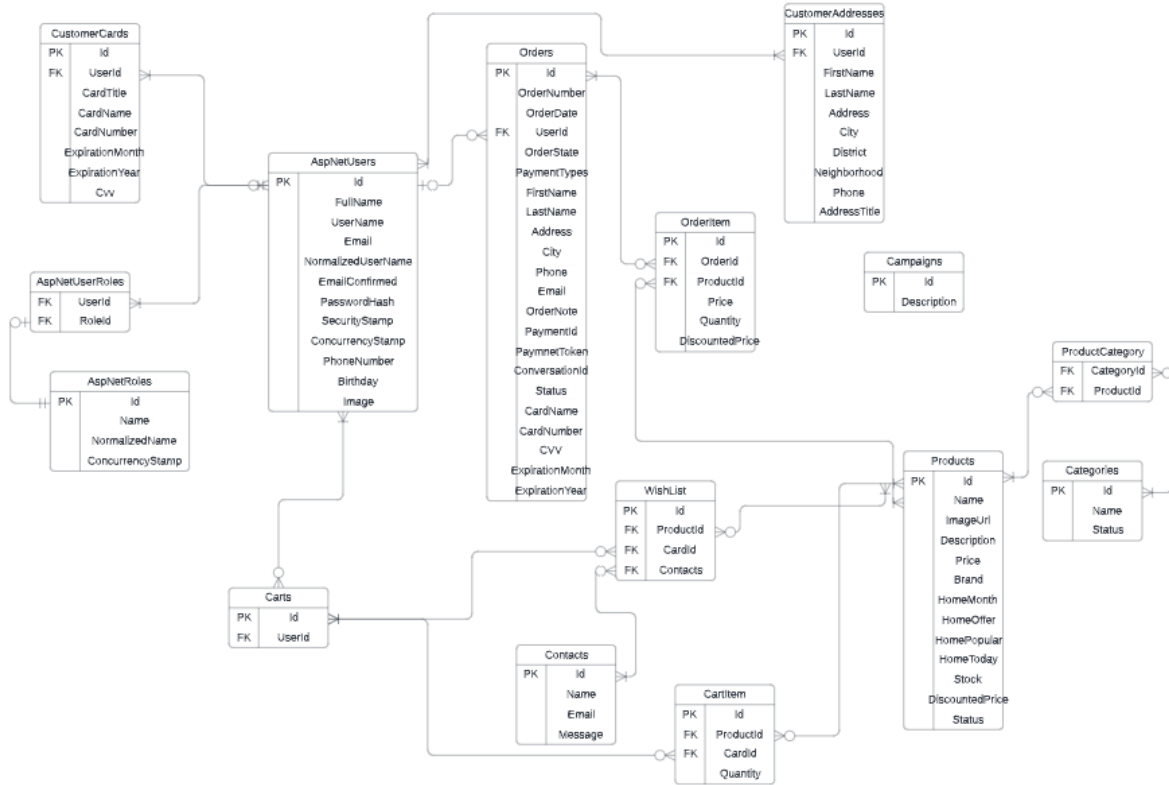
Api ile;

- Get ile girilen max ve min değerler aralığındaki ürünler listelenebilir.
- Get ile tüm ürünler listelenebilir.
- Get ile stoğu tükenen ürünler listelenebilir.

-Post ile gönderilen kategori id'sine ait veri silinebilir. Eğer kategoriye ait ürün var ise hata verir.

```
[HttpDelete("{id}")]
0 references
public JsonResult Delete(int id)
{
    var message = "";
    using (var context = new BorsaContext())
    {
        var y1 = context.Categories.FirstOrDefault(x => x.Id == id);
        if (y1 is null)
        {
            message = "Kategori bulunamadı!";
        }
        var y2 = context.ProductCategory.FirstOrDefault(x => x.CategoryId == id);
        if (y2 is not null)
        {
            message = "Kategoriye bağlı ürünler var, kategori silinemez.";
        }
        if (message == "")
        {
            context.Categories.Remove(y1);
            context.SaveChanges();
            message = "Başarılı";
        }
        return Json(message);
    }
}
```

UML Diyagramı



Tablolar:

-**Products:** Ürünlerin tutulduğu tablodur.

-**Orders:** Siparişlerin tutulduğu tablodur.

- ProductCategory:** Ürün -kategori eşleşmelerinin tutulduğu tablodur.
- Wishlist:** Kullanıcıların ürün isteklerinin listelendiği tablodur.
- OrderItem:** Sipariş miktarlarının tutulduğu tablodur.
- CustomerCards:** Kullanıcı kartlarının tutulduğu tablodur.
- Contacts:** İletişim kurmak isteyen kullanıcıların isteklerinin listelendiği tablodur.
- Categories:** Ürün kategori isimlerinin tutulduğu tablodur.
- Carts:** Kullanıcı kartlarının tutulduğu tablodur.
- CardItem:** Kart-ürün-miktar tablosudur.
- Campaigns:** Kampanyaların tutulduğu tablodur.
- Users:** Kullanıcıların tutulduğu tablodur.
- UserRoles:** Kullanıcı – rol eşleşmelerinin tutulduğu tablodur.
- Roles:** Rollerin tutulduğu tablodur.
- CustomerAddress:** Kullanıcı adreslerinin tutulduğu tablodur.