

**Yalova Üniversitesi**

**Mühendislik Fakültesi**

**Bilgisayar Mühendisliği Bölümü**

**Web Programlama Ödev 2**

**(okuoku.com)**

**Hazırlayan**

**M. Bektaş ÇİMEN**

**Gökhan ŞİŞMAN**

**Ekim-2019**

İçindekiler

[Header 3](#_Toc23280901)

[Menü Navbar 4](#_Toc23280902)

[Content(index) 5](#_Toc23280903)

[Content(ürün) 6](#_Toc23280904)

[Footer 6](#_Toc23280905)

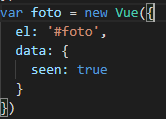
# Header

Bu kısım sitenin header kısmıdır. Burası içerisinde giriş ve üye ol butonları, sepetim, arama butonu, siteye ait bir logo ve menü yer alır.

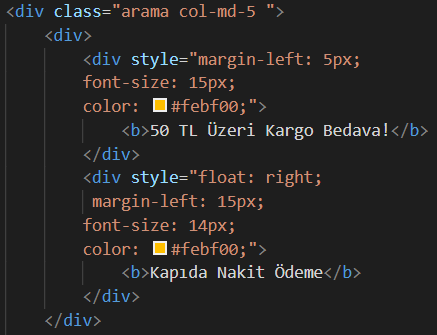
Sitenin açılımında bir yükleme ekranı ile karşılaşmaktayız. Sitenin içeriğinin yüklenmesi ve verilerin veritabanından alınması için geçen süre bu şekilde bekleniyor. Fakat bu ödevde veritabanı olmadığı için kendi yazdığımız js ile bir bekleme ekranı oluşturuldu.



ID si “silinecek” olan div 3 sn sonra setTimeout fonksiyonuyla ekranı kapatma işlemi sağlanıyor.



Burada , sitenin en üstünde gözüken banner için vueJS yazıldı. Seen özelliğini “false” yaparak bannerın kapanmasını sağlıyoruz ve bunuda @click özelliği ile yapıyoruz.



Örnek olarak bir tanesine yer verdiğimizde bir div oluşturuldu. Div içerisine style tagı açarak bu butonlara şekil verildi. Margin-left; div içerisinde sol tarafa doğru boşluk bırakmak için kullanıldı. Font-size; yazının büyüklüğü, Float: right ise bu butonu divin sağına yanaştırmak için kullanıldı.

# Menü Navbar

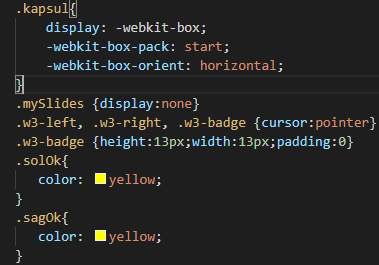
Menü içerisinde kategoriler bulunuyor. Iconlar **(ref)**fontawesome.css ‘inden geliyor.



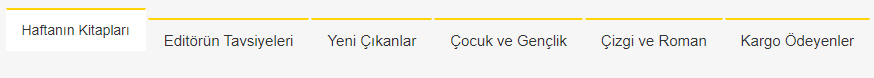
**Text-transform:** uppercase komutu li içerisindeki bütün karakterlerin büyük harf ile yazılmasını sağlar. Bu şekilde menü kısmı oluşturulmuş ve headerın en altına yerleştirilmiş oldu.

# Content(index)

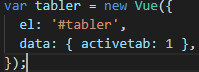
Sitede bulanan sliderlar [w3Schools’dan](https://www.w3schools.com/w3css/w3css_slideshow.asp)(ref: [kod](https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_slideshow_dots2)) alınmıştır.



Kapsül , solOk ve sagOk cssleri yazıldı. Sliderlar’da kullanılan js ‘lerde w3Schoolsdan alındı.



Yukarıda görülen tab’ler vueJs ile birlikte yapıldı. Yararlanılan kaynak([ref](https://vuejsexamples.com/tabbed-content-with-vue-js/)).



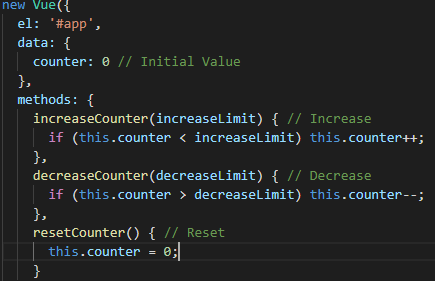
Data içinde başlangıç(initial) değerimizi aktif olan tabımızı 1 olarak tanımlıyoruz. Daha sonra tıklanılan tab’a göre aktif tab’ın değerini değiştiriyoruz.Örneğin tab2 ye tıkladığıysa “activetab = 1” oluyor. Bu sayede güncellenen tab değerine göre kitaplar geliyor. Kitaplarında bilgileri vueJs ile birlikte datadan geliyor. Örnek bir kitap ; {{object.kitapAdi}} şeklinde veriye ulaşabiliyoruz.



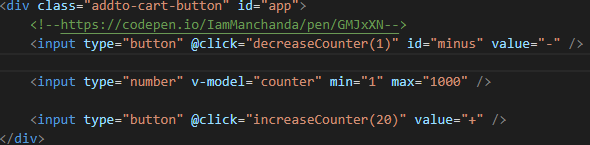
# Content(ürün)



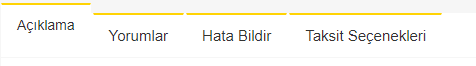
Yukarıda görülen kısmı vueJs yardımıyla yapıldı. Yararlanılan kaynak([ref](https://codepen.io/IamManchanda/pen/GMJxXN))



Tanımlanan methodlar parametre alıyor , azaltırken 1 , üst limit ise geçici olarak 20 yapıldı.



@click özelliği ile decreaseCounter metodunu parametresi 1 vererek her tıkladığında çağrılıyor. “number” inputu v-model ile counter verisine bağlı, değişecek olan kısım burası. IncreaseCounter metodu ise counter’ı bir arttırıyor lakin koşulu counter değişkeninin 20’den küçük olması. Burada pek bir önemi yok lakin ilerde kullanılabilir.



Bu sayfada bulunan tab , index sayfasındaki mantıkla aynıdır.

# Footer